A study on the detection of double faults

Ying Liu

2010

A dissertation presented for fulfillment of the requirements for the award of
Doctor of Philosophy
Abstract

Fault-based testing aims at selecting test cases to detect hypothesized faults in a program. Ideally, such selected test cases can guarantee to detect hypothesized faults if they do appear in the programs. Most fault-based testing strategies derive their test cases based on the assumption that at most one of the hypothesized faults is committed by the programmer. However, empirical studies show that multiple faults occur more often in programs and test case selection strategies developed for single faults cannot guarantee to detect multiple faults.

In this thesis, double faults that might have occurred within Boolean expressions are studied. A double fault is defined as the occurrences of any two single faults. Since the different order of occurrences of two single faults in a double fault may result in different faulty expressions, double faults were classified into two categories, namely double fault with and without ordering. After modelling faulty expressions from these two categories, we compared these faulty expressions and enumerate all possible faulty expressions of double fault in Boolean expressions.

The detection conditions of the considered faulty expression were studied from a theoretical perspective and are characterized based on certain properties of test cases. These conditions helped to identify and develop test case selection strategies to detect such double faults.

After examining the fault detection capability of some existing test case selection strategies developed for single faults, six test case selection strategies were proposed to supplement an existing test case selection strategy to detect double faults. We proved that these strategies are guaranteed to detect double faults and investigate the cost of these strategies for double faults detection.

This thesis helps us to better understand how different faults occur in a Boolean expression and how they interact with each other. This provides further insight for modelling and detecting multiple faults such as triple faults.
Acknowledgement

I would like to thank my supervisor, Dr. Man Fai Lau, for his continuous support, encouragement, and guidance for my research and his wonderful patience and kindness as a friend and mentor.

I also like to thank Australian Government who grants me a scholarship to support my research during this work, and Faculty of Information and Communication Technologies in Swinburne University of Technology, who provide a grateful environment for my studies. Part of the work in this thesis was done in collaboration with Dr. Y.T. Yu, Department of Computer Science, City University of Hong Kong, with support by grants from the Research Grants Council of Hong Kong Special Administrative Region, China (project nos. 123206 and 1166/04E).

I would also thank my working colleagues, DeHao Huang, Diana Kuo, Huai Liu, Irene Moser and Prof. T.Y. Chen for useful discussions and encouragement; James Montgomery and Niger Fisher for reading my thesis.

I also want to thank my family members, especially my parents, JinSheng Liu and Ming Tang, who provide great support during my studies. I also want to thank Jason Brownlee who shows great patience and keeps encouraging me to finish this thesis.
CANDIDATURE DECLARATION

I certify that this thesis entitled:

A study on the detection of double faults

is the result of my own research and that where reference is made to the work of others, due acknowledgment is given. I also certify that this Thesis which has been accepted for the Degree has not, wholly or in part, been submitted for a degree or diploma at any other university or institution.

Ying Liu
Notation

General

\[ \emptyset \] The empty set
\[ A, B, \ldots \] Test sets
\[ |A| \] The cardinality of a set \( A \)
\[ A \setminus B \] The set of complement of \( B \) in \( A \)
\[ A \cap B \] The set intersection of \( A \) and \( B \)
\[ A \cup B \] The set union of \( A \) and \( B \)
\[ f : A \rightarrow B \] A function from a set \( A \) to set \( B \)
\[ \log_2 n \] The logarithm of \( n \) with base 2
\[ \log_e n \] The natural logarithm of \( n \)
Boolean Algebra

- The Boolean AND operator
- The Boolean NOT operator
+ The Boolean OR operator
⊕ The Boolean exclusive OR operator
0 The FALSE truth value
1 The TRUE truth value

\( a, b, c, \ldots, x, \ldots \) Boolean variables

\( B \) The set of all truth values, that is \( B = \{0, 1\} \)

\( B^n \) The \( n \) dimensional Boolean space, that is, \( B^n = B \times B \times \cdots \times B \) (\( n \) copies)

\( E_1, E_2, \ldots \) Subexpressions of a given Boolean expression \( S \)

\( E'_1, E'_2, \ldots \) Resulting subexpressions when a single fault \( F \) occurs on \( E_1, E_2, \ldots \)

\( F_1, F_2, \ldots \) Single fault classes

\( F_1 \times F_2 \) Double fault class formed by \( F_1 \) and \( F_2 \)

\( F_1(E_1 \rightarrow E'_1) \) A single fault \( F \) occurs on \( E_1 \) making it to become \( E'_1 \)

\( I \) An implementation of \( S \)

\( I_{F_1(E_1 \rightarrow E'_1), F_2(E_2 \rightarrow E'_2)} \) An implementation of a double fault \( F_1 \times F_2 \) without ordering for \( S \), where \( F_1 \) occurs on \( E_1 \) and \( F_2 \) occurs on \( E_2 \)

\( I_{F_1(E_1 \rightarrow E'_1) \otimes F_2(E_2 \rightarrow E'_2)} \) An implementation of a double fault \( F_1 \times F_2 \) with ordering for \( S \), where \( F_1 \) occurs on \( E_1 \) and \( F_2 \) occurs on \( E_2 \)

\( I_{F_1(E_1 \rightarrow E'_1) \triangleleft F_2(E_2 \rightarrow E'_2)} \) An implementation of a double fault \( F_1 \triangleleft F_2 \) for \( S \), where \( F_1 \) occurs on \( E_1 \) and \( F_2 \) occurs on \( E_2 \)

\( k_i \) The number of literals in the \( i \)-th term \( p_i \) of a Boolean expression

\( n \) The number of variables in a Boolean expression

\( p_i \) The \( i \)-th term of a Boolean expression

\( p_i x_l \) A term obtained from \( p_i \) by inserting one of its missing literal \( x_l \)

\( p_i, 1, j \) The subexpressions of \( p_i \) obtained by keeping the first \( j \) literals of \( p_i \) and removing the rest
The subexpression obtained from $p_{i,1,j}$ by negating the literal $x^i_j$

The subexpressions of $p_i$ obtained by removing the first $j$ literals of $p_i$ and keeping the rest

A term obtained from $p_i$ by negating its literal $x^i_j$

A term obtained from $p_i$ by negating its literal $x^i_j$ and inserting a literal $x_l$

A term obtained from $p_i$ by omitting its literal $x^i_j$

A term obtained from $p_i$ by replacing its literal $x^i_j$ with one of its missing literal $x_l$

A term obtained from $p_i$ by negating its literals $x^i_{j_1}$ and $x^i_{j_2}$

A term obtained from $p_i$ by negating its literal $x^i_{j_1}$ and omitting its literal $x^i_{j_2}$

A term obtained from $p_i$ by negating its literal $x^i_{j_1}$ and replacing its literal $x^i_{j_2}$ with $x_l$

A term obtained from $p_i$ by negating its literals $x^i_{j_1}$ and $x^i_{j_2}$

A term obtained from $p_i$ by omitting its literal $x^i_{j_1}$ and replacing its literal $x^i_{j_2}$ with $x_l$

A term obtained from $p_i$ by replacing its literal $x^i_{j_1}$ with $x_{l_1}$ and $x^i_{j_2}$ with $x_{l_2}$

The subexpression obtained by negating both the term $p_i$ and its literal $x^i_j$

The subexpression obtained by negating both the term $p_i$ and omitting its literal $x^i_j$

The subexpression obtained by negating the term $p_i$ and replacing its literal $x^i_j$ with $x_l$

Boolean expressions or Boolean specifications

The Boolean difference of two Boolean expressions $S_1$ and $S_2$

Elements in the Boolean space $B^n$

The negation of the Boolean variable $x$

The $j$-th literal of the $i$-th term of a Boolean expression

A literal in a Boolean expression
Sets related to Boolean expression $S$

$FP(S)$  The set of all false points of $S$

$NFP(S)$  The set of all near false points of $S$

$NFP_i(S)$  The set of all near false points for the $i$-th term of $S$

$NFP_{i,j}(S)$  The set of all near false points for the $j$-th literal in the $i$-th term of $S$

$OTP(S)$  The set of all overlapping true points of $S$

$RFP(S)$  The set of all remaining false points of $S$

$TP(S)$  The set of all true points of $S$

$TP_i(S)$  The set of all true points of $p_i$ in $S$

$UTP(S)$  The set of all unique true points of $S$

$UTP_i(S)$  The set of all unique true points for the $i$-th term of $S$
Contents

1 Introduction 1
   1.1 Software Testing .................................................. 1
   1.2 Fault-based Testing .............................................. 2
   1.3 Specification-based Testing ..................................... 3
       1.3.1 Fault-based Testing based on Boolean Specification ....... 4
   1.4 Previous Approach on Multiple Faults ............................. 5
   1.5 Our Approach .......................................................... 6
   1.6 Overview of the Thesis ............................................. 8

2 Preliminary 9
   2.1 Notation ............................................................... 9
   2.2 Single Fault Class .................................................. 11
   2.3 Testing Strategies for Single Fault ................................. 12

3 Double Faults 16
   3.1 Double Faults ........................................................ 16

4 Double Faults Related to Terms Only 19
   4.1 Double Faults without Ordering .................................... 19
       4.1.1 ENF with Other Term Faults ................................. 20
       4.1.2 TNF with Other Term Faults ................................. 22
       4.1.3 TOF with Other Term Faults ................................ 24
       4.1.4 DORF with Other Term Faults ............................... 25
       4.1.5 CORF with Other Term Faults ............................... 26
   4.2 Double Faults with Ordering ....................................... 26
       4.2.1 ENF First, then Other Term Faults ............................ 27
       4.2.2 TNF First, then Other Term Faults ............................ 30
       4.2.3 TOF First, then Other Term Faults ............................ 31
       4.2.4 DORF First, then Other Term Faults ......................... 33
       4.2.5 CORF First, then Other Term Faults ......................... 35
4.3 Relation between Double Faults with and without Ordering on Faults Related to Terms Only ............................... 38
4.4 Summary .................................................................................. 42

5 Detection Conditions on Double Faults Related to Terms only 44
5.1 Detection conditions on 27 Double-fault Expressions Without Ordering 45
  5.1.1 ENF with Other Term Faults .............................................. 46
  5.1.2 TNF with Other Term Faults .............................................. 58
  5.1.3 TOF with Other Term Faults .............................................. 64
  5.1.4 DORF with Other Term Faults .......................................... 66
  5.1.5 CORF with Other Term Faults .......................................... 70
5.2 Detection conditions of 4 Remaining Double-fault Expressions with Ordering ........................................... 72

6 Double Faults Related to Literals Only 81
6.1 Double Faults without Ordering ............................................... 81
  6.1.1 LNF with Other Literals Faults ........................................ 81
  6.1.2 LOF with Other Literals Faults ........................................ 84
  6.1.3 LIF with Other Literals Faults ........................................ 86
  6.1.4 LRF with Other Literals Faults ........................................ 87
6.2 Double Faults with Ordering .................................................... 88
  6.2.1 LNF first, then Other Literals Faults ................................. 89
  6.2.2 LOF first, then Other Literals Faults ................................. 91
  6.2.3 LIF first, then Other Literals Faults ................................. 93
  6.2.4 LRF first, then Other Literals Faults ................................. 95
6.3 Relation between Double Faults with and without Ordering ............ 98
6.4 Summary .................................................................................. 102

7 Detection Conditions on Double Faults Related to Literals Only 104
  7.1 LNF with Other Literal Faults .............................................. 104
  7.2 LOF with Other Literal Faults .............................................. 114
  7.3 LIF with Other Literal Faults .............................................. 121
  7.4 LRF with Other Literal Faults .............................................. 129

8 Double Faults Related to a Term and a Literal 140
  8.1 Double Faults without Ordering ........................................... 140
    8.1.1 ENF with Literal Faults .............................................. 140
    8.1.2 TNF with Literal Faults .............................................. 143
    8.1.3 TOF with Literal Faults .............................................. 145
    8.1.4 DORF with Literal Faults ............................................ 147
8.2 Double Faults with Ordering ................................................. 152
  8.2.1 ENF First, then other Literal Faults .......................... 152
  8.2.2 TNF First, then other Literal Faults .......................... 154
  8.2.3 TOF First, then other Literal Faults .......................... 156
  8.2.4 DORF First, then other Literal Faults ......................... 157
  8.2.5 CORF First, then other Literal Faults ......................... 159
  8.2.6 LOF First, then other Term Faults ............................ 164
  8.2.7 LIF First, then other Term Faults ............................ 166
  8.2.8 LRF First, then other Term Faults ............................ 168

8.3 Relation between Double Faults with and without Ordering .......... 171

8.4 Summary ................................................................. 174

9 Detection Conditions on Double Faults Related to a Term and a Literal 179

  9.1 Detection conditions on 36 Double-fault Expressions based on Double Fault without Ordering ........................................... 179
    9.1.1 ENF with Other Literal Faults ................................. 179
    9.1.2 TNF with Other Literal Faults ................................. 191
    9.1.3 TOF with Other Literal Faults ................................. 199
    9.1.4 DORF with Other Literal Faults ............................... 206
    9.1.5 CORF with Other Literal Faults ............................... 220

  9.2 Detection conditions of 6 Remaining Double-fault Expressions with Ordering ......................................................... 228

  9.3 Summary ................................................................. 233

10 Existing Strategies in Detecting Double Faults 239

  10.1 The BASIC strategy .................................................. 240
  10.2 The MUMCUT strategy ............................................... 254
  10.3 The MAX-A Strategy .................................................. 257
  10.4 The MAX-B strategy .................................................. 262
  10.5 Summary ................................................................. 265

11 New Testing Strategies for Double Fault Detection 267

  11.1 Approach for developing new strategies ........................... 267
  11.2 The base test case selection strategy ............................. 270
  11.3 New Test Case Selection Strategies For Detecting Double Faults Related to Literals Only ......................................................... 271
    11.3.1 The Supplementary Multiple False Point (SMFP) Strategy . 271
    11.3.2 The Supplementary Multiple Overlapping True Point (SMOTP) Strategy ................................................................. 273
11.3.3 The Supplementary Multiple Unique True Point (SMUTP) Strategy 276
11.3.4 The Pairwise Multiple Unique True Point (PMUTP) Strategy 277
11.3.5 The Pairwise Multiple Near False Point (PMNFP) Strategy 282
11.3.6 The Supplementary Pairwise Multiple False Point (SPMFP) Strategy 284
11.4 Test Case Selection Strategies For Detecting Double Faults Related to Term and Literal Only 287
11.5 Summary 295

12 Empirical Study on MUMCUT with New Supplementary Strategies 296
12.1 The Aim 296
12.2 Experimental Method and Subject Expressions 297
12.3 Results and Observations 299
12.3.1 Results for LRU Expressions 299
12.3.2 Results for TCAS Expressions 303
12.3.3 Results for Random Expressions 308
12.3.4 Summary 312

13 Conclusions 314
13.1 Summary 314
13.2 Achievements 316
13.3 Future Work 317

A Works Arising from the Study 318
A.1 Peer-Reviewed Publications 318
A.2 Unpublished Technical Reports 318
List of Figures

10.1 Relationship of Double-Fault Expressions (DFE) detected by each strategy ........................................... 240

12.1 The LRU expressions [11] ................................................. 300
12.2 The TCAS expressions [56] ................................................. 304
12.3 The random expressions [60] ................................................. 309
List of Tables

2.1 Single fault, single-fault expression and detection condition \( S = p_1 + \ldots + p_m \) .......................... 13

3.1 Types of double faults without ordering ........................................ 17
3.2 Types of double faults with ordering ........................................... 18

4.1 Types of double term faults without ordering .......................... 20
4.2 Types of double term faults with ordering ................................. 27
4.3 Comparison of double-fault expressions of ENF and other term faults 39
4.4 Comparison of double-fault expressions of TNF and other term faults 39
4.5 Comparison of double-fault expressions of TOF and other term faults 40
4.6 Comparison of double-fault expressions of DORF and other term faults 40
4.7 Comparison of double-fault expressions of CORF and other term faults 41
4.8 Double fault classes and double-fault expressions \( S = p_1 + \ldots + p_m \) 42

5.1 Organization of detection conditions for double faults without ordering ........................................ 45
5.2 Double fault, double-fault expression and detection condition \( S = p_1 + \ldots + p_m \) .......................... 78

6.1 Types of double literal faults without ordering .......................... 82
6.2 Types of double literal faults with ordering ................................ 89
6.3 Comparison of double-fault expressions of LNF and other literal faults 99
6.4 Comparison of double-fault expressions of LOF and other literal faults 99
6.5 Comparison of double-fault expressions of LIF and other literal faults 101
6.6 Comparison of double-fault expressions of LRF and other literal faults 101
6.7 Double fault classes and double-fault expressions \( S = p_1 + \ldots + p_m \) 103

7.1 Organization of detection conditions for double faults without ordering ........................................ 104
7.2 Double fault, double-fault expression and detection condition \( S = p_1 + \ldots + p_m \) .......................... 138
11.9 All false points of $S$ where $S = ab + cd + ef$ ............... 285

11.10 Detection conditions aimed by the test case selection strategies .... 287

12.1 The mean size of test sets satisfying each strategy for the LRU group 300
12.2 Comparing different testing strategies for LRU expressions ......... 302
12.3 The mean size of randomly generated test sets for each strategy from
   20 TCAS II expressions ............................................. 305
12.4 Comparing different testing strategies for TCAS II expressions .... 307
12.5 The mean size of randomly generated test sets for each strategy from
   40 random expressions ............................................. 310
12.6 Comparing different testing strategies for Random expressions .... 311
12.7 Mean size of test sets (% of exhaustive test set) ................. 312
Chapter 1

Introduction

This study aims to detect double faults within Boolean expressions. Boolean expressions are fundamental in computer science and software engineering. For example, predicates in computer program can be expressed as Boolean expressions, and program specifications can be modelled as Boolean expressions or equivalent forms. Strategies for detecting faults in Boolean expressions have been investigated for decades [8, 9, 12, 18, 26, 33, 49, 51, 53, 56] and most studies assume that there is only one fault in a given Boolean expression. However, multiple faults are more common in practices [38, 55]. Double faults, as a special instance of multiple faults, within Boolean expressions are the focus of this study. By studying the detection of double faults, we hope to gain further insights into detecting more complicated multiple faults such as triple faults.

1.1 Software Testing

Reducing the cost of software and improving software quality are important objectives in software development. Even though the budgets for quality assurance are increasing, software failures are quite common [58]. Failures can have a catastrophic impact, especially in safety-critical software (for example: avionics computer systems, computer-controlled life support systems and industrial control systems).

A fault is a manifestation of human error in a program, while a failure is an observed deviation from the specified behaviour of the program [37]. Software testing, an important software quality assurance methodology [52], is aimed at discovering faults and therefore preventing failures in the operational phase. Moreover, statistical studies show that more than half of the resources of the entire software development process are spent on software testing [4, 19]. Therefore, improved testing can reduce the cost of developing software and improve software quality [40].

Many testing methodologies have been proposed to reveal faults in the program, such as code review, inspection and walkthrough. However, such methodologies are
performed without actually executing a program. Therefore, the dynamic behaviour of the program cannot be examined and validate against its requirement or specification. Dynamic testing is performed by executing the software with suitable input values and comparing the actual output with the expected output. In this study, we only focus on dynamic testing because of following two reasons: (1) dynamic testing can examine the program behaviour and performance while it is executing to verify whether it operated as expected, and (2) dynamic testing is the validation portion of software quality and assurance, and is widely used in the software development process. A test case is a combination of an input value and its expected output. A test case selection strategy is a systematic method to select (or generate) a set of test cases.

Traditionally, there are two main approaches to selecting test cases: white-box testing and black-box testing. White-box testing selects test cases based on the structure of the source code of a program. Two examples of white-box testing are statement testing, in which test cases are selected to execute every statement of a program, and branch testing, in which test cases are required to execute every branch within a program. In black-box testing, also called functional testing, testers treat the program as a black box and ignore the structure of the program. Hence, test cases are derived from the program specification rather than the program source code. In this thesis, we study the black box testing which focuses on testing whether the functional requirements are satisfied and identified those unsatisfied, hence, assure the code meet its’ requirements and fulfills its’ intended purpose.

1.2 Fault-based Testing

One of the major limitations of testing is that it can only show the presence of faults but not their absence [13]. Fault-based testing has arisen as a solution to this problem by guaranteeing that some hypothesized faults do not exist in the program under test [47]. Ideally, a fault-based test case selection strategy aims at selecting test cases to guarantee the detection of certain hypothesized faults that commonly occur during program development. More precisely, the test cases selected by the fault-based test case selection strategy can guarantee to reveal the corresponding failures of the program under test if those hypothesized faults do appear in the program. Therefore, when executing such test cases without causing the program under test to fail, we can claim that the hypothesized faults do not exist in the program. Lau and Yu [33] point out that fault-based testing is especially effective when the hypothesized faults are committed in the program.

Recent studies on fault-based approaches mainly focus on selecting test cases to detect certain well-defined fault classes [9, 18, 50]. The fault-based testing approach
can be applied to source code or specification. For example, mutation testing, a commonly accepted fault-based testing technique, was originally proposed to apply to source code [41, 42, 43]. It works by introducing lots of simple syntactic faults, called mutation operators, into a program to create different versions of mutated programs (or simply mutants), and then collect test cases that can distinguished the original program and mutants. It relies on two assumptions: (1) the competent programmer hypothesis [15], which states that an incorrect program written by a competent programmer will differ from a correct version by relatively simple faults, and (2) the believe that a test set which detects all simple mutants in a program will have a high chance to detect complex mutants [42]. Mutation testing has also been applied to program specifications to automatically generate test cases [6]. In simple terms, mutation analysis is applied to specifications as follows: to create different versions of a specification by injecting many simple faults (one version per fault) and then collect test cases that can distinguish the original specification from the created specifications with injected faults. Empirical studies have shown that even for some simple syntactical faults, this technique can create a large number of programs whose behaviours are as complex as those produced by real faults [14, 36]. As a result, when the nature and types of faults cannot be easily identified, mutation-style faults can be used. However, mutation analysis is very expensive for practical use.

1.3 Specification-based Testing

One of the many goals of software testing is to verify that a program behaves correctly according to its specifications. Even though every statement or branch in a program is executed by test cases generated by code-based testing, the function described by the specification is not always guaranteed to be implemented. Moreover, it has been shown that test cases selected from code-based testing are not reliable failure finders [4, 24].

Recently, specification-based testing, which selects test cases based on the program specification, has received much attention. Test cases selected from specification-based testing can directly reveal whether the program under test behaves according to its specification. Moreover, specification-based testing has the following advantages. First, specification-based testing can reduce required software developing resources. For example, test cases can be generated from the specification when a program is being developed and executed immediately when the program is finished. Moreover, test cases can be reused even after faults have been found and corrected in a program, provided that its specification has not been changed. Second, specification-based testing can also help to confirm the specification’s correctness and consistency because the specification is reviewed during the process of test case
There has been an increasing interest and attention on using a fault-based approach to generate test cases from software specifications \[5, 9, 15, 26, 43, 50, 53\]. The common approach first hypothesizes some fault types that may be committed during programming, and then generates test cases to detect those faults. For example, Chen and Lau [9] study and generalise faults that may occur in a specification, and then propose three test case selection strategies to detect those faults.

1.3.1 Fault-based Testing based on Boolean Specification

Formal specification has been generally accepted in software development industry and various formal specifications have been proposed and used, such as Boolean specification [2, 3], RSML [34], SOFL [35] and Z [48]. Formal specification is defined as the use of mathematical notation to precisely describe the properties that software must have [48]. Many test generation strategies based on formal specifications have been proposed [1, 9, 14, 16, 17, 20, 21, 44, 45, 54, 56]. Carrington [7] argues that a formal specification plays an important role in software testing because a formal specification is a convenient starting point for the systematic derivation of test suites for testing implementation. Moreover, it has been shown that formal specifications help to automatically generate test cases [17, 20, 44, 56, 60].

Boolean specifications are program specifications written in Boolean expressions. In practice, program specifications may use Boolean expressions as part of the formalism. For example, the program specification may include some functional requirements, such as the conditions and events that trigger state transitions [2, 45], and some non-functional requirements, such as safety properties [3, 34], that can be written as Boolean expressions or their equivalent forms. Moreover, some behaviour of software systems can be represented by Boolean expressions. For instance, Chilen- ski [11] extracted over 20,000 Boolean expressions from an airborne software system.

The goal of fault-based testing is to detect hypothesized faults. Since any type of fault may occur in a program, the choice or definition of hypothesized faults affect the effectiveness of fault-based testing techniques. So far, various hypothesized faults related to Boolean expressions have been defined and studied [9, 26, 33, 53, 56], such as expression negation fault (ENF, a Boolean expression is replaced by its negation), variable negation fault (VNF, a variable is replaced by its negation), and variable reference fault (VRF, a variable is replaced by another variable). Kuhn [26] found the relationship among the conditions that detect these three types of faults. The relationship states that any test case that can detect a VRF can also detect its corresponding VNF, which in turn can detect its corresponding ENF, provided the specification is expressed in Boolean expression in disjunctive normal form. Tsuchiya
and Kikuno [53] further analyzed the relationship between a VRF and a missing condition fault (MCF). A MCF is one which causes a condition in a specification to be omitted from the implementation. They found that test cases that can detect a MCF may not be able to detect the corresponding VRF and vice versa. Lau and Yu [33] extend their study to nine types of fault classes that include literal, operator, term, and expression faults within Boolean expressions. By analyzing the relationships between these faults, they discovered an extended fault class hierarchy. A fault class hierarchy establishes relationships between different types of fault classes. For example, if any test case that can detect fault class $A$ can also detect fault class $B$, then $A$ is put in the lower part of the hierarchy than $B$.

Many test case selection strategies and algorithms have been proposed to generate test cases from Boolean expressions [9, 12, 18, 49, 51, 56] and some of them can be used to generate test cases from program specifications, such as the BOR strategy [49], the BASIC, MAX-A and MAX-B meaningful impact strategies (or simply the BASIC, MAX-A and MAX-B strategies) [56] and the MUMCUT strategy [9]. The details of these strategies will be discussed in Chapter 2. The BOR testing strategy can guarantee the detection of Boolean operator faults [50], however it requires every variable in the expression to occur only once. In [33], nine types of fault classes that may occur in Boolean expressions are used to evaluate some well-known test case strategies. It has been shown that the BASIC strategy can detect seven out of nine types of faults, while the MUMCUT, MAX-A and MAX-B strategies can detect all nine types of fault [33]. Moreover, empirical studies [56, 59] have shown that the MUMCUT, MAX-A and MAX-B strategies require, on average, 12.0%, 40.6% and 48.2% of the entire input domain, respectively. Thus the MUMCUT strategy is much more cost-effective in detecting these faults.

1.4 Previous Approach on Multiple Faults

Most fault-based testing strategies mentioned above derive their test cases based on the assumption that at most one of the hypothesized faults is committed by programmers [39]. However, it is quite common that, during software development, programmers make mistakes which in turn inject more than one fault into a program. Moreover, empirical studies on software faults show that, in practice, multiple faults occur more often in programs [38, 55]. Hence, studies related to multiple faults may give further insights into errors committed by programmers.

Previous studies on multiple faults focus mainly on double faults and fault coupling [22, 23, 41, 42]. A program is said to have a double fault if there are two occurrences of single faults. Faults are said to be coupled if they can be detected in isolation but not when combined together.
Offutt [42] performed an empirical study on fault coupling via mutation analysis. Three programs whose sizes ranged from 16 to 28 lines of code were studied. A mutant is a 1-order (2-order in the case of a double fault) mutant if it differs from the original program by one syntactic change (two syntactic changes in the case of a double fault). Test sets that can detect all 1-order mutants were generated and used to detect 2-order mutants. Offutt stated that the experiment performed was a study of the mutation coupling effect, to be precise. It was found that test sets so generated can detect approximately 99.9% of 2-order mutants. He concluded that the effect of two faults coupled together rarely occurred.

How Tai Wah [22, 23] studied fault coupling from a theoretical perspective. He modelled a program as a composition of finite number of functions. A program with a single fault was modelled as a composition of functions in which exactly one function was faulty, while a program with double faults was a composition of functions in which exactly two of them were faulty. A function was considered to be faulty if it produced an incorrect output. In other words, a program with double fault was a combination of two individual faulty functions. He investigated the problem of detecting double faults by test sets that can detect the two faults in isolation. All such test sets considered in the study had at most two elements. It should be noted that if such a test set has one element, that single test case can detect both faults in isolation. Test sets that can detect individual faults of a double fault are defined as proper test sets. Among proper test sets, those that cannot detect the double fault are defined as coupled test sets. He calculated the coupling ratio, defined as the ratio of the number of coupled test sets to that of proper test sets. The coupling ratio is approximately $\frac{1}{|D|}$ and $\frac{1}{|D|^2}$ for test sets of sizes 1 and 2, respectively, where $|D|$ is the size of the input domain $D$. As $|D|$ is usually very large, the ratio is very small. He concluded that fault coupling rarely occurs.

1.5 Our Approach

In this study, we chose to study the problem of multiple faults by using double faults for two reasons: (1) multiple faults are multiple occurrences of single faults that may be of the same or different class, (2) Previous studies [22, 23, 41, 42] on multiple faults have used double faults as a special instance of multiple faults, as they are the simplest form of multiple faults. Therefore, the study of double faults is a good starting point to bring some insight into multiple faults.

This study aims to detect double faults within Boolean expressions. Double faults studied in this thesis are two occurrences of single faults from [33]. Test cases that can detect these double faults are generated from software specifications. This is different from Offutt’s approach [41, 42], which is code based, and How
Tai Wah’s approach [22, 23], which is from a functional viewpoint. Since programs are developed based on specifications, modeling faults from specifications provides better insights on how faults are being injected into programs. The rest of this section gives a brief overview of the approach.

Firstly, all possible double fault classes that may occur in Boolean expressions are identified. Faulty expressions with double fault that are equivalent to the original expression are excluded. Moreover, faulty expressions with double faults that are equivalent to faulty expressions with single faults are discarded because they can be detected by existing test case selection strategies. Since the order of occurrences of two single faults in a double fault may result in different expressions, double faults and their corresponding faulty expressions are classified into two categories, namely double faults with and without ordering. The first case refers to the situation where the ordering of the two single faults may actually make a difference whereas the second case refers to the situation where single faults are independent of each other. This is different from previous studies because they have not explicitly considered how a single fault can affect another single fault. Since two individual faults committed in different orders may result in two different faulty implementations, it is important to study these double faults from both orders for better insights into how these faults interact with each other and, hence, for better double fault detection. Moreover, by examining the relationship of the faulty expressions between double faults with and without ordering, the faulty expressions of all possible double faults studied in this thesis are obtained.

Secondly, the detection conditions of these double faults are proved by using the method presented in [26]. The detection condition of a particular fault that might occur in a program is a condition that makes the actual output of the faulty program different from its original intended output. In short, the method in [26] describes that detection condition can be derived from $S \oplus I$ where $S$ is a Boolean expression, $I$ is created by injecting faults into $S$ and $\oplus$ is the exclusive-or operator XOR. Instead of only presenting the Boolean expression $S \oplus I$ as detection conditions, the detection condition is categorized based on properties of test cases. This helps in identifying and developing test case selection strategies to detect studied double faults.

Thirdly, the fault-detecting capabilities of some well-known test case selection strategies on double faults are studied in this thesis. Moreover, new test case selection strategies are developed. A tool is implemented to automatically generate test cases based on the newly proposed test case selection strategies and some well-known test case selection strategies for each of the 80 Boolean expressions that are taken from [11, 56, 60]. Note that this study aims to guarantee the detection of double faults instead of investigating only the percentage of detected double
faults [22, 23, 42]. Since newly developed strategies are fault-based test case selection strategies, they guarantee to detect hypothesized double faults in this study.

1.6 Overview of the Thesis

Chapter 2 introduces the notation used in this thesis, single fault classes and test case selection strategies for single faults. Chapter 3 discusses double faults studied in this thesis. Chapters 4 through 9 present different types of double fault classes, their corresponding faulty expressions and fault detection conditions of the corresponding faulty expressions. Chapters 4 and 5 present those related to terms only, Chapters 6 and 7 present those related to literals only, and Chapters 8 and 9 present those related to term and literal. Chapter 10 analyses the fault-detecting capabilities of some well-known test case selection strategies in detecting double faults, while Chapter 11 develops new test case selection strategies that supplement the existing ones to detect double faults. Chapter 12 studies the cost-effectiveness of these new test case selection strategies and Chapter 13 concludes the thesis and discusses future work.
Chapter 2

Preliminary

2.1 Notation

In this thesis, 1 and 0 are used to represent ‘TRUE’ and ‘FALSE’ respectively. The Boolean operators AND, OR and NOT, are denoted by ‘·’, ‘+’ and ‘¬’, respectively. Usually, ‘·’ is omitted whenever it is clear from the context. The set of all truth values, that is \{0, 1\}, and the n-dimensional Boolean space are denoted by \(\mathbb{B}\) and \(\mathbb{B}^n\), respectively. Given a Boolean expression \(p_1 + p_2 + \cdots + p_m\) with \(m\) terms and \(n\) variables, it uniquely defines a Boolean function \(f : \mathbb{B}^n \rightarrow \mathbb{B}\). A test case or test point for the Boolean function \(f\) is a point in the \(n\)-dimensional Boolean space \(\mathbb{B}^n\).

Let \(S = p_1 + \cdots + p_m\) be a Boolean expression in irredundant disjunctive normal form (IDNF) (that is, disjunctive normal form with no redundant term or literal) where \(m\) is the number of terms and \(p_i\) is the \(i\)-th term of \(S\). Let \(p_i = x_{i1}^j \cdots x_{ik_i}^j\) be the \(i\)-th term of \(S\) where \(x_{ij}\) is the \(j\)-th literal in \(p_i\) and \(k_i\) is the number of literals in \(p_i\). A literal \(x\) of \(S\) is a missing literal of a term \(p_i\) if both \(x\) and \(\bar{x}\) do not appear in the term. The corresponding software implementing \(S\) as its specification will be referred to as an implementation of \(S\) in this thesis. In this study, the given Boolean expression \(S\) is in IDNF because the expressions in IDNF are more common in practice. For example, Chilenski [11] extracted more than 20,000 logical decisions from the airborne software of five different Line Replaceable Units in five different systems across two airplane models. Over 94% of the occurrences of decision tabulated in [11] are in IDNF [59].

Let \(S = p_1 + \cdots + p_m\) be a Boolean expression in IDNF. A true point of \(S\) is a point in \(\mathbb{B}^n\) such that \(S\) evaluates to 1. We use \(TP(S)\) to denote the set of all true
points of $S$. A true point of the $i$-th term, $p_i$, of $S$ is a point such that $p_i$ evaluates to 1. We use $TP_i(S)$ to denote the set of all true points of $p_i$ in $S$. A unique true point of $p_i$ in $S$ is a true point of $p_i$ such that terms other than $p_i$ evaluate to 0. The set of all unique true points of $p_i$ is denoted by $UTP_i(S)\left(=\left(TP_{i1}(S)\setminus \bigcup_{i\neq i_1} TP_i(S)\right)\right)$.

True points that are not unique true points are overlapping true points. The set of all overlapping true points of $S$ is denoted by $OTP(S)$. For example, let $S=ab+cd+de$. A true point of $p_1=ab$ in $S$ is 11011 (that is, $a = b = 1, c = 0$ and $d = e = 1$). The set $TP_1(S)$ of all true points of $p_1$ is $\{11000, 11001, 11010, 11011, 11100, 11101, 11110, 11111\}$. The point 11000 is a unique true point of $p_1$ in $S$. The point 11110 is an overlapping true point of $S$ because both $p_1$ and $p_2$ evaluate to 1.

A false point of $S$ is a point in $\mathbb{B}^n$ such that $S$ evaluates to 0. We use $FP(S)$ to denote the set of all false points of $S$. A near false point for the $j$-th literal, $x_j$, of the $i$-th term, $p_i$, in $S$ is a false point of $S$ such that (1) $x_j$ evaluates to 0, and (2) all literals in $p_i$ other than $x_j$ evaluate to 1. The set of all near false points for $x_j$ of $p_i$ in $S$ is denoted by $NFP_{i,j}(S)$. False points that are not near false points are remaining false points. The set of all remaining false points of $S$ is denoted by $RFP(S)$. For example, let $S=ab+cd+de$. A near false point for the first literal $a$ of the first term $p_1=ab$ is 01000. The set $NFP_{1,1}(S)$ of all near false points for the first literal $a$ of the first term $ab$ is $\{01000, 01001, 01010, 01100, 01101\}$. The point 00000 is a remaining false point of $S$.

Note that, when $S$ is expressed in IDNF, $UTP_i(S)$ and $NFP_{i,j}(S)$ are always non-empty, as stated in the following lemma.

**Lemma 2.1.1 ([9, Theorem 1])** Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Then,

1. $UTP_i(S) \neq \emptyset$ for all $i = 1, \ldots, m$
2. $NFP_{i,j}(S) \neq \emptyset$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, k_i$, where $k_i$ is the number of literals in the term $p_i$.

Let $S=p_1 + \cdots + p_m$ be a Boolean expression in IDNF. Let $x$ be a missing literal of the $i$-th term, $p_i$, of $S$. Given a test set $T$, $T$ covers all truth values of $x$ (that is, 0 and 1) if there is a point $\vec{t}_1 \in T$ such that $x$ evaluates to 0 on $\vec{t}_1$ and there is another point in $\vec{t}_2 \in T$ such that $x$ evaluates to 1 on $\vec{t}_2$. The literal $x$ has a missing truth value on $T$ when $T$ cannot cover all truth values of $x$.

Let $x_{l_1}$ and $x_{l_2}$ be two different literals of $S$. A given test set $T$ can pairwise cover all truth values of $x_{l_1}$ and $x_{l_2}$, if all possible truth value combinations of $x_{l_1}$ and $x_{l_2}$, that is, 00 ($x_{l_1}=0$ and $x_{l_2}=0$), 01, 10 and 11, are covered. The literals $x_{l_1}$ and $x_{l_2}$ have some missing truth value pairs on $T$ when $T$ cannot pairwise cover all truth
value combinations of $x_{l_1}$ and $x_{l_2}$. For example, let $S = ab + cd$ and $T = \{1100, 1001, 0100, 0000\}$, $T$ pairwise covers all truth values of the missing literals $a$ and $b$ of the second term, $cd$, of $S$, but not the missing literals $c$ and $d$ of the first term, $ab$, of $S$ because $T$ cannot cover the truth value pairs 10 (that is, $c = 1$, $d = 0$) and 11 of $c$ and $d$.

### 2.2 Single Fault Class

Lau and Yu [33] have studied various common types of single faults that may be committed in Boolean expressions in various research literature [9, 26, 31, 53, 56] and proposed nine different fault classes. Among these fault classes, five of them are related to terms in Boolean expressions, while the remaining four are literal faults. They are defined as follows:

1. **Expression Negation Fault** (ENF): The entire Boolean expression or its subexpression is negated. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $\overline{ab} + \overline{cd} + \overline{ae}$ or $ab + \overline{cd} + \overline{ae}$.

2. **Term Negation Fault** (TNF): A term in a Boolean expression is negated. For example, the Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + cd + \overline{ae}$.

3. **Term Omission Fault** (TOF): A term in a Boolean expression is omitted. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + cd$.

4. **Disjunctive Operator Reference Fault** (DORF): The Boolean operator ‘+’ between any two terms in a Boolean expression is implemented as the Boolean operator ‘·’. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + cda$.

5. **Conjunctive Operator Reference Fault** (CORF): The Boolean operator ‘·’ between any two literals of a particular term in a Boolean expression is implemented as the Boolean operator ‘+’. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + c + d + ae$.

6. **Literal Negation Fault** (LNF): A literal of a particular term in a Boolean expression is negated. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + \overline{c} + \overline{d} + \overline{ae}$.

7. **Literal Omission Fault** (LOF): A literal of a particular term in a Boolean expression is omitted. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + \overline{c} + \overline{d} + \overline{ae}$.

---

1Note that TNF is considered as an ENF when there is only one term in the expression.

2Note that if the term consists of a single literal only (when $k_i = 1$), negating the literal will be regarded as a TNF.

3Note that if the term consists of a single literal only (when $k_i = 1$), omitting the literal will be regarded as a TOF.
be wrongly implemented as $ab + cd + e$.

8. **Literal Insertion Fault (LIF):** A missing literal of a particular term of a Boolean expression is inserted into the term. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + cd + ace$.

9. **Literal Reference Fault (LRF):** A literal in a particular term of a Boolean expression is replaced by a missing literal of the term. For example, a Boolean expression $ab + cd + ae$ may be wrongly implemented as $ab + cd + de$ with the literal ‘$a$’ in the third term ‘$ae$‘ being replaced by another literal ‘$d$‘ not in the third term. The literal $a$ is referred to as the *replaced literal* and the literal $d$ is referred to as the *replacing literal*.

For any Boolean specification $S$, $I$ is used to denote the corresponding implementation. It is possible that when a single fault occur in a Boolean expression, the resulting faulty expression may be equivalent to its original expression. For example, a Boolean expression $S = ab + ac + cd$ may be implemented as $I = ab\bar{c} + ac + cd$ by inserting the literal ‘$\bar{c}$‘ into the first term, the resulting expression $I$ is still equivalent to $S$. Therefore, we define a *single-fault expression* with respect to $S$ as an expression which (1) differs from the original expression by one syntactic change and (2) is not equivalent to the original expression. Then, $I = ab\bar{c} + ac + cd$ is not a single-fault expression with respect to $S$. However, $I' = abd + ac + cd$ is a single-fault expression with respect to $S$ because $S$ and $I$ evaluate to 1 and 0 on the point 1100.

Table 2.1 lists these nine fault classes, their corresponding faulty expressions and detection conditions. Let us consider the row related to LIF in Table 2.1 which represents the situation that a LIF is committed in $S$. If a missing literal $x_i$ of the $i$-th term $p_i$ in $S (= p_1 + \cdots + p_m)$ is inserted into that term, the resulting expression $I$ is equivalent to $p_1 + \cdots + p_i \cdot x_i + \cdots + p_m$. During the programming process such a LIF is actually mimicking the situation when an extra condition is wrongly inserted into a predicate. If $S \not\equiv I$, the resulting implementation will behave as $I$ which is different from $S$. Hence, test cases satisfying the detection condition of the corresponding LIF (that is, those test cases that make $p_i$ and only $p_i$ evaluate to true and an extra condition, $x_i$, evaluate to false) will be able to reveal the fault in the program.

### 2.3 Testing Strategies for Single Fault

Many test case selection strategies and algorithms have been developed to generate test cases from Boolean expressions, designed to detect certain types of faults [8, 18, 51, 49, 56], and some of them can be used to generate test case from program specification [8, 49, 56]. In this section, this type test case selection strategies are reviewed.
Table 2.1: Single fault, single-fault expression and detection condition \((S = p_1 + \ldots + p_m)\)

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Single-fault Expression</th>
<th>Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF*</td>
<td>( p_1 + \cdots + p_{i-1} + p_i + \cdots + p_k + p_{k+1} + \cdots + p_m )</td>
<td>any point in ((\bigcup_{i=1}^{m} TP_i(S)) \setminus \bigcup_{i\neq i,...,h}^{m} TP_i(S)) or any point in (FP(S))</td>
</tr>
<tr>
<td>TF</td>
<td>( p_1 + \cdots + p_{i-1} + p_i + \cdots + p_m )</td>
<td>any point in (UTP_i(S)) or any point in (FP(S))</td>
</tr>
<tr>
<td>DORF</td>
<td>( p_1 + \cdots + p_{i-1} + p_i p_1 + p_{i+2} + \cdots + p_m )</td>
<td>any point in (UTP_i(S)) or any point in (UTP_{i+1}(S))</td>
</tr>
<tr>
<td>CORF</td>
<td>( p_1 + \cdots + p_{i-1} + p_i,1,j + p_{i,j+1,k} + p_{i+1} + \cdots + p_m ) where ( p_{i,j+1,k} = x^i_1 \cdots x^i_j j_{i}^k ) and ( p_{i,j+1,k} = 1 )</td>
<td>any point in (FP(S)) such that ( p_{i,1,j} + p_{i,j+1,k} = 1 )</td>
</tr>
<tr>
<td>LNF</td>
<td>( p_1 + \cdots + p_{i-1} + x^i_1 \cdots x^i_j j_{i}^k + p_{i+1} + \cdots + p_m )</td>
<td>any point in (UTP_i(S)) or any point in (NFP_{i,j}(S))</td>
</tr>
<tr>
<td>LOF</td>
<td>( p_1 + \cdots + p_{i-1} + x^i_1 \cdots x^i_{j-1} x^i_{j+1} \cdots x^i_{k_i} + p_{i+1} + \cdots + p_m )</td>
<td>any point in (NFP_{i,j}(S))</td>
</tr>
<tr>
<td>LIF</td>
<td>( p_1 + \cdots + p_{i-1} + p_i x_l + p_{i+1} + \cdots + p_m )</td>
<td>any point in (UTP_i(S)) such that ( x_l = 0 )</td>
</tr>
<tr>
<td>LRF</td>
<td>( p_1 + \cdots + p_{i-1} + x^i_1 \cdots x^i_{j-1,j} x^i_{j+1} \cdots x^i_{k_i} x_l + p_{i+1} + \cdots + p_m )</td>
<td>any point in (UTP_i(S)) such that ( x_l = 0 ) or any point in (NFP_{i,j}(S)) such that ( x_l = 1 )</td>
</tr>
</tbody>
</table>

*When the entire Boolean expression \( S \) is negated, the resulting expression is \( \overline{S} \) and it can be distinguished from \( S \) by any point in \( \mathbb{B}^n \). Here, the negation of the entire Boolean expression is included as a special case of the negation of the subexpression \( p_1 + \cdots + p_k \) when \( i = 1 \) and \( h = m \).

*The detection condition of TNF can also be expressed as “any point such that \( p_1 + \ldots + p_{i-1} + p_{i+1} + \ldots + p_m = 0 \)”. However, for ease of reference and discussion in later sections, we choose to present it in the form shown in the table which is consistent with those of other faults.
Tai and colleagues developed a family of testing strategies for Boolean expressions [46, 50, 51]. One of their strategies is the BOR (Boolean Operator) strategy [51], which guarantees the detection of Boolean operator faults: (1) incorrect AND/OR operators, and (2) missing/extra NOT operators. Since the BOR strategy requires every variable in the expression to occur only once, it is not widely applicable to all Boolean expressions in IDNF [9].

Weyuker and colleagues also developed a family of testing strategies to automatically generate test cases from Boolean expressions [56]. Among these strategies, the basic meaningful impact strategy (simply, the BASIC strategy) is the core member which guarantees the detection of LNF. It is noted that, the MAX-A and MAX-B meaningful impact strategies (simply, the MAX-A and MAX-B strategies) developed in [56] do not target any type of fault, however it has shown in [9] that the MAX-A and MAX-B strategies can guarantee to detect all nine single fault classes considered in this thesis, while the BASIC strategy only detect seven of them. The rest of the section introduces the test case selection criteria of the three above-mentioned test case selection strategies based on a Boolean expression \( S = p_1 + \cdots + p_m \) in IDNF.

1. The BASIC strategy is required to select: (1) one point from \( UTP_i(S) \) for every \( i \); and (2) one point from \( NFP_{i,j}(S) \) for every \( i \) and \( j \).

2. The MAX-A strategy is required to select: (1) all points from \( UTP_i(S) \) for every \( i \); and (2) all points from \( NFP_{i,j}(S) \) for every \( i \) and \( j \).

3. The MAX-B strategy is required to select: (1) all points from \( UTP_i(S) \) for every \( i \); (2) all points from \( NFP_{i,j}(S) \) for every \( i \) and \( j \); (3) \( \lceil \log_2(|OTP(S)|) \rceil \) points from \( OTP(S) \) where \( |OTP(S)| \) denotes the size of \( OTP(S) \) (one point is selected if \( OTP(S) \) is a singleton set); and (4) \( \lceil \log_2(|RFP(S)|) \rceil \) points from \( RFP(S) \) where \( |RFP(S)| \) denotes the size of \( RFP(S) \) (one point is selected if \( RFP(S) \) is a singleton set).

Chen and Lau developed three test case selection strategies which aim to detect LIF and LRF. These strategies are the MUTP (Multiple Unique True Point) which guarantees the detection of LIF, the CUTPNFP (Corresponding Unique True Point and Near False Point Pair) strategy which guarantees the detection of LRF provided that required test points exist, and the MNFP (Multiple Near False Points) strategy which supplement the MUTP strategy to guarantee the detection of LRF when points required by the CUTPNFP strategy cannot be found [9, 10]. Later on, the MUTP, MNFP, and CUTPNFP strategies were integrated to form the MUMCUT strategy [10, 60]. It has been shown that the MUMCUT strategy can also be used to detect all nine single fault classes considered in this thesis [9]. More precisely, the MUMCUT strategy is a combination of these three test case selection strategies.
The three strategies, based on a Boolean expression $S(= p_1 + \cdots + p_m)$ in IDNF, are as follows:

1. The **MUTP** strategy is required to select unique true points from $UTP_i(S)$ such that all possible truth values (that is, 0 and 1) of every literal not occurring in $p_i$ are covered, for every $i$.

2. The **MNFP** strategy is required to select near false points from $NFP_{i,j}(S)$ such that all possible truth values of every literal not occurring in $p_i$ are covered, for every $i$ and $j$.

3. The **CUTPNFP** strategy is required to select one unique true point from $UTP_i(S)$ and one near false point from $NFP_{i,j}(S)$ such that they only differ at the truth value of the $j$-th literal of the $i$-th term, $p_i$, for every possible $i$ and $j$ pair.

From the test case requirements of the four test case selection strategies, it is easy to see that the MAX-B strategy subsumes the MAX-A strategy, which in turn subsumes the MUMCUT strategy, which in turn subsumes the BASIC strategy. A testing criterion $C_1$ is said to subsume another criterion $C_2$ if any test set that satisfies $C_1$ must also satisfy $C_2$ [57].
Chapter 3

Double Faults

3.1 Double Faults

Multiple occurrences of any fault classes may result in faulty expressions which differ from the original Boolean expression by several syntactic changes. A Boolean expression which differs from the original expression by more than one syntactic change is said to contain multiple faults. For example, let \( S = abc + cd + ef \) be a Boolean expression, \( ab\bar{c} + \bar{c}d + ef \) contains multiple faults because it differs from \( S \) by two syntactic changes. This study focuses on double faults defined as two occurrences of single faults which may be of the same or different classes discussed in the Chapter 2. The notation \( F_1 \circ F_2 \) is used to denote the double fault class formed by two single fault classes \( F_1 \) and \( F_2 \).

When two single faults occur in a Boolean expression, the resulting faulty expression may sometimes be equivalent to its original expression or a faulty expression with a single-fault expression. For example, if the first term of \( S \) is negated twice, the resulting expression, \( \bar{\bar{p}}_1 + p_2 + \cdots + p_m \), is equivalent to \( S \). On the other hand, if the first term of \( S \) is first negated and then removed, the resulting expression, \( p_2 + \cdots + p_m \), is equivalent to a single-fault expression where the first term is omitted. Such expressions are not studied in this work because there are strategies that can detect them [9]. As a result, only double-fault expressions are considered in this work. A double-fault expression is defined as an expression containing two single faults such that it (1) differs from the original expression by two syntactic changes and (2) is equivalent to neither the original expression nor any single-fault expression with respect to the original expression.

For the nine single fault classes introduced in Section 2.2, five of them are term faults and the remaining four are literal faults. Therefore, there are altogether 81 different ways to form a double fault. We apply a divide-and-conquer approach to further classify all double fault classes into the following three categories:
1. Double faults related to terms only;
2. Double faults related to literals only; and
3. Double faults related to a term and a literal.

Table 3.1: Types of double faults without ordering

<table>
<thead>
<tr>
<th>Term related faults</th>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
<th>Literal related faults</th>
<th>LNF</th>
<th>LOF</th>
<th>LIF</th>
<th>LRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term related faults</td>
<td>ENF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>LNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>TNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>TNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>TOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>TOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>DORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>DORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>CORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>CORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>LNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>LOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LIF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>LIF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LRF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>LRF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

When a double fault is committed in an expression, it is also possible for the two resulting faulty expressions to be equivalent to each other, but not to the original expression. For example, consider the expression $ab + cd + ef$. If the literal ‘e’ is inserted into the first term ‘$ab$’ before the literal ‘c’ in the second term is replaced by another literal ‘f’, the resulting faulty expression is equivalent to $abe + df + ef$. On the other hand, if these two faults are committed in the reverse order (that is, ‘c’ in the second term ‘$cd$’ is replaced by ‘f’ before ‘e’ is inserted into the first term ‘$ab$’), the resulting faulty expression is still equivalent to $abe + df + ef$. This kind of double fault is referred to as double fault without ordering. In other words, the order of occurrences of these two single faults will not affect the final result. In this case, there is no need to distinguish between $F_1 \Join F_2$ and $F_2 \Join F_1$. Table 3.1 shows all double fault classes without ordering. There are 15, 20 and 10 types of double fault classes related to terms only, double fault classes with one term fault and one literal fault, and double fault classes related to literals only, respectively.

On the other hand, it is possible that the order of occurrences of the two individual faults may result in non-equivalent faulty expressions. For example, the first term ‘$ab$’ of $ab + cd + ef$ is negated before the literal ‘e’ is inserted into it, then the resulting faulty expression is $\overline{abe} + cd + ef$. On the other hand, if the insertion of the literal ‘e’ is committed before the negation of the first term, the resulting faulty expression is $\overline{abe} + cd + ef$ which is not equivalent to the previous faulty expression, $\overline{abe} + cd + ef$. This kind of double fault is referred to as double fault with ordering. In such a case, we need to study both $F_1 \Join F_2$ and $F_2 \Join F_1$. Table 3.2 shows all double fault classes with ordering.
fault classes with ordering. As shown in the table, there are 25, 40 and 16 types of double fault classes related to terms only, double fault classes with one term fault and one literal fault, and double fault classes related to literals only, respectively.

Table 3.2: Types of double faults with ordering

<table>
<thead>
<tr>
<th>Term related faults</th>
<th>Literal related faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF</td>
<td>LNF</td>
</tr>
<tr>
<td>TNF</td>
<td>LOF</td>
</tr>
<tr>
<td>TOF</td>
<td>LIF</td>
</tr>
<tr>
<td>DORF</td>
<td>LRF</td>
</tr>
</tbody>
</table>

In previous work on double fault experimentation [42], double fault mutants have been generated using the following steps: (1) single fault mutants are generated first by making one syntactic change to the original program, (2) those single fault mutants that are equivalent to the original program are then discarded, and (3) double fault mutants are then generated by making one syntactic change to the remaining non-equivalent single fault mutants. However, during our analysis of double fault without ordering, we have the following interesting observation. Given an original program, it is possible to have a double fault mutant such that the corresponding two single fault mutants (which are separately caused by the two individual single faults) are both equivalent to the original program. This is illustrated in Example 3.1.1.

**Example 3.1.1** Let $S = a + b + cd$. If the literal $\bar{b}$ is inserted in the first term $a$ in $S$, the resulting expression $\bar{a}b + b + cd$, denoted as $I_1''$, is equivalent to $S$. Similarly, if the literal $\bar{a}$ is inserted in the second term $b$ in $S$, the resulting expression $a + \bar{a}b + cd$, denoted as $I_2''$, is equivalent to $S$. However, when the above two faults are committed on $S$, the resulting expression $\bar{a}b + \bar{a}b + cd$, denoted by $I''$, is not equivalent to $S$ because $S$ and $I''$ evaluate to 1 and 0 on 1100($a = b = 1, c = d = 0$), respectively.

In other words, by discarding those single fault mutants that are equivalent to the original expression, some double faults mutants that are not equivalent to the original expression could not be generated. Hence, we suggest, when generating double-fault expressions, to keep all expressions with one syntactic difference with $S$ whether it is equivalent to $S$ or not.
Chapter 4

Double Faults Related to Terms Only

In this chapter, double faults related to terms only are studied. A double fault related to terms only is a double fault in which two individual faults of the double fault are term faults within Boolean expressions. For ease of reference, we used double term faults instead. As a reminder, five term fault classes are considered in this chapter, they are ENF, TNF, TOF, DORF and CORF. Since the ordering of the occurrences of two single faults in a double fault may result in different faulty expressions, double term faults are studied from two cases, double fault with and without ordering. Double faults without ordering refers to the situation that two single faults involved in a double fault are independent of each other while double faults with ordering refers to the situation that two single faults occur one after the other in such a way that the first fault may affect the occurrence of the second.

4.1 Double Faults without Ordering

In this section, different types of double term faults without ordering and their corresponding double-fault expressions are introduced. Given a Boolean expression $S$, suppose that two single fault classes $F_1$ and $F_2$ are committed in $S$ changing its subexpressions $E_1$ and $E_2$ to $E'_1$ and $E'_2$, respectively, the resulting double-fault implementation is denoted by $I_{F_1(E_1 \rightarrow E'_1), F_2(E_2 \rightarrow E'_2)}$ because their order of occurrence will result in the same faulty implementation. For example, let $S$ be $abc + cd + ef$. When TNF occurs at the first term $abc$ of $S$ and CORF occurs at the · operator between $c$ and $d$ of the second term $cd$, the corresponding expression is $I_{TNF(abc \rightarrow \overline{abc}), CORF(cd \rightarrow c + d)}$ which is equivalent to $\overline{abc} + c + d + ef$. Table 4.1 lists all 15 types of double faults without ordering for five single fault classes related to terms. In the rest of this section, these 15 double fault classes and their corresponding faulty implementations are discussed.
Table 4.1: Types of double term faults without ordering

<table>
<thead>
<tr>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.1 ENF with Other Term Faults

**ENF and ENF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose two subexpressions $p_1 + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ are negated. We use $I_{ENF(p_1 + \cdots + p_{h_1}, ENF(p_{i_2} + \cdots + p_{h_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The two subexpressions $p_1 + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ are mutually exclusive, that is $\{i_1, \ldots, h_1\} \cap \{i_2, \ldots, h_2\} = \emptyset$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m \quad (4.1)$$

**Case 2.** The two subexpressions $p_1 + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ are such that one subexpression contains another, but they are not equal. Without loss of generality, we can assume $\{i_2, \ldots, h_2\} \subseteq \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1} + \cdots + p_m \quad (4.2)$$

We do not consider the following two situations. First, the two subexpressions $p_1 + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ are exactly the same, that is $\{i_1, \ldots, h_1\} = \{i_2, \ldots, h_2\}$, because the implementation is equivalent to $S$. Second, the two subexpressions $p_1 + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ have some common terms, but one does not contain the other. For example, if $S = ab + cd + ef + gh$, we do not consider the situation that the subexpressions $ab + cd$ and $cd + ef$ can both be negated.

**ENF and TNF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the subexpression $p_1 + \cdots + p_{h_1}$ and the term $p_{i_2}$ are negated. We use $I_{ENF(p_1 + \cdots + p_{h_1}, TNF(p_{i_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:
Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is then equivalent to the following expression
\[
p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + \overline{p_{i_2}} + \cdots + p_m \tag{4.3}
\]

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression
\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \tag{4.4}
\]

**ENF and TOF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the subexpression $p_{i_1} + \cdots + p_{h_1}$ is negated and the term $p_{i_2}$ is omitted. We use $I_{ENF(p_{i_1}+\cdots+p_{h_1}-p_{i_2}=0),TOF(p_{i_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is then equivalent to the following expression
\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \tag{4.5}
\]

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression
\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + \overline{p_{h_1}} + \cdots + p_m \tag{4.6}
\]

**ENF and DORF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the subexpression $p_{i_1} + \cdots + p_{h_1}$ is negated and the subexpression $p_{i_2} + p_{i_2+1}$ is implemented as $p_{i_2}p_{i_2+1}$. We use $I_{ENF(p_{i_1}+\cdots+p_{h_1}-p_{i_2}-p_{i_2}=0),DORF(p_{i_2}p_{i_2+1})}$ to denote the corresponding faulty implementation, which can be further classified into the following four cases:

Case 1. The two subexpressions $p_{i_1} + \cdots + p_{h_1}$ and $p_{i_2} + p_{i_2+1}$ are mutually exclusive, that is $\{i_1, \ldots, h_1\} \cap \{i_2, i_2 + 1\} = \emptyset$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is then equivalent to the following expression
\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \tag{4.7}
\]

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains exactly one of the two terms $p_{i_2}$ and $p_{i_2+1}$, that is either $h_1 = i_2$ or $i_2 + 1 = i_1$. Without loss of generality,
we can assume the latter case, that is \( i_2 = h_1 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} \cdot p_{h_1+1} + \cdots + p_m \quad (4.8)
\]

Case 3. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the subexpression \( p_{i_2} + p_{i_2+1} \), but they are not equal, that is \( \{i_2, i_2 + 1\} \nsubseteq \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2} \cdot p_{i_2+1} + \cdots + p_{h_1} + \cdots + p_m \quad (4.9)
\]

Case 4. The two subexpressions \( p_{i_1} + \cdots + p_{h_1} \) and \( p_{i_2} + p_{i_2+1} \) are exactly same, that is \( \{i_2, i_2 + 1\} = \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} \cdot p_{i_1+1} + \cdots + p_m \quad (4.10)
\]

**ENF and CORF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose the subexpression \( p_{i_1} + \cdots + p_{h_1} \) is wrongly negated and the term \( p_{i_2} \) is wrongly implemented as \( p_{i_2} = p_{i_2,1,j_2} + p_{i_2, j_2+1, k_{i_2}} \), where \( p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2, j_2+1, k_{i_2}} \). We use \( I_{ENF(p_{i_1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{h_1}), CORF(p_{i_2} - p_{i_2},1,j_2 + p_{i_2, j_2+1, k_{i_2}})} \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \notin \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( h_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{h_1} + \cdots + p_{i_2,1,j_2} + p_{i_2, j_2+1, k_{i_2}} + \cdots + p_m \quad (4.11)
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2,1,j_2} + p_{i_2, j_2+1, k_{i_2}} + \cdots + p_{h_1} + \cdots + p_m \quad (4.12)
\]

### 4.1.2 TNF with Other Term Faults

**TNF and TNF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose two different terms \( p_{i_1} \) and \( p_{i_2} \) are negated. We use \( I_{TNF(p_{i_1} - p_{i_1}), TNF(p_{i_2} - p_{i_2})} \) to denote the corresponding faulty implementation. We do not consider the situation where the same term is negated twice (that is, \( i_1 = i_2 \)) because the implementation is then equivalent to the original expression \( S \). Without loss of generality, we can
assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_m$$

**TNF and TOF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the term $p_{i_1}$ is wrongly negated and the term $p_{i_2}$ is wrongly omitted. We use $I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}), \text{TOF}(p_{i_2} \rightarrow \overline{p_{i_2}})$ to denote the corresponding faulty implementation. We do not consider the situation where both TNF and TOF occur at the same term (that is $i_1 = i_2$). Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m$$

**TNF and DORF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the term $p_{i_1}$ is negated and the subexpression $p_{i_2} + p_{i_2+1}$ is implemented as $p_{i_2}p_{i_2+1}$. We use $I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}), \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow \overline{p_{i_2}p_{i_2+1}})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The subexpression $p_{i_2} + p_{i_2+1}$ does not contain the term $p_{i_1}$, that is $i_1 \notin \{i_2, i_2+1\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m$$

**Case 2.** The subexpression $p_{i_2} + p_{i_2+1}$ contains the term $p_{i_1}$, that is $i_1 \in \{i_2, i_2+1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 = i_2 + 1$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}}p_{i_1+1} + \cdots + p_m$$

**TNF and CORF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the term $p_{i_1}$ is negated and the term $p_{i_2}$ is implemented as $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}$, where $p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}}$. We use $I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}), \text{CORF}(p_{i_2} \rightarrow \overline{p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_m$$

23
Case 2. The terms \( p_{i_1} \) and \( p_{i_2} \) are actually the same term, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_i} + \cdots + p_m \tag{4.18}
\]

### 4.1.3 TOF with Other Term Faults

**TOF and TOF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose two different terms \( p_{i_1} \) and \( p_{i_2} \) are omitted. We use \( I_{\text{TOF}(p_{i_1} \rightarrow),\text{TOF}(p_{i_2} \rightarrow)} \) to denote the corresponding faulty implementation. We do not consider the situation where the same term is omitted twice (that is, \( i_1 = i_2 \)) because the implementation is then equivalent to the term being omitted which is a single TOF. Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \tag{4.19}
\]

**TOF and DORF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose the term \( p_{i_1} \) is omitted and the subexpression \( p_{i_2} + p_{i_2+1} \) is implemented as \( p_{i_2} p_{i_2+1} \). We use \( I_{\text{TOF}(p_{i_1} \rightarrow),\text{DORF}(p_{i_2}+p_{i_2+1} \rightarrow p_{i_2} p_{i_2+1})} \) to denote the corresponding faulty implementation. We do not consider the situation where the subexpression contains the omitted term (that is, \( i_1 = i_2 \) or \( i_1 = i_2 + 1 \)) because the implementation is then equivalent to the term being omitted which is a single TOF. Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2} p_{i_2+1} + \cdots + p_m \tag{4.20}
\]

**TOF and CORF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose the term \( p_{i_1} \) is omitted and the term \( p_{i_2} \) is implemented as \( p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \), where

\[
p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}}
\]

We use \( I_{\text{TOF}(p_{i_1} \rightarrow),\text{CORF}(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})} \) to denote the corresponding faulty implementation. We do not consider the situation where both TOF and CORF occur at the same term (that is, \( i_1 = i_2 \)) because the resulting expressions are not the same. Hence, it should be discussed in Section 4.2 when we consider double faults with ordering. Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_m \tag{4.21}
\]
4.1.4 DORF with Other Term Faults

DORF and DORF Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF.
Suppose two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$ are implemented as $p_{i_1}p_{i_1+1}$
and $p_{i_2}p_{i_2+1}$, respectively. We use $I_{DORF}(p_{i_1}+p_{i_1+1} \rightarrow p_{i_1}p_{i_1+1}), DORF(p_{i_2}+p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})$
to denote the corresponding faulty implementation, which can be further classified into
the following two cases:

Case 1. The two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$ are mutually exclusive,
that is $\{i_1, i_1 + 1\} \cap \{i_2, i_2 + 1\} = \emptyset$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (4.22)$$

Case 2. The two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$ have exactly one term
in common, that is $\{i_1, i_1 + 1\} \cap \{i_2, i_2 + 1\} = A$ and there is only one
element in set $A$. Hence, there are two possible cases, namely $i_1 + 1 = i_2$
and $i_2 + 1 = i_1$. Without loss of generality, we can assume $i_1 + 1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1}p_{i_1+2} + \cdots + p_m \quad (4.23)$$

We do not consider the situation where two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$
are exactly the same, because the implementation is equivalent to a single
DORF.

DORF and CORF Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF.
Suppose the subexpression $p_{i_1} + p_{i_1+1}$ of $S$ is implemented as $p_{i_1}p_{i_1+1}$ and the term
$p_{i_2}$ of $S$ is implemented as $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}$, where $p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}}$. We use
$I_{DORF}(p_{i_1}+p_{i_1+1} \rightarrow p_{i_1}p_{i_1+1}), CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})$
to denote the corresponding faulty implementation, which can be further classified into
the following two cases:

Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$, that is $i_2 \notin \{i_1, i_1 + 1\}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_m \quad (4.24)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$.
Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without
loss of generality, we can assume $i_1 = i_2$. The implementation is then
equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_1}p_{i_1+1} + \cdots + p_m \]  

\[ (4.25) \]

### 4.1.5 CORF with Other Term Faults

**CORF and CORF**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose two terms \( p_{i_1} \) and \( p_{i_2} \) are implemented as \( p_{i_1,1,j_1} + p_{i_1,1+1,k_1} \) and \( p_{i_2,1,j_2} + p_{i_2,2+1,k_2} \), respectively, where \( p_{i_1} = p_{i_1,1,j_1}p_{i_1,1+1,k_1} \) and \( p_{i_2} = p_{i_2,1,j_2}p_{i_2,2+1,k_2} \). We use \( I_{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1}p_{i_1,1+1,k_1}, CORF(p_{i_2} \rightarrow p_{i_2,1,j_2}p_{i_2,2+1,k_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The term \( p_{i_1} \) and the term \( p_{i_2} \) are different terms, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,1+1,k_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,2+1,k_2} + \cdots + p_m \]  

\[ (4.26) \]

**Case 2.** The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). We do not consider the situation where the split positions are same (that is, \( j_1 = j_2 \)) because the implementation is then equivalent to a single CORF. Without loss of generality, we can assume \( j_1 < j_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,1+1,j_1} + p_{i_1,2+1,k_1} + \cdots + p_m \]  

\[ (4.27) \]

In summary, there are altogether 27 different double fault expressions among the 15 double fault classes without ordering considered in this section.

### 4.2 Double Faults with Ordering

In this section, different types of double term faults with ordering and their corresponding double-fault expressions are discussed. As previously stated, double faults with ordering is such that two single faults occur one after the other in such a way that the occurrence of the first fault may affect the occurrence of the second fault.

Given a Boolean expression \( S \), suppose that two single fault classes \( F_1 \) and \( F_2 \) are committed in \( S \) changing the subexpressions \( E_1 \) and \( E_2 \) in \( S \) to \( E_1' \) and \( E_2' \), respectively. Suppose that \( F_1 \) is committed before \( F_2 \), the resulting double-fault expression is denoted by \( I_{F_1(E_1 \rightarrow E_1')} \otimes F_2(E_2 \rightarrow E_2') \). For example, let \( S \) be \( abc + cd + ef \). When TNF occurs at the first term \( abc \) of \( S \) and then CORF occurs at the ‘+’ operator
between $a$ and $b$, the corresponding expression is $I_{TNF}(abc \to \overline{abc}) \otimes CORF(\overline{abc} \to a + bc)$ which is equivalent to $a + bc + cd + ef$. A different notation than the one defined in Section 4.1 is used in this section because the order of occurrences of the two single faults may result in different faulty expressions. As shown in Table 4.2, for the five single fault classes related to terms, there are altogether 25 different double fault classes with ordering. In the rest of this section, these double fault classes and corresponding double-fault expressions are discussed.

Table 4.2: Types of double term faults with ordering

<table>
<thead>
<tr>
<th></th>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>TNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>TOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>DORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>CORF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

4.2.1 ENF First, then Other Term Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose an ENF is committed first by negating the subexpression $p_{i_1} + \cdots + p_{h_1}$. The corresponding faulty implementation is

$$I_{ENF}(p_{i_1} + \cdots + p_{h_1} \to p_{i_1} + \cdots + p_{h_1}) = p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_m$$

ENF and ENF After the first ENF is made on $S$, another subexpression $p_{i_2} + \cdots + p_{h_2}$ is then negated. Let $I_{ENF}(p_{i_1} + \cdots + p_{h_1} \to p_{i_1} + \cdots + p_{h_1}) \otimes ENF(p_{i_2} + \cdots + p_{h_2} \to \overline{p_{i_2}} + \cdots + \overline{p_{h_2}})$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The two subexpressions $p_{i_1} + \cdots + p_{h_1}$ and $p_{i_2} + \cdots + p_{h_2}$ are mutually exclusive, that is $\{i_1, \ldots, h_1\} \cap \{i_2, \ldots, h_2\} = \emptyset$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{h_2} + \cdots + p_m \quad (4.28)$$

Case 2. One subexpression is contained in another, but they are not the same.

(a) The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the subexpression $p_{i_2} + \cdots + p_{h_2}$. That is, $\{i_2, \ldots, h_2\} \subsetneq \{i_1, \ldots, h_1\}$. The implementation is then equivalent to

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{h_2} + \cdots + p_{h_1} + \cdots + p_m \quad (4.29a)$$
The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains the subexpression $p_{i_1} + \cdots + p_{h_1}$. That is, $\{i_1, \ldots, h_1\} \subsetneq \{i_2, \ldots, h_2\}$. The implementation is then equivalent to
\begin{equation}
p_1 + \cdots + p_{i_2} + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_{h_2} + \cdots + p_m \tag{4.29b}
\end{equation}

As discussed in Section 4.1.1, we do not consider the situations where (1) the two subexpressions are exactly same, and (2) the two subexpressions have some common terms, but one does not contain the other.

**ENF and TNF** After the ENF is made on $S$, the $p_{i_2}$ term is then negated. Let $I_{ENF}(p_1 + \cdots + p_{i_1} \overline{p_{i_1}} + \cdots + p_{h_1}) \otimes TNF(p_{i_2} \overline{p_{i_2}})$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is then equivalent to the following expression
\begin{equation}
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_m \tag{4.30}
\end{equation}

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression
\begin{equation}
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{i_1} + \cdots + p_m \tag{4.31}
\end{equation}

**ENF and TOF** After the ENF is made on $S$, the $p_{i_2}$ term is then omitted. Let $I_{ENF}(p_1 + \cdots + p_{i_1} \overline{p_{i_1}} + \cdots + p_{h_1}) \otimes TOF(p_{i_2})$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is equivalent to the following expression
\begin{equation}
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{i_2} - 1 + p_{i_2} + 1 + \cdots + p_m \tag{4.32}
\end{equation}

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is equivalent to the following expression
\begin{equation}
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{i_1} + \cdots + p_m \tag{4.33}
\end{equation}

**ENF and DORF** After the ENF is made on $S$, another subexpression $p_{i_2} + p_{i_2+1}$ is then wrongly implemented as $p_{i_2}p_{i_2+1}$. Let
Case 1. The two subexpressions $p_{i_1} + \cdots + p_{h_1}$ and $p_{i_2} + p_{i_2+1}$ are mutually exclusive, that is $\{i_1, \ldots, h_1\} \cap \{i_2, i_2 + 1\} = \emptyset$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (4.34)$$

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains either the term $p_{i_2}$ or the term $p_{i_2+1}$. Without loss of generality, we can assume that the subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$ (that is, $i_2 = h_1$). The implementation is then equivalent to

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1}p_{h_1+1} + \cdots + p_m \quad (4.35)$$

Case 3. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the subexpression $p_{i_2} + p_{i_2+1}$, but they are not same, that is $\{i_2, i_2 + 1\} \subseteq \{i_1, \ldots, h_1\}$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_{h_1} + \cdots + p_m \quad (4.36)$$

Case 4. The two subexpressions $p_{i_1} + \cdots + p_{h_1}$ and $p_{i_2} + p_{i_2+1}$ are exactly same, that is $\{i_2, i_2 + 1\} = \{i_1, \ldots, h_1\}$. Hence, $i_1 = i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_m \quad (4.37)$$

**ENF and CORF** After the ENF is made on $S$, the term $p_{i_2}$ is then implemented as $p_{i_2,1, j_2} + p_{i_2, j_2+1, k_{i_2}}$, where $p_{i_2} = p_{i_2,1, j_2} \cdot p_{i_2, j_2+1, k_{i_2}}$. Let $I_{ENF(p_{i_1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{h_1}) \circ CORF(p_{i_2} - p_{i_2,1, j_2} + p_{i_2, j_2+1, k_{i_2}})}$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \notin \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $h_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2,1, j_2} + p_{i_2, j_2+1, k_{i_2}} + \cdots + p_m \quad (4.38)$$

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$.
The implementation is equivalent to the following expression

\[ p_1 + \cdots + p_{i_1} + \cdots + p_{i_2,j_2} + p_{i_2,j_2+1,k_2} + \cdots + p_{i_1} + \cdots + p_m \]  

(4.39)

### 4.2.2 TNF First, then Other Term Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose a TNF is committed first by negating the term \( p_{i_1} \). The corresponding faulty expression is

\[ I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}) = p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_m. \]

**TNF and ENF**  After the TNF is made on \( S \), the subexpression \( p_{i_2} + \cdots + p_{h_2} \) is then negated. Let \( I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}) \otimes I_{\text{ENF}}(p_{i_2} + \cdots + p_{h_2} \rightarrow \overline{p_{i_2} + \cdots + p_{h_2}}) \) be the corresponding faulty implementation. We have the following two cases:

**Case 1.** The subexpression \( p_{i_2} + \cdots + p_{h_2} \) does not contain the term \( p_{i_1} \), that is \( i_1 \not\in \{ i_2, \ldots, h_2 \} \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is equivalent to the following expression

\[ p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_{h_2} + \cdots + p_m \]  

(4.40)

**Case 2.** The subexpression \( p_{i_2} + \cdots + p_{h_2} \) contains the term \( p_{i_1} \), that is \( i_1 \in \{ i_2, \ldots, h_2 \} \). The implementation is equivalent to the following expression

\[ p_1 + \cdots + \overline{p_{i_2}} + \cdots + \overline{p_{i_1}} + \cdots + p_{h_2} + \cdots + p_m \]  

(4.41)

**TNF and TNF**  After the first TNF is made on \( S \), the \( i_2 \)-th term \( p_{i_2} \) is then negated. Let \( I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}) \otimes I_{\text{TNF}}(p_{i_2} \rightarrow \overline{p_{i_2}}) \) be the corresponding faulty implementation. As discussed in Section 4.1.2, we do not consider the situation that the two terms are exactly the same (that is, \( i_1 = i_2 \)). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2}} + \cdots + p_m \]  

(4.42)

**TNF and TOF**  After the TNF is made on \( S \), the \( i_2 \)-th term \( p_{i_2} \) is then omitted. Let \( I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p_{i_1}}) \otimes I_{\text{TOF}}(p_{i_2} \rightarrow) \) be the corresponding faulty implementation. We do not consider the situation where both TNF and TOF occur at the same term (that is, \( i_1 = i_2 \)). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is equivalent to the following expression

\[ p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \]  

(4.43)
**TNF and DORF** After the TNF is made on $S$, the subexpression $p_{i_2} + p_{i_2+1}$ is then wrongly implemented as $p_{i_2}p_{i_2+1}$. Let $I_{TNF(p_{i_1} \rightarrow \overline{p_{i_1}}) \circ DORF(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})}$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_2} + p_{i_2+1}$ does not contain the term $p_{i_1}$, that is $i_1 \not\in \{i_2, i_2 + 1\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (4.44)$$

Case 2. The subexpression $p_{i_2} + p_{i_2+1}$ contains the term $p_{i_1}$, that is $i_1 \in \{i_2, i_2 + 1\}$. Without loss of generality, we can assume $i_1 = i_2$, the implementation is equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}}p_{i_1+1} + \cdots + p_m \quad (4.45)$$

**TNF and CORF** After the TNF is made on $S$, the term $p_{i_2}$ is then implemented as $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}$, where $p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_2}$. Let $I_{TNF(p_{i_1} \rightarrow \overline{p_{i_1}}) \circ CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2})}$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2} + \cdots + p_m \quad (4.46)$$

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are the same, that is $i_1 = i_2$. The implementation is equivalent to the following expression $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}$

$$p_1 + \cdots + \overline{p_{i_1,1,j_1}} + \overline{p_{i_1,j_1+1,k_1}} + \cdots + p_m \quad (4.47)$$

**4.2.3 TOF First, then Other Term Faults**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose a TOF is committed first by omitting the term $p_{i_1}$. The corresponding faulty expression is $I_{TOF(p_{i_1} \rightarrow \cdot)} = p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_m$.

**TOF and ENF** After the TOF is made on $S$, the subexpression $p_{i_2} + \cdots + p_{i_2}$ is then negated. Let $I_{TOF(p_{i_1} \rightarrow \cdot) \circ ENF(p_{i_2} + \cdots + p_{i_2} \rightarrow \overline{p_{i_2} + \cdots + p_{i_2}})}$ be the corresponding faulty implementation. We have the following two cases:
Case 1. The subexpression $p_{i_2} + \cdots + p_{h_2}$ does not contain the term $p_{i_1}$, that is $i_1 \notin \{i_2, \ldots, h_2\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m \quad (4.48)$$

Case 2. The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains the term $p_{i_1}$, that is $i_1 \in \{i_2, \ldots, h_2\}$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m \quad (4.49)$$

**TOF and TNF** After the TOF is made on $S$, the $i_2$-th term $p_{i_2}$ is then negated. Let $I_{TOF(p_{1} \to) \otimes TNF(p_{2} \to p_{i_2})}$ be the corresponding faulty implementation. We do not consider the situation where the same term is omitted and then negated (that is, $i_1 = i_2$). Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + \overline{p_{i_2}} + \cdots + p_m \quad (4.50)$$

**TOF and TOF** After the first TOF is made on $S$, the $i_2$-th term $p_{i_2}$ is then omitted. Let $I_{TOF(p_{1} \to) \otimes TOF(p_{2} \to -)}$ be the corresponding faulty implementation. We do not consider the situation where the same term is omitted first and then omitted again. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \quad (4.51)$$

**TOF and DORF** After the TOF is made on $S$, the subexpression $p_{i_2} + p_{i_2+1}$ is then wrongly implemented as $p_{i_2}p_{i_2+1}$. Let $I_{TOF(p_{1} \to) \otimes DORF(p_{i_2}p_{i_2+1} \to -)}$ be the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_2} + p_{i_2+1}$ involves any subexpression of two consecutive terms other than $p_{i_1-1} + p_{i_1+1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (4.52)$$

Case 2. The subexpression $p_{i_1-1} + p_{i_1+1}$ is implemented as $p_{i_1-1}p_{i_1+1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1}p_{i_1+1} + \cdots + p_m \quad (4.53)$$
TOF and CORF  After the TOF is made on \( S \), the term \( p_{i_2} \) is wrongly implemented as \( p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2} \), where \( p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_2} \). Let \( I_{\text{TOF}}(p_{i_1} \rightarrow \text{CORF}(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) \) be the corresponding faulty implementation. We do not consider the situation where a term is omitted first and then a CORF is committed at the same term (that is, \( i_1 = i_2 \)). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2} + \cdots + p_m \tag{4.54}
\]

4.2.4 DORF First, then Other Term Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose a DORF is committed first by concatenating two terms \( p_{i_1} \) and \( p_{i_1+1} \) using the ‘+’ operator. The corresponding faulty expression is \( I_{\text{DORF}}(p_{i_1} + p_{i_1+1} \rightarrow p_{i_1}p_{i_1+1}) = p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_m \).

DORF and ENF  After the DORF is made on \( S \), the subexpression \( p_{i_2} + \cdots + p_{h_2} \) is then negated. Let \( I_{\text{DORF}}(p_{i_1} + p_{i_1+1} \rightarrow p_{i_1}p_{i_1+1}) \circ \text{ENF}(p_{i_2} + \cdots + p_{h_2} - \overline{p_{i_2} + \cdots + p_{h_2}}) \) be the corresponding faulty implementation. We have the following four cases:

Case 1. The subexpressions \( p_{i_2} + \cdots + p_{h_2} \) and \( p_{i_1} + p_{i_1+1} \) are mutually exclusive, that is \( \{i_1, i_1 + 1\} \cap \{i_2, \ldots, h_2\} = \emptyset \). Without loss of generality, we can assume \( i_1 + 1 < i_2 \). The implementation is equivalent to the following expression

\[
p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + \overline{p_{i_2} + \cdots + p_{h_2}} + \cdots + p_m \tag{4.55}
\]

Case 2. The subexpression \( p_{i_2} + \cdots + p_{h_2} \) contains either \( p_{i_1} \) or \( p_{i_1+1} \). Without loss of generality, we can assume the subexpression \( p_{i_2} + \cdots + p_{h_2} \) contains \( p_{i_1} \), that is \( i_1 = h_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + \overline{p_{i_2} + \cdots + p_{h_2}}p_{h_2+1} + \cdots + p_m \tag{4.56}
\]

Case 3. The subexpression \( p_{i_2} + \cdots + p_{h_2} \) contains the subexpression \( p_{i_1} + p_{i_1+1} \), but they are not same, that is \( \{i_1, i_1 + 1\} \subseteq \{i_2, \ldots, h_2\} \). The implementation is equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_2} + \cdots + p_{h_2}} + p_{i_1}p_{i_1+1} + \cdots + \overline{p_{h_2}} + \cdots + p_m \tag{4.57}
\]

Case 4. The two subexpressions \( p_{i_2} + \cdots + p_{h_2} \) and \( p_{i_1} + p_{i_1+1} \) are exactly same, that is \( \{i_1, i_1 + 1\} = \{i_2, \ldots, h_2\} \). The implementation is equivalent to the following expression...
following expression
\[ p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_m \] \hspace{1cm} (4.58)

**DORF and TNF** After the DORF is made on \( S \), the \( i_2 \)-th term \( p_{i_2} \) is then negated. Let \( I_{DORF(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})\otimes TNF(p_{i_2}-p_{i_2})} \) be the corresponding faulty implementation. We have the following three cases:

Case 1. The subexpression \( p_{i_1} + p_{i_1+1} \) does not contain the term \( p_{i_2} \). Without loss of generality, we can assume \( i_1 + 1 < i_2 \). The implementation is equivalent to the following expression
\[ p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2} + \cdots + p_m \] \hspace{1cm} (4.59)

Case 2. The subexpression \( p_{i_1} + p_{i_1+1} \) contains the term \( p_{i_2} \). Without loss of generality, we can assume \( i_2 = i_1 \) following expression
\[ p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_m \] \hspace{1cm} (4.60)

Case 3. The term \( p_{i_2} \) is the newly created term \( p_{i_1}p_{i_1+1} \). The implementation is then equivalent to the following expression
\[ p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_m \] \hspace{1cm} (4.61)

**DORF and TOF** After the DORF is made on \( S \), the \( i_2 \)-th term \( p_{i_2} \) is then omitted. Let \( I_{DORF(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})\otimes TOF(p_{i_2})} \) be the corresponding faulty implementation. We do not consider the two situations where the term \( p_{i_2} \) is either \( p_{i_1} \) or \( p_{i_1+1} \) because the net effect is a single TOF. Therefore, we have the following two cases:

Case 1. The subexpression \( p_{i_1} + p_{i_1+1} \) does not contain the term \( p_{i_2} \). Without loss of generality, we can assume \( i_1 + 1 < i_2 \). The implementation is equivalent to the following expression
\[ p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \] \hspace{1cm} (4.62)

Case 2. The term \( p_{i_2} \) is the newly created term \( p_{i_1}p_{i_1+1} \). The implementation is then equivalent to the following expression
\[ p_1 + \cdots + p_{i_1-1} + p_{i_1+2} + \cdots + p_m \] \hspace{1cm} (4.63)
**DORF and DORF** After the first DORF is made on $S$, another subexpression $p_{i_2} + p_{i_2+1}$ is then implemented as $p_{i_2}p_{i_2+1}$. Let $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1}p_{i_1+1}) \otimes DORF(p_{i_2} + p_{i_2+1} - p_{i_2}p_{i_2+1})$ be the corresponding faulty implementation. We do not consider the situation where the two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$ are exactly the same because the implementation is equivalent to a single DORF. Therefore, we have the following two cases:

Case 1. The two subexpressions $p_{i_1} + p_{i_1+1}$ and $p_{i_2} + p_{i_2+1}$ are mutually exclusive, that is $\{i_1, i_1+1\} \cap \{i_2, i_2+1\} = \emptyset$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (4.64)$$

Case 2. Either the term $p_{i_2}$ or the term $p_{i_2+1}$ is the newly created term $p_{i_1}p_{i_1+1}$. Without loss of generality, we can assume the term $p_{i_2}$ is the newly created term $p_{i_1}p_{i_1+1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1}p_{i_1+2} + \cdots + p_m \quad (4.65)$$

**DORF and CORF** After the DORF is made on $S$, the term $p_{i_2}$ is implemented as $p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_2}$, where $p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,2,j_2+1,k_2}$. Let $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1}p_{i_1+1}) \otimes CORF(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_2})$ be the corresponding faulty implementation. We do not consider the situations where the newly created term $p_{i_1}p_{i_1+1}$ is splitted into $p_{i_1} + p_{i_1+1}$ because the resulting expression is equivalent to the original expression. Therefore, we have the following two cases:

Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$, that is $i_2 \notin \{i_1, i_1+1\}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_2} + \cdots + p_m \quad (4.66)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$. Without loss of generality, we can assume $i_2 = i_1$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_2} + p_{i_1,2,j_2+1,k_1} : p_{i_1+1} + \cdots + p_m \quad (4.67)$$

**4.2.5 CORF First, then Other Term Faults**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose a CORF is committed first by splitting the term $p_{i_1}$ into $p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_1}$, where $p_{i_1} = p_{i_1,1,j_1} \cdot p_{i_1,1,j_1+1,k_1}$.
and $1 \leq j_1 < k_{i_1}$. The corresponding faulty expression is $I_{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}) = p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m$.

**CORF and ENF** After the CORF is made on $S$, the subexpression $p_{i_2} + \cdots + p_{h_2}$ is then negated. Let $I_{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \circ ENF(p_{i_2} + \cdots + p_{h_2} \rightarrow \overline{p_{i_2}} + \cdots + \overline{p_{h_2}})$ be the corresponding faulty implementation. We have the following four cases:

- **Case 1.** The subexpression $p_{i_2} + \cdots + p_{h_2}$ does not contain $p_{i_1,1,j_1}$ and $p_{i_1,j_1+1,k_{i_1}}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is equivalent to the following expression

  $$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m \quad (4.68)$$

- **Case 2.** The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains $p_{i_1,1,j_1}$ and $p_{i_1,j_1+1,k_{i_1}}$. The implementation is equivalent to the following expression

  $$p_1 + \cdots + p_{i_2} + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_2} + \cdots + p_m \quad (4.69)$$

- **Case 3.** The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains exactly one of the two newly created term. Without loss of generality, we can assume that the subexpression $p_{i_2} + \cdots + p_{h_2}$ contains $p_{i_1,j_1+1,k_{i_1}}$. Hence, the term $p_{i_2}$ actually refers to $p_{i_1,j_1+1,k_{i_1}}$. The implementation is then equivalent to the following expression

  $$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_2} + \cdots + p_m \quad (4.70)$$

- **Case 4.** The subexpression $p_{i_2} + \cdots + p_{h_2}$ is exactly the newly created subexpression $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}$. That is, the newly created subexpression is then negated. The implementation is equivalent to the following expression

  $$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m \quad (4.71)$$

**CORF and TNF** After the CORF is made on $S$, the $i_2$-th term $p_{i_2}$ is then negated. Let $I_{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \circ TNF(p_{i_2} \rightarrow \overline{p_{i_2}})$ be the corresponding faulty implementation. We have the following three cases:

- **Case 1.** The term $p_{i_2}$ is neither $p_{i_1,1,j_1}$ nor $p_{i_1,j_1+1,k_{i_1}}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

  $$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{i_2} + \cdots + p_m \quad (4.72)$$
Case 2. The negated term is either $p_{i_1,1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume that the negated term is $p_{i_1,j_1+1,k_1}$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m$$

(4.73)

Case 3. The negated term is $p_{i_1}$. The implementation is equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m$$

(4.74)

**CORF and TOF** After the CORF is committed on $S$, the $i_2$-th term $p_{i_2}$ is then omitted. Let $I_{\text{CORF}}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) \otimes \text{TOF}(p_2 \rightarrow \cdot)$ be the corresponding faulty implementation. We do not consider the situation where $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}$ (originally belongs to the term $p_{i_1}$) is then omitted because the net effect is a single TOF. We have the following two cases:

Case 1. The term $p_{i_2}$ is neither $p_{i_1,1,j_1}$ nor $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m$$

(4.75)

Case 2. The term $p_{i_2}$ is either $p_{i_1,1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume that the omitted term is $p_{i_1,j_1+1,k_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1,1,j_1} + p_{i_1+1} + \cdots + p_m$$

(4.76)

**CORF and DORF** After the CORF is made on $S$, another subexpressions $p_{i_2} + p_{i_2+1}$ is wrongly implemented as $p_{i_2} p_{i_2+1}$. Let $I_{\text{CORF}}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) \otimes DORF(p_2 + p_{i_2+1} - p_2 p_{i_2+1})$ be the corresponding faulty implementation. We do not consider the situation where a DORF is committed on $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}$ because the implementation is equivalent to the original expression $S$. We have the following two cases:

Case 1. The subexpression $p_{i_2} + p_{i_2+1}$ does not contain $p_{i_1,1,j_1}$ and $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2} p_{i_2+1} + \cdots + p_m$$

(4.77)
Case 2. The subexpression $p_i + p_{i+1}$ contains either $p_{i1,j1}$ or $p_{i1,j1+1,k1}$. Without loss of generality, we can assume that the subexpression $p_i + p_{i+1}$ contains $p_{i1,j1+1,k1}$. Hence, the term $p_{i+1}$ actually refers to $p_{i1,j1+1,k1}$

The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i1,j1} + p_{i1,j1+1,k1} + \cdots + p_{i2,j2+1,k2} (4.78)$$

**CORF and CORF** After the CORF is made on $S$, the term $p_i$ is implemented as $p_{i1,j1} = p_{i1,j1} \cdot p_{i2,j2+1,k2}$. Let $I_{CORF}(p_i - p_{i1,j1} + p_{i1,j1+1,k1}) \otimes CORF(p_i - p_{i1,j1} + p_{i2,j2+1,k2})$ be the corresponding faulty implementation. We do not consider the situation where $i_1 = i_2$ and $j_1 = j_2$ because the implementation is then equivalent to a single CORF. We have the following two cases:

Case 1. The term $p_i$ is neither $p_{i1,j1}$ nor $p_{i1,j1+1,k1}$. Without loss the generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i1,j1} + p_{i1,j1+1,k1} + \cdots + p_{i2,j2+1,k2} + \cdots + p_m (4.79)$$

Case 2. The term $p_i$ is either $p_{i1,j1}$ or $p_{i1,j1+1,k1}$. Without loss the generality, we can assume that the term $p_i$ is $p_{i1,j1+1,k1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i1,j1} + p_{i1,j1+1,j2} + p_{i1,j2+1,k1} + \cdots + p_m (4.80)$$

In summary, there are altogether 53 double-fault expressions among the 25 double fault classes with ordering considered in this section. The next section shows that these 53 double-fault expressions can be reduced to 31 double-fault expressions.

### 4.3 Relation between Double Faults with and without Ordering on Faults Related to Terms Only

In this section, the relation of double faults with and without ordering are analysed. All the possible faulty expressions of double faults without ordering with respect to those with ordering in the same fault category related to ENF, TNF, TOF, DORF, and CORF are compared. Table 4.3 (respectively, 4.4, 4.5, 4.6 and 4.7) summarizes the situations of double faults with ENF (respectively TNF, TOF, DORF, and CORF) and other term faults. Each row in these tables shows those faulty expressions of a particular type of double fault without ordering and their counterparts in
double faults with ordering.

Table 4.3: Comparison of double-fault expressions of ENF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double-fault without ordering (Expression number)</th>
<th>Double fault with ordering - ENF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF and ENF</td>
<td>4.1, 4.2</td>
<td>4.28, 4.29a, 4.29b</td>
</tr>
<tr>
<td>ENF and TNF</td>
<td>4.3, 4.4</td>
<td>4.30, 4.31</td>
</tr>
<tr>
<td>ENF and TOF</td>
<td>4.5, 4.6</td>
<td>4.32, 4.33</td>
</tr>
<tr>
<td>ENF and DORF</td>
<td>4.7, 4.8, 4.9</td>
<td>4.34, 4.35, 4.36</td>
</tr>
<tr>
<td>ENF and CORF</td>
<td>4.11, 4.12</td>
<td>4.38, 4.39</td>
</tr>
</tbody>
</table>

Expressions 4.2 and 4.29b are dual to each other.

For Table 4.3, in the first row, there are two subcases for ENF and ENF without ordering which are given by Expressions (4.1) and (4.2) in Section 4.1. For the first subcase, Expressions (4.1) is equivalent to the Expression (4.28) which corresponds to the first subcase of ENF and ENF with ordering as discussed in Section 4.2. For the second subcase of ENF and ENF without ordering, the Expressions (4.2) and (4.29a) are equivalent. Moreover, it should be noted that the Expressions (4.2) and (4.29b) are dual to each other. Hence, they are considered to be equivalent. Other rows in Table 4.3 show the equivalent faulty expressions of double faults of ENF and other faults with and without ordering.

Table 4.4: Comparison of double-fault expressions of TNF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double-fault without ordering (Expression number)</th>
<th>Double fault with ordering - TNF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNF and ENF</td>
<td>4.3, 4.4</td>
<td>4.40, 4.41</td>
</tr>
<tr>
<td>TNF and TNF</td>
<td>4.13</td>
<td>4.42</td>
</tr>
<tr>
<td>TNF and TOF</td>
<td>4.14</td>
<td>4.43</td>
</tr>
<tr>
<td>TNF and DORF</td>
<td>4.15, 4.16</td>
<td>4.44, 4.45</td>
</tr>
<tr>
<td>TNF and CORF</td>
<td>4.17, 4.18</td>
<td>4.46, 4.47</td>
</tr>
</tbody>
</table>

In Table 4.4, the rows can also be interpreted in a similar manner as those in

\[ p_1 + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{i_1} + \cdots + p_m \]

(where \( \{i_2, \ldots, h_2\} \subseteq \{i_1, \ldots, h_1\} \)) and

\[ p_1 + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_m \]

(where \( \{i_1, \ldots, h_1\} \nsubseteq \{i_2, \ldots, h_2\} \)), respectively.

39
Table 4.3. For example, for the row of TNF and TNF, the Expression (4.13) is equivalent to the Expression (4.42).

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - TOF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOF and ENF</td>
<td>4.5, 4.6</td>
<td>4.48, 4.49</td>
</tr>
<tr>
<td>TOF and TNF</td>
<td>4.14</td>
<td>4.50</td>
</tr>
<tr>
<td>TOF and TOF</td>
<td>4.19</td>
<td>4.51</td>
</tr>
<tr>
<td>TOF and DORF</td>
<td>4.20, -</td>
<td>4.52, 4.53</td>
</tr>
<tr>
<td>TOF and CORF</td>
<td>4.21</td>
<td>4.54</td>
</tr>
</tbody>
</table>

For the rows in Table 4.5, there are two different situations. First, some of them can be interpreted in a similar manner as those in Tables 4.3 and 4.4. Second, for other rows, the expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class. For example, for the second subcase of TOF and DORF, the Expression (4.53) does not have its counterpart in TOF and DORF without ordering.

Table 4.6: Comparison of double-fault expressions of DORF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - DORF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DORF and ENF</td>
<td>4.7, 4.8, 4.9, 4.10</td>
<td>4.55, 4.56, 4.57, 4.58</td>
</tr>
<tr>
<td>DORF and TNF</td>
<td>4.15, 4.16, -</td>
<td>4.59, 4.60, 4.61 (4.10)</td>
</tr>
<tr>
<td>DORF and TOF</td>
<td>4.20, -</td>
<td>4.62, 4.63 (4.19)</td>
</tr>
<tr>
<td>DORF and DORF</td>
<td>4.22, 4.23</td>
<td>4.64, 4.65</td>
</tr>
<tr>
<td>DORF and CORF</td>
<td>4.24, 4.25</td>
<td>4.66, 4.67</td>
</tr>
</tbody>
</table>

\(^a\)When both \(p_{i_1}\) and \(p_{i_1+1}\) are omitted, Expression (4.19) is then equivalent to Expression (4.63).

Similarly, for the rows in Table 4.6, there are two different situations. First, some of them can be interpreted in a similar manner as those in Tables 4.3 and 4.4. For example, for DORF and DORF, the Expression (4.22) is equivalent to the Expression (4.64). Second, for some other rows, the faulty expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class, but they are equivalent to other faulty expressions.
of double faults without ordering in a different double fault class. For example, for the third subcase of DORF and TNF, the Expression (4.61) does not have its counterpart in DORF and TNF without ordering. However, it is equivalent to the Expression (4.10) in the double fault ENF and DORF without ordering. Another example is the second subcase of DORF and TOF with ordering. The Expression (4.63) is such that the subexpression \( p_{i+1} + p_{i+2} \) is wrongly committed as \( p_{i} \cdot p_{i+1} \) and then this newly created term is omitted. This is, in fact, equivalent to the Expression (4.19) where both terms \( p_{i} \) and \( p_{i+1} \) are omitted.

Table 4.7: Comparison of double-fault expressions of CORF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - CORF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORF and ENF</td>
<td>4.11 4.12 -</td>
<td>4.68 4.69 4.70 4.71 (4.18)</td>
</tr>
<tr>
<td>CORF and TNF</td>
<td>4.17 - 4.18</td>
<td>4.72 4.73 4.74</td>
</tr>
<tr>
<td>CORF and TOF</td>
<td>4.21 -</td>
<td>4.75 4.76</td>
</tr>
<tr>
<td>CORF and DORF</td>
<td>4.24 4.25</td>
<td>4.77 4.78</td>
</tr>
<tr>
<td>CORF and CORF</td>
<td>4.26 4.27</td>
<td>4.79 4.80</td>
</tr>
</tbody>
</table>

For the rows in Table 4.7, there are three different situations. First, some of them can be interpreted in a similar manner as those in Tables 4.3 and 4.4. For example, for CORF and CORF in Table 4.7, the Expression (4.26) is equivalent to the Expression (4.79). Second, in some rows, the faulty expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class. For example, for the third subcase of CORF and ENF in Table 4.7, the Expression (4.70) does not have its counterpart in double faults without ordering. Third, in some rows, the faulty expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class, but they are equivalent to other faulty expressions of double faults without ordering in a different double fault class. For example, for the fourth subcase of CORF and ENF in Table 4.7, the Expression (4.71) does not have its counterpart in CORF and ENF without ordering. However, it is equivalent to the Expression (4.18) in the double fault TNF and CORF without ordering in Section 4.1.
4.4 Summary

As discussed in Sections 4.1 and 4.2, there are 27 possible faulty expressions of double faults without ordering and 53 possible faulty expressions of double faults with ordering. After comparing all these faulty expressions, 49 out of the 53 faulty expressions of double faults with ordering have their equivalent counterparts in double faults without ordering. As a result, the 27 faulty expressions due to double faults without ordering can be used to represent these 49 faulty expressions. The remaining 4 faulty expressions are given by Expressions (4.53), (4.70), (4.73) and (4.76).

Hence, for the five single classes studied in this chapter, there are altogether 31 different double-fault expressions, 27 of them are from double fault classes without ordering and the remaining 4 are from double fault classes with ordering.

Table 4.8 summaries all 31 double-fault expressions and their corresponding double fault classes. The notation $F_1 \Join F_2$ is used to denote the double fault class formed from two single fault classes $F_1$ and $F_2$. For ease of reference, the numbers of those double-fault expressions reported in Section 4.1 and Section 4.2 are used in Table 4.8.

Table 4.8: Double fault classes and double-fault expressions $(S = p_1 + \ldots + p_m)$

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF $\Join$ ENF</td>
<td>Case 1 $(i_1 &lt; h_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2} + \ldots + p_{h_2} + \ldots + p_m$ (4.1)</td>
</tr>
<tr>
<td></td>
<td>Case 2 $({i_2, \ldots, h_2} \subsetneq {i_1, \ldots, h_1}): p_1 + \ldots + p_{i_2} + \ldots + p_{h_2} + \ldots + p_{i_1} + \ldots + p_m$ (4.2)</td>
</tr>
<tr>
<td>ENF $\Join$ TNF</td>
<td>Case 1 $(i_1 &lt; h_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2} + \ldots + p_m$ (4.3)</td>
</tr>
<tr>
<td></td>
<td>Case 2 $(i_1 \leq i_2 \leq h_1$ and $i_1 &lt; h_1): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2} + \ldots + p_m$ (4.4)</td>
</tr>
<tr>
<td>ENF $\Join$ TOF</td>
<td>Case 1 $(i_1 &lt; h_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2-1} + p_{i_2+1} + \ldots + p_m$ (4.5)</td>
</tr>
<tr>
<td></td>
<td>Case 2 $(i_1 \leq i_2 \leq h_1$ and $i_1 &lt; h_1): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2-1} + p_{i_2+1} + \ldots + p_m$ (4.6)</td>
</tr>
<tr>
<td>ENF $\Join$ DORF</td>
<td>Case 1 $(i_1 &lt; h_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} + \ldots + p_{i_2} p_{i_2+1} + \ldots + p_m$ (4.7)</td>
</tr>
<tr>
<td></td>
<td>Case 2 $(i_1 &lt; h_1 &lt; m): p_1 + \ldots + p_{i_1} + \ldots + p_{h_1} p_{h_1+1} + \ldots + p_m$ (4.8)</td>
</tr>
<tr>
<td></td>
<td>Case 3 $(i_1 &lt; m): p_1 + \ldots + p_{i_1} + \ldots + p_{i_2} p_{i_2+1} + \ldots + p_{h_1} + \ldots + p_m$ (4.9)</td>
</tr>
<tr>
<td></td>
<td>Case 4 $(i_1 &lt; i_2 &lt; h_1): p_1 + \ldots + p_{i_1} p_{i_1+1} + \ldots + p_m$ (4.10)</td>
</tr>
<tr>
<td>ENF $\Join$ CORF</td>
<td>Case 1 $(i_1 &lt; h_1 &lt; i_2$ and $1 \leq j_2 &lt; k_{i_2}): p_1 + \ldots + p_{i_1} + \ldots + p_{i_2} p_{i_2+1} + \ldots + p_m$ (4.11)</td>
</tr>
<tr>
<td></td>
<td>Case 2 $(i_1 \leq i_2 \leq h_1$, $i_1 &lt; h_1$ and $1 \leq j_2 &lt; k_{i_2}): p_1 + \ldots + p_{i_1} + \ldots + p_{i_2} 1, p_{i_2+1} 1, k_{i_2} + \ldots + p_{h_1} + \ldots + p_m$ (4.12)</td>
</tr>
<tr>
<td>TNF $\Join$ TNF</td>
<td>$(i_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{i_2} + \ldots + p_m$ (4.13)</td>
</tr>
<tr>
<td>TNF $\Join$ TOF</td>
<td>$(i_1 &lt; i_2): p_1 + \ldots + p_{i_1} + \ldots + p_{i_2-1} + p_{i_2+1} + \ldots + p_m$ (4.14)</td>
</tr>
</tbody>
</table>
Table 4.8 (cont’d) Double fault classes and double-fault expressions

\((S = p_1 + \ldots + p_m)\)

(a) Double-fault expressions (numbered 4.15 - 4.27) due to double fault without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
</table>
| TNF \(\bowtie\) DORF | Case 1 \((i_1 < i_2 < m)\): \(p_1 + \ldots + \bar{p}_{i_1} + \ldots + p_{i_2}p_{i_2+1} + \ldots + p_m\) (4.15)  
Case 2 \((i_1 < m)\): \(p_1 + \ldots + \bar{p}_{i_1}p_{i_1+1} + \ldots + p_m\) (4.16) |
| TNF \(\bowtie\) CORF | Case 1 \((i_1 < i_2\) and \(1 \leq j_2 < k_{i_2}\)): \(p_1 + \ldots + \bar{p}_{i_1}\) + \(p_{i_2,1} + p_{i_2,1} + \ldots + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}} + \ldots + p_m\) (4.17)  
Case 2 (both faults occurred at the same term \(p_i\) and \(1 \leq j < k_i\)): \(p_1 + \ldots + \bar{p}_{i_1,j} + \bar{p}_{i_1,j+1,k_i} + \ldots + p_m\) (4.18) |
| TOF \(\bowtie\) TOF | \((i_1 < i_2): p_1 + \ldots + p_{i_1-1} + p_{i_1+1} + \ldots + p_{i_2-1} + p_{i_2+1} + \ldots + p_m\) (4.19)  
| TOF \(\bowtie\) DORF | \((i_1 < i_2 < m): p_1 + \ldots + p_{i_1-1} + p_{i_1+1} + \ldots + p_{i_2}p_{i_2+1} + \ldots + p_m\) (4.20) |
| TOF \(\bowtie\) CORF | \((i_1 < i_2\) and \(1 \leq j_2 < k_{i_2}\)): \(p_1 + \ldots + p_{i_1-1} + p_{i_1+1} + \ldots + p_{i_2} + p_{i_2,j_2+1,k_{i_2}} + \ldots + p_m\) (4.21) |
| DORF \(\bowtie\) DORF | Case 1 \((i_1 < i_2 < m): p_1 + \ldots + p_{i_1}p_{i_1+1} + \ldots + p_m\) (4.22)  
Case 2 \((i_1 < m - 1): p_1 + \ldots + p_{i_1}p_{i_1+1} + \ldots + p_m\) (4.23) |
| DORF \(\bowtie\) CORF | Case 1 \((i_1 < i_2 - 1\) and \(1 \leq j_2 < k_{i_2}\)): \(p_1 + \ldots + p_{i_1}p_{i_1+1} + \ldots + p_{i_2,j_2+1,k_{i_2}} + \ldots + p_m\) (4.24)  
Case 2 \((i_1 < m\) and \(1 \leq j < k_i): p_1 + \ldots + p_{i_1,j} + p_{i_1,j+1,k_i}p_{i_1+1} + \ldots + p_m\) (4.25) |
| CORF \(\bowtie\) CORF | Case 1 \((i_1 < i_2, 1 \leq j_1 < k_{i_1}\) and \(1 \leq j_2 < k_{i_2}\)): \(p_1 + \ldots + p_{i_1,1,j_1} + p_{i_1,1,j_1} + \ldots + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}} + \ldots + p_m\) (4.26)  
Case 2 (both faults occurred at the same term, \(p_i\) and \(1 \leq j_1 < j_2 < k_i\)): \(p_1 + \ldots + p_{i_1,1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_i} + \ldots + p_m\) (4.27) |

(b) Four double-fault expressions\(^a\) due to double faults with ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOF (\bowtie) DORF</td>
<td>(p_1 + \ldots + p_{i_1-1}p_{i_1+1} + \ldots + p_m) (4.53)</td>
</tr>
<tr>
<td>CORF (\bowtie) ENF</td>
<td>((i_1 &lt; h_1) and (1 \leq j_1 &lt; k_{i_1}: p_1 + \ldots + p_{i_1,1,j_1} + p_{i_1,1,j_1} + \ldots + p_{i_1,j_1+1,k_{i_1}} + \ldots + p_m) (4.70)</td>
</tr>
<tr>
<td>CORF (\bowtie) TNF</td>
<td>((1 \leq j_1 &lt; k_{i_1}): p_1 + \ldots + p_{i_1,j_1} + \bar{p}<em>{i_1,j_1+1,k</em>{i_1}} + \ldots + p_m) (4.73)</td>
</tr>
<tr>
<td>CORF (\bowtie) TOF</td>
<td>((1 \leq j_1 &lt; k_{i_1}): p_1 + \ldots + p_{i_1-1} + p_{i_1,j_1} + p_{i_1,j_1+1} + \ldots + p_m) (4.76)</td>
</tr>
</tbody>
</table>

\(^a\)For ease of cross-reference, the numbering of the faulty expressions follows Section 4.2.
Chapter 5

Detection Conditions on Double Faults Related to Terms only

Recently, the detection conditions of various hypothesized faults have been studied [9, 26, 30, 28, 29, 33, 53]. The detection condition of a particular fault in a program is a condition that makes the actual output of the program different from its expected output.

Previous studies on detection conditions mainly focused in two areas. First, the detection conditions have been used to develop test case selection strategies. In [9], Chen and Lau proposed three test case selection strategies based on the detection conditions of seven types of faults. Second, the detection conditions have been used to develop the fault class hierarchy. A fault class hierarchy establishes relationships between different types of faults. Kuhn [26] and Lau and Yu [33] used fault class hierarchies to explain fault-detecting abilities of various test case selection strategies.

Kuhn [25] presented a method to calculate the predicate differences for analysing the effects of faults in predicates. Later on, in [26], he used a similar method to analyse the effects of faults in Boolean expression. In simple terms, the method is as follows: let $P$ be a correct Boolean expression and $P'$ be another Boolean expression with several syntactical changes with respect to to $P$. The detection condition of $P'$ with respect to $P$ is presented as a result of $P \oplus P'$ where $\oplus$ is the Boolean exclusive-or operator XOR.

In this thesis, the same method is used to obtain the detection conditions of double faults in Boolean expressions. Let $S$ be a given Boolean expression in IDNF and $I$ be a double-fault expression in which differs from $S$ by two syntactic changes. A detection condition of $I$ with respect to $S$ is a condition that makes $S$ and $I$ evaluate to different truth values. In short, the detection condition can be derived from $S \oplus I$. Instead of simply presenting the Boolean expression $S \oplus I$ as detection conditions, they are presented as conditions satisfied by test cases in $\mathbb{B}^n$. Since such categorization is based on certain properties of test sets, it helps in identifying and
developing test case selection strategy to detect studied double faults.

As discussed before, double fault classes studied are classified as following three categories:

1. double faults related to terms only;

2. double faults related to a term and a literal; and

3. double faults related to literals only.

The detection conditions of double faults related to terms only are proved in this chapter. The detection conditions of double faults related to literals only and double faults related to a term and a literal and are proved in Chapters 7 and 9, respectively. Test cases that satisfy all detection conditions in these three chapters can detect all double faults studied in this thesis.

### 5.1 Detection conditions on 27 Double-fault Expressions Without Ordering

As discussed in Chapter 4, for double faults related to terms only, there are 15 classes of double faults without ordering, which result in 27 different double-fault expressions, and 25 classes of double faults with ordering, which result in 53 double-fault expressions. Since 49 out of 53 double-fault expressions are equivalent to the 27 distinct double-fault expressions due to double faults without ordering, and the 4 remaining double-fault expressions are not equivalent to any of the previous group of 27 faulty expressions. Altogether, there are 31 different double-fault expressions among all double-fault classes related to terms only. In this section, the detection conditions of 27 distinct faulty expressions due to double fault without ordering are proved and the discussion are organized into the following 5 subsections as shown in Table 5.1. The detection conditions of the remaining 4 faulty expressions due to double fault with ordering are proved in the next section.

<table>
<thead>
<tr>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
<th>Detailed Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>5.1.1</td>
</tr>
<tr>
<td>TNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>5.1.2</td>
</tr>
<tr>
<td>TOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>5.1.3</td>
</tr>
<tr>
<td>DORF</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td>5.1.4</td>
</tr>
<tr>
<td>CORF</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>5.1.5</td>
</tr>
</tbody>
</table>
5.1.1 ENF with Other Term Faults

Theorem 5.1.1 (ENF $\times$ ENF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two subexpressions $(p_{i_1} + \cdots + p_{i_1})$ and $(p_{i_2} + \cdots + p_{i_2})$ in $S$ are negated where $1 \leq i_1 < i_2 < m$, the resulting expression denoted as $I_{ENF(p_1 + \cdots + p_{i_1} \rightarrow p_1 + \cdots + p_{i_1}) \times ENF(p_{i_2} + \cdots + p_{i_2} \rightarrow p_{i_2} + \cdots + p_{i_2})}$ is equivalent to Expression (4.1) in Table 4.8. Then, we have $S \not\equiv I_{ENF(p_1 + \cdots + p_{i_1} \rightarrow p_1 + \cdots + p_{i_1}) \times ENF(p_{i_2} + \cdots + p_{i_2} \rightarrow p_{i_2} + \cdots + p_{i_2})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \cap \left( \bigcup_{i=i_2}^{h_2} TP_i(S) \right) \setminus \bigcup_{i=1}^{m} TP_i(S)$, or

2. $\bar{t} \in FP(S)$.

Proof: First, we observe that $S \oplus I_{ENF(p_1 + \cdots + p_{i_1} \rightarrow p_1 + \cdots + p_{i_1}) \times ENF(p_{i_2} + \cdots + p_{i_2} \rightarrow p_{i_2} + \cdots + p_{i_2})}$

$\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) + (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}))$

$\cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2})$

$+ (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}))$

$\cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2})$

$+ (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}))$

$\cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) + (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}))$

$\cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) + (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}))$

$\cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$+ (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + \cdots + p_{i_2}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + S$
Now, \( S(\vec{t}) \neq I_{\text{ENF}}(p_1 + \cdots + p_{h_1} \rightarrow \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \times I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_2} \rightarrow \bar{p}_{i_2} + \cdots + \bar{p}_{i_{h_2}}) \) (\( \vec{t} \)) if and only if
\[
S(\vec{t}) \oplus I_{\text{ENF}}(p_1 + \cdots + p_{h_1} \rightarrow \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \times I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_2} \rightarrow \bar{p}_{i_2} + \cdots + \bar{p}_{i_{h_2}}) \) (\( \vec{t} \)) = 1
\]
if and only if
\[
(p_{i_1} + \cdots + p_{h_1})(p_{i_2} + \cdots + p_{h_2}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}
\]
\[
\cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + S \text{ evaluates to 1 on } \vec{t}
\]
if and only if \( \vec{t} \) satisfies any of the following conditions

1. \( \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \cap \left( \bigcup_{i=1}^{h_2} TP_i(S) \right) \setminus \left( \bigcup_{i=i_1, \ldots, i_2, h_1, h_2} TP_i(S) \right) \), or
2. \( \vec{t} \in FP(S) \).

Hence, the result follows. \( \square \)

**Theorem 5.1.2 (ENF \& ENF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two subexpressions \((p_{i_1} + \cdots + p_{h_1})\) and \((p_{i_2} + \cdots + p_{h_2})\) in \( S \) are negated where \( 1 \leq i_1 \leq i_2 \leq h_2 \leq h_1 \leq m \), \( i_1 \neq i_2 \) and \( h_1 \neq h_2 \), the resulting expression denoted as \( I_{\text{ENF}}(p_1 + \cdots + p_{h_1} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \times I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_2} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \) is equivalent to Expression (4.2) in Table 4.8. Then, \( S \neq I_{\text{ENF}}(p_1 + \cdots + p_{h_1} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \times I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_2} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \) if and only if there is a test case \( \vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=i_1}^{m} TP_i(S) \right) \).

**Proof:** First, we observe that
\[
S \oplus I_{\text{ENF}}(p_1 + \cdots + p_{h_1} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \times I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_2} - \bar{p}_{i_1} + \cdots + \bar{p}_{i_{h_2}}) \equiv ((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1}) \oplus (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1}))
\]
\[
\cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv ((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1})(p_{i_1} + \cdots + \bar{p}_{i_2} + \cdots + \bar{p}_{i_{h_2}} + \cdots + p_{h_1}) + (p_{i_1} + \cdots + \bar{p}_{i_2} + \cdots + \bar{p}_{i_{h_2}} + \cdots + p_{h_1})(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv ((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1})(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1}) + (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_{h_1})(p_{i_1} + \cdots + \bar{p}_{i_2} + \cdots + \bar{p}_{i_{h_2}} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv ((p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1} + 0) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\text{(By making use of } (A + B)(A + \overline{B}) \equiv A \text{ and } (A + \overline{B})(A + B) \equiv 0)\]
\[
\equiv (p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]

47
Now, \[ S(\vec{t}) \neq I_{ENF}(p_1 + \cdots + p_{h_1} + \cdots + p_{h_2} + \cdots + p_m - \vec{p}_1 + \cdots + \vec{p}_{i_2} + \cdots + \vec{p}_m) \]
if and only if \[ S(\vec{t}) \oplus I_{ENF}(p_1 + \cdots + p_{h_1} + \cdots + p_{h_2} + \cdots + p_m - \vec{p}_1 + \cdots + \vec{p}_{i_2} + \cdots + \vec{p}_m) \]
if and only if \[ (p_1 + \cdots + p_{i_2 - 1} + p_{i_2 + 1} + \cdots + p_{h_1}) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{h_1 + 1} \cdots \vec{p}_m \]
evaluates to 1 on \( \vec{t} \).

if and only if \[ \vec{t} \in \left( \bigcup_{i=i_2 \ldots h_2} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1 \ldots i_1, h_1} TP_i(S) \right). \]

Hence, the result follows. \( \square \)

As a reminder, Theorem 5.1.2 excludes the case when \( S \) contains just 1 term or 2 terms. When \( S \) contains just 1 term (that is \( S = p_1 \)), negating it twice is equivalent to \( S \). When \( S \) contains just 2 terms (that is \( S = p_1 + p_2 \)), both ENFs require that \( i_1 < h_1 \) and \( i_2 < h_2 \). The net result is to negate \( S \) twice, which is then equivalent to \( S \).

**Theorem 5.1.3 (ENF \( \not\equiv \) TNF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_i + \cdots + p_{h_1}) \) and the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) are negated where \( 1 \leq i_1 < h_1 < i_2 \leq m \), the resulting expression denoted as \( I_{ENF}(p_1 + \cdots + p_{h_1} - \vec{p}_1 + \cdots + \vec{p}_m) \) is equivalent to Expression (4.3) in Table 4.8. Then, \( S \neq I_{ENF}(p_1 + \cdots + p_{h_1} - \vec{p}_1 + \cdots + \vec{p}_m) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \cap \left( \bigcup_{i=i_2}^{h_2} TP_i(S) \right) \) \( \setminus \left( \bigcup_{i \neq i_1 \ldots i_1, i_2} TP_i(S) \right) \), or
2. \( \vec{t} \in FP(S) \).

**Proof**: First, we observe that \( S \oplus I_{ENF}(p_1 + \cdots + p_{h_1} - \vec{p}_1 + \cdots + \vec{p}_m) \not\equiv \text{NF}(p_2 - \vec{p}_{i_2}) \)

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \oplus (p_1 + \cdots + p_{h_1} + p_{i_2})\right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) + (p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot (p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot (p_1 + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]

\[ \equiv \left((p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \cdot (p_i + \cdots + p_{h_1} + p_{i_2}) \right) \cdot \vec{p}_1 \cdots \vec{p}_{i_2 - 1} \cdot \vec{p}_{i_2 + 1} \cdots \vec{p}_m \]
\[
\equiv (p_{i_1} + \cdots + p_{h_{1}}) p_{i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + \bar{S}
\]

Now,
\[
S(\vec{t}) \not\equiv I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_{1}} - p_{i_1} + \cdots + p_{h_{1}}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})(\vec{t})
\]
if and only if
\[
S(\vec{t}) \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_{1}} - p_{i_1} + \cdots + p_{h_{1}}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})(\vec{t}) = 1
\]
if and only if
\[
(p_{i_1} + \cdots + p_{h_{1}}) p_{i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + \bar{S}
\]
evaluates to 1 on \vec{t}
if and only if \vec{t} satisfies any of the following conditions:

1. \[\vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \cap \left( \bigcup_{i=1}^{m} TP_i(S) \right), \text{ or} \]
2. \[\vec{t} \in FP(S).
\]

Hence, the result follows. \(\square\)

**Theorem 5.1.4 (ENF \& TNF - Case 2)**

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the sub-expression \((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1})\) and the \(i_2\)-th term, \(p_{i_2}\), in \(S\) are negated where \(1 \leq i_1 \leq i_2 \leq h_1 \leq m\) and \(i_1 \neq h_1\), the resulting expression denoted as \(I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})\) is equivalent to Expression (4.4) in Table 4.8. Then, \(S \not\equiv I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})\) if and only if there is a test case \(\vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)\).

**Proof:** First, we observe that \(S \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \oplus (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})\)
\[
\equiv \left((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \frac{(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1})}{(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) + (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{i_2} + \cdots + p_{h_1})}ight) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv \left((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv (p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
(By making use of \((A + B)(A + B) \equiv A\) and \((A + B)(A + \bar{B}) \equiv 0\))

Now,
\[
S(\vec{t}) \not\equiv I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})(\vec{t})
\]
if and only if
\[
S(\vec{t}) \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times I_{\text{TNF}}(p_{i_2} - \bar{p}_{i_2})(\vec{t}) = 1
\]
if and only if
\[
(p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
evaluates to 1 on \vec{t}
if and only if \vec{t} satisfies any of the following conditions:

1. \[\vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \cap \left( \bigcup_{i=1}^{m} TP_i(S) \right), \text{ or} \]
2. \[\vec{t} \in FP(S).
\]

Hence, the result follows. \(\square\)
Theorem 5.1.5 (ENF $\bowtie$ TOF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $(p_i + \cdots + p_h)$ in $S$ is negated and the $i_2$-th term, $p_{i_2}$, in $S$ is omitted where $1 \leq i_1 < h_1 < i_2 \leq m$, the resulting expression denoted as $I_{ENF}(p_1 + \cdots + p_h, p_{i_2} \rightarrow) = I_{TOF}(p_{i_2} \rightarrow)$ is equivalent to Expression (4.5) in Table 4.8. Then, $S \not\equiv I_{ENF}(p_1 + \cdots + p_h, p_{i_2} \rightarrow) \equiv I_{TOF}(p_{i_2} \rightarrow)$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in \bigcup_{i=1}^{h_1} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_1, i_2}^{m} TP_i(S)$, or

2. $\vec{t} \in FP(S)$.

Proof: First, we observe that $S \oplus I_{ENF}(p_1 + \cdots + p_h, p_{i_2} \rightarrow) = I_{TOF}(p_{i_2} \rightarrow)$

$$
\equiv ((p_i + \cdots + p_h) \oplus (p_i + \cdots + p_h)) \cdot \bar{p}_1 \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_m
$$

$$
\equiv ((p_i + \cdots + p_h) \cdot (p_i + \cdots + p_h)) \cdot (p_i + \cdots + p_h) \cdot (p_i + \cdots + p_h)
$$

$$
\equiv ((p_i + \cdots + p_h) + (p_i + \cdots + p_h)) \cdot \bar{p}_1 \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_m
$$

$$
\equiv (p_i + \cdots + p_h) \cdot \bar{p}_1 \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_m + S
$$

Now, $S(\vec{t}) \not\equiv I_{ENF}(p_1 + \cdots + p_h, p_{i_2} \rightarrow)(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{ENF}(p_1 + \cdots + p_h, p_{i_2} \rightarrow) = 1$

if and only if $(p_i + \cdots + p_h) \cdot \bar{p}_1 \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_m + S$

evaluates to 1 on $\vec{t}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in \bigcup_{i=1}^{h_1} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_1, i_2}^{m} TP_i(S)$, or

2. $\vec{t} \in FP(S)$.

Hence, the result follows. $\square$

Theorem 5.1.6 (ENF $\bowtie$ TOF - Case 2)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $(p_i + \cdots + p_h)$ in $S$ is negated and the $i_2$-th term, $p_{i_2}$, in $S$ is omitted where $1 \leq i_1 \leq i_2 \leq h_1 \leq m$ and $i_1 \neq h_1$, the resulting expression denoted as
\( I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1} - p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) \) is equivalent to Expression (4.6) in Table 4.8. Then, \( S \neq I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1} - p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{m} TP_i(S) \right) \), or

2. \( \vec{t} \in \text{FP}(S) \).

**Proof:** First, we observe that \( S \oplus I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1} - p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) \equiv (p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1}) \oplus (p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1}) \)

\[
\equiv (p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1}) \cdot \left( p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1} \right) + \left( p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1} \right) \cdot \left( p_{i_1} + \cdots + p_{i_3} + \cdots + p_{i_1} \right)
\]

Now, \( S(\vec{t}) \neq I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1} - p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) \) if and only if \( S(\vec{t}) \oplus I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1} - p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) = 1 \)

if and only if \( (p_{i_1} + \cdots + p_{i_2} + p_{i_2} + \cdots + p_{i_1}) \cdot \bar{p}_{i_1} \cdots \bar{p}_{i_2} \cdots \bar{p}_{i_1} \cdot \bar{p}_{i_3} \cdots \bar{p}_{i_2} \cdots \bar{p}_{i_1} + \bar{S} \) evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{m} TP_i(S) \right) \), or

2. \( \vec{t} \in \text{FP}(S) \).

Hence, the result follows. \( \square \)

**Theorem 5.1.7 (ENF \& DORF - Case 1)**

Let \( S=p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_{i_1} + \cdots + p_{i_1}) \) in \( S \) is negated and the subexpression \( (p_{i_2} + p_{i_2+1}) \) in \( S \) is implemented as \( p_{i_2}p_{i_2+1} \) where \( 1 \leq i_1 < h_1 < i_2 < m \), the resulting expression denoted as
\( I_{\text{ENF}}(\tilde{p}_i_1 \cdots + \tilde{p}_i_n \rightarrow \tilde{p}_i_1 \cdots + \tilde{p}_i_n \times DORF(\tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p)) \) is equivalent to Expression (4.7) in Table 4.8. Then, \( S \neq I_{\text{ENF}}(\tilde{p}_i_1 \cdots + \tilde{p}_i_n \rightarrow \tilde{p}_i_1 \cdots + \tilde{p}_i_n \times DORF(\tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p)) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, i_2, i_2+1} TP_i(S) \right) \cup TP_{i_2}(S) \cap TP_{i_2+1}(S) \), or

2. \( \vec{t} \in FP(S) \).

**Proof:** First, we observe that \( S \oplus I_{\text{ENF}}(\tilde{p}_i_1 \cdots + \tilde{p}_i_n \rightarrow \tilde{p}_i_1 \cdots + \tilde{p}_i_n \times DORF(\tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p)) \) evaluates to Expression (4.7).

Now, \( S(\vec{t}) \neq I_{\text{ENF}}(\tilde{p}_i_1 \cdots + \tilde{p}_i_n \rightarrow \tilde{p}_i_1 \cdots + \tilde{p}_i_n \times DORF(\tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p)) \) if and only if \( S(\vec{t}) \oplus I_{\text{ENF}}(\tilde{p}_i_1 \cdots + \tilde{p}_i_n \rightarrow \tilde{p}_i_1 \cdots + \tilde{p}_i_n \times DORF(\tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p + \tilde{p}_i_p)) = 1 \) if and only if \( (\tilde{p}_i_1 \cdots + \tilde{p}_i_n) \tilde{p}_i_2 \tilde{p}_i_2 + 1 \cdot \tilde{p}_i \cdots \tilde{p}_i - 1 \cdot \tilde{p}_i - 2 \cdots \tilde{p}_m + S \) evaluates to 1 on \( \vec{t} \). If \( \vec{t} \) satisfies any of the following conditions:

1. \( \tilde{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, i_2, i_2+1} TP_i(S) \right) \cup TP_{i_2}(S) \cap TP_{i_2+1}(S) \), or

2. \( \tilde{t} \in FP(S) \).

Hence, the result follows. \( \square \)
Theorem 5.1.8 \((\text{ENF} \triangleleft \text{DORF} - \text{Case 2})\)

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the subexpression \((p_i_1 + \cdots + p_{i_1})\) in \(S\) is negated and the subexpression \((p_{i_2} + p_{i_2+1})\) in \(S\) is implemented as \(p_{i_2}p_{i_2+1}\) where \(1 \leq i_1 < (h_1 = i_2) < m\), the resulting expression denoted as \(I_{\text{ENF}}(p_1 + \cdots + p_{i_1} \rightarrow p_{i_1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow p_{h_1} + p_{i_2+1})\) is equivalent to Expression (4.8) in Table 4.8. Then, \(S \not\equiv I_{\text{ENF}}(p_1 + \cdots + p_{i_1} \rightarrow p_{i_1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow p_{h_1} + p_{i_2+1})\) if and only if there is a test case \(t^*\) in \((\bigcup_{i=i_1}^{h_1} TP_i(S)) \setminus (\bigcup_{i=i_1}^{h_1} TP_i(S))\).

Proof: Since \(i_2 = h_1\), we observe that

\[
S \equiv I_{\text{ENF}}(p_1 + \cdots + p_{i_1} \rightarrow p_{i_1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow p_{h_1} + p_{i_2+1})
\]

\[
\equiv ((p_i + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})
\]

(㎝ByEmail making use of \((A + B)(A + B) = A\) and \((A + B)(A + B) = 0\))

Now, \(S(t^*) \neq I_{\text{ENF}}(p_1 + \cdots + p_{i_1} \rightarrow p_{i_1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow p_{h_1} + p_{i_2+1})(t^*)\) if and only if \(S(t^*) \equiv I_{\text{ENF}}(p_1 + \cdots + p_{i_1} \rightarrow p_{i_1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} \rightarrow p_{h_1} + p_{i_2+1})(t^*) = 1\) if and only if \((p_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p} \cdot \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + (p_{i_2} + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1})\) evaluates to 1 on \(t^*\) if and only if \(t^* \in \left(\bigcup_{i=i_1}^{h_1} TP_i(S)\right) \setminus (\bigcup_{i=i_1}^{h_1} TP_i(S))\).

Hence, the result follows. \(\square\)

Theorem 5.1.9 \((\text{ENF} \triangleleft \text{DORF} - \text{Case 3})\)

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in irredundant disjunctive normal form. Suppose that the subexpression \((p_i + \cdots + p_{i_1})\) in \(S\) is negated and the subexpression \((p_{i_2} + p_{i_2+1})\) in \(S\) is implemented as \(p_{i_2}p_{i_2+1}\) where \(1 \leq i_1 \leq i_2 < h_1 \leq m\), \(i_1 \neq i_2\) and \(h_1 \neq i_2 + 1\), the resulting expression denoted as \(I_{\text{ENF}}(p_1 + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + p_{i_2+1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} - p_{i_2} + p_{i_2+1})\) is equivalent to Expression (4.9) in Table 4.8. Then, \(S \not\equiv I_{\text{ENF}}(p_1 + \cdots + p_{i_1} + p_{i_2} + p_{i_2+1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + p_{i_2+1} + \cdots + p_{h_1}) \triangleleft \text{DORF}(p_{i_2} + p_{i_2+1} - p_{i_2} + p_{i_2+1})\) if and only if there is a test case \(t^*\) that satisfies any of the following conditions:

1. \(t^* \in \left(\bigcup_{i=i_1}^{h_1} TP_i(S)\right) \cup (TP_{i_2}(S) \cap TP_{i_2+1}(S)) \setminus (\bigcup_{i=i_1}^{h_1} TP_i(S)),\) or

53
2. \( \vec{t} \in FP(S) \).

**Proof**: First, we observe that

\[
S \oplus I_{ENF}(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1 - p_1 - \cdots - p_2 - p_{2+1}) \equiv (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \oplus \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)}
\]

\[\equiv ((p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)}) + (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)}\]

\[\equiv ((p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)}) + (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)}\]

\[\equiv (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) + (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)\]

\[\cdot \overline{(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1)} \cdot \overline{p_{h+1} + \cdots + p_m}
\]

(By making use of \((A + B + C)(AB + C) \equiv AB + C\)
and \((A + B + C)(AB + C) \equiv A + B + C\)

\[\equiv (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{p_1 \cdot \overline{p_{h+1} + \cdots + p_m}}
\]

\[+ (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{p_1 \cdot \overline{p_{h+1} + \cdots + p_m}}\]

\[\equiv (p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{p_1 \cdot \overline{p_{h+1} + \cdots + p_m} + S}\]

Now, \(S(\vec{t}) \neq I_{ENF}(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1 - p_1 - \cdots - p_2 - p_{2+1}) \oplus I_{ENF}(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1 - p_1 - \cdots - p_2 - p_{2+1})\)

if and only if \(S(\vec{t}) \oplus I_{ENF}(p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1 - p_1 - \cdots - p_2 - p_{2+1})\)

\[\equiv DORF(p_2 + p_{2+1} \cdots p_{2+1})((\vec{t}) = 1\]

if and only if \((p_1 + \cdots + p_2 + p_{2+1} + \cdots + p_h_1) \cdot \overline{p_1 \cdot \overline{p_{h+1} + \cdots + p_m} + S}\)

evaluates to 1 on \(\vec{t}\)

if and only if \(\vec{t}\) satisfies any of the following conditions:

1. \(\vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \cup (TP_{i2}(S) \cap TP_{i2+1}(S)) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right),\)

2. \(\vec{t} \in FP(S)\).

Hence, the result follows. \(\square\)

**Theorem 5.1.10 (ENF \(\times\) DORF - Case 4)**

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in irredundant disjunctive normal form.
Suppose that the subexpression \((p_{i_1} + \cdots + p_{h_1})\) in \(S\) is negated and the subexpression \((p_{i_2} + p_{i_2+1})\) in \(S\) is implemented as \(p_{i_2}p_{i_2+1}\) where \(1 \leq (i_1 = i_2) < (h_1 = i_2 + 1) \leq m\), the resulting expression denoted as \(I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{DORF}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})\) is equivalent to Expression (4.10) in Table 4.8. Then, \(S \neq I_{\text{ENF}}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{DORF}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})\) if and only if there is a test case \(\vec{t}\) that satisfies any of the following conditions:

1. \(\vec{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1,i_1+1} TP_i(S) \right)\), or

2. \(\vec{t} \in FP(S)\).

**Proof:** Since \(i_1 = i_2\) and \(h_1 = i_2 + 1\), we have \(h_1 = i_1 + 1\). Then, we observe that

\[
S \oplus I_{\text{ENF}}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{DORF}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1}) = \left( (p_{i_1} + p_{i_1+1}) \oplus \left( p_{i_1}p_{i_1+1} \right) \right) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m.
\]

\[
\equiv \left( (p_{i_1} + p_{i_1+1}) \cdot \left( p_{i_1}p_{i_1+1} \right) \right) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m.
\]

\[
\equiv \left( (p_{i_1} + p_{i_1+1}) \cdot \left( p_{i_1}p_{i_1+1} \right) \right) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m.
\]

\[
\equiv (p_{i_1}p_{i_1+1} + (p_{i_1} + p_{i_1+1})) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m.
\]

(By making use of \((A+B)(AB) \equiv AB\) and \((A+B)(AB) \equiv A + B\))

\[
\equiv p_{i_1}p_{i_1+1} \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m + (p_{i_1} + p_{i_1+1}) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m.
\]

\[
\equiv p_{i_1}p_{i_1+1} \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m + S.
\]

Now, \(S(\vec{t}) \neq I_{\text{ENF}}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{DORF}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})\)(\(\vec{t}\)) if and only if \(S(\vec{t}) \oplus I_{\text{ENF}}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{DORF}(p_{i_1} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})\)(\(\vec{t}\)) = 1 if and only if \(p_{i_1}p_{i_1+1} \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \cdot \tilde{p}_{i_1+2} \cdots \tilde{p}_m + S\) evaluates to 1 on \(\vec{t}\) if and only if \(\vec{t}\) satisfies any of the following conditions:

1. \(\vec{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1,i_1+1} TP_i(S) \right)\), or

2. \(\vec{t} \in FP(S)\).

Hence, the result follows. \(\square\)

**Theorem 5.1.11 (ENF \& CORF - Case 1)**

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the subexpression \((p_{i_1} + \cdots + p_{h_1})\) in \(S\) is negated and the \(i_2\)-th term, \(p_{i_2}\), in \(S\) is implemented as \(p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_{i_2}}\) where \(1 \leq i_1 < h_1 < i_2 \leq m\), \(p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,2,j_2+1,k_{i_2}}\) and \(1 \leq j_2 < k_{i_2}\), the resulting expression denoted as \(I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{CORF}(p_{i_2} \rightarrow p_{i_2,2,j_2} + p_{i_2,2,j_2+1,k_{i_2}})\) is equivalent to Expression (4.11) in Table 4.8. Then, \(S \neq I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} \rightarrow p_{i_2} \rightarrow p_{i_2+1}) \times \text{CORF}(p_{i_2} \rightarrow p_{i_2,2,j_2} + p_{i_2,2,j_2+1,k_{i_2}})\) if and only if there is a test case \(\vec{t}\) that satisfies any of the following conditions:
1. $\vec{t} \in (\bigcup_{i=1}^{h_1} TP_i(S)) \setminus (\bigcup_{i=1}^{m} TP_i(S))$ such that $p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2} = 0$, or

2. $\vec{t} \in FP(S)$.

**Proof:** First, we observe that

$$S \oplus I_{ENF}(p_1 + \cdots + p_{h_1} \rightarrow p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \equiv (p_1 + \cdots + p_{h_1} + p_{i_2}) \oplus (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2})$$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) + (p_1 + \cdots + p_{h_1} + p_{i_2})$$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2} + (p_1 + \cdots + p_{h_1} + p_{i_2}))$$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$$

(By making use of $(A+B)A \equiv A$ and $(\bar{A}+\bar{B})(\bar{A}+B+C) \equiv \bar{A}+\bar{B}C$)

$$= (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$$

(By rewriting $A+B$ as $(A+B)(\bar{A}B)$ because they are equivalent)

$$= (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$$

+ $(p_1 + \cdots + p_{h_1} + p_{i_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$ + $S$

$$= (p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + S$

Now, $S(\vec{t}) \neq I_{ENF}(p_1 + \cdots + p_{h_1} \rightarrow p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2})(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{ENF}(p_1 + \cdots + p_{h_1} \rightarrow p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2})(\vec{t})$

= 1

if and only if $(p_1 + \cdots + p_{h_1} + p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2}) \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

+ $S$ evaluates to 1 on $\vec{t}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in \bigcup_{i=1}^{h_1} TP_i(S) \setminus \bigcup_{i=1}^{m} TP_i(S)$ such that $p_{i_2, 1, j_2} + p_{i_2, j_2+1, k_2} = 0$, or

2. $\vec{t} \in FP(S)$.

Hence, the result follows.

**Theorem 5.1.12 (ENF $\otimes$ CORF - Case 2)**

Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the sub-ex-
pression \((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1})\) in \(S\) is negated and the \(i_2\)-th term, \(p_{i_2}\), in \(S\) is implemented as \(p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}\) where \(1 \leq i_1 \leq i_2 \leq h_1 \leq m\), \(i_1 \neq h_1\), \(p_{i_2} = p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}}\) and \(1 \leq j_2 < k_{i_2}\), the resulting expression denoted as \(I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1})\) \(\equiv CORF(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})\) is equivalent to Expression \((4.12)\) in Table \(4.8\).

Then, \(S \not\equiv I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1})\) \(\equiv CORF(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})\) if and only if there is a test case \(\bar{t}\) that satisfies any of the following conditions:

1. \(\bar{t} \in \left( \bigcup_{i=1}^{h_1} \text{TP}_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1} \text{TP}_i(S) \right),\) or

2. \(\bar{t} \in \text{FP}(S)\) such that \(p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} = 0\) on \(\bar{t}\).

**Proof:** First, we observe that

\[
S \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \oplus (p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1})
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv ((p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1}) + (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1})
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1}) + (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot (p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1})
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

(\text{By making use of } (A + BC)(A + B + C) \equiv A + BC

\text{and } (A + BC)(A + B + C) \equiv A + B + C)

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
+ p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1} \cdot p_1 \cdots p_{i_1-1} \cdot p_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
+ p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_{h_1} \cdot p_1 \cdots p_{i_1-1} \cdot p_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

(\text{By rewriting } AB \text{ as } AB(A + B) \text{ because they are equivalent})

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
+ p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m + p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_2} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m + p_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_2} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m + \left( p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \right) \cdot \bar{S}
\]
Now, \[ S(\bar{t}) \neq I_{ENF(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1})} \]
\[ \times CORF(p_2 - p_{i_2,1}, j_2 + p_{i_2,2} + 1, k_{i_2}) (\bar{t}) \]
if and only if \[ S(\bar{t}) \oplus I_{ENF(p_1 + \cdots + p_{i_2} + \cdots + p_{i_1})} \]
\[ \times CORF(p_2 - p_{i_2,1}, j_2 + p_{i_2,2} + 1, k_{i_2}) (\bar{t}) = 1 \]
if and only if \( (p_1 + \cdots + p_{i_2} + \cdots + p_{i_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \)
\[ + (p_{i_2,1}, j_2 + p_{i_2,2} + 1, k_{i_2}) \cdot \bar{S} \]
evaluates to 1 on \( \bar{t} \)
if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1} TP_i(S) \right) \), or

2. \( \bar{t} \in FP(S) \) such that \( p_{i_2,1}, j_2 + p_{i_2,2} + 1, k_{i_2} = 0 \) on \( \bar{t} \).

Hence, the result follows. \( \square \)

5.1.2 TNF with Other Term Faults

Theorem 5.1.13 (\( \text{TNF} \times \text{TNF} \))

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two different terms, \( p_{i_1} \) and \( p_{i_2} \), in \( S \) are negated where \( 1 \leq i_1 < i_2 \leq m \), the resulting expression denoted as \( I_{\text{TNF}(p_{i_1} \rightarrow \bar{p}_{i_1}) \times \text{TNF}(p_{i_2} \rightarrow \bar{p}_{i_2})} \) is equivalent to Expression (4.13) in Table 4.8. Then, \( S \neq I_{\text{TNF}(p_{i_1} \rightarrow \bar{p}_{i_1}) \times \text{TNF}(p_{i_2} \rightarrow \bar{p}_{i_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \), or

2. \( \bar{t} \in FP(S) \).

Proof: First, we observe that \( S \oplus I_{\text{TNF}(p_{i_1} \rightarrow \bar{p}_{i_1}) \times \text{TNF}(p_{i_2} \rightarrow \bar{p}_{i_2})} \)
\[ \equiv ( (p_{i_1} + p_{i_2}) \oplus (\bar{p}_{i_1} + \bar{p}_{i_2}) ) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv ( (p_{i_1} + p_{i_2})(\bar{p}_{i_1} + \bar{p}_{i_2}) + (p_{i_1} + p_{i_2})(\bar{p}_{i_1} + \bar{p}_{i_2}) ) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv ( (p_{i_1} + p_{i_2})(\bar{p}_{i_1} + \bar{p}_{i_2}) + (\bar{p}_{i_1} + \bar{p}_{i_2})(\bar{p}_{i_1} + \bar{p}_{i_2}) ) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1}p_{i_2} + (\bar{p}_{i_1} + \bar{p}_{i_2}) ) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \text{(By making use of } (A + B)(AB) \equiv AB \text{ and } (A + B)(AB) \equiv A + B) \]
\[ \equiv p_{i_1}p_{i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + (\bar{p}_{i_1} + p_{i_2}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv p_{i_1}p_{i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + \bar{S} \]
Now, \( S(\vec{t}) \neq I_{\text{TNF}(p_{i_1} \rightarrow \vec{p}_{1}) \otimes \text{TOF}(p_{i_2} \rightarrow \vec{p}_{2})}(\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{\text{TNF}(p_{i_1} \rightarrow \vec{p}_{1}) \otimes \text{TOF}(p_{i_2} \rightarrow \vec{p}_{2})}(\vec{t}) = 1 \)
if and only if \( p_{i_1} p_{i_2} \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m + \overline{S} \)
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1}^{m} TP_i(S) \), or

2. \( \vec{t} \in FP(S) \).

Hence, the result follows.

\( \Box \)

**Theorem 5.1.14 (TNF \otimes TOF)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 \leq i_1 < i_2 \leq m \). The resulting expression denoted as \( I_{\text{TNF}(p_{i_1} \rightarrow \vec{p}_{1}) \otimes \text{TOF}(p_{i_2} \rightarrow \vec{p}_{2})} \) is equivalent to Expression (4.14) in Table 4.8. Then, \( S \neq I_{\text{TNF}(p_{i_1} \rightarrow \vec{p}_{1}) \otimes \text{TOF}(p_{i_2} \rightarrow \vec{p}_{2})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in (TP_{i_1}(S)) \setminus \bigcup_{i=1}^{m} TP_i(S) \), or

2. \( \vec{t} \in FP(S) \).

**Proof**: First, we observe that \( S \oplus I_{\text{TNF}(p_{i_1} \rightarrow \vec{p}_{1}) \otimes \text{TOF}(p_{i_2} \rightarrow \vec{p}_{2})} \)

\[ \equiv ((p_{i_1} + p_{i_2}) \oplus (\vec{p}_{1}) \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m) \]

\[ \equiv ((p_{i_1} + p_{i_2}) \cdot (\vec{p}_{1}) + (p_{i_1} + p_{i_2}) \cdot (\vec{p}_{1})) \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m \]

\[ \equiv ((p_{i_1} + p_{i_2}) \cdot p_{i_1} + (p_{i_1} + p_{i_2}) \cdot (\vec{p}_{1})) \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m \]

\[ \equiv (p_{i_1} + (p_{i_1} + p_{i_2})) \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m \]

\[ \equiv p_{i_1} \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m + (p_{i_1} + p_{i_2}) \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m \]

\[ \equiv p_{i_1} \cdot \vec{p}_1 \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdot \vec{p}_m + \overline{S} \]
Now, \( S(\bar{t}) \neq I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}) \times I_{\text{TOF}}(\bar{p}_{i_2} \rightarrow ) (\bar{t}) \) if and only if \( S(\bar{t}) \oplus I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}) \times I_{\text{TOF}}(\bar{p}_{i_2} \rightarrow ) (\bar{t}) = 1 \) if and only if \( p_{i_1} \cdot p_{i_1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_{m} + S \) evaluates to 1 on \( \bar{t} \) if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in (TP_{i_1}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \), or
2. \( \bar{t} \in FP(S) \).

Hence, the result follows.

**Theorem 5.1.15 (\( \text{TNF} \otimes \text{DORF} \) - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the subexpression \( (p_{i_2} + p_{i_2+1}) \) in \( S \) is implemented as \( p_{i_2}p_{i_2+1} \) where \( 1 \leq i_1 < i_2 < m \), the resulting expression denoted as \( I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes I_{\text{DORF}}(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1}) \) is equivalent to Expression (4.15) in Table 4.8. Then, \( S \neq I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes I_{\text{DORF}}(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1}) \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \cup (TP_{i_2}(S) \cap TP_{i_2+1}(S)) \), or
2. \( \bar{t} \in FP(S) \).

**Proof:** First, we observe that \( S \oplus I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes I_{\text{DORF}}(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1}) \)

\[ \equiv \left( \left( p_{i_1} \oplus p_{i_2} + p_{i_2+1} \right) \right) \cdot \left( \bar{p}_{i_1} \cdot \bar{p}_{i_2+1} \right) \]

\[ \equiv \left( \left( p_{i_1} \oplus p_{i_2} + p_{i_2+1} \right) \cdot \left( p_{i_1} \cdot p_{i_2+1} \right) + \left( p_{i_1} \cdot p_{i_2} + p_{i_2+1} \right) \right) \cdot \left( p_{i_1} \cdot p_{i_2+1} \right) \]

By making use of \((A+B+C)(A+BC) \equiv A+B+C\)

\[ \equiv p_{i_1} (p_{i_2}p_{i_2+1}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+2} \cdot \bar{p}_m \]

\[ + (p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+2} \cdot \bar{p}_m \]

\[ \equiv \bar{p}_{i_1} (p_{i_2}p_{i_2+1}) \cdot \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+2} \cdot \bar{p}_m + S \]
Now, \[ S(\vec{t}) \neq I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1})(\vec{t}) \]

if and only if \[ S(\vec{t}) \oplus I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1})(\vec{t}) = 1 \]

if and only if \[ p_i \cdot \bar{p}_i \cdot \bar{p}_{i+1} \cdot \bar{p}_{i+2} \cdots \bar{p}_m \text{ evaluates to 1 on } \vec{t} \]

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1, i_1 \pm 1} TP_i(S) \right) \cup \left( TP_{i_2}(S) \cap TP_{i_2+1}(S) \right) \), or
2. \( \vec{t} \in FP(S) \).

Hence, the result follows. \( \square \)

**Theorem 5.1.16 (TNF \times DORF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), is negated and the subexpression \( (p_{i_1} + p_{i+1}) \) is implemented as \( p_{i_1} \bar{p}_{i+1} \) where \( 1 \leq i_1 < m \), the resulting expression denoted as \( I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1}) \) is equivalent to Expression (4.16) in Table 4.8. Then, \( S \neq I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1}) \) if and only if there is a test case \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1, i_1 \pm 1} TP_i(S) \right) \).

**Proof:** We observe that \( S \oplus I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1}) \)

\[ \equiv (p_{i_1} + p_{i+1} + \cdots + p_m) \cdot 0 \]

\[ \equiv (p_{i_1} + p_{i_1} + \cdots + p_m) \cdot (\bar{p}_{i_1} \bar{p}_{i+1} \cdots \bar{p}_m) \]

\[ \equiv (p_{i_1} + p_{i_1} + \cdots + p_m) \cdot (\bar{p}_{i_1} \bar{p}_{i+1} \cdots \bar{p}_m) \]

\[ \equiv (p_{i_1} + p_{i_1} + \cdots + p_m) \cdot (\bar{p}_{i_1} \bar{p}_{i+1} \cdots \bar{p}_m) \]

\[ \equiv (p_{i_1} + 0) \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+1} + \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} + \cdots \]

(\text{By making use of } (A + B)(A + B) \equiv A \text{ and } (A + B)(\bar{A}B) \equiv 0 )

\[ \equiv p_{i_1} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+1} + \bar{p}_{i_1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} + \cdots \bar{p}_m \]

Now, \[ S(\vec{t}) \neq I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1})(\vec{t}) \]

if and only if \[ S(\vec{t}) \oplus I_{TNF(p_i \rightarrow \bar{p}_i)} \times DORF(p_{i+1} \rightarrow \bar{p}_{i+1})(\vec{t}) = 1 \]

if and only if \( p_{i_1} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+1} + \cdots \bar{p}_m \text{ evaluates to 1 on } \vec{t} \]

if and only if \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1, i_1 \pm 1} TP_i(S) \right) \).

Hence, the result follows. \( \square \)

**Theorem 5.1.17 (TNF \times CORF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term,
Proof: First, we observe that $S \oplus I_{T N F(p_{i_1} \rightarrow \bar{p}_{i_1})} \triangleq CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2})$

$\equiv \left(\left(p_{i_1} + p_{i_2}\right) \oplus \left(\bar{p}_{i_1} + p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right)\right) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv \left(\left(p_{i_1} + p_{i_2}\right) \cdot \left(p_{i_1} + p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right) + \left(p_{i_1} + p_{i_2}\right) \cdot \left(p_{i_1} + p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right)\right)$

$\cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv \left(\left(p_{i_1} \cdot \left(p_{i_1} + p_{i_2}\right) + p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right) + \left(p_{i_1} + p_{i_2}\right)\right) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

(By making use of $(A + BC)(A + B + C) \equiv A + BC$)

$\equiv p_{i_1} \cdot \left(p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$+ \left(p_{i_1} + p_{i_2}\right) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv p_{i_1} \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,2,j_2+1,k_i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \left(p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}\right) \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + \bar{S}$

(By rewriting $AB$ as $AB(A + B)$ because they are equivalent)

$\equiv p_{i_1} \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,2,j_2+1,k_i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \left(p_{i_2,1,j_2} \cdot \bar{p}_{i_2,2,j_2+1,k_i_2}\right) \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + \bar{S}$

$\equiv p_{i_1} \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,2,j_2+1,k_i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_m + \bar{S}$

$\equiv p_{i_1} \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,2,j_2+1,k_i_2} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_m + \bar{S}$

Now,

$S(\bar{t}) \neq I_{T N F(p_{i_1} \rightarrow \bar{p}_{i_1})} \triangleq CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2})(\bar{t})$

if and only if

$S(\bar{t}) \oplus I_{T N F(p_{i_1} \rightarrow \bar{p}_{i_1})} \triangleq CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2})(\bar{t}) = 1$

if and only if

$p_{i_1}(p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot p_{i_1+1} \cdots \bar{p}_m + \bar{S}$

evaluates to 1 on $\bar{t}$

if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$ such that $p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_i_2} = 0$, or

2. $\bar{t} \in FP(S)$.

Hence, the result follows. □
Theorem 5.1.18 (TFNF \(\times\) CORF - Case 2)

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term, \(p_{i_1}\), in \(S\) is negated and \(p_{i_1}\) is implemented as \(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}\) where \(1 \leq i_1 \leq m, p_{i_1} = p_{i_1,1,j_1} \cdot p_{i_1,1,j_1+1,k_{i_1}}\) and \(1 \leq j_1 < k_{i_1}\), the resulting expression denoted as \(I_{\text{TFNF}(p_{i_1} \rightarrow \overline{p}_{i_1}) \times \text{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})}\) is equivalent to Expression (4.18) in Table 4.8. Then, \(S \not\equiv I_{\text{TFNF}(p_{i_1} \rightarrow \overline{p}_{i_1}) \times \text{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})}\) if and only if there is a test case \(\overline{t}\) that satisfies any of the following conditions:

1. \(\overline{t} \in UTP_{i_1}(S)\), or
2. \(\overline{t} \in FP(S)\) such that \(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}} = 0\) on \(\overline{t}\).

Proof: We observe that \(S \oplus I_{\text{TFNF}(p_{i_1} \rightarrow \overline{p}_{i_1}) \times \text{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})}\)

\[
\equiv \left( (p_{i_1}) \oplus (p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \right) \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

\[
\equiv \left( (p_{i_1}) \cdot (p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \right) \cdot (\overline{p}_{i_1} \cdot p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

\[
\equiv (p_{i_1} + \overline{p}_{i_1,1,j_1} \overline{p}_{i_1,1,j_1+1,k_{i_1}}) \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

(By making use of \((A \cdot B) \equiv \overline{A} \cdot \overline{B}\) because they are equivalent)

\[
\equiv p_{i_1} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m + p_{i_1,1,j_1} \overline{p}_{i_1,1,j_1+1,k_{i_1}} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

\[
\equiv p_{i_1} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m + p_{i_1,1,j_1} \overline{p}_{i_1,1,j_1+1,k_{i_1}} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot (\overline{p}_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})
\]

\[
\equiv p_{i_1} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m + p_{i_1,1,j_1} \overline{p}_{i_1,1,j_1+1,k_{i_1}} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot (\overline{p}_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})
\]

\[
\equiv p_{i_1} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m + (\overline{p}_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

Now,

\[
S(\overline{t}) \not\equiv I_{\text{TFNF}(p_{i_1} \rightarrow \overline{p}_{i_1}) \times \text{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})}
\]

if and only if

\[
S(\overline{t}) \oplus I_{\text{TFNF}(p_{i_1} \rightarrow \overline{p}_{i_1}) \times \text{CORF}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})}(\overline{t}) = 1
\]

if and only if

\[
p_{i_1} \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m + (p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \cdot \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \cdot \overline{p}_{i_1+1} \cdots \overline{p}_m
\]

evaluates to 1 on \(\overline{t}\) if and only if \(\overline{t}\) satisfies following conditions:

1. \(\overline{t} \in UTP_{i_1}(S)\), or
2. \(\overline{t} \in FP(S)\) such that \(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}} = 0\) on \(\overline{t}\).

Hence, the result follows.
5.1.3 TOF with Other Term Faults

Theorem 5.1.19 (TOF \& TOF)
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two different terms, \( p_{i_1} \) and \( p_{i_2} \), in \( S \) are omitted where \( 1 \leq i_1 < i_2 \leq m \), the resulting expression denoted as \( I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& TOF}(p_{i_2} \rightarrow )} \) is equivalent to Expression (4.19) in Table 4.8. Then, \( S \not\equiv I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& TOF}(p_{i_2} \rightarrow )} \) if and only if there is a test case \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_2}(S)) \setminus \left( \bigcup_{i=1 \atop i \neq i_1, i_2}^m TP_i(S) \right) \).

Proof: First, we observe that \( S \equiv (p_1 + \ldots + p_{i_1-1} + p_{i_1+1} + p_{i_2-1} + p_{i_2+1} + \ldots + p_m) \)
\( \equiv (p_{i_1} + p_{i_2}) \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdots \vec{p}_m \)
(By making use of \((A + B) \oplus B \equiv A \cdot \overline{B})\)

Now, \( S(\vec{t}) \not\equiv I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& TOF}(p_{i_2} \rightarrow )}(\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& TOF}(p_{i_2} \rightarrow )}(\vec{t}) = 1 \)
if and only if \( (p_{i_1} + p_{i_2}) \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+1} \cdots \vec{p}_m \)
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_2}(S)) \setminus \left( \bigcup_{i=1 \atop i \neq i_1, i_2}^m TP_i(S) \right) \).

Hence, the result follows. \( \square \)

Theorem 5.1.20 (TOF \& DORF)
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted and the subexpression \((p_{i_1} + p_{i_2+1})\) in \( S \) is implemented as \( p_{i_1}p_{i_2+1} \) where \( 1 \leq i_1 < i_2 < m \), the resulting expression denoted as \( I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& DORF}(p_{i_2} \rightarrow p_{i_2+1})} \) is equivalent to Expression (4.20) in Table 4.8. Then, \( S \not\equiv I_{\text{TOF}(p_{i_1} \rightarrow ) \text{\& DORF}(p_{i_2} \rightarrow p_{i_2+1})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions

1. \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_2}(S)) \setminus \left( \bigcup_{i=1 \atop i \neq i_1, i_2}^m TP_i(S) \right) \)

2. \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_2+1}(S)) \setminus \left( \bigcup_{i=1 \atop i \neq i_1, i_2+1}^m TP_i(S) \right) \)

Proof: First, we observe that \( S \equiv \big((p_{i_1} + p_{i_2} + p_{i_2+1}) \oplus (p_{i_1}p_{i_2+1}) \big) \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+2} \cdots \vec{p}_m \)
\( \equiv ((p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot (p_{i_1}p_{i_2+1}) \oplus (p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot (p_{i_2}p_{i_2+1})) \cdot \vec{p}_{i_1-1} \cdot \vec{p}_{i_1+1} \cdot \vec{p}_{i_2-1} \cdot \vec{p}_{i_2+2} \cdots \vec{p}_m \)
\[
\begin{align*}
\equiv & \left( (p_{i_1} + p_{i_2} + p_{i_2+1}) \cdot (\overline{p}_{i_2} + \overline{p}_{i_2+1}) \right) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+2} \cdots \overline{p}_m \\
& \quad \text{(By making use of } (A + B + C)(BC) \equiv 0) \\
\equiv & \left( \overline{p}_{i_2} (p_{i_1} + p_{i_2+1}) + \overline{p}_{i_2+1} (p_{i_1} + p_{i_2}) \right) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+2} \cdots \overline{p}_m \\
\equiv & \left( p_{i_1} + p_{i_2} \right) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+1} \cdots \overline{p}_m \\
& \quad + (p_{i_1} + p_{i_2+1}) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+2} \cdots \overline{p}_m
\end{align*}
\]

Now, \[S(\overline{t}) \neq I_{TOF(p_{i_1} \to )DORF(p_{i_2} + p_{i_2+1} - p_{i_2} p_{i_2+1})}(\overline{t})\] if and only if \[S(\overline{t}) \oplus I_{TOF(p_{i_1} \to )DORF(p_{i_2} + p_{i_2+1} - p_{i_2} p_{i_2+1})}(\overline{t}) = 1\] if and only if \[(p_{i_1} + p_{i_2}) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+1} \cdots \overline{p}_m \]

\[+ (p_{i_1} + p_{i_2+1}) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+2} \cdots \overline{p}_m\]
evaluates to \(1\) on \(\overline{t}\) if and only if \(\overline{t}\) satisfies any of the following conditions

1. \(\overline{t} \in (TP_{i_1}(S) \cup TP_{i_2}(S)) \setminus \left( \bigcup_{i_1 \neq i_1, i_2} TP_i(S) \right)\)

2. \(\overline{t} \in (TP_{i_1}(S) \cup TP_{i_2+1}(S)) \setminus \left( \bigcup_{i_1 \neq i_1, i_2+1} TP_i(S) \right)\)

Hence, the result follows. \(\square\)

**Theorem 5.1.21** \((TOF \times CORF)\)

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term, \(p_{i_1}\), in \(S\) is omitted and the \(i_2\)-th term, \(p_{i_2}\), in \(S\) is implemented as \(p_{i_2,1,j_2} \cup p_{i_2,j_2+1,k_2}\) where \(1 \leq i_1 < i_2 \leq m\), \(1 \leq j_2 < k_2\), and \(p_{i_2} = p_{i_2,1,j_2} p_{i_2,j_2+1,k_2}\), the resulting expression denoted as \(I_{TOF(p_{i_1} \to )CORF(p_{i_2} \to p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2})}\) is equivalent to Expression (4.21) in Table 4.8. Then, \(S \neq I_{TOF(p_{i_1} \to )CORF(p_{i_2} \to p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2})}\) if and only if there is a test case \(\overline{t}\) that satisfies any of the following conditions:

1. \(\overline{t} \in UTP_{i_1}(S)\) such that \(p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2} = 0\) on \(\overline{t}\),

2. \(\overline{t} \in FP(S)\) such that \(p_{i_2,1,j_2} = 1\) on \(\overline{t}\), or

3. \(\overline{t} \in FP(S)\) such that \(p_{i_2, j_2+1, k_2} = 1\) on \(\overline{t}\).

**Proof:** First, we observe that \(S \oplus I_{TOF(p_{i_1} \to )CORF(p_{i_2} \to p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2})}\)

\[
\equiv \left( (p_{i_1} + p_{i_2}) \oplus (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) \right) \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+1} \cdots \overline{p}_m \\
\equiv \left( (p_{i_1} + p_{i_2}) \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) + (p_{i_1} + p_{i_2}) \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) \right) \\
\quad \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+1} \cdots \overline{p}_m \\
\equiv \left( (p_{i_1} + p_{i_2}) \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) + (p_{i_1} \cdot p_{i_2}) \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2}) \right) \\
\quad \cdot \overline{p}_{i_1} \cdot \overline{p}_{i_1+1} \cdot \overline{p}_{i_2} \cdot \overline{p}_{i_2+1} \cdots \overline{p}_m \\
\equiv \left( p_{i_1} \overline{p}_{i_2,1,j_2} + p_{i_2,1,j_2} \cdot \overline{p}_{i_2,j_2+1,k_2} + \overline{p}_{i_1} p_{i_2,1,j_2} \cdot \overline{p}_{i_2,j_2+1,k_2} \right)
\]

65
\[ \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By making use of \((A + B \cdot C)(\overline{B} \cdot \overline{C}) \equiv A \cdot \overline{B} \cdot \overline{C}\) and
\[ (A + B \cdot C)(B + C) \equiv A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} \])

\[ \equiv p_{i_1} \bar{p}_{i_2,1,j_2} \cdot \bar{p}_{i_2,2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[ + (p_{i_2,1,j_2} \cdot \bar{p}_{i_2,2+1,k_{i_2}} + p_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[ \equiv p_{i_1} \bar{p}_{i_2,1,j_2} \cdot \bar{p}_{i_2,2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot (p_{i_2,1,j_2} + \bar{p}_{i_2,2+1,k_{i_2}}) \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[ + (p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot (\bar{p}_{i_2,1,j_2} \cdot \bar{p}_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(rewriting \(AB \) as \(AB(A + B)\) because they are equivalent;

\[ \text{and} \ (A\overline{B} + \overline{A}B) \text{ as} \ (A + B)\overline{A}B \text{ because they are equivalent} \])

\[ \equiv p_{i_1} \bar{p}_{i_2,1,j_2} \cdot \bar{p}_{i_2,2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot (p_{i_2,1,j_2} + \bar{p}_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[ + (p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[ + (p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_2,1,j_2} \overline{S} + p_{i_2,2,j_2+1,k_{i_2}} \overline{S} \]

Now,

\[ S(\bar{t}) \neq I_{TOF(p_{i_1} \to \overline{CORF}(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,2+1,k_{i_2}}))}(\bar{t}) \]

if and only if

\[ S(\bar{t}) \oplus I_{TOF(p_{i_1} \to \overline{CORF}(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,2+1,k_{i_2}}))}(\bar{t}) = 1 \]

if and only if

\[ p_{i_1}(p_{i_2,1,j_2} + p_{i_2,2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_2,1,j_2} \overline{S} \]

\[ + p_{i_2,2,j_2+1,k_{i_2}} \overline{S} \text{ evaluates to } 1 \text{ on } \bar{t} \]

if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,1,j_2} + p_{i_2,2+1,k_{i_2}} = 0 \) on \( \bar{t} \),
2. \( \bar{t} \in FP(S) \) such that \( p_{i_2,1,j_2} = 1 \) on \( \bar{t} \), or
3. \( \bar{t} \in FP(S) \) such that \( p_{i_2,2,j_2+1,k_{i_2}} = 1 \) on \( \bar{t} \).

Hence, the result follows. \( \square \)

### 5.1.4 DORF with Other Term Faults

**Theorem 5.1.22** (DORF \( \times \) DORF - Case 1)

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two different subexpressions \((p_{i_1} + p_{i_1+1}) \) and \((p_{i_2} + p_{i_2+1}) \) in \( S \) are implemented as \( p_{i_1}p_{i_1+1} \) and \( p_{i_2}p_{i_2+1} \) respectively, where \( 1 \leq i_1 < i_1+1 < i_2 < m \), the resulting expression denoted as \( I_{DORF(p_{i_1} + p_{i_1+1} \to p_{i_1}p_{i_1+1}) \times DORF(p_{i_2} + p_{i_2+1} \to p_{i_2}p_{i_2+1})} \) is equivalent to Expression (4.22) in Table 4.8. Then, \( S \neq I_{DORF(p_{i_1} + p_{i_1+1} \to p_{i_1}p_{i_1+1}) \times DORF(p_{i_2} + p_{i_2+1} \to p_{i_2}p_{i_2+1})} \) if and
only if there is a test case \( \bar{t} \in ( \bigcup_{i=i_1,i_1+1,i_2,i_2+1} TP_i(S) ) \setminus ( \bigcup_{i=i_1+1} TP_i(S) ) \cup (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \cup (TP_{i_2}(S) \cap TP_{i_2+1}(S)) \).

66
Proof: First, we observe that $S \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \equiv \left((p_{i_1} \cdots p_{i_1+1}) \oplus (p_{i_1+1} \cdots p_{i_1+m})\right) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

$\equiv (p_{i_1} + p_{i_1+1} + p_{i_1+2}) \oplus (p_{i_1+1} + p_{i_1+2}) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

$\equiv (p_{i_1} + p_{i_1+1} + p_{i_1+2}) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

(By making use of $A \oplus B \oplus C \oplus D = 0$)

Now, $S(\bar{t}) \neq I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})$ if and only if $S(\bar{t}) \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m}) = 1$

if and only if

$\bar{t} \in \left( \bigcup_{i=i_1,i_1+1,i_1+2} TP_i(1) \right) \setminus \left( \bigcup_{i\neq i_1,i_1+1,i_1+2} TP_i(1) \right)$

if and only if

$\bar{t} \in \bigcup\left( TP_i(1) \cap TP_{i_1+1}(1) \right) \cup \bigcup\left( TP_{i_2}(1) \cap TP_{i_1+2}(1) \right)$

Hence, the result follows.

Theorem 5.1.23 (DORF $\times$ DORF - Case 2)

Let $S=p_1 \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the two sub-expressions $(p_{i_1} + p_{i_1+1})$ and $(p_{i_1+1} + p_{i_1+2})$ in $S$ are implemented as $p_{i_1} p_{i_1+1}$ and $p_{i_1+1} p_{i_1+2}$, respectively, where $1 \leq i_1 \leq m-2$, the resulting expression denoted as $I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})$ is equivalent to Expression (4.23) in Table 4.8. Then, $S \neq I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})$ if and only if there is a test case $\bar{t} \in \bigcup\left( TP_i(1) \setminus \bigcup_{i\neq i_1,i_1+1,i_1+2} TP_i(1) \right)$.

Proof: We observe that $S \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})$

$\equiv (p_{i_1} \cdots p_{i_1+1}) \oplus (p_{i_1+1} \cdots p_{i_1+m}) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

$\equiv (p_{i_1} \cdots p_{i_1+1}) \cdot (p_{i_1+1} \cdots p_{i_1+m}) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

$\equiv (p_{i_1} \cdots p_{i_1+1}) \cdot (p_{i_1+1} \cdots p_{i_1+m}) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m$

(By making use of $A \oplus B \oplus C \oplus D = 0$)

Now, $S(\bar{t}) \neq I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})(\bar{t})$ if and only if $S(\bar{t}) \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+m}) \oplus I_{DORF}(p_{i_1+1}+p_{i_1+2} \cdots p_{i_1+1+m})(\bar{t}) = 1$

if and only if

$\bar{t} \in \bigcup\left( TP_i(1) \setminus \bigcup_{i\neq i_1,i_1+1,i_1+2} TP_i(1) \right)$

if and only if

$\bar{t} \in \bigcup\left( TP_i(1) \cap TP_{i_1+1}(1) \right) \cup \bigcup\left( TP_{i_2}(1) \cap TP_{i_1+2}(1) \right)$.
Hence, the result follows. □

**Theorem 5.1.24 (DORF ⊗ CORF - Case 1)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $(p_i + p_{i+1})$ is implemented as $p_i p_{i+1}$ and the $i_2$-th term, $p_{i_2}$, is implemented as $p_{i_1} j_2 + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}}$ where $1 \leq i_1 < i_1 + 1 < i_2 \leq m$, $p_{i_2} = p_{i_2,1,j_2} p_{i_2,j_2+1,k_{i_2}}$ and $1 \leq j_2 < k_{i_2}$, the resulting expression denoted as $I_{DORF(p_1 + p_{i+1} - p_{i+1} p_{i+1})} \otimes CORF(p_{i_2} - p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})$ is equivalent to Expression (4.24) in Table 4.8. Then, $S \neq I_{DORF(p_1 + p_{i+1} - p_{i+1} p_{i+1})} \otimes CORF(p_{i_2} - p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_1}(S)$ such that $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} = 0$ on $\vec{t}$,
2. $\vec{t} \in UTP_{i_1+1}(S)$ such that $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} = 0$ on $\vec{t}$,
3. $\vec{t} \in FP(S)$, such that such that $p_{i_2,1,j_2} = 1$ on $\vec{t}$, or
4. $\vec{t} \in FP(S)$, such that such that $p_{i_2,j_2+1,k_{i_2}} = 1$ on $\vec{t}$.

**Proof:** First, we observe that $S \otimes I_{DORF(p_1 + p_{i+1} - p_{i+1} p_{i+1})} \otimes CORF(p_{i_2} - p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})$

$\equiv ((p_1 + p_{i+1} + p_{i_2})(p_1 p_{i+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1}$

$\cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_1 + p_{i+1} + p_{i_2})(p_{i_1+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) + (p_1 + p_{i+1} + p_{i_2}))$

$\cdot (p_1 p_{i+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_1 + p_{i+1} + p_{i_2}) \cdot (p_{i_1+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2})$

$\cdot (p_1 p_{i+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \cdot \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_1 + p_{i+1} + p_{i_2}) \cdot (p_{i_1+1} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot p_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1}$

$\cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1+1} \cdots \bar{p}_{i_2} \cdots \bar{p}_m$

(By making use of $(C + A \cdot B) \cdot \overline{A \cdot B} \equiv C \cdot \overline{A \cdot B} \text{ and } \overline{A \cdot B} \cdot (A \cdot B + C) \equiv \overline{A \cdot B} \cdot C)$

$\equiv (p_1 \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_1 \cdot \bar{p}_{i_1+1}) \cdot \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})$

$\cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \overline{S}$

(By rewriting $AB$ as $AB(A + B)$ because they are equivalent)

$\equiv (p_1 \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_1 \cdot \bar{p}_{i_1+1}) \cdot \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})$

$\cdot \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \overline{S}$

$\equiv (p_1 \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_1+1} \cdot \bar{p}_1 \cdot \bar{p}_{i_1+1}) \cdot \bar{p}_{i_2,1,j_2} \bar{p}_{i_2,j_2+1,k_{i_2}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_{i_2} \cdots \bar{p}_m$

$+ (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \overline{S}$

$\equiv p_1 \cdot (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1+1} (p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1} \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_2,1,j_2} \bar{p}_{i_2,j_2+1,k_{i_2}} \cdot \overline{S}$
Now, \[ S(\vec{t}) \neq I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+p_{i_1+1}}) \times CORF(p_{1_2} \cdots p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2}) \] if and only if

\[ S(\vec{t}) \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+p_{i_1+1}}) \times CORF(p_{1_2} \cdot p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2}) \] (\vec{t}) = 1

if and only if

\[ p_{i_1}(p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2} \cdot \vec{p}_{1_1} \cdots \vec{p}_{1_2,1,j_2} \cdots \vec{p}_{1_2,2,j_2} \cdots \vec{p}_{1_2,j_2}) + p_{i_1}(\vec{p}_{1_2,1,j_2} \cdots \vec{p}_{1_2,2,j_2} \cdots \vec{p}_{1_2,j_2} S) \]

evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2} = 0 \) on \( \vec{t} \),
2. \( \vec{t} \in UTP_{i_1+1}(S) \) such that \( p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2} = 0 \) on \( \vec{t} \),
3. \( \vec{t} \in FP(S) \), such that such that \( p_{1_2,1,j_2} = 1 \) on \( \vec{t} \), or
4. \( \vec{t} \in FP(S) \), such that such that \( p_{1_2,1,j_2} = 1 \) on \( \vec{t} \).

Hence, the result follows. □

**Theorem 5.1.25 (DORF × CORF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_{i_1} + p_{i_1+1}) \) in \( S \) is implemented as \( p_{i_1}p_{i_1+1} \) and the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,1,j_1} \cdots p_{i_1,1,k_1}, \) where \( 1 \leq i_1 < m, 1 \leq j_1 < k_1 \) and \( p_{i_1} = p_{i_1,1,j_1} \cdots p_{i_1,1,k_1} \), the resulting expression denoted as \( I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+p_{i_1+1}}) \times CORF(p_{1_2} \cdots p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2}) \) is equivalent to Expression (4.25) in Table 4.8. Then, \( S \neq I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+p_{i_1+1}}) \times CORF(p_{1_2} \cdots p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1+1}(S) \), such that \( p_{i_1,1,j_1} \cdots p_{i_1,1,k_1} = 0 \) on \( \vec{t} \), or
2. \( \vec{t} \in FP(S) \), such that \( p_{i_1,1,j_1} = 1 \) on \( \vec{t} \).

**Proof**: We observe that \( S \oplus I_{DORF}(p_{i_1}+p_{i_1+1} \cdots p_{i_1+p_{i_1+1}}) \times CORF(p_{1_2} \cdots p_{1_2,1,j_2} \cdots p_{1_2,2,j_2} \cdots p_{1_2,j_2}) \)

\[ \equiv \left((p_{i_1} + p_{i_1+1}) \oplus (p_{i_1,1,j_1} \cdots p_{i_1,1,k_1} \cdot p_{i_1+1}) + p_{i_1} + p_{i_1+1} + p_{i_1,1,j_1} \cdots p_{i_1,1,k_1} \cdot p_{i_1+1}\right) \]

\[ \equiv \left(p_{i_1} \cdot p_{i_1+1} \cdot \vec{p}_{i_1+1} \cdots \vec{p}_{i_1+2} \cdots \vec{p}_m \right) \]

\[ \equiv \left(\vec{p}_{i_1,1,j_1} \vec{p}_{i_1,1,k_1} + p_{i_1,1,j_1} \vec{p}_{i_1,1,k_1} \vec{p}_{i_1+1}\right) \cdot \vec{p}_{i_1} \cdots \vec{p}_{i_1+1} \cdots \vec{p}_{i_1+2} \cdots \vec{p}_m \]

(By making use of \((A \cdot B + C)(A + B \cdot C) \equiv A \cdot B \cdot C \) and \((A \cdot B + C)(A + B \cdot C) \equiv A \cdot B \cdot C \))

\[ \equiv \vec{p}_{i_1,1,j_1} \vec{p}_{i_1,1,k_1} \vec{p}_{i_1+1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_1+2} \cdots \vec{p}_m \]

\[ + p_{i_1,1,j_1} \vec{p}_{i_1,1,k_1} \vec{p}_{i_1+1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_1+2} \cdots \vec{p}_m \]

\[ \equiv \vec{p}_{i_1,1,j_1} \vec{p}_{i_1,1,k_1} \vec{p}_{i_1+1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_1+2} \cdots \vec{p}_m \]

(By rewriting \(AB \) as \(AB(A + B)\) because they are equivalent;
and $AB$ as $A \cdot (AB)$ because they are equivalent.

$\equiv \bar{p}_{i_1,j_1} \bar{p}_{i_1,j_1+1,k_{i_1}} p_{i_1+1} \cdot \bar{p}_1 \cdot \bar{p}_{i_1+2} \cdots \bar{p}_m + \bar{p}_{i_1,1,j_1} \cdot \bar{p}_1 \cdots \bar{p}_m$

$\equiv p_{i_1+1} \bar{p}_{i_1,1,j_1} \bar{p}_{i_1,j_1+1,k_{i_1}} \cdot \bar{p}_1 \cdots \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1,1,j_1} \cdot \bar{S}$

$\equiv p_{i_1+1}(\bar{p}_{i_1,1,j_1} + \bar{p}_{i_1,j_1+1,k_{i_1}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1,1,j_1} \cdot \bar{S}$

Now, $S(\bar{t}) \neq I_{\text{DORF}}(p_{i_1+p_{i_1+1}-p_{i_1,p_{i_1+1}}}) \cdot \text{CORF}(p_{i_1-p_{i_1,1,j_1}+p_{i_1,j_1+1,k_{i_1}}})(\bar{t})$ if and only if $S(\bar{t}) + I_{\text{DORF}}(p_{i_1+p_{i_1+1}-p_{i_1,p_{i_1+1}}}) \cdot \text{CORF}(p_{i_1-p_{i_1,1,j_1}+p_{i_1,j_1+1,k_{i_1}}})(\bar{t}) = 1$

if and only if $p_{i_1+1}(\bar{p}_{i_1,1,j_1} + \bar{p}_{i_1,j_1+1,k_{i_1}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1,1,j_1} \cdot \bar{S}$ evaluates to 1 on $\bar{t}$

if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1+1}(S)$ such that $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} = 0$ on $\bar{t}$, or

2. $\bar{t} \in FP(S)$, such that $p_{i_1,1,j_1} = 1$ on $\bar{t}$.

Hence, the result follows. \(\square\)

### 5.1.5 CORF with Other Term Faults

**Theorem 5.1.26 (CORF $\times$ CORF - Case 1)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two different terms $p_i$ and $p_j$ in $S$ are implemented as $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}$ respectively, where $1 \leq i_1 < i_2 \leq m$, $p_{i_1} = p_{i_1,1,j_1} : p_{i_1,j_1+1,k_{i_1}}$, $p_{i_2} = p_{i_2,1,j_2} : p_{i_2,j_2+1,k_{i_2}}$, $1 \leq j_1 < k_{i_1}$ and $1 \leq j_2 < k_{i_2}$, the resulting expression denoted as $I_{\text{CORF}}(p_{i_1-p_{i_1,1,j_1}+p_{i_1,j_1+1,k_{i_1}}}) \cdot \text{CORF}(p_{i_2-p_{i_2,1,j_2}+p_{i_2,j_2+1,k_{i_2}}})$ is equivalent to Expression (4.26) in Table 4.8. Then, $S \neq I_{\text{CORF}}(p_{i_1-p_{i_1,1,j_1}+p_{i_1,j_1+1,k_{i_1}}}) \cdot \text{CORF}(p_{i_2-p_{i_2,1,j_2}+p_{i_2,j_2+1,k_{i_2}}})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in FP(S)$ such that $p_{i_1,1,j_1} = 1$ on $\bar{t}$,

2. $\bar{t} \in FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$ on $\bar{t}$,

3. $\bar{t} \in FP(S)$ such that $p_{i_2,1,j_2} = 1$ on $\bar{t}$, or

4. $\bar{t} \in FP(S)$ such that $p_{i_2,j_2+1,k_{i_2}} = 1$ on $\bar{t}$.

**Proof**: First, we observe that $S + I_{\text{CORF}}(p_{i_1-p_{i_1,1,j_1}+p_{i_1,j_1+1,k_{i_1}}}) \cdot \text{CORF}(p_{i_2-p_{i_2,1,j_2}+p_{i_2,j_2+1,k_{i_2}}})$

$\equiv (p_1 + p_2) \cdot (p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}})$

$\cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2} \cdots \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv (p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) + (p_1 + p_2)$

$\cdot (p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_{i_2} \cdots \bar{p}_m$
\[ \equiv ((p_{i_1} + p_{i_2}) \bar{p}_{i_1,j_1} \cdot \bar{p}_{i_1,j_1+1,k_{i_1}} \cdot \bar{p}_{i_2,j_2} \cdot \bar{p}_{i_2,j_2+1,k_{i_2}} + (\bar{p}_{i_1} \cdot \bar{p}_{i_2})
\]
\[ \cdot (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})) \cdot \bar{p}_{i_1+1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_{m} \]
\[ \equiv (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}}) \bar{p}_{i_1} \bar{p}_{i_2} \cdot \bar{p}_{i_1} \cdot \bar{p}_{i_2+1} \cdot \bar{p}_{m} \]
\[ \text{(By making use of } (AB + CD)(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) \equiv 0) \]
\[ \equiv (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}}) \cdot \bar{S} \]
\[ \equiv p_{i_1,j_1} \cdot \bar{S} + p_{i_1,j_1+1,k_{i_1}} \cdot \bar{S} + p_{i_2,j_2} \cdot \bar{S} + p_{i_2,j_2+1,k_{i_2}} \cdot \bar{S} \]

Now,
\[ S(\bar{t}) \neq I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \otimes_{\text{CORF}}(p_2 \rightarrow p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})(\bar{t}) \]
if and only if
\[ S(\bar{t}) \oplus I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \otimes_{\text{CORF}}(p_2 \rightarrow p_{i_2,j_2} + p_{i_2,j_2+1,k_{i_2}})(\bar{t}) = 1 \]
if and only if
\[ p_{i_1,j_1} \cdot \bar{S} + p_{i_1,j_1+1,k_{i_1}} \cdot \bar{S} + p_{i_2,j_2} \cdot \bar{S} + p_{i_2,j_2+1,k_{i_2}} \cdot \bar{S} \]
evaluates to 1 on \( \bar{t} \)
if and only if \( \bar{t} \) satisfies following conditions:

1. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1} = 1 \) on \( \bar{t} \),
2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_{i_1}} = 1 \) on \( \bar{t} \),
3. \( \bar{t} \in FP(S) \) such that \( p_{i_2,j_2} = 1 \) on \( \bar{t} \), or
4. \( \bar{t} \in FP(S) \) such that \( p_{i_2,j_2+1,k_{i_2}} = 1 \) on \( \bar{t} \).

Hence, the result follows. \( \square \)

**Theorem 5.1.27 (CORF \( \otimes \) CORF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two different CORFs are committed at a particular term in \( S \). That is, the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_1+1,k_{i_1}} \) where \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,j_2} \cdot p_{i_1,j_1+1,k_{i_1}} \) and \( 1 \leq j_1 < j_2 < k_{i_1} \), the resulting expression denoted as \( I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \otimes_{\text{CORF}}(p_2 \rightarrow p_{i_1,j_2} + p_{i_1,j_2+1,k_{i_1}}) \) is equivalent to Expression \( (4.27) \) in Table 4.8.

Then, \( S \neq I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \otimes_{\text{CORF}}(p_2 \rightarrow p_{i_1,j_2} + p_{i_1,j_2+1,k_{i_1}}) \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1} = 1 \) on \( \bar{t} \),
2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,j_2} = 1 \) on \( \bar{t} \), or
3. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_2+1,k_{i_1}} = 1 \) on \( \bar{t} \).

**Proof:** First, we observe that
\[ S \oplus I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \otimes_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_2} + p_{i_1,j_2+1,k_{i_1}}) \]
\[ \equiv (p_{i_1} \oplus (p_{i_1,i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_1})) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1}(p_{i_1,i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_1})) + \bar{p}_1(p_{i_1,i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_1})) \]
\[ \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1,i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_1}) \cdot \bar{p}_1 \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m \]

(By making use of \((ABC)(A+B+C) \equiv 0\))
\[ \equiv (p_{i_1,i_1,j_1} + p_{i_1,j_1+1,j_2} + p_{i_1,j_2+1,k_1}) \cdot \bar{S} \]
\[ \equiv p_{i_1,i_1,j_1} \cdot S + p_{i_1,j_1+1,j_2} \cdot S + p_{i_1,j_2+1,k_1} \cdot \bar{S} \]

Now, \[ S(\vec{t}) \neq I_{CORF}(p_{i_1} \rightarrow p_{i_1,i_1,j_1} + p_{i_1,j_1+1,k_1}) \cdot \bar{S} \]
if and only if \[ S(\vec{t}) \oplus I_{CORF}(p_{i_1} \rightarrow p_{i_1,i_1,j_1} + p_{i_1,j_1+1,k_1}) \cdot \bar{S} \]
\[ = 1 \]
if and only if \[ p_{i_1,i_1,j_1} \cdot S + p_{i_1,j_1+1,j_2} \cdot S + p_{i_1,j_2+1,k_1} \cdot \bar{S} \]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies following conditions:

1. \( t_1 \in FP(S) \) such that \( p_{i_1,i_1,j_1} = 1 \) on \( t_1 \),
2. \( t_2 \in FP(S) \) such that \( p_{i_1,j_1+1,j_2} = 1 \) on \( t_1 \), or
3. \( t_3 \in FP(S) \) such that \( p_{i_1,j_2+1,k_1} = 1 \) on \( t_1 \).

Hence, the result follows. \( \square \)

5.2 Detection conditions of 4 Remaining Double-fault Expressions with Ordering

As discussed in previous chapter, for 25 types of fault classes of double fault related to terms with ordering, 49 out 53 all possible resulting double-fault expressions have their equivalent counterparts in the 27 double-fault expressions for double fault related to terms only without ordering. The detection conditions of remaining 4 double-fault expressions related to terms which do not have their equivalent counterparts in double fault without ordering are studied in the rest of the section.

\textbf{Theorem 5.2.1 (TOF \( \bowtie \) DORF)}

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), is omitted and then the subexpression \((p_{i_1-1} + p_{i_1+1})\) is implemented as \( p_{i_1-1}p_{i_1+1} \) where \( 1 < i_1 < m \), the resulting expression denoted as \( I_{TOF(p_{i_1}- \bowtie DORF)(p_{i_1-1} + p_{i_1+1} - p_{i_1-1}p_{i_1+1})} \) is equivalent to that given by Expression (4.53) in Table 4.8. Then, \( S \neq I_{TOF(p_{i_1}- \bowtie DORF)(p_{i_1-1} + p_{i_1+1} - p_{i_1-1}p_{i_1+1})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:
1. \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1, i_1+1}^{m} TP_{i}(S) \right) \), or

2. \( \vec{t} \in (TP_{i_1}(S) \cup TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1, i_1+1}^{m} TP_{i}(S) \right) \).

**Proof:** First, we observe that

\( S \oplus I_{TOF(p_{i_1} \rightarrow \mathbb{DORF}(p_{i_1-1}+p_{i_1+1} \rightarrow p_{i_1-1} p_{i_1+1}))} \)

\( = ((p_{i_1-1} + p_{i_1} + p_{i_1+1}) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} + p_{i_1-1} p_{i_1+1} p_{i_1+1}) \cdot \vec{p}_m \)

\( = (p_{i_1-1} p_{i_1+1} + p_{i_1+1} + 0) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} \cdot \vec{p}_m \)

\( = (p_{i_1-1} + p_{i_1}) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} \cdot \vec{p}_m + (p_{i_1-1} + p_{i_1}) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} \cdot \vec{p}_m \)

Now,

\( S(\vec{t}) \neq I_{TOF(p_{i_1} \rightarrow \mathbb{DORF}(p_{i_1-1}+p_{i_1+1} \rightarrow p_{i_1-1} p_{i_1+1}))}(\vec{t}) \)

if and only if

\( S(\vec{t}) \oplus I_{TOF(p_{i_1} \rightarrow \mathbb{DORF}(p_{i_1-1}+p_{i_1+1} \rightarrow p_{i_1-1} p_{i_1+1}))}(\vec{t}) = 1 \)

if and only if

\( (p_{i_1-1} + p_{i_1}) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} \cdot \vec{p}_m + (p_{i_1-1} + p_{i_1}) \cdot \vec{p}_i \cdot \vec{p}_{i-1} \cdot \vec{p}_{i+1} \cdot \vec{p}_m \)

\( \vec{p}_{i+1} \cdot \vec{p}_m \)

evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies following conditions:

1. \( \vec{t} \in (TP_{i_1-1}(S) \cup TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1, i_1+1}^{m} TP_{i}(S) \right) \), or

2. \( \vec{t} \in (TP_{i_1+1}(S) \cup TP_{i_1+2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_1+1}^{m} TP_{i}(S) \right) \).

Hence, the result follows.

\( \square \)

**Theorem 5.2.2** *(CORF with ENF)*

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1} k_i \) where \( 1 \leq i_1 < h_1 < m \), \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1} k_i \) and \( 1 \leq j_1 < k_i \), and then the subexpression \( p_{i_1,j_1+1} k_1 + \cdots + p_{i_1} \) in \( S \) is negated, the resulting expression denoted as

\[
I_{CORF(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1} k_i)} \cdot ENF(p_{i_1,j_1+1} k_1 + \cdots + p_{i_1})
\]

is equivalent to Expression (4.70) in Table 4.8. Then, \( S \neq I_{CORF(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1} k_i)} \cdot ENF(p_{i_1,j_1+1} k_1 + \cdots + p_{i_1}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions
1. \( \bar{t} \in \left( \bigcup_{i=i_1+1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1+1}^{i_1} TP_i(S) \right) \) such that \( p_{i_1,j_1} = 0 \) on \( \bar{t} \), or

2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_{i_1}} = 0 \) on \( \bar{t} \).

**Proof :** First, we observe that

\[
S \equiv I_{CORF(p_{i_1} + p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}})} \times \operatorname{ENV}(p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1,j_1+1,k_{i_1}})
\]

\[
\equiv \left( (p_{i_1} + \cdots + p_{h_1}) + (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv \left( (p_{i_1} + \cdots + p_{h_1})(p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1}) + \bar{p}_1 \cdot \bar{p}_{i_1+1} + \cdots + \bar{p}_{h_1}(p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv \left( (p_{i_1} + \cdots + p_{h_1})(p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1}) + \bar{p}_1 \cdot \bar{p}_{i_1+1} + \cdots + \bar{p}_{h_1}(p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{h_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

(By making use of \( (AB + C)\overline{A} \equiv \overline{AC} \))

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

(By making use of \( (A + \overline{B})C(A + \overline{BC}) \equiv \overline{BC} \))

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
+ \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \left( \bar{p}_{i_1,j_1} + \bar{p}_{i_1,j_1+1,k_{i_1}} \right) \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

(By rewriting \( A \) as \( A(A + B) \) because they are equivalent; and \( B \) as \( B(A + B) \) because they are equivalent)

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \left( \bar{p}_{i_1,j_1} \bar{p}_{i_1,j_1+1,k_{i_1}} \right) \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m
\]

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{i_1} \cdot \bar{p}_{h_1+1} \cdots \bar{p}_m + \bar{p}_{i_1,j_1+1,k_{i_1}} \bar{p}_1 \cdots \bar{p}_{i_1} \cdots \bar{p}_m
\]

\[
\equiv \bar{p}_{i_1,j_1}(p_{i_1+1} + \cdots + p_{h_1}) \bar{p}_1 \cdots \bar{p}_{h_1+1} \cdots \bar{p}_m + \bar{p}_{i_1,j_1+1,k_{i_1}} \nabla
\]

74
Now, \( S(\vec{t}) \not\equiv I_{CORF}(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \times ENF(p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{n,1,j_1+1,k_{i_1}})(\vec{t}) \)

if and only if \( S(\vec{t}) \not\equiv I_{CORF}(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \times ENF(p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{n,1,j_1+1,k_{i_1}})(\vec{t}) = 1 \)

if and only if \( \bar{p}_{i_1,1,j_1}(p_{i_1,1+1} + \cdots + p_{n,1})\bar{p}_1 \cdot \bar{p}_{h_1+1} \cdot \cdots \bar{p}_m \) \( \bar{p}_{i_1,1,j_1+1,k_{i_1}} \cdot \bar{S} \)

evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions

1. \( \vec{t} \in \left( \bigcup_{i=i_1+1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1+1, \cdots, h_1}^{m} TP_i(S) \right) \) such that \( p_{i_1,1,j_1} = 0 \) on \( \vec{t} \), or

2. \( \vec{t} \in FP(S) \) such that \( p_{i_1,1,j_1+1,k_{i_1}} = 0 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

**Theorem 5.2.3 (CORF with TNF)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}} \) where \( 1 \leq i_1 \leq m \), \( p_{i_1,1,j_1+1,k_{i_1}} \) is negated, the expression denoted as \( I_{CORF}(p_{i_1,1,j_1+1,k_{i_1}}) \times ENF(p_{i_1,1,j_1+1,k_{i_1}}) \) is equivalent to Table 4.8. Then,

\( S \not\equiv I_{CORF}(p_{i_1,1,j_1+1,k_{i_1}}) \times ENF(p_{i_1,1,j_1+1,k_{i_1}}) - p_{i_1,1,j_1+1,k_{i_1}} \)

if and only if there is a test case \( \vec{t} \in FP(S) \) such that \( p_{i_1,1,j_1+1,k_{i_1}} = 0 \) on \( \vec{t} \).

**Proof:** First, we observe that

\[
S \equiv I_{CORF}(p_{i_1,1,j_1+1,k_{i_1}}) \times ENF(p_{i_1,1,j_1+1,k_{i_1}} - p_{i_1,1,j_1+1,k_{i_1}})
\equiv ((p_{i_1}) \cup (p_{i_1},1,j_1 + p_{i_1},1,j_1+1,k_{i_1})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv ((p_{i_1})(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) + (p_{i_1})(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv (p_{i_1}(p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv (p_{i_1,1,j_1+1,k_{i_1}} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\text{(By making use of } AB(\bar{A}B) \equiv 0 \text{ and } (\bar{A}B)(A + \bar{B}) \equiv \bar{B})
\equiv (p_{i_1,1,j_1+1,k_{i_1}} \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot (p_{i_1,1,j_1} + p_{i_1,1,j_1+1,k_{i_1}}) \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\text{(By rewriting } A \text{ as } A(A + B) \text{ because they are equivalent})
\equiv (p_{i_1,1,j_1+1,k_{i_1}} \bar{p}_1 \cdots \bar{p}_{i_1-1} \cdot \bar{p}_1 \cdot \bar{p}_{i_1+1} \cdots \bar{p}_m
\equiv (p_{i_1,1,j_1+1,k_{i_1}} \bar{p}_1 \cdots \bar{p}_m
\equiv (p_{i_1,1,j_1+1,k_{i_1}} \cdot \bar{S})
Now, \( S(\vec{t}) \neq I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TNF}} (p_{i_1,j_1+1,k_i} \to p_{i_1,j_1+1,k_i}) \)(\(\vec{t}\))
if and only if
\[
S(\vec{t}) \oplus I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TNF}} (p_{i_1,j_1+1,k_i} \to p_{i_1,j_1+1,k_i}) \)(\(\vec{t}\)) = 1
\]
if and only if \( (p_{i_1,j_1+1,k_i}) \cdot S \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1+1,k_i} = 0 \) on \( \vec{t} \).
Hence, the result follows.

**Theorem 5.2.4 (CORF with TOF)**
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_i} \) where \( 1 \leq i_1 \leq m \), \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_i} \), and \( 1 < j_1 < k_i \), and then the term \( p_{i_1,j_1+1,k_i} \) in \( S \) is omitted from the expression, the resulting expression denoted as \( I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TOF}} (p_{i_1,j_1+1,k_i} \to -) \) is equivalent to Expression (4.76) in Table 4.8. Then, \( S \neq I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TOF}} (p_{i_1,j_1+1,k_i} \to -) \) if and only if there is a test case \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1} = 0 \) on \( \vec{t} \).

**Proof**: First, we observe that \( S \oplus I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TOF}} (p_{i_1,j_1+1,k_i} \to -) \)
\[
\equiv \left( (p_{i_1}) \oplus (p_{i_1,j_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} p_{i_1,j_1} \cdots p_m
\]
\[
\equiv \left( (p_{i_1}) (\bar{p}_{i_1,j_1}) + (p_{i_1,j_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} p_{i_1,j_1} \cdots p_m
\]
\[
\equiv \left( 0 + (\bar{p}_1) (p_{i_1,j_1}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} p_{i_1,j_1} \cdots p_m
\]
\[
\equiv p_{i_1,j_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} p_{i_1,j_1} \cdots p_m
\]
\[
\equiv p_{i_1,j_1} \cdot S
\]
Now, \( S(\vec{t}) \neq I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TNF}} (p_{i_1,j_1+1,k_i} \to p_{i_1,j_1+1,k_i}) \)(\(\vec{t}\))
if and only if
\[
S(\vec{t}) \oplus I_{\text{CORF}}(p_{i_1} \to p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \times_{\text{TNF}} (p_{i_1,j_1+1,k_i} \to p_{i_1,j_1+1,k_i}) \)(\(\vec{t}\)) = 1
\]
if and only if \( p_{i_1,j_1} \cdot S \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1} \) evaluate to 1.
Hence, the result follows.

For ease of reading and understanding, we list all double fault classes, the corresponding double-fault expression numbers and their corresponding fault detection conditions in Table 5.2. For example, the third row of Table 5.2 presents the detection conditions of two double-fault expressions of ENF \( \times \) TOF. For double-fault expression (4.5) (please refer to Table 4.8 for the actual double-fault expression), the detection condition shows that any true point of \( S \) in \(( \bigcup_{i=1}^{h_1} \text{TP}_i(S) \setminus \bigcup_{i=1}^{m} \text{TP}_i(S) \) or any false point of \( S \) can distinguish \( S \) and double-fault expression (4.5). While for double-fault expression (4.6) in Table 4.8, the detection
condition shows that any true point of $S$ in $\bigcup_{i=1}^{h_1} TP_i(S) \setminus \bigcup_{i \neq i_2}^{m} TP_i(S)$ or any false point of $S$ can distinguish $S$ and double-fault expression (4.6).
Table 5.2: Double fault, double-fault expression and detection condition ($S = p_1 + \ldots + p_m$)

(a) Double-fault expressions (numbered 4.1 – 4.14) due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Expression No.: Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF × ENF</td>
<td>(4.1): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>ENF × TNF</td>
<td>(4.2): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>ENF × TOF</td>
<td>(4.3): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>ENF × DORF</td>
<td>(4.4): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>ENF × CORF</td>
<td>(4.5): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>TNF × TNF</td>
<td>(4.6): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>TNF × TOF</td>
<td>(4.7): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>TNF × DORF</td>
<td>(4.8): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>TNF × CORF</td>
<td>(4.9): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
<tr>
<td>TNF × TOF</td>
<td>(4.10): (C1) any point in $\left( \bigcup_{i=1}^{l_1} TP_i(S) \right) \cap \left( \bigcup_{i=2}^{l_2} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $\left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or $(C2)$ any point in $FP(S)$.</td>
</tr>
</tbody>
</table>

78
(a) Double-fault expressions (numbered 4.15 – 4.27) due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.)</th>
<th>Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DORF × DORF</td>
<td>(4.15): (C1) any point in $\bigcap_{i_{1},i_{2},i_{2}+1} TP_{1}(S) \cup (TP_{i_{2}}(S) \cap TP_{i_{2}+1}(S))$, or (C2) any point in $FP(S)$.</td>
<td></td>
</tr>
<tr>
<td>CORF × CORF</td>
<td>(4.16): (C1) any point in $\bigcap_{i=1}^{m} TP_{i}(S)$.</td>
<td></td>
</tr>
<tr>
<td>DORF × DORF</td>
<td>(4.17): (C1) any point in $UTP_{i_{1}}(S)$ such that $p_{i_{2},j_{2}} = 0$, or (C2) any point in $FP(S)$.</td>
<td></td>
</tr>
<tr>
<td>CORF × CORF</td>
<td>(4.18): (C1) any point in $UTP_{i_{1}}(S)$, or (C2) any point in $FP(S)$ such that $p_{i_{1},j_{1}+1,k_{i_{1}}}$ = 0.</td>
<td></td>
</tr>
<tr>
<td>TOF × TOF</td>
<td>(4.19): (C1) any point in $\bigcap_{i=1}^{m} TP_{i}(S)$.</td>
<td></td>
</tr>
<tr>
<td>DORF × DORF</td>
<td>(4.20): (C1) any point in $\bigcap_{i=1}^{m} TP_{i}(S)$, or (C2) any point in $FP(S)$ such that $p_{i_{2},j_{2}} = 1$, or (C3) any point in $FP(S)$ such that $p_{i_{2},j_{2}+1,k_{i_{2}}}$ = 1.</td>
<td></td>
</tr>
<tr>
<td>DORF × DORF</td>
<td>(4.21): (C1) any point in $UTP_{i_{1}}(S)$ such that $p_{i_{2},j_{2}} = 0$, or (C2) any point in $FP(S)$ such that $p_{i_{2},j_{2}} = 1$, or (C3) any point in $FP(S)$ such that $p_{i_{2},j_{2}+1,k_{i_{2}}}$ = 1.</td>
<td></td>
</tr>
<tr>
<td>DORF × DORF</td>
<td>(4.22): (C1) any point in $\bigcap_{i=1}^{m} TP_{i}(S)$ such that $p_{i_{1},j_{1}} = 0$, or (C2) any point in $FP(S)$ such that $p_{i_{1},j_{1}} = 1$, or (C3) any point in $FP(S)$ such that $p_{i_{1},j_{1}+1,k_{i_{1}}}$ = 0, or (C4) any point in $FP(S)$ such that $p_{i_{1},j_{1}+1,k_{i_{1}}}$ = 1.</td>
<td></td>
</tr>
<tr>
<td>CORF × CORF</td>
<td>(4.23): (C1) any point in $\bigcap_{i=1}^{m} TP_{i}(S)$ such that $p_{i_{1},j_{1}} = 1$, or (C2) any point in $FP(S)$ such that $p_{i_{1},j_{1}+1,k_{i_{1}}}$ = 1, or (C3) any point in $FP(S)$ such that $p_{i_{1},j_{1}+1,k_{i_{1}}}$ = 1.</td>
<td></td>
</tr>
</tbody>
</table>
(b) Four double-fault expressions due to double faults with ordering that do not have equivalent counterparts in double-fault expressions due to double fault without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOF DORF</td>
<td>(4.53):(C1) any point in ( (TP_{i_1-1}(S) \cup TP_{i_1}(S)) \setminus \bigcup_{i \neq i_1-1, i_1}^m TP_i(S) ), or (C2) any point in ( (TP_{i_1}(S) \cup TP_{i_1+1}(S)) \setminus \bigcup_{i \neq i_1, i_1+1}^m TP_i(S) ).</td>
</tr>
<tr>
<td>CORF ENF</td>
<td>(4.70):(C1) any point in ( \left( \bigcup_{i = i_1+1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1+1, \ldots, h_1} TP_i(S) \right) ) such that ( p_{i_1,1,j_1} = 0 ), or (C2) any point in ( FP(S) ) such that ( p_{i_1, j_1+1, k_{i_1}} = 0 ).</td>
</tr>
<tr>
<td>CORF TNF</td>
<td>(4.73):(C1) any point in ( FP(S) ) such that ( p_{i_1,j_1+1,k_{i_1}} = 0 ).</td>
</tr>
<tr>
<td>CORF TOF</td>
<td>(4.76):(C1) any point in ( FP(S) ) such that ( p_{i_1,1,j_1} = 1 ).</td>
</tr>
</tbody>
</table>
Chapter 6

Double Faults Related to Literals Only

In this chapter, double faults related to literals only are studied. A *double fault related to literals only* is a double fault in which two individual faults of the double fault are literal faults within Boolean expressions. For easy reference, we used double literal faults instead. Four literal fault classes are considered in this work, they are LNF, LOF, LIF and LRF. Since the ordering of the occurrences of two single faults in a double fault may result in different faulty expressions, double literal faults are studied from two cases, double fault with and without ordering. As a reminder, double faults without ordering refers to the situation that two faults involved in a double fault may be independent of each other while double faults with ordering refers to the situation that two faults occur one after the other in such a way that the first fault may affect the occurrence of the second.

6.1 Double Faults without Ordering

In this section, different types of double literal fault without ordering are introduced. Given a Boolean expression $S$, suppose that two single fault classes $F_1$ and $F_2$ are committed in $S$ changing its subexpressions $E_1$ and $E_2$ to $E'_1$ and $E'_2$, respectively, the resulting double-fault implementation is denoted by $I_{F_1(E_1→E'_1),F_2(E_2→E'_2)}$ because their order of occurrence will result in the same faulty implementation. Table 6.1 lists all 10 types of double faults without ordering for those four single fault classes related to literals within Boolean expressions. In the rest of this section, these 10 double fault classes and their corresponding faulty implementations are discussed.

6.1.1 LNF with Other Literals Faults

**LNF and LNF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x^i_{j_1}$ and $x^i_{j_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$ of $S$, respectively. Sup-
Table 6.1: Types of double literal faults without ordering

<table>
<thead>
<tr>
<th></th>
<th>LNF</th>
<th>LOF</th>
<th>LIF</th>
<th>LRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOF</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIF</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>LRF</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
</tbody>
</table>

pose that both literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are negated. We use $I_{LNF(p_{i_1} \rightarrow p_{i_1,j_1}), LNF(p_{i_2} \rightarrow p_{i_2,j_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$ and $k_{i_2} > 1$. Furthermore, without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m$$ (6.1)

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do not consider the situation when the two literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ are exactly the same (that is, $j_1 = j_2$). It is because when a literal is negated twice, the implementation is then equivalent to the original expression $S$. Without loss of generality, we can assume $j_1 < j_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m$$ (6.2)

where $p_{i_1,j_1,j_2} = x_{j_1}^{i_1} \cdots \bar{x}_{j_1}^{i_1} \cdots x_{j_2}^{i_1} \cdots x_{k_{i_1}}^{i_1}$ denotes the term obtained from $p_{i_1}$ by negating its literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ ($j_1 < j_2$).

**LNF and LOF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively. Suppose that the literal $x_{j_2}^{i_2}$ is negated and the literal $x_{j_2}^{i_2}$ is omitted. We use $I_{LNF(p_{i_1} \rightarrow p_{i_1,j_1}), LOF(p_{i_2} \rightarrow p_{i_2,j_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$ and $k_{i_2} > 1$. Without loss of generality, we can further assume $i_1 < i_2$. Otherwise, we can always interchange the two terms so that the

---

1. Otherwise, if a term contains just one literal, the negation fault is considered to be a term negation fault, which is studied in other reports and is out of the scope of this report. In the sequel, we will make similar notes related to negation and omission faults, and the reason is similar.

2. This implies $k_{i_1} > 1$ because $1 \leq j_1 < j_2 \leq k_{i_1}$.
term with LOF comes after that of LNF.\footnote{In the rest of this thesis, we will make similar assumption whenever we encounter situations with two faults committed at two different terms.} The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \]  \hspace{1cm} (6.3)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). We do not consider the situation when the two literals \( x_{j_1}^{i_1} \) and \( x_{j_2}^{i_1} \) are exactly the same (that is, \( j_1 = j_2 \)). It is because when \( x_{j_1}^{i_1} \) is first negated and then omitted, the implementation is then equivalent to a single LOF with respect to the original expression \( S \). On the other hand, if \( x_{j_2}^{i_1} \) is first omitted, it is impossible to negate it afterwards. As a result, \( x_{j_1}^{i_1} \) and \( x_{j_2}^{i_1} \) are two different literals. Without loss of generality, we can assume \( j_1 < j_2 \). Otherwise, we can always interchange the two literals, so that the literal with LOF comes after that of LNF.\footnote{In the rest of this thesis, we will make similar assumption whenever we encounter situations with two faults committed at two different literals in the same term.} The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m \]  \hspace{1cm} (6.4)

where \( p_{i_1,j_1,j_2} = x_{j_1}^{i_1} \cdots \bar{x}_{j_1}^{i_1} \cdots x_{j_2}^{i_1} x_{j_2+1}^{i_1} \cdots x_{k_{i_1}}^{i_1} \) denotes the term obtained from \( p_{i_1} \) by negating its literal \( x_{j_1}^{i_1} \) and omitting its literal \( x_{j_2}^{i_1} \) (\( j_1 < j_2 \)).

**LNF and LIF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( x_{j_1}^{i_1} \) be a literal in the \( i_1 \)-th term, \( p_{i_1} \), of \( S \) and \( x_{i_2} \) be a missing literal of the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose that the literal \( x_{j_1}^{i_1} \) is negated and the literal \( x_{i_2} \) is inserted into \( p_{i_2} \). We use \( I_{LNF(p_{i_1} \rightarrow p_{i_1,j_1})} \), \( LIF(p_{i_2} \rightarrow p_{i_2} x_{i_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). It should be noted that \( k_{i_1} > 1 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2} x_{i_2} + \cdots + p_m \]  \hspace{1cm} (6.5)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} x_{i_2} + \cdots + p_m \]  \hspace{1cm} (6.6)
LNF and LRF  Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively, and $x_{l_2}$ be a missing literal of $p_{i_2}$ in $S$. Suppose the literal $x_{j_1}^{i_1}$ is negated and the literal $x_{j_2}^{i_2}$ is replaced by the literal $x_{l_2}$. We use $I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_1,j_1})$, $I_{\text{LRF}}(p_{i_2} \rightarrow p_{i_2,j_2} x_{l_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} x_{l_2} + \cdots + p_m \quad (6.7)$$

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do not consider the situation when the two literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ are exactly the same (that is, $j_1 = j_2$). It is because when $x_{j_1}^{i_1}$ is first negated and then replaced by $x_{l_2}$, the implementation is then equivalent to a single LRF with respect to the original expression $S$. On the other hand, if $x_{j_2}^{i_1}$ is first replaced by $x_{l_2}$, it is impossible to negate it afterwards. Without loss of generality, we can assume $j_1 < j_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2} x_{l_2} + \cdots + p_m \quad (6.8)$$

where $p_{i_1,j_1,j_2} x_{l_2} = x_1^{i_1} \cdots \bar{x}_{j_1}^{i_1} \cdots x_{j_2-1}^{i_1} x_{j_2+1}^{i_1} \cdots x_{k_{i_1}}^{i_1} x_{l_2}$ denotes the term obtained from $p_{i_1}$ by negating its literal $x_{j_1}^{i_1}$ and replacing its literal $x_{j_2}^{i_1}$ with $x_{l_2}(j_1 < j_2)$.

6.1.2 LOF with Other Literals Faults

LOF and LOF  Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively. Suppose that both literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are omitted. We use $I_{\text{LOF}}(p_{i_1} \rightarrow p_{i_1,j_1})$, $I_{\text{LOF}}(p_{i_2} \rightarrow p_{i_2,j_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$ and $k_{i_2} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (6.9)$$
Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do not consider the situation when the two literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are exactly the same (that is, $j_1 = j_2$). It is because when a literal is first omitted, it is impossible to omit it afterwards. Furthermore, we do not consider the situation when $p_{i_1}$ contains just only two literals because the net effect will be equivalent to a term omission fault, which is a single term fault. Without loss of generality, we can assume $j_1 < j_2$ and $k_{i_1} > 2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m \quad (6.10)$$

where $p_{i_1,j_1,j_2} = x_1^{i_1} \cdots x_{j_1-1}^{i_1} x_{j_1+1}^{i_1} \cdots x_{j_2-1}^{i_1} x_{j_2+1}^{i_1} \cdots x_{k_{i_1}}^{i_1}$ denotes the term obtained from $p_{i_1}$ by omitting its literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ ($j_1 < j_2$).

**LOF and LIF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ be a literal in the $i_1$-th term, $p_{i_1}$, of $S$, and $x_{i_2}$ be a missing literal of the $i_2$-th term, $p_{i_2}$, of $S$. Suppose the literal $x_{j_1}^{i_1}$ is omitted and the literal $x_{i_2}$ is inserted into $p_{i_2}$ of $S$. We use $I_{LOF(p_{i_1} \leftarrow p_{i_1,j_1})}, LIF(p_{i_2} \leftarrow p_{i_2,x_{i_2}})$ to denote the corresponding faulty implementation.

We do not consider the situation when the LOF and LIF are committed at the same term (that is, $i_1 = i_2$). It is because when $x_{j_1}^{i_1}$ is omitted from $p_{i_1}$ and $x_{i_2}$ is inserted into $p_{i_2}$, the implementation is then equivalent to a single LRF with respect to the original expression $S$. Therefore, two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_m \quad (6.11)$$

**LOF and LRF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively, and $x_{i_2}$ be a missing literal of the $p_{i_2}$ in $S$. Suppose the literal $x_{j_1}^{i_1}$ is omitted and the literal $x_{j_2}^{i_2}$ is replaced by the literal $x_{i_2}$. We use $I_{LOF(p_{i_1} \leftarrow p_{i_1,j_1})}, LRF(p_{i_2} \leftarrow p_{i_2,x_{i_2}})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2,x_{i_2}} + \cdots + p_m \quad (6.12)$$

85
Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do not consider the situation when the two literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ are exactly the same (that is, $j_1 = j_2$). It is because when a literal is first omitted, it is impossible to replace it afterwards. On the other hand, if a literal is first replaced, it is impossible to omit the literal afterwards. Without loss of generality, we assume $j_1 < j_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2}x_{l_2} + \cdots + p_m$$

6.1.3 LIF with Other Literals Faults

**LIF and LIF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{l_1}$ and $x_{l_2}$ be two missing literals of the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively. Suppose two literals $x_{l_1}$ and $x_{l_2}$ are inserted into $p_{i_1}$ and $p_{i_2}$, respectively. We use $I_{LIF}(p_{i_1} \rightarrow p_{i_1}x_{l_1})$, $LIF(p_{i_2} \rightarrow p_{i_2}x_{l_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}x_{l_1} + \cdots + p_{i_2}x_{l_2} + \cdots + p_m$$

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do not consider the following two situations. First, two literals $x_{l_1}$ and $x_{l_2}$ are exactly the same (that is $x_{l_1} = x_{l_2}$). It is because when a literal is inserted into a term twice, the implementation is then equivalent to a single LIF with respect to the original expression $S$. Second, the two literals $x_{l_1}$ and $x_{l_2}$ are negations of each other (that is $x_{l_1} = \bar{x}_{l_2}$). It is because when a literal and its negation are inserted into a term, the implementation is then equivalent to a single TOF with respect to the original expression $S$. Hence, without considering the above two situations, $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$ and are from different Boolean variables. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}x_{l_1}x_{l_2} + \cdots + p_m$$

**LIF and LRF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_2}^{i_2}$ be a literal in the $i_2$-th term, $p_{i_2}$, of $S$, and $x_{l_1}$ and $x_{l_2}$ be two missing literals of the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively. Suppose the literal $x_{l_1}$ is inserted
Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of
generality, we can assume $i_1 < i_2$. The implementation is then equivalent
to the following expression

$$p_1 + \cdots + p_{i_1}x_{i_1} + \cdots + p_{i_2,j_2}x_{i_2} + \cdots + p_m \tag{6.16}$$

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do
not consider the two situations where the literals $x_{i_1}$ and $x_{i_2}$ are exactly the
same or are negations of each other. First, when $x_{i_1}$ and $x_{i_2}$ are exactly
the same, the net effect of inserting $x_{i_1}$ into $p_{i_1}$ and replacing $x_{j_2}^{i_1}$ by $x_{i_1}$ is
equivalent to a single LRF with $x_{j_2}^{i_1}$ in $S$ being replaced by $x_{i_1}$. Second,
when $x_{i_1}$ and $x_{i_2}$ are negations of each other (that is $x_{i_1} = \bar{x}_{i_2}$), the net
effect of inserting $x_{i_1}$ into $p_{i_1}$ and replacing $x_{j_2}^{i_1}$ by $\bar{x}_{i_1}$ is equivalent to a
single TOF with $p_{i_1}$ in $S$ being omitted. Hence, we only consider situations
where $x_{i_1}$ and $x_{i_2}$ are two different missing literals of $p_{i_1}$. As a result, the
implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_2}x_{i_1}x_{i_2} + \cdots + p_m \tag{6.17}$$

### 6.1.4 LRF with Other Literals Faults

**LRF with LRF** Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$
and $x_{j_2}^{i_2}$ be two literals in the $i_1$-th and $i_2$-th terms, $p_{i_1}$ and $p_{i_2}$, of $S$, respectively,
and $x_{i_1}$ and $x_{i_2}$ be two missing literals of $p_{i_1}$ and $p_{i_2}$, respectively. Suppose that
both literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are replaced by literals $x_{i_1}$ and $x_{i_2}$, respectively. We use
$I_{LRF(p_{i_1} \rightarrow p_{j_1}x_{i_1})}$, $LRF(p_{i_2} \rightarrow p_{j_2}x_{i_2})$ to denote the corresponding faulty implementation,
which can be further classified into the following two cases:

Case 1. The two terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of
generality, we can assume $i_1 < i_2$. The implementation is then equivalent
to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_1} + \cdots + p_{i_2,j_2}x_{i_2} + \cdots + p_m \tag{6.18}$$

Case 2. The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. We do
not consider the situation that $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are exactly the same (that is, $j_1 = j_2$). It is because when the literal $x_{j_1}^{i_1}$ is first replaced by another
literal, it is impossible to replace $x_{j_1}^{i_1}$ again. Therefore, $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are different. Without loss of generality, we can assume $j_1 < j_2$. Furthermore, we do not consider the situation when $x_{l_1}$ and $x_{l_2}$ are negation of each other (that is $x_{l_1} = \bar{x}_{l_2}$). It is because when $x_{l_1}$ replaces $x_{j_1}^{i_1}$ and $\bar{x}_{l_1}$ replaces $x_{j_2}^{i_2}$, the implementation is then equivalent to a single TOF with respect to the original expression $S$. As a result, we have the following two subcases:

(a) The two literals $x_{l_1}$ and $x_{l_2}$ are different (that is $x_{l_1} \neq x_{l_2}$). The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,\hat{j}_1}x_{l_1}x_{l_2} + \cdots + p_m \quad (6.19)$$

(b) The two literals $x_{l_1}$ and $x_{l_2}$ are exactly the same (that is $x_{l_1} = x_{l_2}$). The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2}x_{l_1} + \cdots + p_m \quad (6.20)$$

It should be noted that Expression (6.20) is equivalent to that of (6.13), which corresponds to Case 2 of the LOF and LRF.

In summary, there are altogether 19 different double fault expressions among the 10 double fault classes without ordering considered in this section because Expressions (13) and (20) are equivalent.

### 6.2 Double Faults with Ordering

In this section, different types of double literal faults with ordering and their corresponding double-fault expressions are discussed. As previously stated, double faults with ordering is such that two single faults occur one after the other in such a way that the occurrence of the first fault may affect the occurrence of the second fault.

As discussed before, $I_{F_1(E_1 \rightarrow E'_1) \ominus F_2(E_2 \rightarrow E'_2)}$ are used to denote the resulting double-fault expression of double faults with ordering where $F_1$ and $F_2$ are two single fault classes committed in a given Boolean expression $S$, changing the subexpressions $E_1$ and $E_2$ in $S$ to $E'_1$ and $E'_2$, respectively and $F_1$ is committed before $F_2$. Table 6.2 lists all 16 different types of double faults with ordering studied in this section for those four single literal fault classes discussed in Section 2.2. In the rest of the section, these 16 double fault classes and their corresponding double-fault expressions are introduced.
Table 6.2: Types of double literal faults with ordering

<table>
<thead>
<tr>
<th></th>
<th>LNF</th>
<th>LOF</th>
<th>LIF</th>
<th>LRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOF</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIF</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>LRF</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
</tbody>
</table>

6.2.1 LNF first, then Other Literals Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ be a literal in the $i_1$-th term, $p_{i_1}$, of $S$. Suppose the literal $x_{j_1}^{i_1}$ is negated. The corresponding faulty expression, denoted as $I_{LNF(p_{i_1} \rightarrow p_{i_1}, \bar{j_1})}$, is then equivalent to $p_1 + \cdots + p_{i_1, \bar{j_1}} + \cdots + p_m$.

**LNF and LNF** Let $x_{j_2}^{i_2}$ be a literal of the $i_2$-th term, $p_{i_2}$, of $I_{LNF(p_{i_1} \rightarrow p_{i_1}, \bar{j_1})}$. After the first LNF is made on $S$, the literal $x_{j_2}^{i_2}$ is then negated. We use $I_{LNF(p_{i_1} \rightarrow p_{i_1}, \bar{j_1})} \otimes I_{LNF(p_{i_2} \rightarrow p_{i_2}, \bar{j_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$ and $k_{i_2} > 1$. Without loss of generality, we assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1, \bar{j_1}} + \cdots + p_{i_2, \bar{j_2}} + \cdots + p_m$$ (6.21)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. We do not consider the situation where the two literals $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$ are exactly the same (that is $j_1 = j_2$). It is because when a literal is negated twice, the implementation is then equivalent to the original expression $S$. Without loss of generality, we assume $j_1 < j_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1, \bar{j_1}, \bar{j_2}} + \cdots + p_m$$ (6.22)

**LNF and LOF** Let $x_{j_2}^{i_2}$ be a literal of the $i_2$-th term, $p_{i_2}$, of $I_{LNF(p_{i_1} \rightarrow p_{i_1}, \bar{j_1})}$. After the first LNF is made on $S$, the literal $x_{j_2}^{i_2}$ is then omitted from $p_{i_2}$. We use $I_{LNF(p_{i_1} \rightarrow p_{i_1}, \bar{j_1})} \otimes I_{LOF(p_{i_2} \rightarrow p_{i_2}, \bar{j_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$ and $k_{i_2} > 1$. Without loss of generality, we assume
\(i_1 < i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  

(6.23)

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). We do not consider the situation where the two literals \(x_{j_1}^{i_1}\) and \(x_{j_2}^{i_1}\) are exactly the same (that is \(j_1 = j_2\)). It is because when \(x_{j_1}^{i_1}\) is negated and then omitted, the implementation is then equivalent a single LOF with respect to the original expression \(S\). Without loss of generality, we assume \(j_1 < j_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m
\]  

(6.24)

**LNF and LIF** Let \(x_{l_2}\) be a missing literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LNF}(p_{i_1} \rightarrow p_{i_1,j_1})\). After the first LNF is made on \(S\), the literal \(x_{l_2}\) is then inserted into \(p_{i_2}\). We use \(I_{LNF}(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes LIF(p_{i_2} \rightarrow p_{i_2}x_{l_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The \(i_1\)-th and \(i_2\)-th terms are different terms, that is \(i_1 \neq i_2\). It should be noted that \(k_{i_1} > 1\). Without loss of generality, we assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2}x_{l_2} + \cdots + p_m
\]  

(6.25)

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1}x_{l_2} + \cdots + p_m
\]  

(6.26)

**LNF and LRF** Let \(x_{j_2}^{i_2}\) and \(x_{l_2}\) be a literal and a missing literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LNF}(p_{i_1} \rightarrow p_{i_1,j_1})\), respectively. After the first LNF is made on \(S\), the literal \(x_{j_2}^{i_2}\) from \(p_{i_2}\) is then replaced by the literal \(x_{l_2}\). We use \(I_{LNF}(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes LRF(p_{i_2} \rightarrow p_{i_2,j_2}x_{l_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The \(i_1\)-th and \(i_2\)-th terms are different terms, that is \(i_1 \neq i_2\). It should be noted that \(k_{i_1} > 1\). Without loss of generality, we assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2}x_{l_2} + \cdots + p_m
\]  

(6.27)
Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). We do not consider the situation when the two literals \(x_{j_1}^{i_1}\) and \(x_{j_2}^{i_1}\) are exactly the same (that is \(j_1 = j_2\)). It is because when \(x_{j_1}^{i_1}\) is negated and then replaced by \(x_{j_2}\), the implementation is then equivalent to a single LRF with respect to the original expression \(S\). Without loss of generality, we assume \(j_1 < j_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1,j_2} x_{i_2} + \cdots + p_m
\]  

(6.28)

6.2.2 LOF first, then Other Literals Faults

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Let \(x_{j_1}^{i_1}\) be a literal of the \(i_1\)-th term, \(p_{i_1}\), of \(S\). Suppose the literal \(x_{j_1}^{i_1}\) is omitted from \(p_{i_1}\). The corresponding faulty expression, denoted as \(I_{LOF(p_{i_1} → p_{i_1,j_1})}\), is then equivalent to \(p_1 + \cdots + p_{i_1,j_1} + \cdots + p_m\).

LOF and LNF Let \(x_{j_2}^{i_2}\) be a literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LOF(p_{i_1} → p_{i_1,j_1})}\). After the first LOF is made on \(S\), the literal \(x_{j_2}^{i_2}\) is then negated. We use \(I_{LOF(p_{i_1} → p_{i_1,j_1})} \otimes LNF(p_{i_2} → p_{i_2,j_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The \(i_1\)-th and \(i_2\)-th terms are different terms, that is \(i_1 \neq i_2\). It should be noted that \(k_{i_1} > 1\) and \(k_{i_2} > 1\). Without loss of generality, we assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  

(6.29)

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). Since \(x_{j_1}^{i_1}\) is omitted, \(x_{j_1}^{i_1}\) and \(x_{j_2}^{i_1}\) are different (\(j_1 \neq j_2\)). Without loss of generality, we assume \(j_1 < j_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m
\]  

(6.30)

where \(p_{i_1,j_1,j_2} = x_{j_1}^{i_1} \cdots x_{j_1-1}^{i_1} x_{j_1+1}^{i_1} \cdots x_{j_2}^{i_1} \cdots x_{k_1}^{i_1}\) can be obtained from \(p_{i_1}\) by omitting its \(j_1\)-th literal and negating its \(j_2\)-th literal and \(j_1 < j_2\).

LOF and LOF Let \(x_{j_2}^{i_2}\) be a literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LOF(p_{i_1} → p_{i_1,j_1})}\). After the first LOF is made on \(S\), the literal \(x_{j_2}^{i_2}\) is then omitted from \(p_{i_2}\). We use \(I_{LOF(p_{i_1} → p_{i_1,j_1})} \otimes LOF(p_{i_2} → p_{i_2,j_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:
Case 1. The \(i_1\)-th and \(i_2\)-th terms are different terms, that is \(i_1 \neq i_2\). It should be noted that \(k_{i_1} > 1\) and \(k_{i_2} > 1\). Without loss of generality, we assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (6.31) \]

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). Furthermore, we do not consider the situation when \(p_{i_1}\) contains just two literals because the net effect of omitting two literals from \(p_{i_1}\) will be equivalent to a term omission fault, which is a single term fault. Without loss of generality, we assume \(j_1 < j_2\) and \(k_{i_1} > 2\). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m \quad (6.32) \]

**LOF and LIF** Let \(x_{i_2}\) be a missing literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})}\). After the first LOF is made on \(S\), the literal \(x_{i_2}\) is then inserted into \(p_{i_2}\). We use \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})}\) to denote the corresponding faulty implementation.

We do not consider the situation when LOF and LIF are committed at the same term (that is \(i_1 = i_2\)) because of the following three reasons. First, when the omitted literal \(x_{j_1}^{i_1}\) is inserted back into the term (that is, \(x_{i_2} = x_{j_1}^{i_1}\)), the implementation is then equivalent to the original expression \(S\). Second, when the negation of the omitted literal \(x_{j_1}^{i_1}\) is inserted into the term (that is, \(x_{i_2} = \bar{x}_{j_1}^{i_1}\)), the implementation is then equivalent to a single LNF with respect to the original expression \(S\). Third, when the two literals \(x_{j_1}^{i_1}\) and \(x_{i_2}\) are different, the implementation is then equivalent to a single LRF with respect to the original expression \(S\) because the net result of omitting the literal \(x_{j_1}^{i_1}\) from a term and then inserting \(x_{i_2}\) into that term is the same as replacing \(x_{j_1}^{i_1}\) by \(x_{i_2}\).

Hence, we only consider the situation when LOF and LIF are committed at two different terms, say, the \(i_1\)-th and \(i_2\)-th terms (that is \(i_1 \neq i_2\)). It should be noted that \(k_{i_1} > 1\). Without loss of generality, we assume \(i_1 < i_2\).

The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_m \quad (6.33) \]

**LOF and LRF** Let \(x_{j_2}^{i_2}\) and \(x_{i_2}\) be a literal and a missing literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})}\), respectively. After the first LOF is made on \(S\), the literal \(x_{j_2}^{i_2}\) from \(p_{i_2}\) is then replaced by \(x_{i_2}\). We use \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes LRF(p_{i_2} \rightarrow p_{i_2,j_2,x_{i_2}})}\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:
Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. It should be noted that $k_{i_1} > 1$. Without loss of generality, we assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,j_2} x_{l_2} + \cdots + p_m \quad (6.34)$$

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. Since $x_{j_1}^i$ is omitted, $x_{j_1}^{i_1}$ and $x_{j_2}^{i_1}$ are different (that is $j_1 \neq j_2$). Without loss of generality, we assume $j_1 < j_2$. We do not consider the situation when $x_{j_1}^{i_1}$ is first omitted, and then another literal $x_{j_2}^{i_1}$ is replaced by the omitted literal $x_{j_1}^{i_1}$ (that is, $x_{l_2}$ is the same as $x_{j_1}^{i_1}$) because the implementation is then equivalent to a single LOF with respect to the original expression $S$ with $x_{j_2}^{i_1}$ being omitted.

As a result, we have the following two subcases:

(a) The literal $x_{j_2}^{i_1}$ is replaced by the negation of $x_{j_1}^{i_1}$, (that is $x_{l_2} = \bar{x}_{j_1}^{i_1}$).

The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2} + \cdots + p_m \quad (6.35)$$

(b) The literal $x_{j_2}^{i_1}$ is replaced by a literal different from $x_{j_1}^{i_1}$ (that is, $x_{l_2} \neq x_{j_1}^{i_1}$). The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2} x_{l_2} + \cdots + p_m \quad (6.36)$$

### 6.2.3 LIF first, then Other Literals Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{l_1}$ be a missing literal of the $i_1$-th term, $p_{i_1}$, of $S$. Suppose the literal $x_{l_1}$ is inserted into $p_{i_1}$. The corresponding faulty expression, denoted as $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$, is then equivalent to $p_1 + \cdots + p_{i_1} x_{l_1} + \cdots + p_m$.

**LIF and LNF** Let $x_{j_2}^{i_2}$ be a literal of the $i_2$-th term, $p_{i_2}$, of $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$. After the first LIF is made on $S$, the literal $x_{j_2}^{i_2}$ is then negated. We use $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})} \odot LNF(p_{i_2} \rightarrow p_{i_2} x_{j_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. It should be noted that $k_{i_2} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{l_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (6.37)$$
Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. We do not consider $x_{l_1}$ being negated because the implementation will then be equivalent to a single LIF with respect to the original specification $S$ with $x_{l_1}$ being inserted into $p_{l_1}$ of $S$. Therefore, $x^{i_1}_{j_2}$ is a literal in $p_{l_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{l_1,j_2}x_{l_1} + \cdots + p_m$$  \hspace{1cm} (6.38)

**LIF and LOF** Let $x^{i_2}_{j_2}$ be a literal of the $i_2$-th term, $p_{l_2}$, of $I_{LIF(p_{l_1} \rightarrow p_{l_1}x_{l_1})}$. After the first LIF is made on $S$, the literal $x^{i_2}_{j_2}$ is then omitted from $p_{l_2}$. We use $I_{LIF(p_{l_1} \rightarrow p_{l_1}x_{l_1}) \otimes LOF(p_{l_2} \rightarrow p_{l_2,j_2})}$ to denote the corresponding faulty implementation. We do not consider the situation when the LIF and LOF are committed at the same term (that is, $i_1 = i_2$) because of the following two reasons. First, when the inserted literal $x_{l_1}$ is omitted from $p_{l_1}$, the implementation then is equivalent to original expression $S$. Second, when any literal in $p_{l_1}$ is omitted, the implementation is then equivalent to a single LRF with respect to the original expression $S$ because the net result of inserting a literal $x_{l_1}$ into a term and then omitting another literal $x^{i_1}_{j_2}$ in $p_{l_1}$ is the same as replacing $x^{i_1}_{j_2}$ by $x_{l_1}$.

Hence, we only consider the situation when LOF and LIF are committed at two different terms (that is $i_1 \neq i_2$). It should be noted that $k_{i_2} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{l_1}x_{l_1} + \cdots + p_{l_2,j_2} + \cdots + p_m$$  \hspace{1cm} (6.39)

**LIF and LIF** Let $x_{l_2}$ be a missing literal of the $i_2$-th term, $p_{l_2}$, of $I_{LIF(p_{l_1} \rightarrow p_{l_1}x_{l_1})}$. After the first LIF is made on $S$, the literal $x_{l_2}$ is then inserted into $p_{l_2}$. We use $I_{LIF(p_{l_1} \rightarrow p_{l_1}x_{l_1}) \otimes LIF(p_{l_2} \rightarrow p_{l_2}x_{l_2})}$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{l_1}x_{l_1} + \cdots + p_{l_2}x_{l_2} + \cdots + p_m$$  \hspace{1cm} (6.40)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. Since $x_{l_2}$ is a missing literal of $p_{l_1}x_{l_1}$, the $i_1$-th term of $I_{LIF(p_{l_1} \rightarrow p_{l_1}x_{l_1})}$, the two literals $x_{l_1}$ and $x_{l_2}$ are different. It should be noted that $x_{l_2}$ is also a missing literal of $p_{l_1}$. The
implementation is then equivalent to the following expression

$$p_1 + \cdots + p_i x_{l_1} x_{l_2} + \cdots + p_m$$ (6.41)

**LIF and LRF** Let $x_{j_2}^{i_2}$ and $x_{l_2}$ be a literal and a missing literal of the $i_2$-th term, $p_{i_2}$, of $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$, respectively. After the first LIF is made on $S$, the literal $x_{j_2}^{i_2}$ from $p_{i_2}$ is then replaced by the literal $x_{l_2}$. We use $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})} \otimes LRF(p_{i_2} \rightarrow p_{i_2} x_{l_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{l_1} + \cdots + p_{i_2 \cdot j_2} x_{l_2} + \cdots + p_m$$ (6.42)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. We do not consider the situation where the two literals $x_{j_2}^{i_1}$ and $x_{l_1}$ are exactly the same (that is $x_{j_2}^{i_1} = x_{l_1}$). It is because when $x_{l_1}$ is inserted into $p_i$ of $S$ and then $x_{l_1}$ is replaced by another literal $x_{l_2}$, the implementation is then equivalent to a single LIF with respect to $S$ with $x_{l_2}$ being inserted into $p_{i_1}$. It should be noted that $x_{l_2}$ is also a missing literal of $p_{i_1}$ because it is a missing literal of $p_{i_1} x_{l_1}$, the $i_1$-th term of $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$. As a result, both $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$ and the implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1 \cdot j_1} x_{l_1} x_{l_2} + \cdots + p_m$$ (6.43)

### 6.2.4 LRF first, then Other Literals Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $x_{j_1}^{i_1}$ and $x_{l_1}$ be a literal and a missing literal of the $i_1$-th term, $p_{i_1}$, of $S$. Suppose the literal $x_{j_1}^{i_1}$ is replaced by the literal $x_{l_1}$. The corresponding faulty expression, denoted as $I_{LRF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$, is then equivalent to $p_1 + \cdots + p_{i_1 \cdot j_1} x_{l_1} + \cdots + p_m$.

**LRF and LNF** Let $x_{j_2}^{i_2}$ be a literal of the $i_2$-th term, $p_{i_2}$, of $I_{LRF(p_{i_1} \rightarrow p_{i_1} x_{l_1})}$. After the first LRF is made on $S$, the literal $x_{j_2}^{i_2}$ is then negated. We use $I_{LRF(p_{i_1} \rightarrow p_{i_1} x_{l_1})} \otimes LNF(p_{i_2} \rightarrow p_{i_2} x_{l_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. It should be noted that $k_{i_2} > 1$. Without loss of generality, we can assume $i_1 < i_2$. The
 implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1 \cdot j_1} x_{i_1} + \cdots + p_{i_2 \cdot j_2} + \cdots + p_m \]  

(6.44)

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). We do not consider the situation when the two literals \(x_{i_j}^{i_1}\) and \(x_{i_j}^{i_2}\) are exactly the same (that is \(x_{i_j}^{i_1} = x_{i_j}^{i_2}\)). It is because when \(x_{i_j}^{i_1}\) is replaced by \(x_{i_j}^{i_2}\), and then \(x_{i_j}^{i_1}\) is negated, the implementation is then equivalent to a single LRF with respect to the original expression \(S\) with \(x_{i_j}^{i_1}\) being replaced by \(\bar{x}_{i_1}\). Since \(x_{i_j}^{i_1}\) does not exist in \(p_{i_1 \cdot j_1} x_{i_1}\) in \(I_{LRF}(p_{1-i_1} x_{i_1})\) and \(x_{i_j}^{i_2}\) is different from \(x_{i_1}^{i_1}\), \(x_{i_2}^{i_1}\) and \(x_{i_2}^{i_2}\) are two different literals of \(p_{i_1}\) (that is \(j_1 \neq j_2\)). Without loss of generality, we can assume \(j_1 < j_2\). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1 \cdot j_1} x_{i_1} + \cdots + p_m \]  

(6.45)

LRF and LOF  Let \(x_{i_j}^{i_2}\) be a literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LRF}(p_{1-i_1} x_{i_1})\). After the first LRF is made on \(S\), the literal \(x_{i_j}^{i_2}\) is then omitted from \(p_{i_2}\). We use \(I_{LRF}(p_{1-i_1} x_{i_1}) \otimes \text{LOF}(p_{2-i_2} x_{i_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The \(i_1\)-th and \(i_2\)-th terms are different terms, that is \(i_1 \neq i_2\). It should be noted that \(k_{i_2} > 1\). Without loss of generality, we can assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1 \cdot j_1} x_{i_1} + \cdots + p_{i_2 \cdot j_2} + \cdots + p_m \]  

(6.46)

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). We do not consider the situation when the literal \(x_{i_1}^{i_1}\) is omitted because the implementation is then equivalent to a single LOF with respect to the original expression \(S\) with \(x_{i_1}^{i_1}\) being omitted. Hence, \(x_{i_j}^{i_1}\) is a literal in \(p_{1-i_1} x_{i_1}\) and the two literals \(x_{i_1}^{i_1}\) and \(x_{i_2}^{i_1}\) are different (that is \(j_1 \neq j_2\)). Without loss of generality, we can assume \(j_1 < j_2\). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1 \cdot j_1 \cdot j_2} x_{i_1} + \cdots + p_m \]  

(6.47)

LRF and LIF  Let \(x_{i_2}\) be a missing literal of the \(i_2\)-th term, \(p_{i_2}\), of \(I_{LRF}(p_{1-i_1} x_{i_1})\). After the first LRF is made on \(S\), the literal \(x_{i_2}\) is then inserted into \(p_{i_2}\). We use \(I_{LRF}(p_{1-i_1} x_{i_1}) \otimes \text{LIF}(p_{2-i_2} x_{i_2})\) to denote the corresponding faulty implementation, which can be further classified into the following two cases:
Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_1} + \cdots + p_{i_2}x_{i_2} + \cdots + p_m$$ \hfill (6.48)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. We do not consider the situation when $x_{i_1}^{j_1}$ is inserted back to $p_{i_1,j_1}x_{i_1}$, the $i_1$-th term of $I_{LRF}(p_1 \rightarrow p_{i_1,j_1}x_{i_1})$ (that is $x_{i_2} = x_{i_1}^{j_1}$), because the implementation is then equivalent to a single LIF with respect to the original expression $S$ with $x_{i_1}$ being inserted into $p_{i_1}$. As a result, we have the following two subcases:

(a) The negation of the literal $x_{i_1}^{j_1}$ is inserted (that is $x_{i_2} = \overline{x_{i_1}^{j_1}}$). The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_1} + \cdots + p_m$$ \hfill (6.49)

(b) The two literals $x_{i_1}$ and $x_{i_1}^{j_1}$ are different (that is, $x_{i_2} \neq x_{i_1}^{j_1}$ and $x_{i_2} \neq \overline{x_{i_1}^{j_1}}$). Thus, $x_{i_2}$ is a missing literal of $p_{i_1}$ because it is a missing literal of $p_{i_1,j_1}x_{i_1}$, the $i_1$-th term of $I_{LRF}(p_1 \rightarrow p_{i_1,j_1}x_{i_1})$. As a result, both $x_{i_1}$ and $x_{i_2}$ are two different missing literals of $p_{i_1}$ and the implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_1}x_{i_2} + \cdots + p_m$$ \hfill (6.50)

**LRF and LRF** Let $x_{i_2}^{j_2}$ and $x_{i_2}$ be a literal and a missing literal of the $i_2$-th term, $p_{i_2}$, of $I_{LRF}(p_1 \rightarrow p_{i_1,j_1}x_{i_1})$, respectively. After the first LRF is made on $S$, the literal $x_{i_2}^{j_2}$ from $p_{i_2}$ is then replaced by $x_{i_2}$. We use $I_{LRF}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \otimes LRF(p_2 \rightarrow p_{i_2,j_2}x_{i_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different terms, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_1} + \cdots + p_{i_2,j_2}x_{i_2} + \cdots + p_m$$ \hfill (6.51)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. We do not consider the situation when $x_{i_1}$ is replaced by $x_{i_2}$ because of the following three reasons. During the discussion, please noted that, in the first LRF, $x_{i_1}^{j_1}$ is replaced by $x_{i_1}$. First, when $x_{i_2}$ is now replaced by $x_{i_1}^{j_1}$, the implementation
is then equivalent to the original expression $S$. Second, when $x_{l_1}$ is now replaced by the negation of $x_{j_1}^i$, the implementation is then equivalent to a single LNF with respect to the original expression $S$ with $x_{j_1}^i$ being negated. Third, when $x_{l_1}$ is now replaced by $x_{l_2}$ which is different from $x_{j_1}^i$ (that is, $x_{l_2} \neq x_{j_1}^i$ and $x_{l_2} \neq \bar{x}_{j_1}^i$), the implementation is then equivalent to a single LRF with respect to the original expression $S$ with $x_{j_1}^i$ being replaced by $x_{l_2}$.

As a result, we only need to consider literals in $p_{i_1,j_1}$ being replaced by $x_{l_2}$. Therefore, $x_{j_1}^i$ and $x_{j_2}^i$ are two different literals of $p_{i_1}$. Without loss of generality, we can assume $j_1 < j_2$. Furthermore, we do not consider the situation when $x_{j_1}^i$ is replaced by $x_{j_2}^i$ because the implementation is then equivalent to a single LRF with respect to the original expression $S$ with $x_{j_1}^i$ being replaced by $x_{l_1}^i$. Hence, we have the following two subcases:

(a) The literal $x_{j_2}^i$ is replaced by $\bar{x}_{j_1}^i$, the implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2}x_{l_1} + \cdots + p_m$$ (6.52)

(b) The literal $x_{j_2}^i$ is replaced by a literal different from $x_{j_1}^i$ (that is, $x_{l_2} \neq x_{j_1}^i$ and $x_{l_2} \neq \bar{x}_{j_1}^i$). Since $x_{l_2}$ is also a missing literal of $p_{i_1,j_1}x_{l_1}$, the $i_1$-th term of $I_{LRF(p_{i_1} - p_{i_1,j_1}x_{l_1})}$, and it is different from $x_{j_1}^i$, $x_{l_2}$ is a missing literal of $p_{i_1}$. As a result, both $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$ and the implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1,j_2}x_{l_1}x_{l_2} + \cdots + p_m$$ (6.53)

In summary, there are altogether 33 double-fault expressions among the 16 double fault classes with ordering considered in this report. In the next section, we show that these 33 double-fault expressions can be reduced to 19 double fault expressions discussed in Section 1.

### 6.3 Relation between Double Faults with and without Ordering

In this section, the relation of double faults with and without ordering are analysed. The possible faulty expressions of double faults without ordering with respect to those with ordering in the same fault category related to LNF, LOF, LIF and LRF
are compared. Table 6.3 (respectively, 6.4, 6.5 and 6.6) summarizes the situations of double faults with LNF (respectively LOF, LIF, and LRF) and other faults. Each row in these tables shows those faulty expressions of a particular type of double fault without ordering and their counterparts in double faults with ordering.

Table 6.3: Comparison of double-fault expressions of LNF and other literal faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LNF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF and LNF</td>
<td>6.1</td>
<td>6.21</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>6.22</td>
</tr>
<tr>
<td>LNF and LOF</td>
<td>6.3</td>
<td>6.23</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>6.24</td>
</tr>
<tr>
<td>LNF and LIF</td>
<td>6.5</td>
<td>6.25</td>
</tr>
<tr>
<td></td>
<td>6.6</td>
<td>6.26</td>
</tr>
<tr>
<td>LNF and LRF</td>
<td>6.7</td>
<td>6.27</td>
</tr>
<tr>
<td></td>
<td>6.8</td>
<td>6.28</td>
</tr>
</tbody>
</table>

For the first row in Table 6.3, there are two subcases for LNF and LNF without ordering which are given by Expressions (6.1) and (6.2) in Section 6.1. For the first subcase, Expressions (6.1) is equivalent to Expression (6.21) which corresponds to the first subcase of LNF and LNF with ordering as discussed in Section 6.2. Similarly, other rows in Table 6.3 show the equivalent faulty expressions of double faults of LNF and other faults with and without ordering.

Table 6.4: Comparison of double-fault expressions of LOF and other literal faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LOF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF and LNF</td>
<td>6.3</td>
<td>6.29</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>6.30</td>
</tr>
<tr>
<td>LOF and LOF</td>
<td>6.9</td>
<td>6.31</td>
</tr>
<tr>
<td></td>
<td>6.10</td>
<td>6.32</td>
</tr>
<tr>
<td>LOF and LIF</td>
<td>6.11</td>
<td>6.33</td>
</tr>
<tr>
<td>LOF and LRF</td>
<td>6.12</td>
<td>6.34</td>
</tr>
<tr>
<td></td>
<td>6.13</td>
<td>6.35(6.4)</td>
</tr>
</tbody>
</table>

For the rows in Table 6.4, there are four different situations. First, some of them can be interpreted in a similar manner as those in Table 6.3. For example, for LOF and LOF in Table 6.4, Expression (6.9) is equivalent to Expression (6.31). Second, for some rows, the faulty expressions of double faults with ordering have their counterparts in double faults without ordering in the same double fault class. For example, let us consider the first row in Table 6.4, there are two subcases for LOF and LNF. In the first subcase, Expression (6.3) is derived from LNF and LOF.
without ordering with LNF being committed at the $i_1$-th term and LOF committed at the $i_2$-th term. However, Expression (6.29) is derived from LOF and LNF with ordering with LOF being committed at $i_1$-th term and LNF being committed at the $i_2$-th term. By interchanging the position of these terms, Expressions (6.3) and (6.29) are equivalent. Hence, they are considered as counterparts of each other. Similar situations occurred at Expressions (6.4) and (6.30) in the second subcase of LOF and LNF, and other double fault classes in Tables 6.4, 6.5 and 6.6. In later discussions, we assume that the reader can make this adjustment accordingly. Third, for some rows, the faulty expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class, but they are equivalent to other faulty expressions of double faults without ordering in a different double fault class. For example, let us consider the second subcase of LOF and LRF of double fault with ordering, the faulty expression is given by Expression (6.35). It does not have its counterpart in LOF and LIF without ordering. However, it is equivalent to Expression (6.4) in the double fault LOF and LIF without ordering in Section 6.1. Fourth, for some rows, the faulty expressions of double faults with ordering do have their counterparts in double faults without ordering in the same double fault class, but they are not equivalent. Expression (6.11), $p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2}x_{l_2} + \cdots + p_m$, is derived from LOF and LIF without ordering, where LOF being committed at $i_1$-th term and LIF being committed at the $i_2$-th term. Expression (6.33), $p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2}x_{l_2} + \cdots + p_m$, is derived from LOF and LIF with ordering, where LOF being committed at $i_1$-th term and LIF being committed at the $i_2$-th term. As mentioned in Section 6.1, when considering LOF $\times$ LIF without ordering, the literal $x_{l_2}$ is a missing literal of the $i_2$-th term of $S$. More clearly, $x_{l_2}$ is a variable of $S$, but both $x_{l_2}$ and $\bar{x}_{l_2}$ do not appear in $i_2$-th term. For example, $S = ab + cd$, for the second term $cd$, the missing literals are $a, \bar{a}, b$ and $\bar{b}$. However, as mentioned in Section 6.2, the literal $x_{l_2}$ is a missing literal of the $i_2$-th term, $p_{i_2}$, of $I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})}$. It means $x_{l_2}$ is a variable of $I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})}$. For $S = ab + cd$, if $a$ is omitted from the first term $ab$, the resulting expression is $I = b + cd$. Then, for the second term $cd$ of $I$, the missing literals are $b$ and $\bar{b}$. Clearly, the faulty expressions resulting from LOF $\times$ LIF with ordering is a subset of those resulting from LOF $\times$ LIF without ordering.

For the rows in Table 6.5, there are two different situations. First, some of them can be interpreted in a similar manner as those in Table 6.3. For example, for LIF and LIF in Table 6.5, Expression (6.14) is equivalent to Expression (6.40). Second, for some rows, the faulty expressions of double faults with ordering have their counterparts in double faults without ordering in the same double fault class. For example, Expressions (6.5) and (6.37) are considered to be counterparts of each other as discussed previously for Expressions (6.3) and (6.29) in Table 6.4.
Table 6.5: Comparison of double-fault expressions of LIF and other literal faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LIF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIF and LNF</td>
<td>6.5</td>
<td>6.37</td>
</tr>
<tr>
<td></td>
<td>6.6</td>
<td>6.38</td>
</tr>
<tr>
<td>LIF and LOF</td>
<td>6.11</td>
<td>6.39</td>
</tr>
<tr>
<td>LIF and LIF</td>
<td>6.14</td>
<td>6.40</td>
</tr>
<tr>
<td></td>
<td>6.15</td>
<td>6.41</td>
</tr>
<tr>
<td>LIF and LRF</td>
<td>6.16</td>
<td>6.42</td>
</tr>
<tr>
<td></td>
<td>6.17</td>
<td>6.43</td>
</tr>
</tbody>
</table>

Table 6.6: Comparison of double-fault expressions of LRF and other literal faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LRF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRF and LNF</td>
<td>6.7</td>
<td>6.44</td>
</tr>
<tr>
<td></td>
<td>6.8</td>
<td>6.45</td>
</tr>
<tr>
<td>LRF and LOF</td>
<td>6.12</td>
<td>6.46</td>
</tr>
<tr>
<td></td>
<td>6.13</td>
<td>6.47</td>
</tr>
<tr>
<td>LRF and LIF</td>
<td>6.16</td>
<td>6.48</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>6.49(6.6)</td>
</tr>
<tr>
<td></td>
<td>6.17</td>
<td>6.50</td>
</tr>
<tr>
<td>LRF and LRF</td>
<td>6.18</td>
<td>6.51</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>6.52(6.8)</td>
</tr>
<tr>
<td></td>
<td>6.19</td>
<td>6.53</td>
</tr>
<tr>
<td></td>
<td>6.20(same as 6.13 as discussed in Section 6.1)</td>
<td>-</td>
</tr>
</tbody>
</table>
For the rows in Table 6.6, there are four different situations. First, some of them can be interpreted in a similar manner to those in Table 6.3. For example, for LRF and LRF in Table 6.6, Expression (6.18) is equivalent to Expression (6.51). Second, for some rows, the faulty expressions of double faults with ordering have their counterparts in double faults without ordering in the same double fault class. For example, Expressions (6.7) and (6.44) are considered to be counterparts of each other. Third, for some rows, the faulty expressions of double faults with ordering do not have their counterparts in double faults without ordering in the same double fault class, but they are equivalent to other faulty expressions of double faults without ordering in a different double fault class. For example, for the second subcase of LRF and LRF of double fault with ordering, Expression (6.52) does not have its counterpart in LRF and LRF without ordering. However, it is equivalent to Expression (6.8) in the double fault LNF and LRF without ordering in Section 6.1. Fourth, for some rows, the faulty expressions of double faults without ordering do not have their counterparts in double faults with ordering in the same double fault class, but they are equivalent to other faulty expressions of double faults with ordering in a different double fault class. For example, for the third subcase of LRF and LRF of double fault without ordering, Expression (6.20) does not have its counterpart in LRF and LRF with ordering. But it is equivalent to Expression (6.47) in the double fault LOF and LRF without ordering.

6.4 Summary

In summary, all expressions of double faults with ordering have their counterparts in double faults without ordering. Therefore, for the four single fault classes studied in this chapter, there are 19 different double-fault expressions and all of them fall into the category of double fault without ordering.

Table 6.7 summaries all 19 double-fault expressions and their corresponding double fault classes related to literal only. Similarly, the notation $F_1 \bowtie F_2$ is used to denote the double fault class formed from two single fault classes $F_1$ and $F_2$. For ease of reference, the numbers of those double-fault expressions reported in Section 6.1 and Section 6.2 are used.
Table 6.7: Double fault classes and double-fault expressions ($S = p_1 + \ldots + p_m$)

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF $\ltimes$ LNF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.1)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.2)</td>
</tr>
<tr>
<td>LNF $\ltimes$ LOF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.3)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.4)</td>
</tr>
<tr>
<td>LNF $\ltimes$ LIF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.5)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.6)</td>
</tr>
<tr>
<td>LNF $\ltimes$ LRF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.7)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.8)</td>
</tr>
<tr>
<td>LOF $\ltimes$ LOF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.9)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.10)</td>
</tr>
<tr>
<td>LOF $\ltimes$ LIF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.11)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.12)</td>
</tr>
<tr>
<td>LIF $\ltimes$ LIF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.13)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.14)</td>
</tr>
<tr>
<td>LIF $\ltimes$ LRF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.15)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.16)</td>
</tr>
<tr>
<td>LRF $\ltimes$ LRF</td>
<td>Case 1 ($i_1 \neq i_2$): $p_1 + \cdots + p_{i_1,\bar{i}<em>1} + \cdots + p</em>{i_2,\bar{i}_2} + \cdots + p_m$ (6.17)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 \neq j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.18)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 = i_2$ and $j_1 = j_2$): $p_1 + \cdots + p_{i_1,\bar{i}_1} + \cdots + p_m$ (6.19)</td>
</tr>
</tbody>
</table>
Chapter 7

Detection Conditions on Double Faults Related to Literals Only

In this chapter, the detection conditions of the double faults related to literals are studied. Chapter 6 shows that there are altogether 19 different double-fault expressions related to double fault on literals. In this chapter, the detection conditions of double faults related to literals are studied. The discussion are organized into the following 4 subsections as shown in Table 7.1. The $I_{F_1(E_1\rightarrow E_1')}\& F_2(E_2\rightarrow E_2')$ is used to denote the resulting double-fault expression where two single fault classes $F_1$ and $F_2$ are committed in a given Boolean expression $S$ changing its subexpressions $E_1$ and $E_2$ to $E_1'$ and $E_2'$, respectively.

7.1 LNF with Other Literal Faults

Theorem 7.1.1 (LNF with LNF - Case 1)

Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{j_1}^{i_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ and $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ are negated where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $1 \leq j_2 \leq k_{i_2}$ and $k_{i_1}(>1)$ and $k_{i_2}(>1)$ are the numbers of literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting expression denoted as $I_{LNF(p_{i_1}\rightarrow p_{i_1,j_1})\& LNF(p_{i_2}\rightarrow p_{i_2,j_2})}$ is equivalent to Expression (6.1) in Table 6.7. Then, we have $S \neq I_{LNF(p_{i_1}\rightarrow p_{i_1,j_1})\& LNF(p_{i_2}\rightarrow p_{i_2,j_2})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

Table 7.1: Organization of detection conditions for double faults without ordering

<table>
<thead>
<tr>
<th></th>
<th>LNF</th>
<th>LOF</th>
<th>LIF</th>
<th>LRF</th>
<th>Detailed Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>7.1</td>
</tr>
<tr>
<td>LOF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>7.2</td>
</tr>
<tr>
<td>LIF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>7.3</td>
</tr>
<tr>
<td>LRF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>7.4</td>
</tr>
</tbody>
</table>
1. \( \vec{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( p_{i_2, i_2} = 0 \),

2. \( \vec{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( p_{i_1, i_1} = 0 \),

3. \( \vec{t} \in NFP_{i_1, i_1}(S) \), or

4. \( \vec{t} \in NFP_{i_2, i_2}(S) \).

**Proof:** First, we observe that

\[
\begin{aligned}
S(S(\vec{t})) &\neq I_{\text{NLP}(p_{i_1} - p_{i_1, i_1}) \times \text{NLP}(p_{i_2} - p_{i_2, i_2})}(\vec{t}) \\
\text{if and only if } S(S(\vec{t}) \oplus I_{\text{NLP}(p_{i_1} - p_{i_1, i_1}) \times \text{NLP}(p_{i_2} - p_{i_2, i_2})}(\vec{t})) &= 1 \\
\text{if and only if } p_{i_1, i_1} p_{i_2} \vec{p}_1 \cdots \vec{p}_{i_1 - 1} \cdots \vec{p}_{i_2 - 1} \cdots \vec{p}_m + p_{i_1, i_1} \vec{p}_1 \cdots \vec{p}_{i_1 - 1} \cdots \vec{p}_{i_2 - 1} \cdots \vec{p}_m \\
&= 1 
\end{aligned}
\]

if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1} TP_i(S) \right) \) such that \( p_{i_2, i_2} = 0 \),

2. \( \vec{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1} TP_i(S) \right) \) such that \( p_{i_1, i_1} = 0 \),

3. \( \vec{t} \in NFP_{i_1, i_1}(S) \), or

4. \( \vec{t} \in NFP_{i_2, i_2}(S) \).

Hence, the result follows.
**Theorem 7.1.2** (LNF with LNF - Case 2)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{i_1}^{i_1}$ and $x_{i_2}^{i_2}$ in the $i_1$-th term, $p_{i_1}$, in $S$ are negated where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_{i_1}$ and $k_{i_1}(>1)$ is the number of literals of $p_{i_1}$, the resulting expression denoted as $I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LNF}}(p_{i_1} - p_{i_1,j_2})$ is equivalent to Expression (6.2) in Table 6.7. Then, we have $S \neq I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LNF}}(p_{i_1} - p_{i_1,j_2})$ if and only if there is a test case $\bar{\ell}$ that satisfies any of the following conditions:

1. $\bar{\ell} \in UTP_{i_1}(S)$, or

2. $\bar{\ell} \in FP(S)$ such that $p_{i_1,j_1,j_2} = 1$.

where $p_{i_1,j_1,j_2}$ denotes the term obtained from $p_{i_1}$ by negating its $j_1$-th and $j_2$-th literals.

**Proof:** First, we observe that $S \oplus I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LNF}}(p_{i_1} - p_{i_1,j_2})$

\[
\equiv (p_{i_1} \oplus p_{i_1,j_1,j_2})\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m \\
\equiv (p_{i_1,\bar{p}_{i_1,j_1,j_2}} + p_{i_1,j_1,j_2})\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m \\
\equiv (p_{i_1} + p_{i_1,j_1,j_2})\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m \\
\quad \text{(By making use of} (ABC)(A \cdot \overline{B} \cdot \overline{C}) \equiv ABC 	ext{)}
\]

\[
\equiv p_{i_1}\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m + p_{i_1,j_1,j_2}\bar{p}_1 \cdots \bar{p}_{i_1} \bar{p}_m \\
\equiv p_{i_1}\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m + p_{i_1,j_1,j_2} S
\]

Now, $S(\bar{\ell}) \neq I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LNF}}(p_{i_1} - p_{i_1,j_2})(\bar{\ell})$

if and only if $S(\bar{\ell}) \oplus I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LNF}}(p_{i_1} - p_{i_1,j_2})(\bar{\ell}) = 1$

if and only if $p_{i_1}\bar{p}_1 \cdots \bar{p}_{i_1} - 1\bar{p}_{i_1} + 1 \cdots \bar{p}_m + p_{i_1,j_1,j_2} S$ evaluates to 1 on $\bar{\ell}$

if and only if $\bar{\ell}$ satisfies any of the following conditions:

1. $\bar{\ell} \in UTP_{i_1}(S)$, or

2. $\bar{\ell} \in FP(S)$ such that $p_{i_1,j_1,j_2} = 1$.

Hence, the result follows. \qed

**Theorem 7.1.3** (LNF with LOF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{j_1}^{i_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ is negated and the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ is omitted from $p_{i_2}$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $1 < j_2 \leq k_{i_2}$, and $k_{i_2}(>1)$ are the numbers of literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting expression denoted as $I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LOF}}(p_{i_2} - p_{i_2,j_2})$ is equivalent to Expression (6.3) in Table 6.7. Then, we have $S \neq I_{\text{LNF}}(p_{i_1} - p_{i_1,j_1}) \times I_{\text{LOF}}(p_{i_2} - p_{i_2,j_2})$ if and only if there is a test case $\bar{\ell}$ that satisfies any of the following conditions:

1. $\bar{\ell} \in UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$, 

106
2. \( \vec{t} \in NFP_{i,j_1}(S) \), or
3. \( \vec{t} \in NFP_{i,j_2}(S) \).

**Proof**: First, we observe that \( S \oplus I_{LNF(p_1 \rightarrow p_{i,j_1}) \& LOF(p_2 \rightarrow p_{i,j_2})} \)
\( \equiv \left( (p_{i_1} + p_{i_2}) \oplus (p_{i_{j_1}} + p_{i_{j_2}}) \right) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_m \)
\( \equiv \left( (p_{i_1} + p_{i_2}) (p_{i_{j_1}} + p_{i_{j_2}}) + (p_{i_1} + p_{i_2}) (p_{i_{j_1}} + p_{i_{j_2}}) \right) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_m \)
\( \equiv (p_{i_1} \vec{p}_{i_{j_2}} + 0 + p_{i_{j_1}} \vec{p}_i + p_{i_{j_2}} \vec{p}_i) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_m \)

(By making use of \((AB)(AB) \equiv AB \) and \(AB(A) \equiv 0 \) )
\( \equiv (p_{i_1} \vec{p}_{i_{j_2}} \vec{p}_i + p_{i_{j_1}} \vec{p}_i + p_{i_{j_2}} \vec{p}_i) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_m \)

(By rewriting \( A \) as \((A)(AB) \) because they are equivalent)
\( \equiv p_{i_1} \vec{p}_{i_{j_2}} \vec{p}_i \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_{i_1} \vec{p}_{i_1+1} \cdots \vec{p}_{i_2-1} \vec{p}_{i_2+1} \cdots \vec{p}_m \)
\( \equiv p_{i_1} \vec{p}_{i_{j_2}} \vec{p}_i \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_{j_1}} \vec{S} + p_{i_{j_2}} \vec{S} \)

Now, \( S(\vec{t}) \neq I_{LNF(p_1 \rightarrow p_{i_1,j_1}) \& LOF(p_2 \rightarrow p_{i_2,j_2})}(\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{LNF(p_1 \rightarrow p_{i_1,j_1}) \& LOF(p_2 \rightarrow p_{i_2,j_2})}(\vec{t}) = 1 \)
if and only if \( p_{i_1} \vec{p}_{i_{j_2}} \vec{p}_i \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_{j_1}} \vec{S} + p_{i_{j_2}} \vec{S} \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} = 0 \),
2. \( \vec{t} \in NFP_{i_1,j_1}(S) \), or
3. \( \vec{t} \in NFP_{i_2,j_2}(S) \).

Hence, the result follows. \( \square \)

**Theorem 7.1.4 (LNF with LOF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the literal \( x_{j_1}^{i_1} \)
in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the literal \( x_{j_2}^{i_1} \)
in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted where \( 1 \leq i_1 \leq m, 1 \leq j_1 < j_2 \leq k_{i_1} \) and \( k_{i_1} (> 1) \) is the number of literals of \( p_{i_1} \), the resulting expression denoted as \( I_{LNF(p_1 \rightarrow p_{i_1,j_1}) \& LOF(p_1 \rightarrow p_{i_1,j_2})} \) is equivalent to Expression (6.4) in Table 6.7. Then, we have \( S \neq I_{LNF(p_1 \rightarrow p_{i_1,j_1}) \& LOF(p_1 \rightarrow p_{i_1,j_2})} \)
if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \), or
2. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1,j_2} = 1 \).

where \( p_{i_1,j_1,j_2} \) denotes the term obtained from \( p_{i_1} \) by negating its \( j_1 \)-th literal and omitting its \( j_2 \)-th literal.
Proof: First, we observe that $S \oplus I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LOF(p_1 \leftarrow p_{i_1,j_2})$
\[ \equiv (p_i \oplus p_{i_1,j_1,j_2}) \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m \]
\[ \equiv (p_i \bar{p}_{i_1,j_1,j_2} + \bar{p}_i p_{i_1,j_1,j_2}) \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m \]
\[ \equiv (p_i + p_{i_1,j_1,j_2}) \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m \]

(By making use of $(ABC)\bar{AB} \equiv ABC$)
\[ \equiv p_i \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m + p_{i_1,j_1,j_2} \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m \]
\[ \equiv p_i \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m + p_{i_1,j_1,j_2} S \]

Now, $S(\bar{t}) \neq I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LOF(p_1 \leftarrow p_{i_1,j_2})(\bar{t})$
if and only if $S(\bar{t}) \oplus I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LOF(p_1 \leftarrow p_{i_1,j_2})(\bar{t}) = 1$
if and only if $p_i \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m + p_{i_1,j_1,j_2} S$ evaluates to 1 on $\bar{t}$
if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$, or
2. $\bar{t} \in FP(S)$ such that $p_{i_1,j_1,j_2} = 1$.

Hence, the result follows. \qed

Theorem 7.1.5 (LNF with LIF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{i_1}^{i_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ is negated and the literal $x_{i_2}$ is inserted in the $i_2$-th term, $p_{i_2}$, in $S$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $k_{i_1}(> 1)$ is the number of literals of $p_{i_1}$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting expression denoted as $I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LIF(p_{i_2} \leftarrow p_{i_2,x_{i_2}})$ is equivalent to Expression (6.5) in Table 6.7. Then, we have $S \neq I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LIF(p_{i_2} \leftarrow p_{i_2,x_{i_2}})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$,
2. $\bar{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{1 \leq i_1 < i_2 \leq m} TP_{i_1}(S) \right)$ such that $x_{i_2} = \emptyset$,
3. $\bar{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{1 \leq i_1 < i_2 \leq m} TP_{i_1}(S) \right)$ such that $p_{i_1,j_1} + x_{i_2} = \emptyset$, or
4. $\bar{t} \in NFP_{i_1,j_1}(S)$.

Proof: First, we observe that $S \oplus I_{LNF(p_1 \leftarrow p_{i_1,j_1})} \bullet LIF(p_{i_2} \leftarrow p_{i_2,x_{i_2}})$
\[ \equiv \left( (p_i + p_{i_2}) \oplus (p_{i_1,j_1} + p_{i_2,x_{i_2}}) \right) \bar{p}_1 \cdots \bar{p}_{i-1} \bar{p}_{i+1} \cdots \bar{p}_m \]

108
\[(p_{i_1} + p_{i_2})(p_{i_1,j_1} + p_{i_2}x_{i_2}) + (p_{i_1} + p_{i_2})(p_{i_1,j_1} + p_{i_2}x_{i_2}) p_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[(p_{i_1} + p_{i_2})(\bar{p}_{i_1,j_1}x_{i_2}) + p_{i_1}\bar{p}_{i_2}(p_{i_1,j_1} + p_{i_2}x_{i_2}) p_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m \]

\[(p_{i_1}(p_{i_2}x_{i_2}) + p_{i_1,j_1}p_{i_2}x_{i_2} + p_{i_1,j_1}\bar{p}_{i_2} + 0) p_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By making use of \(AB(AB) = AB\) and \(A(AB) = AB\))

\[(p_{i_1}\bar{p}_{i_2} + p_{i_1}p_{i_2}\bar{p}_{i_2} + p_{i_1,j_1}p_{i_2}x_{i_2} + p_{i_1,j_1}\bar{p}_{i_2} + 0) p_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By rewriting \((AB)\) as \(\bar{A} + AB\) because they are equivalent)

\[p_{i_1}\bar{p}_{i_1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1}p_{i_2}\bar{p}_{i_1} p_{i_2} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1,j_1} \bar{S} \]

S(\(\vec{\iota}\)) \(\neq\) \(I_{LNF}(p_{i_1} - p_{i_1,j_1}) \times LIF(p_{i_2} - p_{i_2}x_{i_2})(\vec{\iota})\)

if and only if

S(\(\vec{\iota}\)) \(\oplus\) \(\det_{LNF}(p_{i_1} - p_{i_1,j_1}) \times LIF(p_{i_2} - p_{i_2}x_{i_2})(\vec{\iota}) = 1\)

if and only if

\[p_{i_1}\bar{p}_{i_1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1,j_1}p_{i_2}\bar{p}_{i_1} \bar{p}_{i_2} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1,j_1} \bar{S}\]

evaluates to 1 on \(\vec{\iota}\)

if and only if \(\vec{\iota}\) satisfies any of the following conditions:

1. \(\vec{\iota} \in UTP_{i_1}(S)\),
2. \(\vec{\iota} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_{i}(S) \right)\) such that \(x_{i_2} = 0\),
3. \(\vec{\iota} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_{i}(S) \right)\) such that \(p_{i_1,j_1} + x_{i_2} = 0\),
4. \(\vec{\iota} \in NFP_{i_1,j_1}(S)\).

Hence, the result follows.

**Theorem 7.1.6** (LNF with LIF - Case 2)

Let \(S = p_{i_1} + \cdots + p_{i_m}\) be a Boolean specification in IDNF. Suppose that the literal \(x_{i_1}\) in the \(i_1\)-th term, \(p_{i_1}\) in \(S\) is negated and the literal \(x_{i_2}\) is inserted in \(p_{i_1}\), where \(1 \leq i_1 \leq m, 1 \leq j_1 \leq k_{i_1}, k_{i_1} (> 1)\) is the number of literals of \(p_{i_1}\), and \(x_{i_2}\) is a missing literal of \(p_{i_1}\), the resulting expression denoted as \(I_{LNF}(p_{i_1} - p_{i_1,j_1}) \times LIF(p_{i_1} - p_{i_1}x_{i_2})\) is equivalent to Expression (6.6) in Table 6.7. Then, we have \(S \neq I_{LNF}(p_{i_1} - p_{i_1,j_1}) \times LIF(p_{i_1} - p_{i_1}x_{i_2})\) if and only if there is a test case \(\vec{\iota}\) that satisfies any of the following conditions:

1. \(\vec{\iota} \in UTP_{i_1}(S)\), or
2. \(\vec{\iota} \in NFP_{i_1,j_1}(S)\) such that \(x_{i_2} = 1\).
Proof: First, we observe that $S \oplus I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LIF(p_1 \rightarrow p_1,x_2)}$

\[ \equiv (p_{1,i} \oplus p_{1,j_1}, x_2) \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m \]

\[ \equiv (p_{1,i} (\bar{p}_1, x_2) + p_{1,j_1}, x_2) \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m \]

\[ \equiv (p_{1,i} + p_{1,j_1}, x_2) \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m \]

\[ \equiv (\bar{p}_1, \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m + p_{1,j_1}, x_2 S) \]

(By making use of \((AB)(ABC) \equiv AB\))

\[ \equiv p_{1,i} \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m + p_{1,j_1}, x_2 \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m \]

\[ \equiv p_{1,i} \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m + p_{1,j_1}, x_2 S \]

Now, $S(\bar{t}) \neq I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LIF(p_1 \rightarrow p_1,x_2)}(\bar{t})$

if and only if $S(\bar{t}) + I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LIF(p_1 \rightarrow p_1,x_2)}(\bar{t}) = 1$

if and only if $p_{1,i} \bar{p}_1 \cdot \bar{p}_i - \bar{p}_{i+1} \cdot \bar{p}_m + p_{1,j_1}, x_2 S$ evaluates to 1 on $\bar{t}$

if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$, or
2. $\bar{t} \in NFP_{i_1,j_1}(S)$ such that $x_{i_2} = 1$.

Hence, the result follows. \(\square\)

Theorem 7.1.7 (LNF with LRF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{j_1}^i$ in the $i_1$-th term, $p_{i_1}$, in $S$ is negated and the literal $x_{j_2}^i$ in the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by $x_{i_2}$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_1}(> 1)$ and $k_{i_2}$ are the numbers of literals of $p_{i_1}$ and $p_{i_2}$, respectively, and $x_{i_2}$ is a missing literal of $p_{i_2}$.

(a) When $k_{i_2} > 1$, the resulting expression denoted as $I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LRF(p_{i_2} \rightarrow p_{i_2,j_2}, x_{i_2})}$ is equivalent to Expression (6.7) in Table 6.7. Then, $S \neq I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LRF(p_{i_2} \rightarrow p_{i_2,j_2}, x_{i_2})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$,
2. $\bar{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} \right) TP_i(S)$ such that $x_{i_2} = 0$,
3. $\bar{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} \right) TP_i(S)$ such that $p_{i_1,j_1} + x_{i_2} = 0$,
4. $\bar{t} \in NFP_{i_1,j_1}(S)$, or
5. $\bar{t} \in NFP_{i_2,j_2}(S)$ such that $x_{i_2} = 1$.

(b) When $k_{i_2} = 1$, the resulting expression denoted as $I_{LNF(p_1 \rightarrow p_{1,j_1}) \times LRF(p_{i_2} \rightarrow x_{i_2})}$ is equivalent to Expression (6.7) in Table 6.7 without $p_{i_2,j_2}$. Then,
\[ S \neq I_{\text{LNF}(p_1 \rightarrow p_{i_1, j_1}) \times \text{LRF}(p_2 \rightarrow x_{l_2})} \] if and only if there is a test case \( \tilde{t} \) that satisfies any of the following conditions:

1. \( \tilde{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i = 1}^{m} TP_i(S) \right) \) such that \( x_{l_2} = 0 \),

2. \( \tilde{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i = 1}^{m} TP_i(S) \right) \) such that \( p_{i_1, j_1} + x_{l_2} = 0 \),

3. \( \tilde{t} \in NFP_{i_1, j_1}(S) \), or

4. \( \tilde{t} \in FP(S) \) such that \( x_{l_2} = 1 \).

**Proof:** (a) First, we observe that

\[
S \oplus I_{\text{LNF}(p_1 \rightarrow p_{i_1, j_1}) \times \text{LRF}(p_2 \rightarrow x_{l_2})} \equiv (p_1 + p_{i_2}) \oplus (p_{i_1, j_1} + p_{i_2, j_2}x_{l_2}) \equiv (p_1 + p_{i_2})(p_{i_1, j_1} + p_{i_2, j_2}x_{l_2}) + (p_{p_1} + p_{i_2})(p_{i_1, j_1} + p_{i_2, j_2}x_{l_2})p_1 \cdots p_{i_2-1}p_{i_2}p_{i_2+1} \cdots p_m \equiv (p_1 + p_{i_2})p_{i_1, j_1}(p_{i_2, j_2}x_{l_2}) + p_{i_1, j_1}p_{i_2}x_{l_2} + p_{i_1, j_1}p_{i_2} + p_{i_1, j_1}p_{i_2}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2}p_{i_2+1} \cdots p_m \]

(By making use of \( AB(\overline{AB}) \equiv AB \) and \( \overline{(AB)(AC)} \equiv \overline{ABC} \))

\[
\equiv p_{i_1}p_{i_2, j_2}p_1 \cdots p_{i_1-1}p_{i_1+1}p_{i_2-1}p_{i_2+1}p_{i_2-1}p_{i_2+1}p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m \]

(By making use of \( (\overline{A})(\overline{A}) \) because they are equivalent)

\[
\equiv p_{i_1}p_{i_2, j_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m + p_{i_1, j_1}x_{l_2}p_1 \cdots p_{i_1-1}p_{i_1+1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_m
\]
Now, $S(\vec{t}) \neq I_{LNF(p_{i_1} \to p_{i_1}, j_1)}(\chi_{LRF(p_{i_2} \to p_{i_2}, x_{l_2})})(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{LNF(p_{i_1} \to p_{i_1}, j_1)}(\chi_{LRF(p_{i_2} \to p_{i_2}, x_{l_2})})(\vec{t}) = 1$

if and only if $p_{i_1}\vec{p}_{i_2,j_2} \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m + \nabla p_{i_1} x_{l_2} \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m + p_{i_1,j_1} \vec{S} + p_{i_2,j_2} x_{l_2} \vec{S}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$,

2. $\vec{t} \in TP_{i_1}(S) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)$ such that $x_{l_2} = 0$,

3. $\vec{t} \in TP_{i_2}(S) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)$ such that $p_{i_1,j_1} + x_{l_2} = 0$,

4. $\vec{t} \in NFP_{i_1,j_1}(S)$, or

5. $\vec{t} \in NFP_{i_2,j_2}(S)$ such that $x_{l_2} = 1$.

(b) The proof is similar to part (a) except that the term $p_{i_2,j_2}$ does not appear in the proof.

First, we observe that $S \oplus I_{LNF(p_{i_1} \to p_{i_1}, j_1)}(\chi_{LRF(p_{i_2} \to x_{l_2})})$

$\equiv (\chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}}) + \chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}})) \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m$

$\equiv (\chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}}) + \chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}})) \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m$

$\equiv (\chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}}) + \chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}})) \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m$

$\equiv (\chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}}) + \chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}})) \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m$

$\equiv (\chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}}) + \chi_{p_{i_1} + p_{i_2}}(\chi_{p_{i_1,j_1} + x_{l_2}})) \vec{p}_1 \cdot \cdot \cdot \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdot \cdot \cdot \vec{p}_m$
Now, 

\[ S(\bar{t}) \neq I_{LNF}(p_{i_1} - p_{i_1,j_1}) \cdot LRF(p_{i_2} - x_{l_2})(\bar{t}) \]

if and only if 

\[ S(\bar{t}) \oplus I_{LNF}(p_{i_1} - p_{i_1,j_1}) \cdot LRF(p_{i_2} - x_{l_2})(\bar{t}) = 1 \]

if and only if 

\[
\begin{align*}
& p_{i_1}x_{l_2}p_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m \\
+ & p_{i_2}(p_{i_1,j_1} + x_{l_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2+1} \cdots \bar{p}_m + \\
& p_{i_1,j_1}\vec{t} + x_{l_2}\vec{S} \text{ evaluates to 1 on } \bar{t}
\end{align*}
\]

if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( x_{l_2} = 0 \),

2. \( \bar{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( p_{i_1,j_1} + x_{l_2} = 0 \),

3. \( \bar{t} \in NFP_{i_1,j_1}(S) \), or

4. \( \bar{t} \in FP(S) \) such that \( x_{l_2} = 1 \).

Hence, the result follows. \( \square \)

It should be noted that there are two differences between the detection conditions of Theorem 7.1.7 (a) and (b). First, detection condition 1 of Theorem 7.1.7 (a) is related to term \( p_{i_2,j_2} \) which does not exist in detection conditions of Theorem 7.1.7(b) when \( k_{i_2} = 1 \) (that is, when \( p_{i_2} \) contains just one literal). Second, detection condition 4 of Theorem 7.1.7(a) (that is, \( "\bar{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{l_2} = 1" \)) differs from detection condition 4 of Theorem 7.1.7(b) syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when \( k_{i_2} = 1 \). It is because of the following reason

\[
\bar{t} \in FP(S) \text{ such that } x_{l_2} = 1 \\
\equiv \bar{t} \in FP(S) \text{ such that } p_{i_2} = x_{l_2}^0 = 0 \text{ and } x_{l_2} = 1 \\
\equiv \bar{t} \in FP(S) \text{ such that } p_{i_2,1} = x_{l_2}^1 = 1 \text{ and } x_{l_2} = 1 \\
\equiv \bar{t} \in NFP_{i_2,1}(S) \text{ such that } x_{l_2} = 1
\]

(Please be noted that \( j_2 = 1 \) when \( k_{i_2} = 1 \))

Hence, without loss of generality, we can still use the five detection conditions in Theorem 7.1.7 (a) to represent the detection conditions of Expression (6.7) in Table 6.7 for both (a) and (b) (that is when \( k_{i_2} \geq 1 \)), bearing in mind that, when \( k_{i_2} = 1 \), \( p_{i_2,j_2}x_{l_2} \) degenerates to \( x_{l_2} \) and \( "\bar{t} \in FP(S) \) such that \( x_{l_2} = 1" \) is equivalent to \( "\bar{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{l_2} = 1" \). We will make similar comments in theorems related to special situations when either \( k_{i_1} \) or \( k_{i_2} \) is equal to 1 or 2 in the sequel.

**Theorem 7.1.8 (LNF with LRF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the literal \( x_{j_1}^{i_1} \) in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the literal \( x_{j_2}^{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in
$S$ is replaced by $x_{i_2}$, where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_{i_1}$, $k_{i_1}$ is the number of literals of $p_{i_1}$, and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting expression denoted as $I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times \text{LRF}(p_{i_1} \rightarrow p_{i_2}, j_2)$ is equivalent to Expression (6.8) in Table 6.7. Then, we have $S \neq I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times \text{LRF}(p_{i_1} \rightarrow p_{i_2}, j_2)$ if and only if there is a test case $\tilde{t}$ that satisfies any of the following conditions:

1. $\tilde{t} \in \text{UTP}_{i_1}(S)$, or
2. $\tilde{t} \in \text{FP}(S)$ such that $p_{i_1,j_1,j_2,x_{i_2}} = 1$.

**Proof**: First, we observe that $S \oplus I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times \text{LRF}(p_{i_1} \rightarrow p_{i_2}, j_2)$

\[
\equiv (p_{i_1} \oplus p_{i_1,j_1,j_2,x_{i_2}})\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m}
\equiv (p_{i_1}\bar{p}_{i_1,j_1,j_2,x_{i_2}} + \bar{p}_{i_1} p_{i_1,j_1,j_2,x_{i_2}})\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m}
\equiv (p_{i_1} + \bar{p}_{i_1} p_{i_1,j_1,j_2,x_{i_2}})\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m}
\]

(By making use of $ABC(\overline{ABD}) \equiv ABC$)

\[
\equiv p_{i_1}\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m} + p_{i_1,j_1,j_2,x_{i_2}}\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m}
\equiv p_{i_1}\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m} + p_{i_1,j_1,j_2,x_{i_2}}\overline{S}
\]

Now, $S(\tilde{t}) \neq I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times \text{LRF}(p_{i_1} \rightarrow p_{i_2}, j_2)(\tilde{t})$

if and only if $S(\tilde{t}) \oplus I_{\text{LNF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times \text{LRF}(p_{i_1} \rightarrow p_{i_2}, j_2)(\tilde{t}) = 1$

if and only if $p_{i_1}\bar{p}_{1} \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_{m} + p_{i_1,j_1,j_2,x_{i_2}}\overline{S}$ evaluates to 1 on $\tilde{t}$

if and only if $\tilde{t}$ satisfies any of the following conditions:

1. $\tilde{t} \in \text{UTP}_{i_1}(S)$, or
2. $\tilde{t} \in \text{FP}(S)$ such that $p_{i_1,j_1,j_2,x_{i_2}} = 1$.

Hence, the result follows. □

### 7.2 LOF with Other Literal Faults

**Theorem 7.2.1 (LOF with LOF - Case 1)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{j_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ and $x_{j_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ are omitted from $p_{i_1}$ and $p_{i_2}$, respectively, where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $1 \leq j_2 \leq k_{i_2}$ and $k_{i_1} (> 1)$ and $k_{i_2} (> 1)$ are the numbers of literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting expression denoted as $I_{\text{LOF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times I_{\text{LOF}}(p_{i_2} \rightarrow p_{i_2}, j_2)$ is equivalent to Expression (6.9) in Table 6.7. Then, we have $S \neq I_{\text{LOF}}(p_{i_1} \rightarrow p_{i_2}, j_1) \times I_{\text{LOF}}(p_{i_2} \rightarrow p_{i_2}, j_2)$ if and only if there is a test case $\tilde{t}$ that satisfies any of the following conditions:

1. $\tilde{t} \in \text{NFP}_{i_1,j_1}(S)$, or
2. $\tilde{t} \in \text{NFP}_{i_2,j_2}(S)$.
Proof: First, we observe that $S \oplus I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_2 \rightarrow p_{i2,j2})}$

\[ \equiv ((p_{i1} + p_{i2}) \oplus (p_{i1,j1} + p_{i2,j2})) \bar{p}_1 \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

\[ \equiv ((p_{i1} + p_{i2}) \bar{p}_{i1,j1} + p_{i2,j2}) \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

\[ = ((p_{i1} + p_{i2}) \bar{p}_{i1,j1} \bar{p}_{i2,j2} + p_{i2} \bar{p}_i (p_{i1,j1} + p_{i2,j2})) \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

\[ \equiv (0 + 0 + p_{i1,j1} \bar{p}_{i1} \bar{p}_{i2} + \bar{p}_{i1} \bar{p}_{i2} p_{i2,j2}) \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

(By making use of $AB(A) \equiv 0$ and rewriting $(AB)(A)$ as $(AB)(AB)$)

because they are equivalent)

\[ \equiv p_{i1,j1} \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

\[ + p_{i2,j2} \bar{p}_{i1} \bar{p}_{i2} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_{i2-1} \bar{p}_{i2} \cdots \bar{p}_m \]

\[ \equiv p_{i1,j1} \bar{S} + p_{i2,j2} \bar{S} \]

Now, $S(\vec{t}) \neq I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_2 \rightarrow p_{i2,j2})}(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_2 \rightarrow p_{i2,j2})}(\vec{t}) = 1$

if and only if $p_{i1,j1} \bar{S} + p_{i2,j2} \bar{S}$ evaluates to 1 on $\vec{t}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in NFP_{i1,j1}(S)$, or
2. $\vec{t} \in NFP_{i2,j2}(S)$.

Hence, the result follows. □

Theorem 7.2.2 (LOF with LOF - Case 2)

Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{j1}^i$ and $x_{j2}^i$ in the $i$-th term, $p_{i1}$, in $S$ are omitted from $p_{i1}$ where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_{i1}$ and $k_{i1} > 2$ is the number of literals in $p_{i1}$, the resulting expression denoted as $I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_1 \rightarrow p_{i1,j2})}$ is equivalent to Expression (6.10) in Table 6.7. Then, we have $S \neq I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_1 \rightarrow p_{i1,j2})}$ if and only if there is a test case $\vec{t} \in FP(S)$ such that $p_{i1,j1,j2}=1$.

Proof: First, we observe that $S \oplus I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_1 \rightarrow p_{i1,j2})}$

\[ \equiv (p_{i1} \oplus p_{i1,j1,j2}) \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_m \]

\[ \equiv (p_{i1} \bar{p}_{i1,j1,j2} + p_{i1} p_{i1,j1,j2}) \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_m \]

\[ \equiv p_{i1,j1,j2} \bar{p}_{i1} \cdots \bar{p}_{i1-1} \bar{p}_{i1} \cdots \bar{p}_m \]

\[ \equiv p_{i1,j1,j2} \bar{S} \]

Now, $S(\vec{t}) \neq I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_1 \rightarrow p_{i1,j2})}(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{LOF(p_1 \rightarrow p_{i1,j1})} \cdot I_{LOF(p_1 \rightarrow p_{i1,j2})}(\vec{t}) = 1$

if and only if $p_{i1,j1,j2} \bar{S}$ evaluates to 1 on $\vec{t}$

if and only if $\vec{t} \in FP(S)$ such that $p_{i1,j1,j2}=1$.

Hence, the result follows. □
Theorem 7.2.3 (LOF with LIF - Case 1)
Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{j_1}^{i_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ is omitted from $p_{i_1}$ and the literal $x_{i_2}$ is inserted in the $i_2$-th term, $p_{i_2}$, in $S$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $k_{i_1} (> 1)$ is the number of literals of $p_{i_1}$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting expression denoted as $I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \times LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})$ is equivalent to Expression (6.11) in Table 6.7. Then, we have $S \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \times LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_2}(S)$ such that $p_{i_1,j_1} + x_{i_2} = 0$, or
2. $\bar{t} \in NFP_{i_1,j_1}(S)$.

Proof: First, we observe that $S \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \times LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})$

\[
= ((p_1 + p_2) \oplus (p_{i_1,j_1} + p_{i_2,x_{i_2}})) p_1 \cdots p_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
= ((p_1 + p_2)(p_{i_1,j_1} + p_{i_2,x_{i_2}}) + (p_1 + p_2)(p_{i_1,j_1} + p_{i_2,x_{i_2}})) p_1 \cdots p_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
= ((p_1 + p_2) \bar{p}_{i_1,j_1}(\bar{p}_{i_2,x_{i_2}}) + \bar{p}_{i_1} p_{i_2}(p_{i_1,j_1} + p_{i_2,x_{i_2}})) p_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
= (0 + p_2 \bar{p}_{i_1,j_1} \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2}(p_{i_1,j_1} + 0)) p_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

(By making use of $AB(\bar{A}) \equiv 0$ and $A(\bar{A}B) \equiv \bar{A}\bar{B}$)

\[
= p_2 \bar{p}_{i_1,j_1} \bar{x}_{i_2} p_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
+ p_{i_1,j_1} \bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
= p_2(p_{i_1,j_1} + x_{i_2}) p_1 \cdots p_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1,j_1} \bar{S}
\]

Now, $S(\bar{t}) \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \times LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})(\bar{t})$

if and only if $S(\bar{t}) \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \times LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})(\bar{t}) = 1$

if and only if $p_2(p_{i_1,j_1} + x_{i_2}) p_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1,j_1} \bar{S}$ evaluates to 1 on $\bar{t}$

if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_2}(S)$ such that $p_{i_1,j_1} + x_{i_2} = 0$, or
2. $\bar{t} \in NFP_{i_1,j_1}(S)$.

Hence, the result follows. \qed

Theorem 7.2.4 (LOF with LRF - Case 1)
Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{j_1}^{i_1}$ in the $i_1$-th term, $p_{i_1}$, in $S$ is omitted from $p_{i_1}$ and the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $k_{i_1} (> 1)$ is the number of literals of $p_{i_1}$ and $x_{i_2}$ is a missing literal of $p_{i_2}$.
(a) When \( k_{i_2} > 1 \), the resulting expression denoted as \( I_{LOF(p_1 \rightarrow p_{i_1,j_1})}^{LRF(p_2 \rightarrow p_{i_2,j_2} \cdot x_{i_2})} \) is equivalent to Expression (6.12) in Table 6.7. Then, \( S \not= I_{LOF(p_1 \rightarrow p_{i_1,j_1})}^{LRF(p_2 \rightarrow p_{i_2,j_2} \cdot x_{i_2})} \) if and only if there is a test case \( \tilde{i} \) that satisfies any of the following conditions:

1. \( \tilde{i} \in UTP_{i_2}(S) \) such that \( p_{i_1,j_1} + x_{i_2} = 0 \),
2. \( \tilde{i} \in NFP_{i_1,j_1}(S) \), or
3. \( \tilde{i} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \).

(b) When \( k_{i_2} = 1 \), the resulting expression denoted as \( I_{LOF(p_1 \rightarrow p_{i_1,j_1})}^{LRF(p_2 \rightarrow x_{i_2})} \) is equivalent to Expression (6.12) in Table 6.7. Then, \( S \not= I_{LOF(p_1 \rightarrow p_{i_1,j_1})}^{LRF(p_2 \rightarrow x_{i_2})} \) if and only if there is a test case \( \tilde{i} \) that satisfies any of the following conditions:

1. \( \tilde{i} \in UTP_{i_2}(S) \) such that \( p_{i_1,j_1} + x_{i_2} = 0 \),
2. \( \tilde{i} \in NFP_{i_1,j_1}(S) \), or
3. \( \tilde{i} \in FP(S) \) such that \( x_{i_2} = 1 \).

**Proof:** (a) First, we observe that \( S \oplus I_{LOF(p_1 \rightarrow p_{i_1,j_1})}^{LRF(p_2 \rightarrow p_{i_2,j_2} \cdot x_{i_2})} \)

\[
\equiv (p_{i_1} \oplus p_{i_2}) \oplus (p_{i_1,j_1} \oplus p_{i_2,j_2} \cdot x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv (p_{i_1} \oplus p_{i_2})(p_{i_1,j_1} \oplus p_{i_2,j_2} \cdot x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv (0 + p_{i_2}(p_{i_1,j_1} \bar{x}_{i_2} + \bar{p}_{i_1} \bar{p}_{i_2} p_{i_1,j_1} + \bar{p}_{i_1} p_{i_2,j_2} \cdot x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

(By making use of \((AB)(\bar{A}) \equiv 0\) and \((AB)(AC) \equiv ABC\))

\[
\equiv (p_{i_2}(p_{i_1,j_1} \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2,j_2} \cdot x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

(By rewriting \((AB)A\) as \((AB)AB\) because they are equivalent)

\[
\equiv p_{i_2}(p_{i_1,j_1} \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
+ p_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
+ p_{i_2,j_2} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv p_{i_2}(p_{i_1,j_1} \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1,j_1} \bar{S} + p_{i_2,j_2} x_{i_2} \bar{S}
\]
Now, \[ S(\vec{t}) \neq I_{LOF(p_{i_1} \rightarrow p_{i_1 \cdot j_1}) \ast LRF(p_{i_2} \rightarrow x_{i_2}})(\vec{t}) \]
if and only if \[ S(\vec{t}) \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1 \cdot j_1}) \ast LRF(p_{i_2} \rightarrow x_{i_2}})(\vec{t}) = 1 \]
if and only if \[ p_{i_2}(p_{i_1 \cdot j_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1 \cdot j_1} \bar{S} + p_{i_2 \cdot j_2} x_{i_2} \bar{S} \]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_2}(S) \) such that \( p_{i_1 \cdot j_1} + x_{i_2} = 0 \),
2. \( \vec{t} \in NFP_{i_1 \cdot j_1}(S) \), or
3. \( \vec{t} \in NFP_{i_2 \cdot j_2}(S) \) such that \( x_{i_2} = 1 \).

Hence, the result follows.

(b) The proof is similar to that of (a) above except that the terms \( p_{i_2 \cdot j_2} \) does not appear in the proof.

First, we observe that \[ S \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1 \cdot j_1}) \ast LRF(p_{i_2} \rightarrow x_{i_2}} \]
evaluates to 0 on \( \vec{t} \).

\[ \equiv (p_{i_1} + p_{i_2})(p_{i_1 \cdot j_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1} + p_{i_2})(p_{i_1 \cdot j_1} + x_{i_2}) (p_{i_1 \cdot j_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1} + p_{i_2})p_{i_1 \cdot j_1}x_{i_2} + \bar{p}_1 \bar{p}_2 p_{i_1 \cdot j_1} + \bar{p}_1 \bar{p}_2 x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (0 + p_{i_2}p_{i_1 \cdot j_1} x_{i_2} + \bar{p}_1 \bar{p}_2 p_{i_1 \cdot j_1} + \bar{p}_1 \bar{p}_2 x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \text{By making use of } (AB)(A) \equiv 0 \]
\[ \equiv (p_{i_2}p_{i_1 \cdot j_1} x_{i_2} + \bar{p}_1 \bar{p}_2 p_{i_1 \cdot j_1} + \bar{p}_1 \bar{p}_2 x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \text{By rewriting } (AB)(A) \text{ as } (AB)AB \text{ because they are equivalent) \]
\[ \equiv p_{i_2}p_{i_1 \cdot j_1} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ + p_{i_1 \cdot j_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ + x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv p_{i_2}(p_{i_1 \cdot j_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1 \cdot j_1} \bar{S} + x_{i_2} \bar{S} \]

Now, \[ S(\vec{t}) \neq I_{LOF(p_{i_1} \rightarrow p_{i_1 \cdot j_1}) \ast LRF(p_{i_2} \rightarrow x_{i_2}})(\vec{t}) \]
if and only if \[ S(\vec{t}) \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1 \cdot j_1}) \ast LRF(p_{i_2} \rightarrow x_{i_2}})(\vec{t}) = 1 \]
if and only if \[ p_{i_2}(p_{i_1 \cdot j_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_1 \cdot j_1} \bar{S} + x_{i_2} \bar{S} \]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_2}(S) \) such that \( p_{i_1 \cdot j_1} + x_{i_2} = 0 \),
2. \( \vec{t} \in NFP_{i_1 \cdot j_1}(S) \), or
3. \( \vec{t} \in FP(S) \) such that \( x_{i_2} = 1 \).

Hence, the result follows. \( \square \)
It should be noted that there is a difference between detection conditions of Theorem 7.2.4(a) and (b). Detection condition 3 of Theorem 7.2.4(a) (that is, “$\vec{t} \in NFP_{i_2,j_2}(S) \text{ such that } x_{l_2} = 1$”) differs from detection condition 3 in Theorem 7.2.4(b) syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when $k_{l_2} = 1$ (that is, $p_{l_2}$ contains just one literal). It is because of the following reason

\[ \vec{t} \in FP(S) \text{ such that } x_{l_2} = 1 \]
\[ \equiv \vec{t} \in FP(S) \text{ such that } p_{l_2} = x^{l_2}_{i_2} = 0 \text{ and } x_{l_2} = 1 \]
\[ \equiv \vec{t} \in FP(S) \text{ such that } p_{l_2} = x^{l_2}_{i_2} = 1 \text{ and } x_{l_2} = 1 \]
\[ \equiv \vec{t} \in NFP_{i_2,j_2}(S) \text{ such that } x_{l_2} = 1 \]

(Please be noted that $j_2 = 1$ when $k_{l_2} = 1$)

Hence, without loss of generality, we can still use the three detection conditions in Theorem 7.2.4 (a) to represent the detection conditions of Expression (6.12) in Table 6.7 for both situations in Theorem 7.2.4 (a) to represent the detection conditions of Expression (6.12) in Table 6.7 for both situations in Theorem 7.2.4 (a) to represent the detection conditions of Expression (6.12) in Table 6.7.

**Theorem 7.2.5 (LOF with LRF - Case 2)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x^{i_1}_{j_1}$ of the $i_1$-th term, $p_{i_1}$, in $S$ is omitted from $p_{i_1}$ and the literal $x^{i_1}_{j_2}$ of the $i_1$-th term, $p_{i_1}$, in $S$ is replaced by the literal $x_{l_2}$ where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_{i_1}$, $k_{i_1}$ is the number of literals of $p_{i_1}$ and $x_{l_2}$ is a missing literal of $p_{i_1}$.

(a) When $k_{l_2} > 2$, the resulting expression denoted as $I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \ast LRF(p_{i_1} \rightarrow p_{i_1,j_2,x_{l_2}})}$ is equivalent to Expression (6.13) in Table 6.7. Then, $S \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \ast LRF(p_{i_1} \rightarrow p_{i_1,j_2,x_{l_2}})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_1}(S)$ such that $x_{l_2} = 0$,
2. $\vec{t} \in NFP_{i_1,j_1}(S)$ such that $x_{l_2} = 1$,
3. $\vec{t} \in NFP_{i_1,j_2}(S)$ such that $x_{l_2} = 1$, or
4. $\vec{t} \in FP(S)$ such that $p_{i_1,j_1,j_2,x_{l_2}} = 1$.

(b) When $k_{l_2} = 2$, the resulting expression denoted as $I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \ast LRF(p_{i_1} \rightarrow p_{i_1,j_2,x_{l_2}})}$ is equivalent to Expression (6.13) in Table 6.7 without $p_{i_1,j_1,j_2}$ because $p_{i_1}$ contains just two literals. Then, we have $S \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1}) \ast LRF(p_{i_1} \rightarrow p_{i_1,j_2,x_{l_2}})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_1}(S)$ such that $x_{l_2} = 0$, or
2. $\vec{t} \in FP(S)$ such that $x_{l_2} = 1$.
Proof: (a) First, we observe that $S \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})$
\[\equiv (p_{i_1} \oplus p_{i_1,j_1,j_2} x_{i_2}) \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv (p_{i_1} \bar{x}_{i_2} + \bar{\bar{p}}_1 p_{i_1,j_1,j_2} x_{i_2}) \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv (p_{i_1} \bar{x}_{i_2} + \bar{\bar{p}}_1 p_{i_1,j_1,j_2} x_{i_2} + \bar{\bar{p}}_1 p_{i_1,j_1,j_2} x_{i_2}) \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
(By making use of $(ABC)(AD) \equiv ABCD$ and $(ABC)A \equiv (ABC)(ABC + ABD + ABC)$)
\[\equiv p_{i_1} \bar{x}_{i_2} p_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + p_{i_1,j_1} x_{i_2} p_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[+ p_{i_1,j_2} x_{i_2} p_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + p_{i_1,j_1,j_2} x_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv p_{i_1} \bar{x}_{i_2} p_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + p_{i_1,j_1} x_{i_2} \bar{\bar{S}} + p_{i_1,j_2} x_{i_2} \bar{\bar{S}} + p_{i_1,j_1,j_2} x_{i_2} \bar{\bar{S}} \]

Now, $S(\bar{\bar{t}}) \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})(\bar{\bar{t}})$
if and only if $S(\bar{\bar{t}}) \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})(\bar{\bar{t}}) = 1$
if and only if $p_{i_1} \bar{x}_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + p_{i_1,j_1} x_{i_2} \bar{\bar{S}} + p_{i_1,j_2} x_{i_2} \bar{\bar{S}} + p_{i_1,j_1,j_2} x_{i_2} \bar{\bar{S}}$ evaluates to 1 on $\bar{\bar{t}}$
if and only if $\bar{\bar{t}}$ satisfies any of the following conditions:

1. $\bar{\bar{t}} \in UTP_{i_1}(S)$ such that $x_{i_2} = 0$,
2. $\bar{\bar{t}} \in NFP_{i_1,j_1}(S)$ such that $x_{i_2} = 1$,
3. $\bar{\bar{t}} \in NFP_{i_1,j_2}(S)$ such that $x_{i_2} = 1$, or
4. $\bar{\bar{t}} \in FP(S)$ such that $p_{i_1,j_1,j_2} x_{i_2} = 1$.

Hence, the result follows.

(b) The proof is similar to that of (a) above except that the term $p_{i_1,j_1,j_2}$ does not appear in the proof.

First, we observe that $S \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})$
\[\equiv (p_{i_1} \oplus x_{i_2}) \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv (p_{i_1} \bar{x}_{i_2} + \bar{\bar{p}}_1 x_{i_2}) \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv p_{i_1} \bar{x}_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + x_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m \]
\[\equiv p_{i_1} \bar{x}_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + x_{i_2} \bar{\bar{S}} \]

Now, $S(\bar{\bar{t}}) \neq I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})(\bar{\bar{t}})$
if and only if $S(\bar{\bar{t}}) \oplus I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \cdot LRF(p_{i_1} \rightarrow p_{i_1,j_2} x_{i_2})(\bar{\bar{t}}) = 1$
if and only if $p_{i_1} \bar{x}_{i_2} \bar{\bar{p}}_1 \cdots \bar{\bar{p}}_{i_1} \bar{\bar{p}}_{i_1+1} \cdots \bar{\bar{p}}_m + x_{i_2} \bar{\bar{S}}$ evaluates to 1 on $\bar{\bar{t}}$
if and only if $\bar{\bar{t}}$ satisfies any of the following conditions:

1. $\bar{\bar{t}} \in UTP_{i_1}(S)$ such that $x_{i_2} = 0$, or
2. $\bar{\bar{t}} \in FP(S)$ such that $x_{i_2} = 1$.

Hence, the result follows. \qed
Although detection conditions 2, 3 and 4 of Theorem 7.2.5(a) are syntactically different from the detection condition 2 of Theorem 7.2.5(b), they are actually equivalent to each other when $k_{i_1} = 2$ (that is $p_{i_1}$ contains just two literals) because of the following reason

\[
\tilde{t} \in FP(S) \text{ such that } x_{l_2} = 1
\]
\[
\equiv \tilde{t} \in FP(S) \text{ such that } p_{i_1} = x_1^{i_1}x_2^{i_1} = 0 \text{ and } x_{l_2} = 1
\]
\[
(p_{i_1} \text{ contains just two literals when } k_{i_1} = 2)
\]
\[
\equiv \tilde{t} \in FP(S) \text{ such that } \bar{p}_{i_1} = x_1^{i_1}x_2^{i_1} = 1 \text{ and } x_{l_2} = 1
\]
\[
\equiv \tilde{t} \in FP(S) \text{ such that } \bar{p}_{i_1} = x_1^{i_1}x_2^{i_1} + x_1^{i_1}\bar{x}_2^{i_1} + \bar{x}_1^{i_1}x_2^{i_1} = 1 \text{ and } x_{l_2} = 1
\]
\[
\equiv \tilde{t} \in FP(S) \text{ such that }
\]
\[
(1) p_{i_1,1} = \bar{x}_1^{i_1}x_2^{i_1} = 1 \text{ and } x_{l_2} = 1,
\]
\[
(2) p_{i_1,2} = x_1^{i_1}\bar{x}_2^{i_1} = 1 \text{ and } x_{l_2} = 1, \text{ or}
\]
\[
(3) p_{i_1,1,2} = \bar{x}_1^{i_1}x_2^{i_1} = 1 \text{ and } x_{l_2} = 1
\]
\[
\equiv \tilde{t} \in FP(S) \text{ such that }
\]
\[
(1) \tilde{t} \in NFP_{i_1,1}(S) \text{ such that } x_{l_2} = 1,
\]
\[
(2) \tilde{t} \in NFP_{i_1,2}(S) \text{ such that } x_{l_2} = 1, \text{ or}
\]
\[
(3) \tilde{t} \in FP(S) \text{ such that } p_{i_1,1,2}x_{l_2} = 1
\]

Hence, without loss of generality, we can still use the four detection conditions in Theorem 7.2.5(a) to represent those of Expression (6.13) in Table 6.7 for both (a) and (b), bearing in mind that detection conditions 2, 3 and 4 degenerate to “$\tilde{t} \in FP(S)$ such that $x_{l_1}x_{l_2} = 1$” when $k_{i_1} = 2$.

### 7.3 LIF with Other Literal Faults

**Theorem 7.3.1 (LIF with LIF - Case 1)**

Let $S = p_1 \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{i_1}$ and $x_{i_2}$ are inserted in the $i_1$-th term, $p_{i_1}$ and the $i_2$-th term, $p_{i_2}$, in $S$, respectively, where $1 \leq i_1 < i_2 \leq m$, and $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting expression denoted as $I_{LIF(p_{i_1} \rightarrow p_{i_1}x_{i_1})} \times LIF(p_{i_2} \rightarrow p_{i_2}x_{i_2})$ is equivalent to Expression (6.14) in Table 6.7. Then, we have $S \neq I_{LIF(p_{i_1} \rightarrow p_{i_1}x_{i_1})} \times LIF(p_{i_2} \rightarrow p_{i_2}x_{i_2})$ if and only if there is a test case $\tilde{t}$ that satisfies any of the following conditions:

1. $\tilde{t} \in UTP_{i_1}(S)$ such that $x_{i_1} = 0$,
2. $\tilde{t} \in UTP_{i_2}(S)$ such that $x_{i_2} = 0$,
3. $\tilde{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i=1}^{m} \bigcup_{j \neq i_1, i_2} TP_i(S) \right)$ such that $x_{i_1} + x_{i_2} = 0$.

**Proof:** First, we observe that $S \oplus I_{LIF(p_{i_1} \rightarrow p_{i_1}x_{i_1})} \times LIF(p_{i_2} \rightarrow p_{i_2}x_{i_2})$

\[
\equiv ((p_{i_1} + p_{i_2}) \oplus (p_{i_1}x_{i_1} + p_{i_2}x_{i_2}))\bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+1} \cdots \bar{p}_{i_2-1}p_{i_2+1} \cdots \bar{p}_m
\]
\( \equiv (p_{i_1} + p_{i_2})(p_{i_1}x_{i_1} + p_{i_2}x_{i_2}) + (p_{i_1} + p_{i_2})(p_{i_1}x_{i_1} + p_{i_2}x_{i_2}) \)
\( \equiv (p_{i_1} + p_{i_2})p_{i_1}x_{i_1} + p_{i_1}p_{i_2}(p_{i_1}x_{i_1} + p_{i_2}x_{i_2}) \)
\( \equiv (p_{i_1} + p_{i_2})p_{i_1}x_{i_1} + p_{i_1}p_{i_2}(p_{i_1}x_{i_1} + p_{i_2}x_{i_2}) + 0 + 0 \)
\( \equiv (p_{i_1}x_{i_2}p_{i_2}x_{i_2} + p_{i_2}x_{i_2}p_{i_1}x_{i_1}) \)
\( \equiv (p_{i_1}x_{i_2}p_{i_2}x_{i_2} + p_{i_2}x_{i_2}p_{i_1}x_{i_1}) + \bar{p}_{i_1} \cdots \bar{p}_{i_1-1}p_{i_1+1} \cdots \bar{p}_{i_2-1}p_{i_2+1} \cdots \bar{p}_m \)

(by making use of \( A(\overline{AB}) = \overline{A} \))
\( \equiv (p_{i_1}x_{i_1}p_{i_2}x_{i_2} + p_{i_2}x_{i_2}p_{i_1}x_{i_1}) - p_{i_1} \cdots \bar{p}_{i_1-1}p_{i_1+1} \cdots \bar{p}_{i_2-1}p_{i_2+1} \cdots \bar{p}_m \)

(by rewriting \( \overline{AB} \) as \( \overline{A} + A \cdot \overline{B} \) because they are equivalent)

\[ S(\overline{t}) \neq I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_1}) \cdot I_{\text{LIF}}(p_{i_2} \rightarrow p_{i_2}x_{i_2})(\overline{t}) \]

if and only if

\[ S(\overline{t}) \oplus I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_1}) \cdot I_{\text{LIF}}(p_{i_2} \rightarrow p_{i_2}x_{i_2})(\overline{t}) = 1 \]

if and only if

\[ p_{i_1}x_{i_1}p_{i_2} \cdots p_{i_1-1}p_{i_1+1} \cdots \bar{p}_{i_2-1}p_{i_2+1} \cdots \bar{p}_m \]

\[ + p_{i_2}x_{i_2}p_{i_1} \cdots p_{i_2-1}p_{i_2+1} \cdots p_{i_1-1}p_{i_1+1} \cdots \bar{p}_{i_2-1}p_{i_2+1} \cdots \bar{p}_m \]

Now, \( \overline{t} \) satisfies any of the following conditions:

1. \( \overline{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),
2. \( \overline{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \), or
3. \( \overline{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} \bigcup_{i \neq i_1, i_2} TP_i(S) \), such that

\[ x_{i_1} + x_{i_2} = 0. \]

Hence, the result follows. \( \square \)

**Theorem 7.3.2 (LIF with LIF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two literals \( x_{i_1} \) and \( x_{i_2} \) are inserted in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) where \( 1 \leq i_1 \leq m \), and \( x_{i_1} \) and \( x_{i_2} \) are two different missing literals of \( p_{i_1} \), the resulting expression denoted as \( I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_1}) \cdot I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_2}) \) is equivalent to Expression (6.15) in Table 6.7. Then, we have \( S \neq I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_1}) \cdot I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_2}) \) if and only if there is a test case \( \overline{t} \) that satisfies any of the following conditions:

1. \( \overline{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \), or
2. \( \overline{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \).

**Proof:** First, we observe that \( S \oplus I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_1}) \cdot I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1}x_{i_2}) \)
\( \equiv (p_{i_1} \oplus p_{i_1}x_{i_1}x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+1} \cdots \bar{p}_m \)
\[ \equiv (p_1 \bar{x}_1 x_1 x_2 + \bar{p}_1 (p_1 x_1 x_2)) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]
\[ \equiv (p_1 x_1 x_2 + 0) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]
(\text{By making use of } A(ABC) \equiv A(BC))

Now, \[ S(\bar{t}) \neq I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LIF}(p_1 \rightarrow p_1 x_2)}(\bar{t}) \]
if and only if \[ S(\bar{t}) \oplus I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LIF}(p_1 \rightarrow p_1 x_2)}(\bar{t}) = 1 \]
if and only if \[ p_1 \bar{x}_1 \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_1 \bar{x}_2 \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]
evaluates to 1 on \( \bar{t} \)
if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} = 0, \)
2. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_2} = 0. \)

Hence, the result follows.

\textbf{Theorem 7.3.3 (LIF with LRF - Case 1)}

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the literal \( x_{i_1} \) is inserted in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) and the literal \( x_{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is replaced by \( x_{i_2} \) where \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_2 \leq k_{i_2}, k_{i_2} \) is the number of literals of \( p_{i_2} \), and \( x_{i_1} \) and \( x_{i_2} \) are missing literals of \( p_{i_1} \) and \( p_{i_2} \), respectively.

(a) When \( k_{i_2} > 1 \), the resulting expression denoted as \( I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LRF}(p_2 \rightarrow p_{i_2} x_{i_2})} \) is equivalent to Expression (6.16) in Table 6.7. Then, \( S \neq I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LRF}(p_2 \rightarrow p_{i_2} x_{i_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( p_{i_2} \bar{x}_{j_2} + x_{i_1} = 0, \)
2. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0, \)
3. \( \bar{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0, \)
4. \( \bar{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1, \) or
5. \( \bar{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i_2 = 1}^{m} TP_i(S) \right) \) such that \( x_{i_1} + x_{i_2} = 0. \)

(b) When \( k_{i_2} = 1 \), the resulting expression denoted as \( I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LRF}(p_2 \rightarrow x_{i_2})} \) is equivalent to Expression (6.16) in Table 6.7 without \( p_{i_2,j_2} \) because \( p_{i_2} \) contains just one literal. Then, we have \( S \neq I_{\text{LIF}(p_1 \rightarrow p_1 x_1)} \cdot I_{\text{LRF}(p_2 \rightarrow x_{i_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0, \)
2. \( \bar{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0, \)
3. $i \in FP(S)$ such that $x_{t_2} = 1$, or

4. $i \in (TP_{t_1}(S) \cap TP_{t_2}(S)) \setminus \left( \bigcup_{i \neq t_1, t_2} TP_i(S) \right)$ such that $x_{t_1} + x_{t_2} = 0$.

**Proof:** (a) First, we observe that $S \oplus I_{LIF}(p_1 \rightarrow p_{t_1}, x_{t_1}) \cong LRF(p_{t_2} \rightarrow p_{t_2}, x_{t_2})$

$$\equiv \left( (p_1 + p_{t_2}) \oplus (p_{t_1} x_{t_1} + p_{t_2} x_{t_2}) \right) \bar{p}_1 \cdots \bar{p}_{t_1} - 1 \bar{p}_{t_1} + 1 \cdots \bar{p}_m$$

$$\equiv \left( (p_{t_1} + p_{t_2}) (p_{t_1} x_{t_1} + p_{t_2} x_{t_2}) \right) \bar{p}_1 \cdots \bar{p}_{t_1} - 1 \bar{p}_{t_1} + 1 \cdots \bar{p}_m$$

$$\equiv \left( (p_{t_1} x_{t_1} + p_{t_2} x_{t_2}) + \bar{p}_1 \bar{p}_2 (p_{t_1} x_{t_1} + p_{t_2} x_{t_2}) \right) \bar{p}_1 \cdots \bar{p}_{t_1} - 1 \bar{p}_{t_1} + 1 \cdots \bar{p}_m$$

(By making use of $A(AB) \equiv AB$ and $AB(AC) \equiv ABC$)

$$\equiv \left( p_{t_1} \bar{x}_1 \bar{p}_{t_2,j_2} \bar{p}_{t_2} + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + \bar{p}_1 \bar{p}_{t_2} \bar{p}_{t_2,j_2} \bar{x}_2 \bar{t}_2 \right)$$

$$\equiv (p_{t_1} \bar{x}_1 \bar{p}_{t_2,j_2} \bar{p}_{t_2} + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + \bar{p}_1 \bar{p}_{t_2} \bar{p}_{t_2,j_2} \bar{x}_2 \bar{t}_2)$$

(By making use of $AC \equiv A \cdot (AB) + (AB) \cdot C + ABC$, $AB \equiv \bar{A} + A \cdot \bar{B}$ and $AB \equiv AB \cdot AB$)

$$\equiv \left( p_{t_1} \bar{x}_1 \bar{p}_{t_2,j_2} \bar{p}_{t_2} + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + \bar{p}_1 \bar{p}_{t_2} \bar{p}_{t_2,j_2} \bar{x}_2 \bar{t}_2 \right)$$

$$\equiv \left( p_{t_1} \bar{x}_1 \bar{p}_{t_2,j_2} \bar{p}_{t_2} + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + \bar{p}_1 \bar{p}_{t_2} \bar{p}_{t_2,j_2} \bar{x}_2 \bar{t}_2 \right)$$

$$\equiv \left( p_{t_1} \bar{x}_1 \bar{p}_{t_2,j_2} \bar{p}_{t_2} + p_{t_1} \bar{p}_{t_2,j_2} \bar{x}_1 \bar{t}_2 + \bar{p}_1 \bar{p}_{t_2} \bar{p}_{t_2,j_2} \bar{x}_2 \bar{t}_2 \right)$$
Now, \( S(\vec{t}) \neq I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes I_{\text{LRF}}(p_{i_2} \rightarrow p_{i_2} x_{i_2})(\vec{t}) \) if and only if
\[
S(\vec{t}) \oplus I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes I_{\text{LRF}}(p_{i_2} \rightarrow p_{i_2} x_{i_2})(\vec{t}) = 1
\]
if and only if
\[
p_{i_1}(\overline{x_{i_1} + x_{i_2}}) \overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m} \\
+ p_{i_1}(\overline{x_{i_1} + x_{i_2}}) \overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m} \\
+ p_{i_1}(\overline{x_{i_1} + x_{i_2}}) \overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m} \\
+ p_{i_1}(\overline{x_{i_1} + x_{i_2}}) \overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m}
\]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in \text{UTP}_{i_1}(S) \) such that \( p_{i_2, i_3} + x_{i_1} = 0 \),
2. \( \vec{t} \in \text{UTP}_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \),
3. \( \vec{t} \in \text{UTP}_{i_2}(S) \) such that \( x_{i_2} = 0 \),
4. \( \vec{t} \in \text{NFP}_{i_2, i_3}(S) \) such that \( x_{i_2} = 1 \), or
5. \( \vec{t} \in (\text{TP}_{i_1}(S) \cap \text{TP}_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2}^{m} \text{TP}_i(S) \) such that \( x_{i_1} + x_{i_2} = 0 \).

Hence, the result follows.

(b) The proof is similar to that in (a) except that the term \( p_{i_2, i_3} \) does not appear in the proof.

First, we observe that \( S \oplus I_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes I_{\text{LRF}}(p_{i_2} \rightarrow x_{i_2}) \)
\[
\equiv ((p_{i_1} + p_{i_2}) \oplus (p_{i_1}, x_{i_1} + x_{i_2}))(\overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m})
\]
\[
\equiv ((p_{i_1} + p_{i_2})(p_{i_1}, x_{i_1} + x_{i_2}) + (p_{i_1} + p_{i_2})(p_{i_1}, x_{i_1} + x_{i_2}))(\overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m})
\]
\[
\equiv (p_{i_1} + p_{i_2})(p_{i_1}, x_{i_1} + x_{i_2})(\overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m})
\]
\[
\equiv p_{i_1}(\overline{x_{i_1} + x_{i_2}}) \overline{\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m} \ni \text{UTP}_{i_1}(S)
\]
\[
\ni \text{UTP}_{i_2}(S)
\]
\[
\ni \text{UTP}_{i_2}(S)
\]
\[
\ni \text{NFP}_{i_2, i_3}(S)
\]
\[
\ni (\text{TP}_{i_1}(S) \cap \text{TP}_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2}^{m} \text{TP}_i(S)
\]
\[
\ni x_{i_1} + x_{i_2} = 0.
\]

125
+p_{i_2} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2} \cdots \bar{p}_m + p_{i_2} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2} \cdots \bar{p}_m \\
≡ p_{i_1} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2} \cdots \bar{p}_m + p_{i_2} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2} \cdots \bar{p}_m + x_{i_2} \overline{S} \\
Now, \quad S(\overline{t}) \not= I_{\text{LIIF}}(p_{i_1} \rightarrow x_{i_1}) \wedge L_{\text{RF}}(p_{i_2} \rightarrow x_{i_2})(\overline{t}) \\
if and only if \quad S(\overline{t}) \oplus I_{\text{LIIF}}(p_{i_1} \rightarrow x_{i_1}) \wedge L_{\text{RF}}(p_{i_2} \rightarrow x_{i_2})(\overline{t}) = 1 \\
if and only if \quad p_{i_1} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m \\
+ p_{i_2} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2} \cdots \bar{p}_m + x_{i_2} \overline{S} \\
+ p_{i_2} p_1 (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2} \cdots \bar{p}_m \\n\text{evaluates to 1 on } \overline{t} \\
if and only if \quad \overline{t} \text{ satisfies any of the following conditions:} \\

1. \overline{t} \in UTP_{i_1}(S) \text{ such that } x_{i_1} + x_{i_2} = 0, \\
2. \overline{t} \in UTP_{i_2}(S) \text{ such that } x_{i_2} = 0, \\
3. \overline{t} \in FP(S) \text{ such that } x_{i_2} = 1, \text{ or} \\
4. \overline{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \text{ such that} \\
\quad x_{i_1} + x_{i_2} = 0.

Hence, the result follows. \( \square \)

It should be noted that there are two differences between detection conditions of Theorem 7.3.3(a) and (b). First, detection condition 1 of Theorem 7.3.3(a) is related to term \( p_{i_2,j_2} \) which does not exist in Theorem 7.3.3(b) when \( k_{i_2} = 1 \) (that is, when \( p_{i_2} \) contains just one literal). Second, detection condition 4 of Theorem 7.3.3(a) (that is, \( \overline{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \)) differs from detection condition 3 in Theorem 7.3.3(b) syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when \( k_{i_2} = 1 \). It is because

\( \overline{t} \in FP(S) \) such that \( x_{i_2} = 1 \) \\
\quad \equiv \quad \overline{t} \in FP(S) \) such that \( p_{i_2} = x_{i_2}^1 = 0 \) and \( x_{i_2} = 1 \) \\
\quad \equiv \quad \overline{t} \in FP(S) \) such that \( p_{i_2,1} = \bar{x}_{i_2}^1 = 1 \) and \( x_{i_2} = 1 \) \\
\quad \equiv \quad \overline{t} \in NFP_{i_2,1}(S) \) such that \( x_{i_2} = 1 \)

(Please be noted that \( j_2 = 1 \) when \( k_{i_2} = 1 \))

Hence, without loss of generality, we can still use the five detection conditions in Theorem 7.3.3 (a) to represent those of Expression (6.16) in Table 6.7 for both situations in Theorem 7.3.3 (a) and (b), bearing in mind the non-existence of the detection condition 1 of (a) and the equivalence between \( \overline{t} \in FP(S) \) such that \( x_{i_2} = 1 \) and \( \overline{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \) when \( k_{i_2} = 1 \).
Theorem 7.3.4 (LIF with LRF - Case 2)

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the literal \( x_{i_1} \) is inserted in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) and the literal \( x_{j_2} \) in \( p_{i_1} \) is replaced by \( x_{i_2} \) where \( 1 \leq i_1 \leq m \), \( 1 \leq j_2 \leq k_{i_1} \), \( k_{i_1} \) is the number of literals in \( p_{i_1} \) and \( x_{i_1} \) and \( x_{i_2} \) are two different missing literals of \( p_{i_1} \) from different Boolean variables (that is, \( x_{i_1} \neq x_{i_2} \) and \( x_{i_1} \neq \bar{x}_{i_2} \)).

(a) When \( k_{i_1} > 1 \), the resulting expression denoted as \( I_{LIF(p_{i_1} - p_{i_1} x_{i_1}) \times LRF(p_{i_1} - p_{i_1} x_{i_1} x_{i_2})} \) is equivalent to Expression (6.17) in Table 6.7. Then, \( S \equiv I_{LIF(p_{i_1} - p_{i_1} x_{i_1}) \times LRF(p_{i_1} - p_{i_1} x_{i_1} x_{i_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),
2. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_2} = 0 \), or
3. \( \bar{t} \in NFP_{i_1,j_2}(S) \) such that \( x_{i_1} x_{i_2} = 1 \).

(b) When \( k_{i_1} = 1 \), the resulting expression denoted as \( I_{LIF(p_{i_1} - p_{i_1} x_{i_1}) \times LRF(p_{i_1} - x_{i_2})} \) is equivalent to that Expression (6.17) in Table 6.7 without \( p_{i_1,j_1} \) because \( p_{i_1} \) contains just one literal. Then, \( S \equiv I_{LIF(p_{i_1} - p_{i_1} x_{i_1}) \times LRF(p_{i_1} - x_{i_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),
2. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_2} = 0 \), or
3. \( \bar{t} \in NFP_{i_1,j_2}(S) \) such that \( x_{i_1} x_{i_2} = 1 \).

Proof: (a) First, we observe that 
\[
S \oplus I_{LIF(p_{i_1} - p_{i_1} x_{i_1}) \times LRF(p_{i_1} - p_{i_1} x_{i_1} x_{i_2})}
\]
\[
\equiv (p_{i_1} \oplus p_{i_1,j_2} x_{i_1} x_{i_2}) p_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv \left( p_{i_1} \bar{p}_{i_1,j_2} x_{i_1} x_{i_2} + p_{i_1,j_2} x_{i_1} x_{i_2} \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv \left( p_{i_1} x_{i_1} x_{i_2} + \bar{p}_{i_1,j_2} x_{i_1} x_{i_2} \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\text{(By making use of } AB(ACD) = A\overline{B}(CD) \text{ and } \overline{A}B(A) = \overline{A}B(AB))
\]
\[
\equiv p_{i_1} \overline{x_{i_1} x_{i_2}} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1,j_2} x_{i_1} x_{i_2} \overline{S}
\]
\[
\equiv p_{i_1} \overline{x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1,j_2} x_{i_1} x_{i_2} \overline{S}
\]
Now, \( S(\vec{t}) \neq I_{LIF(p_1 \rightarrow p_1 x_1)} \times LRF(p_1 \rightarrow p_{1,j_i} x_{i_j}) (\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{LIF(p_1 \rightarrow p_1 x_1)} \times LRF(p_1 \rightarrow p_{1,j_i} x_{i_j}) (\vec{t}) = 1 \)
if and only if \( p_{i_1} \vec{x}_{i_1} p_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_1} \vec{x}_{i_2} p_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_1,j_i} x_{i_1} x_{i_2} S \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1} (S) \) such that \( x_{i_1} = 0 \),
2. \( \vec{t} \in UTP_{i_2} (S) \) such that \( x_{i_2} = 0 \), or
3. \( \vec{t} \in NFP_{i_1,j_i} (S) \) such that \( x_{i_1} x_{i_2} = 1 \).

Hence, the result follows.

(b) The proof is similar to that in (a) except that the term \( p_{i_1,j_i} \) does not appear in the proof.

First, we observe that \( S \oplus I_{LIF(p_1 \rightarrow p_1 x_1)} \times LRF(p_1 \rightarrow x_{i_1}) \)
\[ \equiv (p_{i_1} \oplus x_{i_1} x_{i_2}) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m \]
\[ \equiv (p_{i_1} \vec{x}_{i_1} x_{i_2} + \vec{p}_{i_1} x_{i_1} x_{i_2}) \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m \]
\[ \equiv p_{i_1} \vec{x}_{i_1} p_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_1} \vec{x}_{i_2} p_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + p_{i_1,j_i} x_{i_1} x_{i_2} \vec{S} \]

Now, \( S(\vec{t}) \neq I_{LIF(p_1 \rightarrow p_1 x_1)} \times LRF(p_1 \rightarrow x_{i_1}) (\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{LIF(p_1 \rightarrow p_1 x_1)} \times LRF(p_1 \rightarrow x_{i_1}) (\vec{t}) = 1 \)
if and only if \( p_{i_1} \vec{x}_{i_1} x_{i_2} \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1+1} \cdots \vec{p}_m + x_{i_1} x_{i_2} \vec{S} \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1} (S) \) such that \( x_{i_1} = 0 \),
2. \( \vec{t} \in UTP_{i_2} (S) \) such that \( x_{i_2} = 0 \), or
3. \( \vec{t} \in FP(S) \) such that \( x_{i_1} x_{i_2} = 1 \).

Hence, the result follows. \( \square \)

It should be noted that detection condition 3 in Theorem 7.3.4(a) (that is, \( \vec{t} \in NFP_{i_1,j_i} (S) \) such that \( x_{i_1} x_{i_2} = 1 \)) is equivalent to that in Theorem 7.3.4(b) (that is, \( \vec{t} \in FP(S) \) such that \( x_{i_1} x_{i_2} = 1 \)). The reason is similar to those given in the paragraph after Theorem 7.2.5. Hence, without loss of generality, we can still use the detection conditions in Theorem 7.3.4(a) to represent the detection conditions of double-fault expression (6.17) in Table 6.7 for \( k_{i_2} \geq 1 \), bearing in mind the equivalence between \( \vec{t} \in FP(S) \) such that \( x_{i_1} x_{i_2} = 1 \) and \( \vec{t} \in NFP_{i_1,j_i} (S) \) such that \( x_{i_1} x_{i_2} = 1 \) when \( k_{i_2} = 1 \).
7.4 LRF with Other Literal Faults

Theorem 7.4.1 (LRF with LRF - Case 1)
Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. the literal $x_{j_1}^i$ in the $i_1$-th term, $p_{i_1}$, is replaced by $x_{i_1}$ and the literal $x_{j_2}^i$ in the $i_2$-th term, $p_{i_2}$, is replaced by $x_{i_2}$ where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_1 \leq k_{i_1}$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_1}$ and $k_{i_2}$ are the numbers of literals of $p_{i_1}$ and $p_{i_2}$, respectively, and $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively.

(a) When $k_{i_1}, k_{i_2} > 1$, the resulting expression denoted as $I_{LRF(p_1 \rightarrow p_{i_1} x_{i_1})} \otimes_{LRF(p_2 \rightarrow p_{i_2} x_{i_2})}$ is equivalent to Expression (6.18) in Table 6.7. Then, $S \not\equiv I_{LRF(p_1 \rightarrow p_{i_1} x_{i_1})} \otimes_{LRF(p_2 \rightarrow p_{i_2} x_{i_2})}$ if and only if there is a test case $\bar{i}$ that satisfies any of the following conditions:

1. $\bar{i} \in UTP_{i_1}(S)$ such that $p_{i_2 j_2} + x_{i_1} = 0$,
2. $\bar{i} \in UTP_{i_1}(S)$ such that $x_{i_1} + x_{i_2} = 0$,
3. $\bar{i} \in UTP_{i_2}(S)$ such that $p_{i_1 j_1} + x_{i_2} = 0$,
4. $\bar{i} \in UTP_{i_2}(S)$ such that $x_{i_1} + x_{i_2} = 0$,
5. $\bar{i} \in NFP_{i_1,j_1}(S)$ such that $x_{i_1} = 1$,
6. $\bar{i} \in NFP_{i_2,j_2}(S)$ such that $x_{i_2} = 1$, or
7. $\bar{i} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1}^m TP_i(S)$ such that $x_{i_1} + x_{i_2} = 0$.

(b) When $k_{i_1} = 1$ and $k_{i_2} > 1$, the resulting expression denoted as $I_{LRF(p_1 \rightarrow x_{i_1})} \otimes_{LRF(p_2 \rightarrow x_{i_2})}$ is equivalent to Expression (6.18) in Table 6.7 without $p_{i_1,j_1}$ because $p_{i_1}$ contains just one literal. Then, $S \not\equiv I_{LRF(p_1 \rightarrow x_{i_1})} \otimes_{LRF(p_2 \rightarrow x_{i_2})}$ if and only if there is a test case $\bar{i}$ that satisfies any of the following conditions:

1. $\bar{i} \in UTP_{i_1}(S)$ such that $p_{i_2 j_2} + x_{i_1} = 0$,
2. $\bar{i} \in UTP_{i_1}(S)$ such that $x_{i_1} + x_{i_2} = 0$,
3. $\bar{i} \in UTP_{i_2}(S)$ such that $x_{i_1} + x_{i_2} = 0$,
4. $\bar{i} \in FP(S)$ such that $x_{i_1} = 1$,
5. $\bar{i} \in NFP_{i_2,j_2}(S)$ such that $x_{i_2} = 1$, or
6. $\bar{i} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1}^m TP_i(S)$ such that $x_{i_1} + x_{i_2} = 0$.

(c) When $k_{i_1} > 1$ and $k_{i_2} = 1$, the resulting expression denoted as $I_{LRF(p_1 \rightarrow p_{i_1} x_{i_1})} \otimes_{LRF(p_2 \rightarrow x_{i_2})}$ is equivalent to Expression (6.18) in Table 6.7 without $p_{i_2,j_2}$ because $p_{i_2}$ contains just one literal. Then, we have $S \not\equiv I_{LRF(p_1 \rightarrow p_{i_1} x_{i_1})} \otimes_{LRF(p_2 \rightarrow x_{i_2})}$ if and only if there is a test case $\bar{i}$ that satisfies any of the following conditions:
Proof: (a) First, we observe that 
\[ \vec{t} \in UTP_{i_1}(S) \text{ such that } x_{i_1} + x_{i_2} = 0, \]
\[ \vec{t} \in UTP_{i_2}(S) \text{ such that } p_{i_1,j_1} + x_{i_2} = 0, \]
\[ \vec{t} \in UTP_{i_2}(S) \text{ such that } x_{i_1} + x_{i_2} = 0, \]
\[ \vec{t} \in NFP_{i_1,j_1}(S) \text{ such that } x_{i_1} = 1, \]
\[ \vec{t} \in FP(S) \text{ such that } x_{i_2} = 1, \text{ or } \]
\[ \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{j=1}^{m} TP_j(S) \right) \text{ such that } x_{i_1} + x_{i_2} = 0. \]

(d) When \( k_{i_1} = k_{i_2} = 1 \), the resulting expression denoted as \( I_{LRF}(p_{i_1} \rightarrow x_{i_1}) \)
\( \overset{\text{m}}{\bowtie} LRF(p_{i_2} \rightarrow x_{i_2}) \) is equivalent to Expression (6.18) in Table 6.7 without \( p_{i_1,j_1} \) and \( p_{i_2,j_2} \) because both \( p_{i_1} \) and \( p_{i_2} \) contain just 1 literal. Then, we have \( S \neq I_{LRF}(p_{i_1} \rightarrow x_{i_1}) \overset{\text{m}}{\bowtie} LRF(p_{i_2} \rightarrow x_{i_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0, \)
2. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0, \)
3. \( \vec{t} \in FP(S) \) such that \( x_{i_1} = 1, \)
4. \( \vec{t} \in FP(S) \) such that \( x_{i_2} = 1, \text{ or } \)
5. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{j=1}^{m} TP_j(S) \right) \text{ such that } x_{i_1} + x_{i_2} = 0. \)

(Continued on next page.)
+\( p_{i_2} \bar{x}_{i_1} \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
+\( p_{i_1} p_{i_2} \bar{x}_{i_1} \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
+\( p_{i_1,j_1} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
+\( p_{i_2,j_1} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
\
\[ \equiv p_{i_1} (\bar{p}_{i_2,j_2} + x_{i_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]
+\( p_{i_2} (\bar{p}_{i_1,j_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_2} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_1,j_1} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1,j_1} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_2,j_1} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
\
Now, \( S(\tilde{t}) \neq \text{LRF}(p_{i_1} \rightarrow p_{i_1,j_1} x_{i_1}) \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2}) (\tilde{t}) \)
if and only if \( S(\tilde{t}) \oplus \text{LRF}(p_{i_1} \rightarrow p_{i_1,j_1} x_{i_1}) \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2})(\tilde{t}) = 1 \)
if and only if \( p_{i_1} (\bar{p}_{i_2,j_2} + x_{i_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_2} (\bar{p}_{i_1,j_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_2} (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_1,j_1} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1,j_1} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_2,j_1} x_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
+\( p_{i_1,j_2} x_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
valu-ates to 1 on \( \tilde{t} \)
if and only if \( \tilde{t} \) satisfies any of the following conditions:

1. \( \tilde{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} + x_{i_1} = 0 \), or
2. \( \tilde{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
3. \( \tilde{t} \in UTP_{i_2}(S) \) such that \( p_{i_1,j_1} + x_{i_2} = 0 \), or
4. \( \tilde{t} \in UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
5. \( \tilde{t} \in NFP_{i_1,j_1}(S) \) such that \( x_{i_1} = 1 \), or
6. \( \tilde{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \), or
7. \( \tilde{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( x_{i_1} + x_{i_2} = 0 \).

Hence, the result follows.

(b) The proof is similar to that of (a) above except that the term \( p_{i_1,j_1} \) does not appear in the proof.

First, we observe that \( S \oplus \text{LRF}(p_{i_1} \rightarrow x_{i_1}) \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2}) \)
\[ \equiv (p_{i_1} + p_{i_2}) (x_{i_1} + p_{i_2,j_2} x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1} + p_{i_2}) (x_{i_1} + p_{i_2,j_2} x_{i_2}) + (p_{i_1} + p_{i_2}) (x_{i_1} + x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1} + p_{i_2}) x_{l_1} \cdot \overline{p}_{l_2,j_2} x_{l_2} + \overline{p}_{l_1} \overline{p}_{l_2} (x_{l_1} + p_{l_2,j_2} x_{l_2}) \cdot \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ \equiv (p_{i_1} \overline{x}_{l_1} (\overline{p}_{l_2,j_2} x_{l_2}) + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} + \overline{p}_{l_1} \overline{p}_{l_2} x_{l_1} + \overline{p}_{l_1} \overline{p}_{l_2} p_{l_2,j_2} x_{l_2}) \cdot \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

(By making use of $AB(AC) \equiv A \overline{B} \overline{C}$)

\[ \equiv (p_{i_1} \overline{x}_{l_1} (\overline{p}_{l_2,j_2} \overline{p}_{l_2} x_{l_2} + p_{i_2} \overline{x}_{l_2} + p_{i_2} \overline{x}_{l_2}) + p_{i_2} \overline{x}_{l_1} (\overline{p}_{l_1} \overline{x}_{l_1} + p_{i_1} \overline{x}_{l_1}) + \overline{p}_{l_1} \overline{p}_{l_2} x_{l_1} + \overline{p}_{l_1} \overline{p}_{l_2} p_{l_2,j_2} x_{l_2}) \]

\[ \cdot \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

(By making use of $\overline{ABC} \equiv \overline{A} \cdot (\overline{AB}) + (\overline{AB}) \cdot \overline{C} + ABC \overline{A} \equiv \overline{AB}$

\[ \equiv (p_{i_1} \overline{x}_{l_1} \overline{x}_{l_2} + p_{i_2} \overline{x}_{l_2} + p_{i_2} \overline{x}_{l_2}) + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} + \overline{p}_{l_1} \overline{p}_{l_2} x_{l_1} + p_{i_2} \overline{p}_{l_2} x_{l_1} + \overline{p}_{l_1} \overline{p}_{l_2} \overline{p}_{l_2,j_2} x_{l_2} \]

\[ + \overline{p}_{l_1} \overline{p}_{l_2} p_{l_2,j_2} x_{l_2} \cdot \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ \equiv (p_{i_1} \overline{x}_{l_1} \overline{x}_{l_2} + p_{i_2} \overline{x}_{l_2} + p_{i_2} \overline{x}_{l_2}) + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} + \overline{p}_{l_1} \overline{p}_{l_2} x_{l_1} + p_{i_2} \overline{p}_{l_2} x_{l_1} + \overline{p}_{l_1} \overline{p}_{l_2} p_{l_2,j_2} x_{l_2} \]

\[ + \overline{p}_{l_1} \overline{p}_{l_2} p_{l_2,j_2} x_{l_2} \cdot \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ \equiv p_{i_1} \overline{x}_{l_1} \overline{x}_{l_2} \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + p_{i_1} \overline{x}_{l_1} \overline{x}_{l_2} \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} \overline{p}_{1} \cdots \overline{p}_{l_1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + x_{l_1} \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + p_{i_2} \overline{x}_{l_1} \overline{x}_{l_2} \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ \equiv p_{i_1} (\overline{p}_{l_2,j_2} + x_{l_1}) \overline{p}_{1} \cdots \overline{p}_{l_1+1} \cdots \overline{p}_m + p_{i_1} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_m \]

\[ + p_{i_2} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_2-1} \overline{p}_{l_2+1} \cdots \overline{p}_m + p_{i_1} p_{l_2} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2-1} \cdots \overline{p}_{l_2+1} \cdots \overline{p}_m \]

\[ + x_{l_1} \overline{S} + p_{i_2,j_2} x_{l_2} \overline{S} \]

\[ \equiv p_{i_1} (\overline{p}_{l_2,j_2} + x_{l_1}) \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_m + p_{i_1} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_m \]

\[ + p_{i_2} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_2-1} \overline{p}_{l_2+1} \cdots \overline{p}_m + x_{l_1} \overline{S} + p_{i_2,j_2} x_{l_2} \overline{S} \]

\[ + p_{i_1} p_{l_2} (x_{l_1} + x_{l_2}) \overline{p}_{1} \cdots \overline{p}_{l_1-1} \overline{p}_{l_1+1} \cdots \overline{p}_{l_2-1} \overline{p}_{l_2+1} \cdots \overline{p}_m \]
Now, \( S(\vec{t}) \neq I_{\text{LRF}(p_{i_1} \rightarrow x_{i_1}) \& \text{LRF}(p_{i_2} \rightarrow x_{i_2})}(\vec{t}) \)
if and only if
\[
S(\vec{t}) \oplus I_{\text{LRF}(p_{i_1} \rightarrow x_{i_1}) \& \text{LRF}(p_{i_2} \rightarrow x_{i_2})}(\vec{t}) = 1
\]
if and only if
\[
p_{i_1} \left( p_{i_2, j_2} + x_{i_1} \right) \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_m + p_{i_1} \left( x_{i_1} + x_{i_2} \right) \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_m
+ p_{i_2} \left( x_{i_1} + x_{i_2} \right) \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m + x_{i_1} \bar{S} + p_{i_2, j_2} x_{i_2} \bar{S}
+ p_{i_1} p_{i_2} \left( x_{i_1} + x_{i_2} \right) \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\]
evaluates to 1 on \( \vec{t} \).

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( p_{i_2, j_2} + x_{i_1} = 0 \), or
2. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
3. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
4. \( \vec{t} \in FP(S) \) such that \( x_{i_1} = 1 \), or
5. \( \vec{t} \in NFP_{i_2, j_2}(S) \) such that \( x_{i_2} = 1 \), or
6. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S) \) such that \( x_{i_1} + x_{i_2} = 0 \).

Hence, the result follows. 

(c) The proof is similar to that of (b) except that \( p_{i_2, j_2} \) does not appear in the proof instead of \( p_{i_1, j_1} \). We will omit the proof here.

(d) The proof is similar to that of (a) above except that both \( p_{i_1, j_1} \) and \( p_{i_2, j_2} \) do not appear in the proof. We proceed the proof as follows.

First, we observe that \( S \oplus \text{LRF}(p_{i_1} \rightarrow x_{i_1}) \& \text{LRF}(p_{i_2} \rightarrow x_{i_2}) \)

\[
\equiv \left( (p_{i_1} + p_{i_2}) \oplus (x_{i_1} + x_{i_2}) \right) \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\equiv \left( (p_{i_1} + p_{i_2}) \left( x_{i_1} + x_{i_2} \right) + (p_{i_1} + p_{i_2}) \right) \cdot \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\equiv \left( (p_{i_1} + p_{i_2}) \bar{x}_{i_1} \bar{x}_{i_2} + p_{i_1} \bar{p}_{i_2} \left( x_{i_1} + x_{i_2} \right) \right) \cdot \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\equiv \left( p_{i_1} \bar{x}_{i_1} \bar{x}_{i_2} + p_{i_2} \bar{x}_{i_1} \bar{x}_{i_2} + p_{i_1} \bar{p}_{i_2} x_{i_1} + \bar{p}_{i_1} \bar{p}_{i_2} x_{i_2} \right) \cdot \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\equiv p_{i_1} \bar{x}_{i_1} \bar{x}_{i_2} \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
+ p_{i_2} \bar{x}_{i_1} \bar{x}_{i_2} \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
+ p_{i_1} p_{i_2} \bar{x}_{i_1} \bar{x}_{i_2} \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
+ x_{i_1} \bar{p}_{i_1} \ldots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \ldots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \ldots \bar{p}_m
\]

(By rewriting \( \bar{A} \bar{B} \bar{C} \bar{D} + \bar{B} \bar{C} \bar{D} \) as \( \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} \bar{C} \bar{D} \) because they are equivalent.)
\[ +x_{i_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2}\bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv p_{i_1}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2}\bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_2-1}\bar{p}_{i_2}\bar{p}_{i_2+1} \cdots \bar{p}_m + x_{i_2}\bar{S} + x_{i_2}\bar{S} \]
\[ \equiv p_{i_1}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1}\bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1}\bar{p}_{i_2}\bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_2-1}\bar{p}_{i_2}\bar{p}_{i_2+1} \cdots \bar{p}_m + x_{i_2}\bar{S} \]

Now, if and only if \( S(\bar{t}) \neq I_{LRF(p_{i_1} - x_{i_2})} \cup I_{LRF(p_{i_2} - x_{i_2})}(\bar{t}) \)

if and only if \( S(\bar{t}) + I_{LRF(p_{i_1} - x_{i_1})} \cup I_{LRF(p_{i_2} - x_{i_2})}(\bar{t}) = 1 \)

if and only if \( p_{i_1}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1} \cdots \bar{p}_m + p_{i_2}(x_{i_1} + x_{i_2})\bar{p}_1 \cdots \bar{p}_{i_2-1}\bar{p}_{i_2} \cdots \bar{p}_m + x_{i_2}\bar{S} + x_{i_2}\bar{S} \)

if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
2. \( \bar{t} \in UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \), or
3. \( \bar{t} \in FP(S) \) such that \( x_{i_1} = 1 \), or
4. \( \bar{t} \in FP(S) \) such that \( x_{i_2} = 1 \), or
5. \( \bar{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i_2 \neq i_1, i_2} TP_i(S) \) such that \( x_{i_1} + x_{i_2} = 0 \).

Hence, the result follows. \( \square \)

It should be noted that there are two differences between detection conditions of Theorem 7.4.1(a) and (b). First, detection condition 3 of Theorem 7.4.1(a) is related to term \( p_{i_1,j_1} \) which does not exist in Theorem 7.4.1(b) when \( k_{i_1} = 1 \) (that is, when \( p_{i_1} \) contains just one literal). Second, detection condition 5 of Theorem 7.4.1(a) (that is, \( \bar{t} \in NFP_{i_1,j_1}(S) \) such that \( x_{i_1} = 1 \)) differs from detection condition 4 in Theorem 7.4.1(b) syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when \( k_{i_1} = 1 \). It is because of the following reason

\( \bar{t} \in FP(S) \) such that \( x_{i_1} = 1 \)
\[ \equiv \bar{t} \in FP(S) \) such that \( p_{i_1} = x_{i_1}^j = 1 \) and \( x_{i_1} = 1 \)
\[ \equiv \bar{t} \in FP(S) \) such that \( p_{i_1,1} = x_{i_1}^j = 1 \) and \( x_{i_1} = 1 \)
\[ \equiv \bar{t} \in NFP_{i_1,1}(S) \) such that \( x_{i_1} = 1 \)

(Please be noted that \( j_1 = 1 \) when \( k_{i_1} = 1 \))

Hence, without loss of generality, we can still use the seven detection conditions in Theorem 7.4.1(a) to represent the detection conditions of Expression (6.18) in Table 6.7 for both situations in Theorem 7.4.1(a) and (b), bearing in mind the non-
existence of the term \( p_{i_1,j_1} \) and the equivalence between “\( \vec{t} \in FP(S) \) such that \( x_{i_1} = 1 \)” and “\( \vec{t} \in NFP_{i_1,j_1}(S) \) such that \( x_{i_1} = 1 \)” when \( k_{i_1} = 1 \). Similarly, we can use the seven detection conditions in Theorem 7.4.1(a) to represent the detection conditions of Expression (6.18) in Table 6.7 for situations in Theorem 7.4.1(a), (b), (c) and (d), bearing in mind the non-existence of the term \( p_{i_2,j_2} \) and the equivalence between “\( \vec{t} \in FP(S) \) such that \( x_{i_2} = 1 \)” and “\( \vec{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \)” when \( k_{i_2} = 1 \).

**Theorem 7.4.2 (LRF with LRF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that two literals \( x_{j_1}^{i_1} \) and \( x_{j_2}^{i_2} \) in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) are replaced by \( x_{i_1} \) and \( x_{i_2} \), respectively, where \( 1 \leq i_1 \leq m, 1 \leq j_1 < j_2 \leq k_{i_1}, k_{i_1} \) is the number of literals of \( p_{i_1} \) and \( x_{i_1} \) and \( x_{i_2} \) are two different missing literals of \( p_{i_1} \) from different Boolean variables.

(a) When \( k_{i_1} > 2 \), the resulting expression denoted as \( I_{\text{LRF}}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \times LRF(p_1 \rightarrow p_{i_1,j_2}x_{i_2}) \) is equivalent to Expression (6.19) in Table 6.7. Then, \( S \neq I_{\text{LRF}}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \times LRF(p_1 \rightarrow p_{i_1,j_2}x_{i_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \text{UTP}_{i_1}(S) \) such that \( x_{i_1} = 0 \).
2. \( \vec{t} \in \text{UTP}_{i_1}(S) \) such that \( x_{i_2} = 0 \).
3. \( \vec{t} \in \text{NFP}_{i_1,j_1} \) such that \( x_{i_1}x_{i_2} = 1 \).
4. \( \vec{t} \in \text{NFP}_{i_1,j_2} \) such that \( x_{i_1}x_{i_2} = 1 \), or
5. \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1,j_2} = 1 \) and \( x_{i_1}x_{i_2} = 1 \).

(b) When \( k_{i_1} = 2 \), the resulting expression denoted as \( I_{\text{LRF}}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \times LRF(p_1 \rightarrow p_{i_1,j_2}x_{i_2}) \) is equivalent to Expression (6.19) in Table 6.7 without \( p_{i_1,j_1,j_2} \) because \( p_{i_1} \) contains just two literals. Then, \( S \neq I_{\text{LRF}}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \times LRF(p_1 \rightarrow p_{i_1,j_2}x_{i_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \text{UTP}_{i_1}(S) \) such that \( x_{i_1} = 0 \).
2. \( \vec{t} \in \text{UTP}_{i_1}(S) \) such that \( x_{i_2} = 0 \).
3. \( \vec{t} \in \text{FP}(S) \) such that \( x_{i_1}x_{i_2} = 1 \).

**Proof:** (a) First, we observe that \( S \oplus I_{\text{LRF}}(p_1 \rightarrow p_{i_1,j_1}x_{i_1}) \times LRF(p_1 \rightarrow p_{i_1,j_2}x_{i_2}) \)

\[
\equiv (p_1 \oplus p_{i_1,j_1,j_2}x_{i_1}x_{i_2})\overline{p_1} \cdots \overline{p_{i_1-1}}\overline{p_{i_1}} \cdots \overline{p_m} \\
\equiv (p_1 \overline{p_{i_1,j_1,j_2}x_{i_1}x_{i_2}} + \overline{p_{i_1}}p_{i_1,j_1,j_2}x_{i_1}x_{i_2})\overline{p_1} \cdots \overline{p_{i_1-1}}\overline{p_{i_1}} \cdots \overline{p_m} \\
\equiv (p_1x_{i_1}x_{i_2} + \overline{p_{i_1}}p_{i_1,j_1,j_2}x_{i_1}x_{i_2})\overline{p_1} \cdots \overline{p_{i_1-1}}\overline{p_{i_1}} \cdots \overline{p_m} \\
\text{(By making use of } ABC(\overline{DE}) \equiv ABC(DE)) \\
\equiv (p_1x_{i_1}x_{i_2} + \overline{p_{i_1}}p_{i_1,j_1,j_2}x_{i_1}x_{i_2} + \overline{p_{i_1}}p_{i_1,j_2}x_{i_1}x_{i_2} + \overline{p_{i_1}}p_{i_1,j_2}x_{i_1}x_{i_2})\overline{p_1} \cdots \overline{p_{i_1-1}}\overline{p_{i_1}} \cdots \overline{p_m} \\
\text{(By rewriting } ABCA \text{ as } (ABC)A(BC) + (ABC)AB(UC))
+ \langle ABC \rangle A(\mathcal{B})(\mathcal{C}) \text{ because they are equivalent) }
\equiv p_{i_1} x_{i_1} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1, j_1} x_{i_1} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m
+ p_{i_1, j_2} x_{i_1} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1, j_2} x_{i_1} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdot \cdots \tilde{p}_m
\equiv p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1, j_1} x_{i_1} x_{i_2} \mathcal{S}
+ p_{i_1, j_2} x_{i_1} x_{i_2} \mathcal{S} + p_{i_1, j_1, j_2} x_{i_1} x_{i_2} \mathcal{S}

Now, 
S(\vec{t}) \neq I_{LRF}(p_{i_1, j_1} x_{i_1}) \otimes I_{LRF}(p_{i_1, j_2} x_{i_2})(\vec{t})
if and only if
S(\vec{t}) \oplus I_{LRF}(p_{i_1, j_1} x_{i_1}) \otimes I_{LRF}(p_{i_1, j_2} x_{i_2})(\vec{t}) = 1
if and only if
p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m
+ p_{i_1, j_1} x_{i_1} x_{i_2} \mathcal{S} + p_{i_1, j_2} x_{i_1} x_{i_2} \mathcal{S} + p_{i_1, j_1, j_2} x_{i_1} x_{i_2} \mathcal{S} \text{ evaluates to 1 on } \vec{t}
if and only if \vec{t} \text{ satisfies any of the following conditions: }
1. \vec{t} \in UTP_{t_1}(S) \text{ such that } x_{i_1} = 0,
2. \vec{t} \in UTP_{t_2}(S) \text{ such that } x_{i_2} = 0,
3. \vec{t} \in NFP_{t_1, j_1} \text{ such that } x_{i_1} x_{i_2} = 1,
4. \vec{t} \in NFP_{t_2, j_2} \text{ such that } x_{i_1} x_{i_2} = 1, \text{ or}
5. \vec{t} \in FP(S) \text{ such that } p_{i_1, j_1, j_2} = 1 \text{ and } x_{i_1} x_{i_2} = 1.

Hence, the result follows.

(b) The proof of this part is similar to (a) above except that the term \(p_{i_1, j_1, j_2}\) does not appear in the proof. We proceed the proof as follows.

First, we observe that 

\[ S \oplus I_{LRF}(p_{i_1, j_1, j_2} x_{i_1}) \otimes I_{LRF}(p_{i_1, j_2} x_{i_2}) \]
\[ \equiv (p_{i_1} \oplus x_{i_1} x_{i_2}) \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m \]
\[ \equiv (p_{i_1} x_{i_1} x_{i_2} + p_{i_1} x_{i_1} x_{i_2}) \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m \]
\[ \equiv p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1, j_1} x_{i_1} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m \]
\[ \equiv p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1} x_{i_1} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + x_{i_1} x_{i_2} \mathcal{S} \]
Now, \( S(\vec{t}) \neq \mathcal{ILRF}(p_{i_1,j_1}, x_{l_1}) \kappa \mathcal{ILRF}(p_{i_1,j_2}, x_{l_2})(\vec{t}) \)
if and only if
\[ S(\vec{t}) \oplus \mathcal{ILRF}(p_{i_1,j_1}, x_{l_1}) \kappa \mathcal{ILRF}(p_{i_1,j_2}, x_{l_2})(\vec{t}) = 1 \]
if and only if
\[ p_{i_1,x_{l_1}} p_{i_2,x_{l_2}} \cdots p_{i_m,x_{l_m}} + x_{l_1+j_1} x_{l_2} \sum S \] evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{l_1} = 0 \),
2. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{l_2} = 0 \) or
3. \( \vec{t} \in FP(S) \) such that \( x_{l_1} x_{l_2} = 1 \).

Hence, the result follows. \( \square \)

It should be noted that detection conditions 3, 4 and 5 of Theorem 7.4.2(a) are just syntactically different from the detection condition 3 of Theorem 7.4.2(b). In fact, they are actually equivalent to each other when \( k_{i_1} = 2 \) (that is \( p_{i_1} \) contains just two literals). The reason is similar to that in the paragraph after Theorem 7.2.5. Hence, without loss of generality, we can still use the five detection conditions in Theorem 7.4.2(a) to represent the detection conditions of Expression (6.19) in Table 6.7 for \( k_{i_1} \geq 2 \), bearing in mind that the detection conditions 3, 4 and 5 degenerate to “\( \vec{t} \in FP(S) \) such that \( x_{l_1} x_{l_2} = 1 \)” when \( k_{i_1} = 2 \).

For ease of reading and understanding, Table 7.2 lists all the double fault classes studied in this chapter and their corresponding fault detection conditions. For example, the third row of Table 7.2 presents the detection conditions of two double-fault expressions of \( \mathcal{LNF} \kappa \mathcal{LIF} \). For double-fault expression (6.5) (please refer to Table 6.7 for the actual double-fault expression), the detection condition shows that any point of \( S \) in \( \left( TP_{i_1}(S) \setminus \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( p_{i_2,x_{l_2}} = 0 \), or \( \left( TP_{i_2}(S) \setminus \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( p_{i_1,j_1} + x_{l_2} = 0 \), or \( NFP_{i_1,j_1}(S) \) can distinguish \( S \) and the double-fault expression (6.5). While for double-fault expression (6.6) in Table 6.7, the detection condition shows that any point of \( S \) in \( UTP_{i_1}(S) \) or \( NFP_{i_1,j_1}(S) \) such that \( x_{l_2} = 1 \) can distinguish \( S \) and the double-fault expression.

137
Table 7.2: Double fault, double-fault expression and detection condition ($S = p_1 + \ldots + p_m$)

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF LNF</td>
<td>(6.1): (C1) any point in (\left{ TP_{i_j}(S) \right} \cup \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (p_{i_2,j_2} = 0), (C2) any point in (\left{ TP_{i_j}(S) \right} \cup \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (p_{1,j_1} = 0), (C3) any point in (NFP_{i_2,j_2}(S)), or (C4) any point in (NFP_{i_1,j_1}(S)).</td>
</tr>
<tr>
<td>LNF LOF</td>
<td>(6.2): (C1) any point in (UTP_i(S)), or (C2) any point in (FP(S)) such that (p_{1,j_1,j_2} = 1).</td>
</tr>
<tr>
<td>LNF LIF</td>
<td>(6.3): (C1) any point in (UTP_i(S)) such that (p_{i_2,j_2} = 0), (C2) any point in (NFP_{i_1,j_1}(S)), or (C3) any point in (NFP_{i_2,j_2}(S)).</td>
</tr>
<tr>
<td>LNF LRF</td>
<td>(6.4): (C1) any point in (UTP_i(S)) or (C2) any point in (FP(S)) such that (p_{1,j_1,j_2} = 1).</td>
</tr>
<tr>
<td>LNF</td>
<td>(6.5): (C1) any point in (UTP_i(S)), (C2) any point in (\left{ TP_{i_j}(S) \right} \cap \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (x_{i_2} = 0), (C3) any point in (\left{ TP_{i_j}(S) \right} \cup \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (p_{i_1,j_1} + x_{i_2} = 0), or (C4) any point in (NFP_{i_1,j_1}(S)).</td>
</tr>
<tr>
<td>LNF LRF</td>
<td>(6.6): (C1) any point in (UTP_i(S)) or (C2) any point in (NFP_{i_1,j_1}(S)) such that (x_{i_2} = 1).</td>
</tr>
<tr>
<td>LNF</td>
<td>(6.7): (C1) any point in (UTP_i(S)) such that (p_{i_2,j_2} = 0), (C2) any point in (\left{ TP_{i_j}(S) \right} \cup \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (x_{i_2} = 0), (C3) any point in (\left{ TP_{i_j}(S) \right} \cup \left( \bigcup_{i=1, i \neq j}^m TP_i(S) \right)) such that (p_{i_1,j_1} + x_{i_2} = 0), or (C4) any point in (NFP_{i_1,j_1}(S)), or (C5) any point in (NFP_{i_2,j_2}(S)) such that (x_{i_2} = 1).</td>
</tr>
<tr>
<td>LNF LRF</td>
<td>(6.8): (C1) any point in (UTP_i(S)), or (C2) any point in (FP(S)) such that (p_{1,j_1,j_2}x_{i_2} = 1).</td>
</tr>
<tr>
<td>LOF LOF</td>
<td>(6.9): (C1) any point in (NFP_{i_1,j_1}(S)), or (C2) any point in (NFP_{i_2,j_2}(S)).</td>
</tr>
<tr>
<td>LOF LIF</td>
<td>(6.10): (C1) any point in (FP(S)) such that (p_{1,j_1,j_2} = 1).</td>
</tr>
<tr>
<td>LOF LIF</td>
<td>(6.11): (C1) any point in (UTP_i(S)) such that (p_{i_1,j_1} + x_{i_2} = 0), or (C2) any point in (NFP_{i_1,j_1}(S)).</td>
</tr>
<tr>
<td>LOF LRF</td>
<td>(6.12): (C1) any point in (UTP_{i_2}(S)) such that (p_{i_1,j_1} + x_{i_2} = 0), (C2) any point in (NFP_{i_1,j_1}(S)), or (C3) any point in (NFP_{i_2,j_2}(S)) such that (x_{i_2} = 1).</td>
</tr>
<tr>
<td>LOF LRF</td>
<td>(6.13): (C1) any point in (UTP_{i_2}(S)) such that (x_{i_2} = 0), (C2) any point in (NFP_{i_1,j_1}(S)) such that (x_{i_2} = 1), (C3) any point in (\in NFP_{i_2,j_2}(S)) such that (x_{i_2} = 1), or (C4) any point in (FP(S)) such that (p_{1,j_1,j_2}x_{i_2} = 1).</td>
</tr>
</tbody>
</table>
Table 7.2 (cont’d) Double fault, double-fault expression and detection condition

\[ (S = p_1 + \ldots + p_m) \]

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
</table>
| LIF ⊕ LIF | (6.14): (C1) any point in \( UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \), or  
(C3) any point in \( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( x_{i_1} + x_{i_2} = 0 \). |
| LIF ⊕ LRF | (6.15): (C1) any point in \( UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \), or  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \). |
| LRF ⊕ LRF | (6.16): (C1) any point in \( UTP_{i_1}(S) \) such that \( p_{i_1,j_2} + x_{i_1} = 0 \),  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \),  
(C3) any point in \( UTP_{i_3}(S) \) such that \( x_{i_3} = 0 \), or  
(C4) any point in \( NFP_{i_3,j_2}(S) \) such that \( x_{i_1} = 1 \), or  
(C5) any point in \( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( x_{i_1} + x_{i_2} = 0 \). |
| LRF ⊕ LRF | (6.17): (C1) any point in \( UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \), or  
(C3) any point in \( NFP_{i_1,j_2}(S) \) such that \( x_{i_1} x_{i_2} = 1 \). |
| LRF ⊕ LRF | (6.18): (C1) any point in \( UTP_{i_1}(S) \) such that \( p_{i_2,j_2} + x_{i_1} = 0 \),  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \),  
(C3) any point in \( UTP_{i_3}(S) \) such that \( p_{i_1,j_1} + x_{i_2} = 0 \),  
(C4) any point in \( UTP_{i_4}(S) \) such that \( x_{i_1} + x_{i_2} = 0 \),  
(C5) any point in \( NFP_{i_2,j_3}(S) \) such that \( x_{i_1} = 1 \),  
(C6) any point in \( NFP_{i_3,j_2}(S) \) such that \( x_{i_2} = 1 \), or  
(C7) any point in \( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2} TP_i(S) \right) \) such that \( x_{i_1} + x_{i_2} = 0 \). |
| LRF ⊕ LRF | (6.19): (C1) any point in \( UTP_{i_1}(S) \) such that \( x_{i_1} = 0 \),  
(C2) any point in \( UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \),  
(C3) any point in \( NFP_{i_1,j_3}(S) \) such that \( x_{i_1} x_{i_2} = 1 \),  
(C4) any point in \( NFP_{i_2,j_2}(S) \) such that \( x_{i_1} x_{i_2} = 1 \), or  
(C5) any point in \( FP(S) \) such that \( p_{i_2,j_1,j_2} = 1 \) and \( x_{i_1} x_{i_2} = 1 \). |
Chapter 8

Double Faults Related to a Term and a Literal

In this chapter, double faults related to term and literal are studied. A double fault related to term and literal is a double fault in which one fault is a term fault and the other one is a literal fault within Boolean expressions. Since the ordering of the occurrences of two single faults in a double fault may result in different faulty expressions, double faults related to term and literal are studied from two cases, double fault with and without ordering. As previously discussed, double faults without ordering refers to the situation that two faults involved in a double fault may be independent of each other while double faults with ordering refers to the situation that two faults occur one after the other in such a way that the first fault may affect the occurrence of the second.

8.1 Double Faults without Ordering

In this section, different types of double fault without ordering related to term and literal are introduced. Similarly, $I_{F_1(E_1 \rightarrow E'_1), F_2(E_2 \rightarrow E'_2)}$ is used to denote the resulting double-fault expression where two single fault classes $F_1$ and $F_2$ are committed in a given $S$ changing its subexpressions $E_1$ and $E_2$ to $E'_1$ and $E'_2$, respectively. Since the order of occurrence of the two faults will result in the same double-fault expression, Table 8.1 lists all 20 types of double faults without ordering for those nine single fault classes discussed in Section 2.2.

8.1.1 ENF with Literal Faults

ENF and LNF  Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1} + \cdots + p_{i_h}$ be a subexpression of $S$ and $x_{j_2}^{i_2}$ be a literal in the $i_2$-th term, $p_{i_2}$, of $S$. Suppose the subexpression $p_{i_1} + \cdots + p_{i_h}$ and the literal $x_{j_2}^{i_2}$ of the term $p_{i_2}$ are negated.
Table 8.1: Types of Double Faults without Ordering

<table>
<thead>
<tr>
<th></th>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LOF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LIF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LRF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

As a reminder, we do not consider the situation of \( p_{i_2} \) containing just one literal because the negation fault is considered as a TNF. Hence, the double fault is then considered as a double term fault (ENF and TNF) which has been considered in Section 4.1.1. Hence, we may assume that \( k_{i_2} > 1 \) where \( k_{i_2} \) is the number of literals of \( p_{i_2} \). Suppose the subexpression \( p_{i_1} + \cdots + p_{h_1} \) and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) are negated. We use \( I_{ENF(p_{i_1} + \cdots + p_{h_1} \rightarrow p_{i_1} + \cdots + p_{h_1}), LNF(p_{i_2} - p_{i_2,j_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \notin \{i_1, \ldots, h_1\} \). There are two possible cases, namely \( i_1 < h_1 < i_2 \) and \( i_2 < i_1 < h_1 \). Without loss of generality, we can assume \( i_1 < h_1 < i_2 \). Otherwise, we can always rearrange the terms in \( S \) so that the LNF comes after the ENF.\(^2\) The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1} + \cdots + p_{h_1}} + p_{i_2,j_2} + \cdots + p_m \quad (8.1)
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1} + \cdots + p_{i_2,j_2} + \cdots + p_{h_1}} + \cdots + p_m \quad (8.2)
\]

where \( \overline{p_{i_1} + \cdots + p_{i_2,j_2} + \cdots + p_{h_1}} \) denotes the subexpression obtained by negating the subexpression \( p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} \) and the literal \( x_{j_2}^{i_2} \) in \( p_{i_2} \).

**ENF and LOF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} + \cdots + p_{h_1} \) be a subexpression of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the subexpression \( p_{i_1} + \cdots + p_{h_1} \) is negated and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is omitted. As a remainder, the \( p_{i_2} \) has more than one literals that is

\(^1\)We will make similar assumption related to LNF and LOF in the rest of the thesis.

\(^2\)In the rest of this thesis, we will make similar assumptions whenever a literal fault does not occur in the subexpression \( p_{i_1} + \cdots + p_{h_1} \) where ENF is committed.
\( k_{i_2} > 1 \). Otherwise, the omission of the only literal in \( p_{i_2} \) is considered as a TOF and the resulting implementation is then equivalent to a \( \text{ENF} \times \text{TOF} \) which has been discussed in Section 4.1.1. ³ We use \( I_{\text{ENF}(p_{i_1} + \cdots + p_{h_1} = p_{i_2} + \cdots + p_{h_1})}, \text{LOF}(p_{i_2} = p_{i_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \not\in \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( i_1 < h_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{h_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \tag{8.3}
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2,j_2}} + \cdots + p_{h_1} + \cdots + p_m \tag{8.4}
\]

where \( \overline{p_{i_1}} + \cdots + \overline{p_{i_2,j_2}} + \cdots + p_{h_1} \) denotes the subexpression obtained by negating the subexpression \( p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} \) and omitting the literal \( x_{i_2}^{i_2} \) in \( p_{i_2} \).

**ENF and LIF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} + \cdots + p_{h_1} \) be a subexpression of \( S \) and \( x_{i_2} \) be a missing literal of the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the subexpression \( p_{i_1} + \cdots + p_{h_1} \) is negated and the literal \( x_{i_2} \) is inserted into the term \( p_{i_2} \). We use \( I_{\text{ENF}(p_{i_1} + \cdots + p_{h_1} = p_{i_2} + \cdots + p_{h_1})}, \text{LIF}(p_{i_2} = p_{i_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \not\in \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( i_1 < h_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{h_1} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_m \tag{8.5}
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_{h_1} + \cdots + p_m \tag{8.6}
\]

³In the rest of the thesis, we will make similar assumptions when LOF occurs with other term faults.
where \( p_{i_1} + \cdots + p_{i_2} x_{i_2} + \cdots + p_{h_1} \) denotes the subexpression obtained by negating the subexpression \( p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} \) and inserting the literal \( x_{i_2} \) into \( p_{i_2} \).

**ENF and LRF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} + \cdots + p_{h_1} \) be a subexpression of \( S \), \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) and \( x_{i_2} \) be a missing literal of \( p_{i_2} \). Suppose the subexpression \( p_{i_1} + \cdots + p_{h_1} \) is negated and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is replaced by the literal \( x_{i_2} \). We use \( I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1}, LNF(p_{i_2} - p_{j_2}^{i_2} x_{i_2})) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \not\in \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( i_1 < h_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{h_1}} + \cdots + p_{i_2 j_2} x_{i_2} + \cdots + p_m \tag{8.7}
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( i_2 = i_1 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p_{i_1}} + \cdots + \overline{p_{i_2 j_2}} x_{i_2} + \cdots + \overline{p_{h_1}} + \cdots + p_m \tag{8.8}
\]

where \( \overline{p_{i_1}} + \cdots + \overline{p_{i_2 j_2}} + \cdots + \overline{p_{h_1}} \) denotes the subexpression obtained by negating the subexpression \( p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} \) and replacing the literal \( x_{j_2}^{i_2} \) in \( p_{i_2} \) by \( x_{i_2} \).

**8.1.2 TNF with Literal Faults**

**TNF and LNF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) are negated. We use \( I_{\text{TNF}}(p_{i_1} - p_{i_1}, LNF(p_{i_2} - p_{j_2}^{i_2})) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). Otherwise, we can always interchange the two terms so that the term with LNF comes after that with TNF.\(^4\) The implementation

\(^4\)In the rest of this thesis, we will make similar assumptions whenever we encounter situations with a term fault and a literal fault committed at two different terms.
is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2, j_2} + \cdots + p_m \]  \hspace{1cm} (8.9)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1, j_2} + \cdots + p_m \]  \hspace{1cm} (8.10)

where \( \bar{p}_{i_1, j_2} = \overline{x_{i_1}^{j_1} \cdots x_{j_2}^{i_1} \cdots x_{k_1}^{i_1}} \) denotes the subexpression obtained by negating both the term \( p_{i_1} \) and its literal \( x_{i_1}^{j_2} \).

**TNF and LOF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{i_2}^{j_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) is wrongly negated and the literal \( x_{i_2}^{j_2} \) of the term \( p_{i_2} \) is omitted. We use \( I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}), LOF(p_{i_2} \rightarrow p_{i_2, j_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2, j_2} + \cdots + p_m \]  \hspace{1cm} (8.11)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1, j_2} + \cdots + p_m \]  \hspace{1cm} (8.12)

where \( \bar{p}_{i_1, j_2} = \overline{x_{i_1}^{j_1} \cdots x_{j_2}^{i_1} \cdots x_{k_1}^{i_1}} \) denotes the subexpression obtained by negating the term \( p_{i_1} \) and omitting its literal \( x_{i_2}^{j_2} \).

**TNF and LIF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{i_2}^{j_2} \) be a missing literal of the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) is wrongly negated and the literal \( x_{i_2}^{j_2} \) is inserted into the term \( p_{i_2} \). We use \( I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}), LIF(p_{i_2} \rightarrow p_{i_2, j_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the
following expression

\[ p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2}x_{l_2} + \cdots + p_m \]  

(8.13)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1}x_{l_2} + \cdots + p_m \]  

(8.14)

**TNF and LRF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \), \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) and \( x_{l_2} \) be a missing literal of \( p_{i_2} \). Suppose the term \( p_{i_1} \) is wrongly negated and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is replaced by the literal \( x_{l_2} \). We use \( I_{\text{TNF}(p_{i_1} \rightarrow \bar{p}_{i_1}), \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2}^{i_2}x_{l_2})} \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2,j_2}x_{l_2} + \cdots + p_m \]  

(8.15)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \bar{p}_{i_1,j_2}x_{l_2} + \cdots + p_m \]  

(8.16)

where \( \bar{p}_{i_1,j_2}x_{l_2} = x_{1}^{i_1} \cdots x_{j_2-1}^{i_1}x_{j_2+1}^{i_1} \cdots x_{k_1}^{i_1}x_{l_2} \) denotes the subexpression obtained by negating the term \( p_{i_1} \) and replacing its literal \( x_{j_2}^{i_1} \) with \( x_{l_2} \).

**8.1.3 TOF with Literal Faults**

**TOF and LNF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) is omitted and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is negated. We use \( I_{\text{TOF}(p_{i_1} \rightarrow \bar{p}_{i_1}), \text{LNF}(p_{i_2} \rightarrow \bar{p}_{i_2,j_2}x_{j_2}^{i_2})} \) to denote the corresponding faulty implementation. We do not consider the situation when TOF and LNF are committed at the same term (that is, \( i_1 = i_2 \)). It is because when \( p_{i_1} \) is omitted, it is impossible to negate its literal afterwards. On the other hand, if the literal \( x_{j_2}^{i_1} \) is first negated and then the \( i_1 \)-th term is omitted, the implementation is then equivalent to a single TOF with respect to the original expression \( S \) and, hence, is not a double-fault expression.
As a result, the terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  

(8.17)

**TOF and LOF**  Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) is omitted and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is omitted. We use \( I_{TOF(p_{i_1} \rightarrow \cdot),LOF(p_{i_2} \rightarrow p_{i_2,j_2})} \) to denote the corresponding faulty implementation. We do not consider the situation when TOF and LOF are committed at the same term (that is, \( i_1 = i_2 \)). It is because when \( p_{i_1} \) is omitted, it is impossible to omit its literal afterwards. On the other hand, if the literal \( x_{j_2}^{i_1} \) is first omitted and then the \( i_1 \)-th term is omitted, the implementation is then equivalent to a single TOF with respect to the original expression \( S \) and, hence, is not a double-fault expression. As a result, the terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  

(8.18)

**TOF and LIF**  Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2} \) be a missing literal of the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) is omitted and the literal \( x_{j_2} \) is inserted into the term \( p_{i_2} \). We use \( I_{TOF(p_{i_1} \rightarrow \cdot),LIF(p_{i_2} \rightarrow p_{i_2,j_2})} \) to denote the corresponding faulty implementation. We do not consider the situation when TOF and LIF are committed at the same term (that is, \( i_1 = i_2 \)). It is because when \( p_{i_1} \) is omitted, it is impossible to insert a literal in it. On the other hand, if the literal \( x_{j_2} \) is first inserted into the \( i_1 \)-th term, and then the \( i_1 \)-th term is omitted, the implementation is then equivalent to a single TOF with respect to the original expression \( S \) and, hence, is not a double-fault expression. As a result, the terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2}x_{j_2} + \cdots + p_m
\]  

(8.19)

**TOF and LRF**  Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \), \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) and \( x_{j_2} \) be a missing literal of \( p_{i_2} \). Suppose the term \( p_{i_1} \) is omitted and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is replaced by the literal \( x_{j_2} \). We use \( I_{TOF(p_{i_1} \rightarrow \cdot),LRF(p_{i_2} \rightarrow p_{i_2,j_2}x_{j_2})} \) to denote the corresponding faulty implementation. We do not consider the situation when TOF
and LRF are committed at the same term (that is, \( i_1 = i_2 \)). It is because when \( p_{i_1} \) is omitted, it is impossible to replace its literal afterwards. On the other hand, if the literal \( x_{j_2}^{i_2} \) is first replaced by \( x_{i_2} \) and then the \( i_1 \)-th is omitted, the implementation is then equivalent to a single TOF with respect to the original expression \( S \) and, hence, is not a double-fault expression. As a result, the terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,i_2} x_{i_2} + \cdots + p_m
\]  
(8.20)

### 8.1.4 DORF with Literal Faults

**DORF and LNF**  
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) and \( p_{i_1+1} \) be the \( i_1 \)-th and \( (i_1+1) \)-th terms of \( S \), and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the subexpression \( p_{i_1} + p_{i_1+1} \) of \( S \) is implemented as \( p_{i_1} p_{i_1+1} \) and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is negated. We use \( I_{DORF}(p_{i_1} + p_{i_1+1} \rightarrow p_{i_1} p_{i_1+1}, LNF(p_{i_2} \rightarrow p_{i_2,j_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The subexpression \( p_{i_1} + p_{i_1+1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \not\in \{i_1, i_1 + 1\} \). Without loss of generality, we can assume \( i_1 + 1 < i_2 \). Otherwise, we can always rearrange the terms in \( S \) so that the term with LNF comes after that with DORF.\(^5\) The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1} p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  
(8.21)

**Case 2.** The subexpression \( p_{i_1} + p_{i_1+1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, i_1 + 1\} \). Hence, there are two possible cases, namely \( i_1 = i_2 \) and \( i_1 + 1 = i_2 \). Without loss of generality, we can assume \( i_1 = i_2 \). Otherwise, we can always interchange the \( i_1 \)-th and \( (i_1+1) \)-th terms so that the term with LNF occurs at the first term of the subexpression where DORF is committed.\(^6\) The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_2} p_{i_1+1} + \cdots + p_m
\]  
(8.22)

**DORF and LOF**  
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) and \( p_{i_1+1} \) be the \( i_1 \)-th and \( (i_1+1) \)-th terms of \( S \), and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term,

\(^5\)In the rest of this thesis, we will make similar assumption whenever a literal fault does not occur in the subexpression \( p_{i_1} + p_{i_1+1} \) where DORF is committed.

\(^6\)In the rest of this thesis, we will make similar assumption when a literal fault occurs in the subexpression \( p_{i_1} + p_{i_1+1} \) where DORF is committed.
Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$, that is $i_2 \notin \{i_1, i_1 + 1\}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_1p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (8.23)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_2}p_{i_1+1} + \cdots + p_m \quad (8.24)$$

**DORF and LIF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1}$ and $p_{i_1+1}$ be the $i_1$-th and $(i_1 + 1)$-th terms of $S$ and $x_{i_2}$ be a missing literal of the $i_2$-th term, $p_{i_2}$ of $S$. Suppose the subexpression $p_{i_1} + p_{i_1+1}$ of $S$ is implemented as $p_{i_1}p_{i_1+1}$ and the literal $x_{i_2}$ of the term $p_{i_2}$ is omitted. We use $I_{DORF}(p_{i_1} + p_{i_1+1} - p_1p_{i_1+1}, LOF(p_{i_2} - p_{i_2,j_2}))$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$, that is $i_2 \notin \{i_1, i_1 + 1\}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_2}p_{i_1+1} + \cdots + p_m \quad (8.25)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,x_{i_2}}p_{i_1+1} + \cdots + p_m \quad (8.26)$$

**DORF and LRF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1}$ and $p_{i_1+1}$ be the $i_1$-th and $(i_1 + 1)$-th terms of $S$, $x_{i_2}^{\hat{j}_2}$ be a literal in the $i_2$-th term, $p_{i_2}$, of $S$ and $x_{j_2}$ be a missing literal of $p_{i_2}$. Suppose the subexpression
p_{i_1} + p_{i_1+1} \text{ of } S \text{ is implemented as } p_{i_1}p_{i_1+1} \text{ and the literal } x_{j_2}^{i_2} \text{ of the term } p_{i_2} \text{ is replaced by the literal } x_{i_2}.

We use \( I_{DORF}(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1}, LRF(p_{i_2}-p_{i_2,j_2}x_{i_2})) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The subexpression \( p_{i_1} + p_{i_1+1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \notin \{ i_1, i_1 + 1 \} \). Without loss of generality, we can assume \( i_1 + 1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_m
\]  

(8.27)

Case 2. The subexpression \( p_{i_1} + p_{i_1+1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{ i_1, i_1 + 1 \} \). Hence, there are two possible cases, namely \( i_1 = i_2 \) and \( i_1 + 1 = i_2 \). Without loss of generality, we can assume \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1}x_{i_2} p_{i_1+1} + \cdots + p_m
\]  

(8.28)

**CORF with Literal Faults**

**CORF and LNF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} (= p_{i_1,1,j_1} \cdot p_{i_1,j_1+1,1,k_1}) \) is implemented as \( p_{i_1,1,j_1} + p_{i_1,j_1+1,1,k_1} \) where \( p_{i_1,1,j_1} = x_{1}^{i_1} \cdots x_{j_1}^{i_1} \) and \( p_{i_1,j_1+1,1,k_1} = x_{j_1+1}^{i_1} \cdots x_{k_1}^{i_1} \) denote the terms obtained from \( p_{i_1} \) by keeping its first \( j_1 \) literals and its last \( (k_1 - j_1) \) literals, respectively.\(^7\) Suppose further that the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is negated. We use \( I_{CORF}(p_{i_1}+p_{i_1,1,j_1}+p_{i_1,j_1+1,1,k_1}, LNF(p_{i_2}-p_{i_2,j_2})) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

Case 1. The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,1,k_1} + \cdots + p_{i_2,j_2} + \cdots + p_m
\]  

(8.29)

Case 2. The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). When CORF and LNF are committed at the same term, there are two possible cases, namely, the LNF is committed at a literal in \( p_{i_1,1,j_1} \) or \( p_{i_1,j_1+1,1,k_1} \). Without loss of generality, we can assume that LNF is committed at a

\(^7\)In the rest of this thesis, notations similar to \( p_{i_1,1,j_1} \) and \( p_{i_1,j_1+1,1,k_1} \) should be interpreted similarly.
literal in \( p_{i_1,j_1} \). Otherwise, we can always rearrange the literals in \( p_{i_1,j_1} \) and \( p_{i_1,j_1+1,k_1} \) (or, literals in \( p_{i_1,j_1} \)) so that LNF can occur at a particular literal in \( p_{i_1,j_1} \). Furthermore, without loss of generality, we can assume \( j_1 = j_2 \), that is the literal being negated in \( p_{i_1,j_1} \) is its last literal. Otherwise, we can always rearrange the literals in \( p_{i_1,j_1} \) so that the literal being negated is the last literal of \( p_{i_1,j_1} \). As a reminder, we may assume that \( j_1 > 1 \). Otherwise, \( p_{i_1,j_1} \) only contains 1 literal and the fault of negating such literal is considered as a TNF. As a result, the double fault is considered as a CORF and TNF together, which has been considered in Section 4.2.5. The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m \tag{8.30}
\]

where \( p_{i_1,j_1} = x_{i_1}^{j_1} \cdots x_{j_1}^{i_1} \) denotes the subexpression obtained by negating the literal \( x_{j_1}^{i_1} \) in \( p_{i_1,j_1} \).

**CORF and LOF** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} \) be the \( i_1 \)-th term of \( S \) and \( x_{j_2}^{i_2} \) be a literal in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \). Suppose the term \( p_{i_1} \) (\( = p_{i_1,j_1} : p_{i_1,j_1+1,k_1} \)) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_1} \) and the literal \( x_{j_2}^{i_2} \) of the term \( p_{i_2} \) is omitted. We use \( I_{\text{CORF}}(p_1 \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \cdot I_{\text{LOF}}(p_2 \rightarrow p_{i_2,j_2}) \) to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The terms \( p_{i_1} \) and \( p_{i_2} \) are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2,j_2} + \cdots + p_m \tag{8.31}
\]

**Case 2.** The two terms \( p_{i_1} \) and \( p_{i_2} \) are exactly the same, that is \( i_1 = i_2 \). Similar to the situation of Case 2 of CORF and LNF, we can assume, without loss of generality, that LOF is committed at \( x_{j_1}^{i_1} \) in \( p_{i_1,j_1} \) and \( j_1 > 1 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1-1} + p_{i_1,j_1+1,k_1} + \cdots + p_m \tag{8.32}
\]

where \( p_{i_1,j_1-1} = x_{i_1}^{j_1} \cdots x_{j_1-1}^{i_1} \) denotes the term obtained from \( p_{i_1,j_1} \) by omitting its literal \( x_{j_1}^{i_1} \).

---

8 In the rest of this thesis, we will make similar assumptions whenever a literal fault occurs at the same term where the CORF occurs.

9 Again, in the rest of the thesis, we will make similar assumptions related to LNF and LOF.
**CORF and LIF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1}$ be the $i_1$-th term of $S$ and $x_{i_2}$ be a missing literal of $p_{i_2}$. Suppose the term $p_{i_1} (= p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_{i_1}})$ is implemented as $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and the literal $x_{i_2}$ is inserted into the term $p_{i_2}$. We use $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}), I_{\text{LIF}}(p_{i_2} \rightarrow x_{i_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_m \quad (8.33)$$

**Case 2.** The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. When CORF and LIF are committed at the same term, there are two possible cases, namely, the LIF is committed at either $p_{i_1,j_1}$ or $p_{i_1,j_1+1,k_{i_1}}$. Without loss of generality, we can assume that LIF is committed at $p_{i_1,j_1}$. Otherwise, we can always rearrange the literals in $p_{i_1,j_1}$ and $p_{i_1,j_1+1,k_{i_1}}$ so that LIF occurs at $p_{i_1,j_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1}x_{i_2} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m \quad (8.34)$$

**CORF and LRF** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1}$ be the $i_1$-th term of $S$, $x_{j_2}^{i_2}$ be a literal in the $i_2$-th term, $p_{i_2}$, of $S$ and $x_{i_2}$ be a missing literal of $p_{i_2}$. Suppose the term $p_{i_1} (= p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_{i_1}})$ is implemented as $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and the literal $x_{j_2}^{i_2}$ of the term $p_{i_2}$ is replaced by the literal $x_{i_2}$. We use $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}), I_{\text{LRF}}(p_{i_2} \rightarrow x_{j_2}^{i_2})$ to denote the corresponding faulty implementation, which can be further classified into the following two cases:

**Case 1.** The terms $p_{i_1}$ and $p_{i_2}$ are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_m \quad (8.35)$$

**Case 2.** The two terms $p_{i_1}$ and $p_{i_2}$ are exactly the same, that is $i_1 = i_2$. Without loss of generality, we can assume that LRF is committed at $x_{j_2}^{i_1}$ in $p_{i_1,j_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1-1}x_{i_2} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m \quad (8.36)$$
8.2 Double Faults with Ordering

In this section, different types of double faults with ordering related to term and literal and their corresponding double-fault expressions are discussed. As previously stated, double faults with ordering is such that two single faults occur one after the other in such a way that the occurrence of the first fault may affect the occurrence of the second fault.

As discussed before, \( I_{F_1(E_1 \rightarrow E_1') \otimes F_2(E_2 \rightarrow E_2')} \) is used to denote the resulting double-fault expression of double faults with ordering where \( F_1 \) and \( F_2 \) are two single fault classes committed in a given Boolean expression \( S \), changing the subexpressions \( E_1 \) and \( E_2 \) in \( S \) to \( E_1' \) and \( E_2' \), respectively and \( F_1 \) is committed before \( F_2 \). Table 8.2 lists all 40 different types of double faults with ordering studied in this report for those nine single fault classes discussed in Section 2.2. We now introduce these 40 double fault classes and their corresponding double-fault expressions.

Table 8.2: Types of Double Faults with Ordering Studied in This Section

<table>
<thead>
<tr>
<th>ENF</th>
<th>TNF</th>
<th>TOF</th>
<th>DORF</th>
<th>CORF</th>
<th>LNF</th>
<th>LOF</th>
<th>LIF</th>
<th>LRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

8.2.1 ENF First, then other Literal Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Let \( p_{i_1} + \cdots + p_{h_1} \) be a subexpression of \( S \). Suppose an ENF is committed first by negating the subexpression \( p_{i_1} + \cdots + p_{h_1} \). The corresponding faulty implementation is \( I_{ENF(p_{i_1} + \cdots + p_{h_1} \rightarrow \overline{p_{i_1} + \cdots + p_{h_1}})} = p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_m \).\(^{10}\)

**ENF and LNF** After the ENF is made on \( S \), the literal \( x_{j_2}^{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) is then negated. Let \( I_{ENF(p_{i_1} + \cdots + p_{h_1} \rightarrow \overline{p_{i_1} + \cdots + p_{h_1}}) \otimes LNF(p_{i_2} \rightarrow \overline{p_{i_2}})} \) denote the corresponding faulty implementation. We have the following two cases:

\(^{10}\)For simplicity, in the rest of this report, we will use the following short form “The corresponding faulty implementation is \( I_{ENF(p_{i_1} + \cdots + p_{h_1} \rightarrow \overline{p_{i_1} + \cdots + p_{h_1}})} = p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_m \)” instead.
Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $i_1 < h_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2 j_2} + \cdots + p_m \quad (8.37)$$

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{i_2 j_2} + \cdots + p_{h_1} + \cdots + p_m \quad (8.38)$$

**ENF and LOF** After the ENF is made on $S$, the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then omitted. Let $I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{h_1}) \otimes I_{\text{LOF}}(p_{i_2} - p_{i_2 j_2})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $i_1 < h_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2 j_2} + \cdots + p_m \quad (8.39)$$

Case 2. The subexpression $p_{i_1} + \cdots + p_{h_1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, \ldots, h_1\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{i_2 j_2} + \cdots + p_{h_1} + \cdots + p_m \quad (8.40)$$

**ENF and LIF** After the ENF is made on $S$, a literal $x_{l_2}$ of $S$ is inserted into the $i_2$-th term, $p_{i_2}$, of $S$. Let $I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{h_1}) \otimes I_{\text{LIF}}(p_{i_2} - p_{i_2 l_2})$ denote the corresponding faulty implementation. We have the following three cases:

Case 1. The subexpression $p_{i_1} + \cdots + p_{h_1}$ does not contain the term $p_{i_2}$, that is $i_2 \not\in \{i_1, \ldots, h_1\}$. Without loss of generality, we can assume $i_1 < h_1 < i_2$. It is noted that the inserted literal $x_{l_2}$ is a missing literal of the $p_{i_2}$ because of the following two reasons. First, if $x_{l_2}$ is a literal of $p_{i_2}$, the implementation is equivalent to a single ENF with respect to $S$ and, hence, is not a double-fault expression. Second, if $\bar{x}_{l_2}$ is a literal of $p_{i_2}$, the implementation is equivalent to a double fault ENF and TOF with respect to $S$. Such a fault has been considered in [27]. As a result, neither $x_{l_2}$ nor its negation occur in $p_{i_2}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2 x_{l_2}} + \cdots + p_m \quad (8.41)$$
Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). It is noted that the inserted literal \( x_{i_2} \) is a missing literal of the \( p_{i_2} \) because of the following two reasons. First, if \( x_{i_2} \) is a literal of \( p_{i_2} \), the implementation is equivalent to a single ENF with respect to \( S \) and, hence, is not a double-fault expression. Second, if \( \overline{x}_{i_2} \) is a literal of \( p_{i_2} \), the implementation is equivalent to a double fault ENF and LNF with respect to \( S \). Such fault has been considered previously. As a result, neither \( x_{l_2} \) nor its negation occur in \( p_{i_2} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{i_2} \cdot x_{i_2} + \cdots + p_{h_1} + \cdots + p_m \tag{8.42}
\]

Case 3. The term \( p_{i_2} \) is the newly created term \( \overline{p}_{i_1} + \cdots + p_{h_1} \). The literal \( x_{l_2} \) is inserted into the newly created term. The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{h_1} x_{l_2} + \cdots + p_m \tag{8.43}
\]

**ENF and LRF**

After the ENF is committed on \( S \), the literal \( x_{l_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) is then replaced by the literal \( x_{l_2} \) where \( x_{l_2} \) is a missing literal of \( p_{i_2} \). Let \( I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} \rightarrow \overline{p}_{i_1} + \cdots + p_{h_1}) \odot I_{\text{LRF}}(p_{i_2} \rightarrow \overline{p}_{i_2} x_{l_2}) \) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) does not contain the term \( p_{i_2} \), that is \( i_2 \notin \{i_1, \ldots, h_1\} \). Without loss of generality, we can assume \( i_1 < h_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{h_1} + \cdots + p_{i_2} x_{l_2} + \cdots + p_m \tag{8.44}
\]

Case 2. The subexpression \( p_{i_1} + \cdots + p_{h_1} \) contains the term \( p_{i_2} \), that is \( i_2 \in \{i_1, \ldots, h_1\} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{i_2} x_{l_2} + \cdots + p_{h_1} + \cdots + p_m \tag{8.45}
\]

### 8.2.2 TNF First, then other Literal Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose a TNF is committed first by negating the \( i_1 \)-th term, \( p_{i_1} \), of \( S \). The corresponding faulty implementation is \( I_{\text{TNF}}(p_{i_1} \rightarrow \overline{p}_{i_1}) = p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_m \).
TNF and LNF  After the TNF is made on $S$, the literal $x_{i_2}^{j_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then negated. Let $I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \otimes LN(p_{i_2} \rightarrow \overline{p}_{i_2,j_2})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{i_2,j_2} + \cdots + p_m$$

(8.46)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p}_{i_1,j_2} + \cdots + p_m$$

(8.47)

TNF and LOF  After the TNF is made on $S$, the literal $x_{i_2}^{j_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then omitted. Let $I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \otimes LOF(p_{i_2} \rightarrow p_{i_2,j_2})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p}_{i_1} + \cdots + p_{i_2,j_2} + \cdots + p_m$$

(8.48)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p}_{i_1,j_2} + \cdots + p_m$$

(8.49)

TNF and LIF  After the TNF is made on $S$, the literal $x_{i_2}$ of $S$ is inserted into the term $p_{i_2}$. Let $I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \otimes LIF(p_{i_2} \rightarrow p_{i_2,x_{i_2}})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. It is noted that the inserted literal $x_{i_2}$ is a missing literal of the $p_{i_2}$ because of the following two reasons. First, if $x_{i_2}$ is a literal of $p_{i_2}$, the implementation is equivalent to a single TNF with respect to $S$. Second, if $\overline{x}_{i_2}$ is a literal of $p_{i_2}$, the implementation is equivalent to a double fault TNF and TOF with respect to $S$. Such a fault has been considered in [27]. As a result, neither $x_{i_2}$ nor its negation occur
in \( p_{i_2} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2} x_{i_2} + \cdots + p_m \quad (8.50)
\]

Case 2. The \( i_1 \)-th and \( i_2 \)-th terms are exactly same, that is \( i_1 = i_2 \). It is noted that the inserted literal \( x_{i_2} \) is a missing literal of the \( p_{i_1} \) because of the following two reasons. First, if \( x_{i_2} \) is a literal of \( p_{i_1} \), the implementation is equivalent to a single TNF with respect to \( S \) and, hence, is not a double-fault expression. Second, if \( \bar{x}_{i_2} \) is a literal of \( p_{i_1} \), the implementation is equivalent to 1 because \( \bar{p}_{i_1} \cdot x_{i_2} = \bar{0} = 1 \). In that case, any false point of \( S \) will be able to detect it. As a result, neither \( x_{i_2} \) nor its negation occur in \( p_{i_1} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1} \cdot x_{i_2} + \cdots + p_m \quad (8.51)
\]

Case 3. The \( p_{i_2} \) is the newly created term \( \bar{p}_{i_1} \). The literal \( x_{i_2} \) is inserted into the newly created term. The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1} x_{i_2} + \cdots + p_m \quad (8.52)
\]

**TNF and LRF** After the TNF is made on \( S \), the literal \( x_{i_2}^{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), of \( S \) is then replaced by the literal \( x_{i_2} \) where \( x_{i_2} \) is a missing literal of \( p_{i_2} \). Let \( I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LRF(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2})} \) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The \( i_1 \)-th and \( i_2 \)-th terms are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1} + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_m \quad (8.53)
\]

Case 2. The \( i_1 \)-th and \( i_2 \)-th terms are the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1,j_2} x_{i_2} + \cdots + p_m \quad (8.54)
\]

### 8.2.3 TOF First, then other Literal Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose a TOF is committed first by omitting the \( i_1 \)-th term, \( p_{i_1} \), of \( S \). The corresponding faulty implementation is \( I_{TOF(p_{i_1} \rightarrow)} = p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_m \).
**TOF and LNF** After the TOF is committed on $S$, the literal $x_{j_2}^{i_2}$ of the $i_2$-th term, $p_{i_2}$ of $S$ is then negated. Let $I_{TOF(p_{i_1} \rightarrow ) \otimes LNF(p_{i_2} \rightarrow p_{j_2}^{i_2})}$ denote the corresponding faulty implementation. We do not consider the situation when TOF and LNF are committed at the same term because it is impossible to negate a literal in an omitted term. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (8.55)$$

**TOF and LOF** After the TOF is committed on $S$, the literal $x_{j_2}^{i_2}$ of the $i_2$-th term, $p_{i_2}$ of $S$ is omitted. Let $I_{TOF(p_{i_1} \rightarrow ) \otimes LOF(p_{i_2} \rightarrow p_{j_2}^{i_2})}$ denote the corresponding faulty implementation. We do not consider the situation when TOF and LOF are committed at the same term because it is impossible to omit a literal in an omitted term. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,j_2} + \cdots + p_m \quad (8.56)$$

**TOF and LIF** Suppose that $x_{i_2}$ is a missing literal of $p_{i_2}$. After the TOF is committed on $S$, the literal $x_{i_2}$ is inserted into the term $p_{i_2}$. Let $I_{TOF(p_{i_1} \rightarrow ) \otimes LIF(p_{i_2} \rightarrow p_{i_2} x_{i_2})}$ denote the corresponding faulty implementation. We do not consider the situation when TOF and LIF are committed at the same term because it is impossible to insert a literal into an omitted term. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2, i_2} + \cdots + p_m \quad (8.57)$$

**TOF and LRF** Suppose that $x_{i_2}$ is a missing literal of $p_{i_2}$. After the TOF is committed on $S$, the literal $x_{j_2}^{i_2}$ of the $i_2$-th term, $p_{i_2}$ of $S$ is then replaced by the literal $x_{i_2}$. Let $I_{TOF(p_{i_1} \rightarrow ) \otimes LRF(p_{i_2} \rightarrow p_{i_2} x_{i_2})}$ denote the corresponding faulty implementation. We do not consider the situation when TOF and LRF are committed at the same term because it is impossible to replace a literal in an omitted term with another literal. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_m \quad (8.58)$$

### 8.2.4 DORF First, then other Literal Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Let $p_{i_1}$ and $p_{i_1+1}$ be the $i_1$-th and the $(i_1+1)$-th terms of $S$, respectively. Suppose a DORF is committed first by
concatenating the two terms $p_{i_1}$ and $p_{i_1+1}$ using the `·' operator. The corresponding faulty implementation is $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) = p_1 + \cdots + p_{i_1} p_{i_1+1} + \cdots + p_m$.

**DORF and LNF** After the DORF is made on $S$, the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then negated. Let $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) \otimes LNF(p_{i_2} - p_{i_2} j_2)$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} p_{i_1+1} + \cdots + p_{i_2} j_2 + \cdots + p_m \quad (8.59)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} j_2 p_{i_1+1} + \cdots + p_m \quad (8.60)$$

**DORF and LOF** After the DORF is made on $S$, the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then omitted. Let $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) \otimes \overline{LOF}(p_{i_2} - p_{i_2} j_2)$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_1} + p_{i_1+1}$ does not contain the term $p_{i_2}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} p_{i_1+1} + \cdots + p_{i_2} j_2 + \cdots + p_m \quad (8.61)$$

Case 2. The subexpression $p_{i_1} + p_{i_1+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} j_2 p_{i_1+1} + \cdots + p_m \quad (8.62)$$

**DORF and LIF** After the DORF is made on $S$, the literal $x_{l_2}$ is then inserted in the $i_2$-th term, $p_{i_2}$ of $S$ where $x_{l_2}$ is a missing literal of $p_{i_2}$. Let $I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) \otimes LIF(p_{i_2} - p_{i_2} x_{l_2})$ denote the corresponding faulty implementation. We have the following two cases:
Case 1. The subexpression $p_i + p_{i+1}$ does not contain the term $p_{i_2}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2}x_{i_2} + \cdots + p_m \quad (8.63)$$

Case 2. The subexpression $p_i + p_{i+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}x_{i_2}p_{i_1+1} + \cdots + p_m \quad (8.64)$$

**DORF and LRF**  After the DORF is made on $S$, the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then replaced by the literal $x_{i_2}$ where $x_{i_2}$ is a missing literal of $p_{i_2}$. Let $I_{DORF}(p_1 + p_{i_1+1} - p_{i_1}p_{i_1+1}) \otimes LRF(p_{i_2} - p_{i_2,x_{j_2}^{i_2}})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_i + p_{i+1}$ does not contain the term $p_{i_2}$. Without loss of generality, we can assume $i_1 + 1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1}p_{i_1+1} + \cdots + p_{i_2,j_2}x_{i_2} + \cdots + p_m \quad (8.65)$$

Case 2. The subexpression $p_i + p_{i+1}$ contains the term $p_{i_2}$, that is $i_2 \in \{i_1, i_1 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 + 1 = i_2$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_2}x_{i_2}p_{i_1+1} + \cdots + p_m \quad (8.66)$$

### 8.2.5 CORF First, then other Literal Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose the term $p_{i_1}$ ($= p_{i_1,1,j_1}p_{i_1,j_1+1,k_1}$) is implemented as $p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}$ where $p_{i_1,1,j_1} = x^{i_1}_1 \cdots x^{i_1}_{j_1}$ and $p_{i_1,j_1+1,k_1} = x^{i_1}_{j_1+1} \cdots x^{i_1}_{k_1}$ denote the terms obtained from $p_{i_1}$ by keeping its first $j_1$ literals and its last $(k_1 - j_1)$ literals, respectively.\(^{11}\) The corresponding faulty implementation is $I_{CORF}(p_1 + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) = p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m$.

\(^{11}\)In the rest of this report, notations similar to $p_{i_1,1,j_1}$ and $p_{i_1,j_1+1,k_1}$ should be interpreted similarly.
CORF and LNF  After the CORF is committed on $S$, the literal $x_{i_2}^{j_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then negated. Let $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) \odot L\text{NF}(p_{i_2} \rightarrow p_{i_2,j_2})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The term $p_{i_2}$ is neither $p_{i_1,1,j_1}$ nor $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2,j_2} + \cdots + p_m$$  \hspace{1cm} (8.67)

Case 2. The term $p_{i_2}$ is either $p_{i_1,1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume LNF is committed at $x_{j_1}^{i_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m$$  \hspace{1cm} (8.68)

CORF and LOF  After the CORF is committed on $S$, the literal $x_{i_2}^{j_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then omitted. Let $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) \odot L\text{OF}(p_{i_2} \rightarrow p_{i_2,j_2})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The term $p_{i_2}$ is neither $p_{i_1,1,j_1}$ nor $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2,j_2} + \cdots + p_m$$  \hspace{1cm} (8.69)

Case 2. The term $p_{i_2}$ is either $p_{i_1,1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume LOF is committed at $x_{j_1}^{i_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_1} + \cdots + p_m$$  \hspace{1cm} (8.70)

CORF and LIF  After the CORF is committed on $S$, the literal $x_{i_2}$ is then inserted into the term $p_{i_2}$ where $x_{i_2}$ is a missing literal of $p_{i_2}$. Let $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1}) \odot L\text{IF}(p_{i_2} \rightarrow p_{i_2,x_{i_2}})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The term $p_{i_2}$ is neither $p_{i_1,1,j_1}$ nor $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2,x_{i_2}} + \cdots + p_m$$  \hspace{1cm} (8.71)
Case 2. The term $p_{i_2}$ is either $p_{i_1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume that LIF is committed at $p_{i_1,j_1}$. Since $x_{i_2}$ is a missing literal of $p_{i_1,j_1}$, we have the following three subcases:

(a) $x_{i_2}$ is a missing literal of $p_{i_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1} + \cdots + p_m \quad (8.72)$$

(b) $x_{i_2}$ is a literal in $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume the $x_{i_2} = x_{j_1+1}^{i_1}$. Otherwise, we can always rearrange the literals of $p_{i_1,j_1+1,k_1}$ so that $x_{i_2}$ is the first literal of $p_{i_1,j_1+1,k_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} x_{j_1+1}^{i_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m \quad (8.73)$$

(c) $x_{i_2}$ is the negation of a literal in $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume the $x_{i_2} = \bar{x}_{j_1+1}^{i_1}$. Otherwise, we can always rearrange the literals of $p_{i_1,j_1+1,k_1}$ so that $x_{i_2}$ is the negation of the first literal of $p_{i_1,j_1+1,k_1}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} \bar{x}_{j_1+1}^{i_1} + p_{i_1,j_1+1,k_1} + \cdots + p_m \quad (8.74)$$

**CORF and LRF** After the CORF is committed on $S$, the literal $x_{j_2}^{i_2}$ in the $i_2$-th term, $p_{i_2}$, of $S$ is then replaced by the literal $x_{i_2}$ where $x_{i_2}$ is a missing literal of $p_{i_2}$. Let $ICORF(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \otimes LRF(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The term $p_{i_2}$ is neither $p_{i_1,j_1}$ nor $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_m \quad (8.75)$$

Case 2. The term $p_{i_2}$ is either $p_{i_1,j_1}$ or $p_{i_1,j_1+1,k_1}$. Without loss of generality, we can assume that LRF is committed at $x_{j_1}^{i_1}$. Since $x_{i_2}$ is a missing literal of $p_{i_1,j_1}$, we have the following three subcases:

(a) $x_{i_2}$ is a missing literal of $p_{i_1}$. The implementation is then equivalent
Case 2. The subexpression $I_LNF$ is made on $LNF$ and $TNF$.

After the LNF is made on $LNF$ and $TNF$, and other Term Faults corresponding faulty implementation is committed first by negating the literals so that $x_{i_2}$ is the first literal of $p_i,j_{i_1+1,k_{i_1}}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_{i_1}+1}x_{i_2} + p_{i_1,j_{i_1}+1,k_{i_1}} + \cdots + p_m \quad (8.76)$$

(b) $x_{i_2}$ is a literal in $p_{i_1,j_{i_1}+1,k_{i_1}}$. Without loss of generality, we can assume the $x_{i_2} = x_{i_2}^{i_1}_{j_{i_1}+1}$. Otherwise, we can always rearrange the literals so that $x_{i_2}$ is the first literal of $p_{i_1,j_{i_1}+1,k_{i_1}}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_{i_1}+1}x_{i_2}^{i_1}_{j_{i_1}+1} + p_{i_1,j_{i_1}+1,k_{i_1}} + \cdots + p_m \quad (8.77)$$

(c) $x_{i_2}$ is the negation of a literal in $p_{i_1,j_{i_1}+1,k_{i_1}}$. Without loss of generality, we can assume the $x_{i_2} = \overline{x}_{i_2}^{i_1}_{j_{i_1}+1}$. Otherwise, we can always rearrange the literals so that $x_{i_2}$ is the negation of the first literal of $p_{i_1,j_{i_1}+1,k_{i_1}}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_{i_1}+1}\overline{x}_{i_2}^{i_1}_{j_{i_1}+1} + p_{i_1,j_{i_1}+1,k_{i_1}} + \cdots + p_m \quad (8.78)$$

**LNF First, then other Term Faults**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose a LNF is committed first by negating the $j_1$-th literal $x_{j_1}^{i_1}$ of the $i_1$-th term, $p_{i_1}$, of $S$. The corresponding faulty implementation is $I_{LNF}(p_{i_1}) = p_1 + \cdots + p_{i_1,j_1} + \cdots + p_m$.

**LNF and ENF** After the LNF is made on $S$, the subexpression $p_{i_1} + \cdots + p_{i_2}$ is then negated. Let $I_{LNF}(p_{i_1} - p_{i_1,j_1}) \otimes ENF(p_{i_2} + \cdots + p_{i_2} - p_{i_2} + \cdots + p_{i_2})$ denote the corresponding faulty implementation. We have the following two cases:

**Case 1.** The subexpression $p_{i_2} + \cdots + p_{i_2}$ does not contain the term $p_{i_1}$, that is $i_1 \notin \{i_2, \ldots, h_2\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2} + \cdots + p_{i_2} + \cdots + p_m \quad (8.79)$$

**Case 2.** The subexpression $p_{i_2} + \cdots + p_{i_2}$ contains the $i_1$-th term, that is $i_1 \in \{i_2, \ldots, h_2\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_2} + \cdots + p_{i_2,j_1} + \cdots + p_{i_2} + \cdots + p_m \quad (8.80)$$

**LNF and TNF** After the LNF is made on $S$, the term $p_{i_2}$ is then negated. Let $I_{LNF}(p_{i_1} - p_{i_1,j_1}) \otimes TNF(p_{i_2} - p_{i_2})$ denote the corresponding faulty implementation. We have the following two cases:
Case 1. The \( i_1 \)-th and \( i_2 \)-th terms are different, that is \( i_1 \neq i_2 \). Without loss of
generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent
to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + \bar{p}_{i_2} + \cdots + p_m \quad (8.81)
\]

Case 2. The \( i_1 \)-th and \( i_2 \)-th terms are the same, that is \( i_1 = i_2 \). The implementation
is then equivalent to the following expression

\[
p_1 + \cdots + \bar{p}_{i_1,j_1} + \cdots + p_m \quad (8.82)
\]

**LNF and TOF**  After the LNF is committed on \( S \), the term \( p_{i_2} \) is then omitted.
Let \( I_{\text{LNFI}(p_1 \rightarrow p_1,j_1)} \otimes \text{TOF}(p_{i_2} \rightarrow ) \) denote the corresponding faulty implementation. We
do not consider the situation where the term with the negated literal is then omitted
(that is, \( i_1 = i_2 \)) because the implementation is then equivalent to a single TOF
with respect to the original expression \( S \). Without loss of generality, we can assume
\( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \quad (8.83)
\]

**LNF and DORF**  After the LNF is made on \( S \), the subexpression \( p_{i_2} + p_{i_2+1} \)
is implemented as \( p_{i_2}p_{i_2+1} \). Let \( I_{\text{LNFI}(p_1 \rightarrow p_1,j_1)} \otimes \text{DORF}(p_{i_2}+p_{i_2+1} \rightarrow -p_{i_2}p_{i_2+1}) \)
denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression \( p_{i_2} + p_{i_2+1} \) does not contain the term \( p_{i_1} \). Without loss
of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent
to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2}p_{i_2+1} + \cdots + p_m \quad (8.84)
\]

Case 2. The subexpression \( p_{i_2} + p_{i_2+1} \) contains the term \( p_{i_1} \), that is \( i_1 \in \{i_2, i_2+1\} \).
Hence, there are two possible cases, namely \( i_1 = i_2 \) and \( i_1 = i_2 + 1 \). Without
loss of generality, we can assume \( i_1 = i_2 \). The implementation is then equivalent
to the following expression

\[
p_1 + \cdots + p_{i_1,j_1}p_{i_1+1} + \cdots + p_m \quad (8.85)
\]

**LNF and CORF**  After the LNF is committed on \( S \), the term \( p_{i_2} (= p_{i_2,1,j_2} \cdot
p_{i_2,j_2+1,k_{i_2}}) \) is then implemented as \( p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \) where \( p_{i_2,1,j_2} = x_1^{i_2} \cdots x_{j_2}^{i_2} \) and
\( p_{i_2,j_2+1,k_{i_2}} = x_{j_2+1}^{i_2} \cdots x_{k_{i_2}}^{i_2} \) denote the terms obtained from \( p_{i_2} \) by keeping its first \( j_2 \)}
literals and its last \((k_2 - j_2)\) literals, respectively.\(^{12}\) Let \(I_{LNF(p_{i_1} \rightarrow p_{i_1,j_1})} \odot CORF(p_{i_2} \rightarrow p_{i_2,j_2} + p_{i_2,j_2+1,k_2})\) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The terms \(p_{i_1}\) and \(p_{i_2}\) are different, that is \(i_1 \neq i_2\). Without loss of generality, we can assume \(i_1 < i_2\). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_2} + \cdots + p_m \quad (8.86)
\]

Case 2. The \(i_1\)-th and \(i_2\)-th terms are the same, that is \(i_1 = i_2\). Since LNF and CORF occur at the \(j_1\)-th and \(j_2\)-th literals of \(p_{i_1}\), we have two subcases. If \(j_1 \leq j_2\), the negated literal will be in \(p_{i_1,1,j_2}\). Otherwise, the negated literal will be in \(p_{i_1,j_2+1,k_2}\). Without loss of generality, we can assume the negated literal is in \(p_{i_1,1,j_2}\). Otherwise, we can always rearrange the literals in \(p_{i_1,1,j_2}\) and \(p_{i_1,j_2+1,k_2}\) (or the literals in \(p_{i_1}\)) so that negated literal is in \(p_{i_1,1,j_2}\). Furthermore, without loss of generality, we can assume that the negated literal is the last literal of \(p_{i_1,1,j_2}\). Otherwise, we can always rearrange the literals in \(p_{i_1,1,j_2}\) so that the negated literal is last literal of \(p_{i_1,1,j_2}\).\(^{13}\) The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,1,j_2} + p_{i_1,j_2+1,k_1} + \cdots + p_m \quad (8.87)
\]

where \(p_{i_1,1,j_2} = x_1^{i_1} \cdots x_{j_2-1}^{i_1} \cdot \bar{x}_{j_2}^{i_1}\) denotes the term obtained from \(p_{i_1,1,j_2}\) by negating its last literal \(x_{j_2}^{i_1} \).

### 8.2.6 LOF First, then other Term Faults

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose a LOF is committed first by omitting the \(j_1\)-th literal \(x_{j_1}^{i_1}\) of the \(i_1\)-th term, \(p_{i_1}\), of \(S\). The corresponding faulty implementation is \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} = p_1 + \cdots + p_{i_1,j_1} + \cdots + p_m\).

**LOF and ENF** After the LOF is committed on \(S\), the subexpression \(p_{i_2} + \cdots + p_{h_2}\) is then negated. Let \(I_{LOF(p_{i_1} \rightarrow p_{i_1,j_1})} \odot ENF(p_{i_2} + \cdots + p_{h_2} \rightarrow p_{i_2} + \cdots + p_{h_2})\) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression \(p_{i_2} + \cdots + p_{h_2}\) does not contain the term \(p_{i_1}\), that is \(i_1 \notin \{i_2, \ldots, h_2\}\). Without loss of generality, we can assume \(i_1 < i_2\). The

\[12\] In the rest of this thesis, notations similar to \(p_{i_2,1,j_2}\) and \(p_{i_2,j_2+1,k_2}\) should be interpreted similarly.

\[13\] In the rest of this thesis, we will make similar assumptions whenever a literal fault and CORF occur in the same term.
implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + \overline{p}_{i_2} + \cdots + p_{h_2} + \cdots + p_m \]  

(8.88)

Case 2. The subexpression \( p_{i_2} + \cdots + p_{h_2} \) contains the \( i_1 \)-th term, that is \( i_1 \in \{i_2, \ldots, h_2\} \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \overline{p}_{i_2} + \cdots + p_{i_1,j_1} + \cdots + p_{h_2} + \cdots + p_m \]  

(8.89)

**LOF and TNF** After the LOF is committed on \( S \), the term \( p_{i_2} \) is then negated. Let \( I_{LOF}(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes TNF(p_{i_2} \rightarrow \overline{p}_{i_2}) \) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The \( i_1 \)-th and \( i_2 \)-th terms are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + \overline{p}_{i_2} + \cdots + p_m \]  

(8.90)

Case 2. The \( i_1 \)-th and \( i_2 \)-th terms are the same, that is \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + \overline{p}_{i_1,j_1} + \cdots + p_m \]  

(8.91)

**LOF and TOF** After the LOF is committed on \( S \), the term \( p_{i_2} \) is then omitted. Let \( I_{LOF}(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes TOF(p_{i_2} \rightarrow ) \) denote the corresponding faulty implementation. We do not consider the situation of omitting the term with LOF being committed before (that is, \( i_1 = i_2 \)) because the implementation is then equivalent to a single TOF with respect to the original expression \( S \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m \]  

(8.92)

**LOF and DORF** After the LOF is made on \( S \), the subexpression \( p_{i_2} + p_{i_2+1} \) is implemented as \( p_i p_{i+1} \). Let \( I_{LOF}(p_{i_1} \rightarrow p_{i_1,j_1}) \otimes DORF(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2} p_{i_2+1}) \) denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression \( p_{i_2} + p_{i_2+1} \) does not contain the term \( p_{i_1} \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,j_1} + \cdots + p_{i_2} p_{i_2+1} + \cdots + p_m \]  

(8.93)
Case 2. The subexpression \( p_{i_2} + p_{i_2+1} \) contains the term \( p_{i_1} \), that is \( i_1 \in \{i_2, i_2+1\} \).

Hence, there are two possible cases, namely \( i_1 = i_2 \) and \( i_1 = i_2 + 1 \).

Without loss of generality, we can assume \( i_1 = i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1, j_1} p_{i_1+1} + \cdots + p_m \quad (8.94)
\]

**LOF and CORF** After the LOF is committed on \( S \), the term \( p_{i_2} (= p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_{i_2}}) \) is then implemented as \( p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} \) where \( p_{i_2,1,j_2} = x_{i_2}^1 \cdots x_{i_2}^{j_2} \) and \( p_{i_2,j_2+1,k_{i_2}} = x_{j_2+1}^{i_2} \cdots x_{k_{i_2}}^{i_2} \) denote the terms obtained from \( p_{i_2} \) by keeping its first \( j_2 \) literals and its last \((k_{i_2} - j_2)\) literals, respectively. We have the following two cases:

Case 1. The \( i_1\)-th and \( i_2\)-th terms are different, that is \( i_1 \neq i_2 \). Without loss of generality, we can assume \( i_1 < i_2 \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1, j_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_{i_2}} + \cdots + p_m \quad (8.95)
\]

Case 2. The \( i_1\)-th and \( i_2\)-th terms are the same, that is \( i_1 = i_2 \). Since LOF and CORF occur at the \( j_1\)-th and \( j_2\)-th literals of \( p_{i_1} \), we have two subcases. If \( j_1 \leq j_2 \), the omitted literal will be in \( p_{i_1,1,j_2} \). Otherwise, the omitted literal will be in \( p_{i_1,j_2+1,k_{i_1}} \). Without loss of generality, we can assume the omitted literal is in \( p_{i_1,1,j_2} \). Otherwise, we can always rearrange the literals in \( p_{i_1,1,j_2} \) and \( p_{i_1,j_2+1,k_{i_1}} \) (or the literals in \( p_{i_1} \)) so that omitted literal is in \( p_{i_1,1,j_2} \). Furthermore, without loss of generality, we can assume that the omitted literal is the last literal of \( p_{i_1,1,j_2} \), that is \( x_{j_2}^{i_1} \). Otherwise, we can always rearrange the literals in \( p_{i_1,1,j_2} \) so that the omitted literal is \( x_{j_2}^{i_1} \). The implementation is then equivalent to the following expression

\[
p_1 + \cdots + p_{i_1,1,j_2-1} + p_{i_1,j_2+1,k_{i_1}} + \cdots + p_m \quad (8.96)
\]

where \( p_{i_1,1,j_2-1} = x_1^{i_1} \cdots x_{j_2-1}^{i_1} \) denotes the term obtained from \( p_{i_1,1,j_2} \) by omitting the literal \( x_{j_2}^{i_1} \).

### 8.2.7 LIF First, then other Term Faults

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose a LIF is committed first by inserting the literal \( x_{i_1} \) into the \( i_1\)-th term, \( p_{i_1} \), of \( S \) where \( x_{i_1} \) is a missing literal of \( p_{i_1} \). The corresponding faulty implementation is \( I_{LIF(p_{i_1}-p_{i_1} x_{i_1})} = p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_m \).
**LIF and ENF** After the LIF is committed on $S$, the subexpression $p_{i_2} + \cdots + p_{h_2}$ is then negated. Let $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes ENF(p_{i_2} + \cdots + p_{h_2} \rightarrow p_{i_2} + \cdots + p_{h_2})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_2} + \cdots + p_{h_2}$ does not contain the term $p_{i_1}$, that is $i_1 \notin \{i_2, \ldots, h_2\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m$$  \hspace{1cm} (8.97)

Case 2. The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains the $i_1$-th term, that is $i_1 \in \{i_2, \ldots, h_2\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \bar{p}_{i_2} + \cdots + p_{i_1} x_{i_1} + \cdots + p_{h_2} + \cdots + p_m$$  \hspace{1cm} (8.98)

**LIF and TNF** After the LIF is committed on $S$, the term $p_{i_2}$ is then negated. Let $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes TNF(p_{i_2} \rightarrow \bar{p}_{i_2})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + \bar{p}_{i_2} + \cdots + p_m$$  \hspace{1cm} (8.99)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_m$$  \hspace{1cm} (8.100)

**LIF and TOF** After the LIF is committed on $S$, the term $p_{i_2}$ is then omitted. Let $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes TOF(p_{i_2} \rightarrow )}$ denote the corresponding faulty implementation. We do not consider the situation of omitting the term with the LIF being committed before (that is, $i_1 = i_2$) because the implementation is then equivalent to a single TOF with respect to the original expression $S$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m$$  \hspace{1cm} (8.101)

**LIF and DORF** After the LIF is committed on $S$, the subexpression $p_{i_2} + p_{i_2+1}$ is implemented as $p_{i_2}p_{i_2+1}$. Let $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{i_1}) \otimes DORF(p_{i_2} + p_{i_2+1} \rightarrow p_{i_2}p_{i_2+1})}$ denote the corresponding faulty implementation. We have the following two cases:
Case 1. The subexpression $p_{i_2} + p_{i_2+1}$ does not contain the term $p_{i_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_{i_2} p_{i_2+1} + \cdots + p_m$$  \hspace{1cm} (8.102)

Case 2. The subexpression $p_{i_2} + p_{i_2+1}$ contains the term $p_{i_1}$, that is $i_1 \in \{i_2, i_2+1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 = i_2 + 1$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} p_{i_1+1} + \cdots + p_m$$  \hspace{1cm} (8.103)

**LIF and CORF** After the LIF is committed on $S$, the term $p_{i_2}$ ($= p_{i_2,1,j_2} \cdot p_{i_2,j_2+1,k_i}$) is then implemented as $p_{i_2,1,j_2} + p_{i_2,j_2+1,k_i}$ where $p_{i_2,1,j_2} = x_{i_1}^{\alpha} \cdots x_{j_2}^{\alpha}$ and $p_{i_2,j_2+1,k_i} = x_{j_2+1}^{\alpha} \cdots x_{k_i}^{\alpha}$ denote the terms obtained from $p_{i_2}$ by keeping its first $j_2$ literals and its last $(k_i - j_2)$ literals, respectively. Let $I_{LIF(p_{i_1} \rightarrow p_{i_1} x_{i_1})} \circ CORF(p_{i_2} \rightarrow p_{i_2,1,j_2} + p_{i_2,j_2+1,k_i})$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1} x_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,j_2+1,k_i} + \cdots + p_m$$  \hspace{1cm} (8.104)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. There are two possible cases, namely, the inserted literal is inserted before $j_2$-th literal or after. If the inserted literal is before $j_2$-th literal, the literal will be in $p_{i_1,1,j_2}$. Otherwise, the inserted literal will be in $p_{i_1,j_2+1,k_i}$. Without loss of generality, we can assume the inserted literal is in $p_{i_1,1,j_2}$. Otherwise, we can always rearrange the literals in $p_{i_1,1,j_2}$ and $p_{i_1,j_2+1,k_i}$ (or the literals in $p_{i_1}$) so that inserted literal is in $p_{i_1,1,j_2}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,1,j_2} x_{i_1} + p_{i_1,j_2+1,k_i} + \cdots + p_m$$  \hspace{1cm} (8.105)

### 8.2.8 LRF First, then other Term Faults

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose a LRF is committed first by replacing the $j_1$-th literal $x_{j_1}^{\alpha}$ of the $i_1$-th term, $p_{i_1}$, of $S$ with the literal
$x_{i_1}$ where $x_{i_1}$ is a missing literal of $p_{i_1}$. The corresponding faulty implementation is $I_{LRF(p_{i_1} \rightarrow p_{i_1}, x_{i_1})} = p_1 + \cdots + p_{i_1, j_1} x_{i_1} + \cdots + p_m$.

**LRF and ENF** After the LRF is committed on $S$, the subexpression $p_{i_2} + \cdots + p_{h_2}$ is then negated. Let $I_{LRF(p_{i_1} \rightarrow p_{i_1}, x_{i_1}) \odot ENF(p_{i_2} + \cdots + p_{h_2} \rightarrow \overline{p_{i_2}} + \cdots + \overline{p_{h_2}})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_2} + \cdots + p_{h_2}$ does not contain the term $p_{i_1}$, that is $i_1 \not\in \{i_2, \ldots, h_2\}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1, j_1} x_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_2} + \cdots + p_m$$  \hspace{1cm} (8.106)

Case 2. The subexpression $p_{i_2} + \cdots + p_{h_2}$ contains the $i_1$-th term, that is $i_1 \in \{i_2, \ldots, h_2\}$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_2} + \cdots + p_{i_1, j_1} x_{i_1} + \cdots + p_{h_2} + \cdots + p_m$$  \hspace{1cm} (8.107)

**LRF and TNF** After the LRF is committed on $S$, the term $p_{i_2}$ is then negated. Let $I_{LRF(p_{i_1} \rightarrow p_{i_1}, x_{i_1}) \odot TNF(p_{i_2} \rightarrow \overline{p_{i_2}})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1, j_1} x_{i_1} + \cdots + p_{i_2} + \cdots + p_m$$  \hspace{1cm} (8.108)

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + \overline{p_{i_1, j_1}} x_{i_1} + \cdots + p_m$$  \hspace{1cm} (8.109)

**LRF and TOF** After the LRF is committed on $S$, the term $p_{i_2}$ is then omitted. Let $I_{LRF(p_{i_1} \rightarrow p_{i_1}, x_{i_1}) \odot TOF(p_{i_2} \rightarrow)}$ denote the corresponding faulty implementation. We do not consider the situation of omitting the term with the LRF being committed before (that is, $i_1 = i_2$) because the implementation is then equivalent to a single TOF with respect to the original expression $S$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1, j_1} x_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_m$$  \hspace{1cm} (8.110)
**LRF and DORF** After the LRF is committed on $S$, the subexpression $p_{i_2} + p_{i_2 + 1}$ is implemented as $p_{i_2}p_{i_2 + 1}$. Let $I_{LRF(p_{i_1} - p_{i_1,j_1} x_{i_1}) \odot DORF(p_{i_2} + p_{i_2} p_{i_2 + 1})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The subexpression $p_{i_2} + p_{i_2 + 1}$ does not contain the term $p_{i_1}$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} x_{i_1} + \cdots + p_{i_2} + p_{i_2 + 1} + \cdots + p_m \quad (8.111)$$

Case 2. The subexpression $p_{i_2} + p_{i_2 + 1}$ contains the term $p_{i_1}$, that is $i_1 \in \{i_2, i_2 + 1\}$. Hence, there are two possible cases, namely $i_1 = i_2$ and $i_1 = i_2 + 1$. Without loss of generality, we can assume $i_1 = i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} x_{i_1} p_{i_1 + 1} + \cdots + p_m \quad (8.112)$$

**LRF and CORF** After the LRF is committed on $S$, the term $p_{i_2} \ (= p_{i_2,1,j_2} + p_{i_2,2+1,j_2})$ is then implemented as $p_{i_2,1,j_2} + p_{i_2,2+1,j_2}$ where $p_{i_2,1,j_2} = x_{i_2}^{j_2} \cdots x_{j_2}$ and $p_{i_2,2+1,j_2} = x_{j_2+1}^{j_2} \cdots x_{k_{i_2}}$ denote the terms obtained from $p_{i_2}$ by keeping its first $j_2$ literals and its last $(k_{i_2} - j_2)$ literals, respectively. Let $I_{LRF(p_{i_1} - p_{i_1,j_1} x_{i_1}) \odot CORF(p_{i_2} - p_{i_2,1,j_2} + p_{i_2,2+1,j_2})}$ denote the corresponding faulty implementation. We have the following two cases:

Case 1. The $i_1$-th and $i_2$-th terms are different, that is $i_1 \neq i_2$. Without loss of generality, we can assume $i_1 < i_2$. The implementation is then equivalent to the following expression

$$p_1 + \cdots + p_{i_1,j_1} x_{i_1} + \cdots + p_{i_2,1,j_2} + p_{i_2,2+1,j_2} + \cdots + p_m \quad (8.113)$$

Case 2. The $i_1$-th and $i_2$-th terms are the same, that is $i_1 = i_2$. Since LOF and CORF occur at the $j_1$-th and $j_2$-th literals of $p_{i_1}$, we have two subcases. If $j_1 \leq j_2$, the replaced literal will be in $p_{i_1,1,j_2}$. Otherwise, the replaced literal will be in $p_{i_1,j_2+1,k_{i_1}}$. Without loss of generality, we can assume the replaced literal is in $p_{i_1,1,j_2}$. Otherwise, we can always rearrange the literals in $p_{i_1,1,j_2}$ and $p_{i_1,j_2+1,k_{i_2}}$ (or the literals in $p_{i_1}$) so that the replaced literal is in $p_{i_1,1,j_2}$. Furthermore, without loss of generality, we can assume that the replaced literal is the last literal of $p_{i_1,1,j_2}$, that is $x_{j_2}^{i_1}$. Otherwise, we can always rearrange the literals in $p_{i_1,1,j_2}$ so that the replaced literal is $x_{j_2}^{i_1}$. The
Implementation is then equivalent to the following expression

\[ p_1 + \cdots + p_{i_1,1,j_2-1}x_{l_1} + p_{i_1,j_2+1,k_{i_1}} + \cdots + p_m \]  

(8.114)

8.3 Relation between Double Faults with and without Ordering

In this section, the relationships of double faults with and without ordering based on those faulty expressions discussed in Sections 8.1 and 8.2 are analysed.

First, double faults with ordering where the term fault is being committed first are considered. Table 8.3 (respectively, 8.4, 8.5, 8.6 and 8.7) summarizes the situations of double faults with ENF (respectively TNF, TOF, DORF, and CORF) and other four types of literal faults. Each row in these tables shows those faulty expressions of a particular type of double fault without ordering and their counterparts in double faults with ordering.

Table 8.3: Comparison of double-fault expressions of ENF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - ENF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF and LNF</td>
<td>8.1, 8.2</td>
<td>8.37, 8.38</td>
</tr>
<tr>
<td>ENF and LOF</td>
<td>8.3, 8.4</td>
<td>8.39, 8.40</td>
</tr>
<tr>
<td>ENF and LIF</td>
<td>8.5, 8.6, -</td>
<td>8.41, 8.42, 8.43</td>
</tr>
<tr>
<td>ENF and LRF</td>
<td>8.7, 8.8</td>
<td>8.44, 8.45</td>
</tr>
</tbody>
</table>

For Table 8.3, there are two different situations. First, faulty expressions of the double faults with and without ordering in the same double fault class are equivalent. For example, in the first row, Expressions (8.37) and (8.1) are equivalent faulty expressions of ENF and LNF with and without ordering, respectively. Similarly, Expressions (8.38) and (8.2) are equivalent. Second, faulty expression of a double fault with ordering does not have its equivalent counterparts in double faults without ordering. For example, for the second subcase of ENF and LIF with ordering, Expression (8.43) does not have its equivalent counterpart in double faults without ordering.

Rows in Table 8.4 can be interpreted in a similar manner as those in Table 8.3. For example, Expressions (8.9) and (8.46) are equivalent. Moreover, Expression (8.52) does not have its equivalent counterparts in double faults without
Table 8.4: Comparison of double-fault expressions of TNF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - TNF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNF and LNF</td>
<td>8.9, 8.10</td>
<td>8.46, 8.47</td>
</tr>
<tr>
<td>TNF and LOF</td>
<td>8.11, 8.12</td>
<td>8.48, 8.49</td>
</tr>
<tr>
<td>TNF and LIF</td>
<td>8.13, 8.14</td>
<td>8.50, 8.51</td>
</tr>
<tr>
<td>TNF and LRF</td>
<td>8.15, 8.16</td>
<td>8.53, 8.54</td>
</tr>
</tbody>
</table>

Table 8.5: Comparison of double-fault expressions of TOF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - TOF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOF and LNF</td>
<td>8.17</td>
<td>8.55</td>
</tr>
<tr>
<td>TOF and LOF</td>
<td>8.18</td>
<td>8.56</td>
</tr>
<tr>
<td>TOF and LIF</td>
<td>8.19</td>
<td>8.57</td>
</tr>
<tr>
<td>TOF and LRF</td>
<td>8.20</td>
<td>8.58</td>
</tr>
</tbody>
</table>

In Tables 8.5 and 8.6, there are two situations. First, the faulty expressions of double fault with and without ordering in the same double fault class are equivalent. For example, for TOF and LNF, Expressions (8.55) and (8.17) are equivalent. Second, for some rows, the faulty expressions of double faults with ordering do have their counterparts in double faults without ordering in the same double fault class, but they are not equivalent. Expression (8.19), $p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2} x_{l_2} + \cdots + p_m$, is derived from TOF and LIF without ordering, where TOF being committed at $i_1$-th term and LIF being committed at the $i_2$-th term. Expression (8.57), $p_1 + \cdots + p_{i_1-1} + p_{i_1+1} + \cdots + p_{i_2} x_{l_2} + \cdots + p_m$, is derived from TOF and LIF with...
ordering, where TOF being committed at $i_1$-th term and LIF being committed at the $i_2$-th term. As mentioned in Section 8.1, when considering TOF $\times$ LIF without ordering, the literal $x_{i_2}$ is a missing literal of the $i_2$-th term of $S$. More clearly, $x_{i_2}$ is a variable of $S$, but both $x_{i_2}$ and $\bar{x}_{i_2}$ do not appear in $i_2$-th term. For example, $S = ab + cd + de$, for the second term $cd$, the missing literals are $a$, $\bar{a}$, $b$, $\bar{b}$, $e$ and $\bar{e}$. However, as mentioned in Section 6.2, the literal $x_{i_2}$ is a missing literal of the $i_2$-th term, $p_{i_2}$, of $I_{LOF}(p_{i_1} \rightarrow p_{i_1,j_1})$. It means $x_{i_2}$ is a variable of $I_{LOF}(p_{i_1} \rightarrow p_{i_1,j_1})$. For $S = ab + cd + de$, if the first term $ab$ is omitted, the resulting expression is $I = cd + de$. Then, for the term $cd$ of $I$, the missing literals are $e$ and $\bar{e}$. Clearly, the faulty expressions resulting from TOF $\times$ LIF with ordering is a subset of those resulting from TOF $\times$ LIF without ordering.

Table 8.7: Comparison of double-fault expressions of CORF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - CORF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORF and LNF</td>
<td>8.29, 8.30</td>
<td>8.67, 8.68</td>
</tr>
<tr>
<td>CORF and LOF</td>
<td>8.31, 8.32</td>
<td>8.69, 8.70</td>
</tr>
<tr>
<td>CORF and LIF</td>
<td>8.33, 8.34, -</td>
<td>8.71, 8.72, 8.73, 8.74</td>
</tr>
<tr>
<td>CORF and LRF</td>
<td>8.35, 8.36, -</td>
<td>8.75, 8.76, 8.77, 8.78</td>
</tr>
</tbody>
</table>

In Table 8.7, there are two different situations. First, the faulty expressions of double faults with and without ordering in the same double fault class are equivalent. For example, for CORF and LNF, Expressions (8.67) and (8.29) are equivalent. Second, faulty expression of a double fault with ordering does not have its equivalent counterparts in double faults without ordering. For example, for the third subcase of CORF and LIF with ordering, Expression (8.73) does not have its equivalent counterparts in double faults without ordering.

Now, the situation for double fault with ordering with the literal fault being committed first are considered. Table 8.8 (respectively, 8.9, 8.10 and 8.11) summarizes the situations of double faults with LNF (respectively LOF, LIF, and LRF) and other five types of term faults.

In Table 8.8, there is one situation. The faulty expressions of double faults with and without ordering in the same double fault class are equivalent. For example,
since Expressions (8.79) and (8.1) are in the form of

\[ p_1 + \ldots + p_{i_1,j_1} + \ldots + \overline{p_{i_2}} + \ldots + p_{i_1,j_1} + \ldots + p_{h_2} + \ldots + p_m \quad (1 \leq i_1 < i_2 < h_2 \leq m) \quad (7.79) \]

and

\[ p_1 + \ldots + p_{i_1} + \ldots + \overline{p_{i_2}} + \ldots + p_{i_2,j_2} + \ldots + p_m \quad (1 \leq i_1 < h_1 < i_2 \leq m) \quad (7.1) \]

we can always rearrange the terms and literals in these expressions so that they are equivalent counterparts of each other. Similar situations occurred at other expressions. Finally, rows in Tables 8.9 to 8.11 can be interpreted similarly.

Table 8.8: Comparison of double-fault expressions of LNF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LNF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF and ENF</td>
<td>8.1, 8.2</td>
<td>8.79, 8.80</td>
</tr>
<tr>
<td>LNF and TNF</td>
<td>8.9, 8.10</td>
<td>8.81, 8.82</td>
</tr>
<tr>
<td>LNF and TOF</td>
<td>8.17</td>
<td>8.83</td>
</tr>
<tr>
<td>LNF and DORF</td>
<td>8.21, 8.22</td>
<td>8.84, 8.85</td>
</tr>
<tr>
<td>LNF and CORF</td>
<td>8.29, 8.30</td>
<td>8.86, 8.87</td>
</tr>
</tbody>
</table>

Table 8.9: Comparison of double-fault expressions of LOF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LOF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF and ENF</td>
<td>8.3, 8.4</td>
<td>8.88, 8.89</td>
</tr>
<tr>
<td>LOF and TNF</td>
<td>8.11, 8.12</td>
<td>8.90, 8.91</td>
</tr>
<tr>
<td>LOF and TOF</td>
<td>8.18</td>
<td>8.92</td>
</tr>
<tr>
<td>LOF and DORF</td>
<td>8.23, 8.24</td>
<td>8.93, 8.94</td>
</tr>
<tr>
<td>LOF and CORF</td>
<td>8.31, 8.32</td>
<td>8.95, 8.96</td>
</tr>
</tbody>
</table>

8.4 Summary

As discussed in Sections 8.1 and 8.2, there are 36 possible faulty expressions of double faults without ordering and 78 possible faulty expressions of double faults with ordering. After comparing all these faulty expressions, 72 out of the 78 faulty
Table 8.10: Comparison of double-fault expressions of LIF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LIF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIF and ENF</td>
<td>8.5, 8.6</td>
<td>8.97, 8.98</td>
</tr>
<tr>
<td>LIF and TNF</td>
<td>8.13, 8.14</td>
<td>8.99, 8.100</td>
</tr>
<tr>
<td>LIF and TOF</td>
<td>8.19</td>
<td>8.101</td>
</tr>
<tr>
<td>LIF and DORF</td>
<td>8.25, 8.26</td>
<td>8.102, 8.103</td>
</tr>
<tr>
<td>LIF and CORF</td>
<td>8.33, 8.34</td>
<td>8.104, 8.105</td>
</tr>
</tbody>
</table>

Table 8.11: Comparison of double-fault expressions of LRF and other term faults

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Double fault without ordering (Expression number)</th>
<th>Double fault with ordering - LRF first (Expression number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRF and ENF</td>
<td>8.7, 8.8</td>
<td>8.106, 8.107</td>
</tr>
<tr>
<td>LRF and TNF</td>
<td>8.15, 8.16</td>
<td>8.108, 8.109</td>
</tr>
<tr>
<td>LRF and TOF</td>
<td>8.20</td>
<td>8.110</td>
</tr>
<tr>
<td>LRF and DORF</td>
<td>8.27, 8.28</td>
<td>8.111, 8.112</td>
</tr>
<tr>
<td>LRF and CORF</td>
<td>8.35, 8.36</td>
<td>8.113, 8.114</td>
</tr>
</tbody>
</table>
expressions of double faults with ordering have their equivalent counterparts in double faults without ordering. As a result, the 36 faulty expressions due to double faults without ordering can be used to represent these 72 faulty expressions. The remaining 6 faulty expressions are given by Expressions (8.43), (8.52), (8.73), (8.74), (8.77) and (8.78). Hence, for the nine single classes studied in this chapter, there are altogether 42 different double-fault expressions, 36 of them are from double fault without ordering and the remaining 6 are from double fault with ordering.

Table 8.12 summarizes all 42 double-fault expressions and their corresponding double fault classes related to term and literal. Similarly, the notation $F_1 \triangleright F_2$ is used to denote the double fault class formed from two single fault classes $F_1$ and $F_2$. For ease of reference, the numbers of those double-fault expressions reported in Section 8.1 and Section 8.2 are used.
Table 8.12: Double fault classes and double-fault expressions ($S = p_1 + \ldots + p_m$) due to double fault without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF $\times$ LNF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.1)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 \leq i_2 \leq h_1$ and $i_1 \neq h_1$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{i_2.2} + \ldots + p_m$ (8.2)</td>
</tr>
<tr>
<td>ENF $\times$ LOF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.3)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 \leq i_2 \leq h_1$ and $i_1 \neq h_1$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{i_2.2} + \ldots + p_m$ (8.4)</td>
</tr>
<tr>
<td>ENF $\times$ LIF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.5)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 \leq i_2 \leq h_1$ and $i_1 \neq h_1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{i_2.2} + \ldots + p_m$ (8.6)</td>
</tr>
<tr>
<td>ENF $\times$ LRF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.7)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($i_1 \leq i_2 \leq h_1$ and $i_1 \neq h_1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{i_2.2} + \ldots + p_m$ (8.8)</td>
</tr>
<tr>
<td>TNF $\times$ LNF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.9)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($k_{i_1} &gt; 1$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.10)</td>
</tr>
<tr>
<td>TNF $\times$ LOF</td>
<td>Case 1 ($i_1 &lt; h_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.11)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($k_{i_1} &gt; 1$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.12)</td>
</tr>
<tr>
<td>TNF $\times$ LIF</td>
<td>Case 1 ($i_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.13)</td>
</tr>
<tr>
<td></td>
<td>Case 2: $p_1 + \ldots + p_{12.2} + \ldots + p_m$ (8.14)</td>
</tr>
<tr>
<td>TNF $\times$ LRF</td>
<td>Case 1 ($i_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.15)</td>
</tr>
<tr>
<td></td>
<td>Case 2: $p_1 + \ldots + p_{12.2} + \ldots + p_m$ (8.16)</td>
</tr>
<tr>
<td>TOF $\times$ LNF</td>
<td>($i_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + \ldots + p_{12.2} + \ldots + p_m$ (8.17)</td>
</tr>
<tr>
<td>TOF $\times$ LOF</td>
<td>($i_1 &lt; i_2$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + \ldots + p_{12.2} + \ldots + p_m$ (8.18)</td>
</tr>
<tr>
<td>TOF $\times$ LIF</td>
<td>($i_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.19)</td>
</tr>
<tr>
<td>TOF $\times$ LRF</td>
<td>($i_1 &lt; i_2$): $p_1 + \ldots + p_{i_1} + \ldots + p_{12.2} + \ldots + p_m$ (8.20)</td>
</tr>
<tr>
<td>DORF $\times$ LNF</td>
<td>Case 1 ($1 &lt; i_1 + 1 &lt; i_2 \leq m$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + \ldots + p_{12.2} + \ldots + p_m$ (8.21)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq i_1 &lt; m$, $k_{i_1} &gt; 1$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.22)</td>
</tr>
<tr>
<td>DORF $\times$ LOF</td>
<td>Case 1 ($1 &lt; i_1 + 1 &lt; i_2 \leq m$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + \ldots + p_{12.2} + \ldots + p_m$ (8.23)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq i_1 &lt; m$, $k_{i_1} &gt; 1$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.24)</td>
</tr>
<tr>
<td>DORF $\times$ LIF</td>
<td>Case 1 ($1 &lt; i_1 + 1 &lt; i_2 \leq m$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + \ldots + p_{12.2} + \ldots + p_m$ (8.25)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq i_1 &lt; m$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.26)</td>
</tr>
<tr>
<td>DORF $\times$ LRF</td>
<td>Case 1 ($1 &lt; i_1 + 1 &lt; i_2 \leq m$): $p_1 + \ldots + p_{i_1} + 1 + \ldots + p_{i_2.2} + \ldots + p_m$ (8.27)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq i_1 &lt; m$): $p_1 + \ldots + p_{i_1.2} + \ldots + p_m$ (8.28)</td>
</tr>
<tr>
<td>CORF $\times$ LNF</td>
<td>Case 1 ($i_1 &lt; i_2$, $j_1 &lt; k_{i_1}$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.29)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq i_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.30)</td>
</tr>
<tr>
<td>CORF $\times$ LOF</td>
<td>Case 1 ($i_1 &lt; i_2$, $j_1 &lt; k_{i_1}$, $k_{i_2} &gt; 1$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.31)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($1 \leq j_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.32)</td>
</tr>
<tr>
<td>CORF $\times$ LIF</td>
<td>Case 1 ($i_1 &lt; i_2$, $j_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1} + 1 + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.33)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($j_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1.1} + p_{i_1.2} \ldots + 1 + \ldots + p_m$ (8.34)</td>
</tr>
<tr>
<td>CORF $\times$ LRF</td>
<td>Case 1 ($i_1 &lt; i_2$, $j_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1} + p_{i_1.1} + p_{i_1.2} \ldots + p_{i_2.2} + \ldots + p_m$ (8.35)</td>
</tr>
<tr>
<td></td>
<td>Case 2 ($j_1 &lt; k_{i_1}$): $p_1 + \ldots + p_{i_1.1} + p_{i_1.2} \ldots + p_{i_2.2} + \ldots + p_m$ (8.36)</td>
</tr>
</tbody>
</table>

177
Table 8.12 (cont’d) Double fault classes and double-fault expressions \((S = p_1 + \ldots + p_m)\)

(b) Six Double-fault expressions\(^a\) due to double faults with ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>Double-fault Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF ⋊ LIF</td>
<td>Case 3((i_1 &lt; k_1)): (p_1 + \cdots + p_{i_1} + \cdots + p_{i_1}x_{i_2} + \cdots + p_m) (8.43)</td>
</tr>
<tr>
<td>TNF ⋊ LIF</td>
<td>Case 3: (p_1 + \cdots + p_{i_1}x_{i_2} + \cdots + p_m) (8.52)</td>
</tr>
<tr>
<td>CORF ⋊ LIF</td>
<td>Case 2(b)(1 (\leq j_1 &lt; k_1)): (p_1 + \cdots + p_{i_1,1,j_1+1} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m) (8.73)</td>
</tr>
<tr>
<td></td>
<td>Case 2(c)(1 (\leq j_1 &lt; k_1)): (p_1 + \cdots + p_{i_1,1,j_1,x_{j_1+1}} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m) (8.74)</td>
</tr>
<tr>
<td>CORF ⋊ LRF</td>
<td>Case 2(b)(1 (\leq j_1 &lt; k_1)): (p_1 + \cdots + p_{i_1,1,j_1-1,x_{j_1+1}} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m) (8.77)</td>
</tr>
<tr>
<td></td>
<td>Case 2(c)(1 (\leq j_1 &lt; k_1)): (p_1 + \cdots + p_{i_1,1,j_1-1,x_{j_1+1}} + p_{i_1,j_1+1,k_{i_1}} + \cdots + p_m) (8.78)</td>
</tr>
</tbody>
</table>

\(^a\)For ease of cross-reference, the numbering of the faulty expressions follows Section 8.2.
Chapter 9

Detection Conditions on Double Faults Related to a Term and a Literal

In this chapter, the detection conditions of the double faults related to a term and a literal are studied. Chapter 8 shows that there are altogether 42 (=36+6) different double-fault expressions related to double fault on term and literal. In this chapter, the detection conditions of 36 double-fault expressions and 6 remaining double-fault expressions are presented in Section 9.1 and 9.2, respectively. The $IF_1(E_1 \rightarrow E'_1) \bowtie IF_2(E_2 \rightarrow E'_2)$ is used to denote the resulting double-fault expression where two single fault classes $F_1$ and $F_2$ are committed in a given Boolean expression $S$ changing its subexpressions $E_1$ and $E_2$ to $E'_1$ and $E'_2$, respectively.

9.1 Detection conditions on 36 Double-fault Expressions based on Double Fault without Ordering

In this section, the detection conditions of double faults related to a term and a literal are discussed. As a reminder, there are 5 fault classes related to terms and 4 fault classes related to literal within Boolean expressions considered in this thesis. The discussion are organized into the following 5 subsections as shown in Table 9.1.

9.1.1 ENF with Other Literal Faults

In this section, we study the detection conditions of double faults in which one of the single fault is an ENF.

**Theorem 9.1.1** (ENF \(\bowtie\) LNF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpres-
sion $p_1 + \cdots + p_{n_1}$ in $S$ and the $j_2$-th literal $x_{j_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ are negated where $1 \leq i_1 < h_1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2} > 1$ and $k_{i_2}$ is the number of literals in $p_{i_2}$, the resulting implementation denoted as $I_{\text{ENF}(p_1 + \cdots + p_{h_1} + p_{h_1 + 1}) \times \text{LNF}(p_{i_2} + p_{i_2 - 1})}$ is equivalent to Expression (8.1) in Table 8.12. Then, $S \neq I_{\text{ENF}(p_1 + \cdots + p_{h_1} + p_{h_1 + 1}) \times \text{LNF}(p_{i_2} + p_{i_2 - 1})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i 
eq 1, \ldots, h_1, i_2}^{m} TP_i(S) \right)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$, or

2. $\vec{t} \in FP(S)$.

**Proof:** First, we observe that $S \oplus I_{\text{ENF}(p_1 + \cdots + p_{h_1} + p_{h_1 + 1}) \times \text{LNF}(p_{i_2} + p_{i_2 - 1})}$

$\equiv ((p_1 + \cdots + p_{h_1} + p_{i_2}) \oplus (p_1 + \cdots + p_{h_1} + p_{i_2 - 1})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_1 + \cdots + p_{h_1} + p_{i_2}) ((p_1 + \cdots + p_{h_1} + p_{i_2 - 1})) (p_1 + \cdots + p_{h_1} + p_{i_2 - 1})) (p_1 + \cdots + p_{h_1} + p_{i_2 - 1})$

$\cdot (p_1 + \cdots + p_{h_1} + p_{i_2 - 1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv ((p_1 + \cdots + p_{h_1}) \bar{p}_{i_2 - 1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

(By making use of $(A + B)A = A$ and $A(A + B) = A$)

$\equiv (p_1 + \cdots + p_{h_1}) \bar{p}_{i_2 - 1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

$+ (p_1 + \cdots + p_{h_1}) \bar{p}_{i_2 - 1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

$\equiv (p_1 + \cdots + p_{h_1}) \bar{p}_{i_2 - 1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + S$
Now, $$S(\vec{t}) \neq I_{ENF(p_1 + \cdots + p_{h_1} - \bar{p}_1 + \cdots + \bar{p}_{h_1}) \times LNF(p_{i_2} - p_{i_2, j_2})}(\vec{t})$$ if and only if

$$S(\vec{t}) \oplus I_{ENF(p_1 + \cdots + p_{h_1} - \bar{p}_1 + \cdots + \bar{p}_{h_1}) \times LNF(p_{i_2} - p_{i_2, j_2})}(\vec{t}) = 1$$

if and only if

$$(p_{i_1} + \cdots + p_{h_1})\bar{p}_{i_2,j_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}p_{h_1} + 1 \cdots \bar{p}_{i_2-1}p_{i_2} + 1 \cdots \bar{p}_m + \bar{S}$$

evaluates to 1 on $\vec{t}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$, or
2. $\vec{t} \in FP(S)$.

Hence, the result follows. \hfill \square

In Theorem 9.1.1, ENF occurs at the subexpression $p_{i_1} + \cdots + p_{h_1}$ and LNF occurs at a term outside the subexpression, we can assume without loss of generality that $i_1 < h_1$. However, in the next theorem, when LNF occurs at a term within the subexpression $p_{i_1} + \cdots + p_{h_1}$, it is possible that $i_1 \leq h_1$. In fact, when $i_1 = h_1$, the expression $S$ contains one and only one term. Otherwise, the negation of $p_{i_1}$ is considered as a TNF in $S$ rather than an ENF. In the rest of the thesis, we will make similar observations related to a literal fault occurring within a subexpression.

**Theorem 9.1.2 (ENF \& LNF - Case 2)**

(a) Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}$ in $S$ and the $j_2$-th literal $x_2^{j_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ are negated where $1 \leq i_1 \leq i_2 \leq h_1 \leq m$, $i_1 \neq h_1$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2} > 1$ and $k_{i_2}$ is the number of literals in $p_{i_2}$, the resulting implementation denoted as $I_{ENF(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times LNF(p_{i_2} - p_{i_2, j_2})}$ is equivalent to Expression (8.2) in Table 8.12. Then, $S \neq I_{ENF(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1}) \times LNF(p_{i_2} - p_{i_2, j_2})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or
2. $\vec{t} \in FP(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$.

(b) Let $S = p_{i_2}$.\footnote{In this part of the theorem, we keep using notations like $p_{i_2}$ instead of $p_{1}$ for consistency with part (a) above and ease of comparison between the detection conditions of both parts.} Suppose that $S$ and its $j_2$-th literal are negated where $1 \leq j_2 \leq k_{i_2}$, $k_{i_2} > 1$ and $k_{i_2}$ is the number of literals in $p_{i_2}$, the resulting implementation,
denoted as $I_{\text{ENF}((S \implies S) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j}))}$, is equivalent to Expression (8.2) in Table 8.12 when $m = 1$. Then, $S \not\equiv I_{\text{ENF}((S \implies S) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j}))}$ if and only if there is a $\bar{t} \in FP(S)$ and $p_{i_2 \cdot j_2} = 0$ on $\bar{t}$.

**Proof**: (a) First, we observe that

$$S \oplus I_{\text{ENF}(p_1 + \cdots + p_2 + \cdots + p_{i_2} \implies \bar{p_1} + \cdots + p_2 + \cdots + p_{i_2}) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j}))} \equiv ((p_1 + \cdots + p_2 + \cdots + p_{i_2} \oplus (p_1 + \cdots + p_{i_2} + \cdots + p_{i_2})) \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m \equiv ((p_1 + \cdots + p_2 + \cdots + p_{i_2} \oplus p_{i_2+1} + \cdots + p_{i_2}) \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m $$

$$\equiv ((p_1 + \cdots + p_2 + \cdots + p_{i_2} \oplus p_{i_2+1} + \cdots + p_{i_2}) \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_2} \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m \equiv (p_1 + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{i_2}) \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m + \bar{p}_{i_2 \cdot j_2} \bar{S} \quad \text{(By making use of } (AB + C)(AB + C) \equiv C; \text{ and rewriting } (AB + C)(AB + C) \text{ as } (AB + C)A \text{ because they are equivalent)}$$

Now, $S(\bar{t}) \not\equiv I_{\text{ENF}(p_1 + \cdots + p_2 + \cdots + p_{i_2} \implies \bar{p_1} + \cdots + p_2 + \cdots + p_{i_2}) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j}))}((\bar{t}))$ if and only if $S(\bar{t}) \oplus I_{\text{ENF}(p_1 + \cdots + p_2 + \cdots + p_{i_2} \implies \bar{p_1} + \cdots + p_2 + \cdots + p_{i_2}) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j})) \equiv 1$ if and only if $(p_1 + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{i_2}) \bar{p_1} \cdots \bar{p}_{i_1} \bar{p}_{i_1+1} \cdots \bar{p}_m + \bar{p}_{i_2 \cdot j_2} \bar{S}$ evaluates to $1$ on $\bar{t}$ if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in \left( \bigcup_{i = i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1} TP_i(S) \right)$, or

2. $\bar{t} \in FP(S)$ such that $p_{i_2 \cdot j_2} = 0$ on $\bar{t}$.

Hence, the result follows. (b) First, we observe that $S \oplus I_{\text{ENF}((S \implies S) \land \text{LNF}(p_2 \implies p_{i_2} \implies \bar{j}))}$

$$\equiv p_{i_2} \oplus p_{i_2 \cdot j_2} \equiv p_{i_2} \bar{p}_{i_2 \cdot j_2} + \bar{p}_{i_2} \cdot p_{i_2 \cdot j_2} \equiv p_{i_2} p_{i_2 \cdot j_2} + \bar{p}_{i_2} \cdot p_{i_2 \cdot j_2} \equiv \bar{p}_{i_2} \cdot p_{i_2 \cdot j_2} \quad \text{(By making use of } (AB)(AB) \equiv 0; \text{ and rewriting } (AB)(AB) \text{ as } (AB)A \text{ because they are equivalent to } A)$$

$$\equiv \bar{p}_{i_2 \cdot j_2} \bar{S}$$
Now, \( S(\vec{t}) \neq I_{\text{ENF}(S \rightarrow \overline{S}) \times \text{LNF}(p_{i_2} \rightarrow p_{i_2}, j_2)}(\vec{t}) \)
if and only if 
\( S(\vec{t}) \oplus I_{\text{ENF}(S \rightarrow \overline{S}) \times \text{LNF}(p_{i_2} \rightarrow p_{i_2}, j_2)}(\vec{t}) = 1 \)

if and only if \( \overline{p_{i_2,j_2}} \overline{S} \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \in FP(S) \) and \( p_{i_2,j_2} = 0 \) on \( \vec{t} \).

Hence, the result follows.

We now compare the detection conditions between Theorem 9.1.2(a) and (b). First, we observe that the detection condition 2 in Theorem 9.1.2(a) and the detection condition in Theorem 9.1.2(b) are the same. Second, if we just look at the syntax of detection condition 1 in Theorem 9.1.2(a) when \( m = 1 \), the condition degenerates to \( \vec{t} \in \emptyset \) which can never be satisfied. Hence, without loss of generality, we can still use detection conditions 1 and 2 in Theorem 9.1.2(a) to represent the detection conditions of Expression (8.2) in Table 8.12 for \( m \geq 1 \). In the rest of this report, we will make similar observations.

**Theorem 9.1.3** *(\text{ENF \times LOF - Case 1})*

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the sub-expression \( p_{i_1} + \cdots + p_{i_2} \) in \( S \) is negated and the \( j_2 \)-th literal \( x_{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 \leq i_1 < h_1 < i_2 \leq m, 1 \leq j_2 \leq k_{i_2}, k_{i_2} > 1 \)
and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the resulting implementation denoted as \( I_{\text{ENF}(p_{i_1} + \cdots + p_{i_1} \rightarrow p_{i_2} + \cdots + p_{i_2}) \times \text{LOF}(p_{i_2} \rightarrow p_{i_2}, j_2)} \) is equivalent to Expression (8.3) in Table 8.12. Then, \( S \neq I_{\text{ENF}(p_{i_1} + \cdots + p_{i_1} \rightarrow p_{i_2} + \cdots + p_{i_2}) \times \text{LOF}(p_{i_2} \rightarrow p_{i_2}, j_2)} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \), or

2. \( \vec{t} \in FP(S) \).

**Proof:** First, we observe that \( S \oplus I_{\text{ENF}(p_{i_1} + \cdots + p_{i_1} \rightarrow p_{i_2} + \cdots + p_{i_2}) \times \text{LOF}(p_{i_2} \rightarrow p_{i_2}, j_2)} \)

\[ \equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2}) \oplus (p_{i_1} + \cdots + p_{i_1} + p_{i_2}, j_2)) \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1-2} \cdots \overline{p}_{i_2} \overline{p}_{i_2+1} \cdots \overline{p}_m \]

\[ \equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2}) (p_{i_1} + \cdots + p_{i_1} + p_{i_2}, j_2) + (p_{i_1} + \cdots + p_{i_1} + p_{i_2}, j_2)) \]

\[ \cdot (p_{i_1} + \cdots + p_{i_1} + p_{i_2}, j_2) \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1-2} \cdots \overline{p}_{i_2} \overline{p}_{i_2+1} \cdots \overline{p}_m \]

\[ \equiv ((p_{i_1} + \cdots + p_{i_1} + p_{i_2}) \overline{p}_{i_2,j_2} + (p_{i_1} + \cdots + p_{i_1} + p_{i_2}) \overline{p}_{i_2,j_2} + (p_{i_1} + \cdots + p_{i_1} + p_{i_2})) \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1-2} \cdots \overline{p}_{i_2} \overline{p}_{i_2+1} \cdots \overline{p}_m \]

(By making use of \( (A+B)A = A \) and \( (\overline{A} + BC)(\overline{A} + B) = \overline{A} + BC \))

\[ \equiv (p_{i_1} + \cdots + p_{i_1}) \overline{p}_{i_2,j_2} \overline{p}_{i_2} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1-2} \cdots \overline{p}_{i_2} \overline{p}_{i_2+1} \cdots \overline{p}_m \]

\[ + (p_{i_1} + \cdots + p_{i_1}) \overline{p}_{i_2,j_2} \overline{p}_{i_2} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1-2} \cdots \overline{p}_{i_2} \overline{p}_{i_2+1} \cdots \overline{p}_m \]

(By rewriting \( \overline{A} \) as \( \overline{(A)} \overline{(AB)} \) because they are equivalent)
\[ \equiv (p_{i_1} + \cdots + p_{h_1})\bar{p}_{i_2,j_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{h_1+1} \cdots \bar{p}_m + \bar{S} \]

Now, \[ S(\vec{t}) \neq I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} - p_{i_1+1} - \cdots - p_{h_1}) \times \text{LOF}(p_{i_2} - p_{i_2,j_2})(\vec{t}) \]
if and only if \[ S(\vec{t}) \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{h_1} - p_{i_1+1} - \cdots - p_{h_1}) \times \text{LOF}(p_{i_2} - p_{i_2,j_2})(\vec{t}) = 1 \]
if and only if \[ (p_{i_1} + \cdots + p_{h_1})\bar{p}_{i_2,j_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{h_1+1} \cdots \bar{p}_m + \bar{S} \text{ evaluates to 1 on } \vec{t} \]
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \[ \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{m} TP_i(S) \right) \text{ such that } p_{i_2,j_2} = 0 \text{ on } \vec{t}, \text{ or} \]
2. \( \vec{t} \in FP(S) \).

Hence, the result follows. \( \square \)

Similar to the situation of Theorem 9.1.2, we need to split the discussion of Theorem 9.1.4 in two cases depending on \( m > 1 \) or \( m = 1 \). Afterwards, the detection condition of Expression (8.4) in Table 8.12 can be summarized as one set of detection conditions.

**Theorem 9.1.4 (\textbf{ENF} \times \textbf{LOF} - Case 2)**

(a) Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the sub-expression \( p_{i_1} + \cdots + p_{i_2} + \cdots + p_{i_1} \) in \( S \) is negated and the \( j_2 \)-th literal \( x_{i_2}^{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 \leq i_1 \leq i_2 \leq h_1 \leq m, i_1 \neq h_1, 1 \leq j_2 \leq k_{i_2}, \) \( k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the resulting implementation denoted as \( I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1+1} - \cdots - p_{h_1}) \times \text{LOF}(p_{i_2} - p_{i_2,j_2}) \) is equivalent to Expression (8.4) in Table 8.12. Then, \( S \neq I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_1+1} - \cdots - p_{h_1}) \times \text{LOF}(p_{i_2} - p_{i_2,j_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

(a) \[ \vec{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{m} TP_i(S) \right), \text{ or} \]
(b) \( \vec{t} \in FP(S) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \).

(b) Let \( S = p_{i_2} \).\(^2\) Suppose that \( S \) is negated and its \( j_2 \)-th literal is omitted where \( 1 \leq j_2 \leq k_{i_2}, k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the resulting implementation, denoted as \( I_{\text{ENF}}(S - \bar{S}) \times \text{LOF}(p_{i_2} - p_{i_2,j_2}) \), is equivalent to that given by \( ^2\)Similar to Theorem 9.1.2, we use notations like \( p_{i_2} \) instead of \( p_1 \) for consistency and ease of comparison.
Expression (8.4) in Table 8.12 when \( m = 1 \). Then, \( S \not\equiv I_{\mathrm{ENF}(S \to \overline{S})} \times \mathrm{LOF}(p_{ij} \rightarrow p_{ij}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in TP(S) \), or
2. \( \vec{t} \in FP(S) \) such that \( p_{ij} = 0 \) on \( \vec{t} \).

Proof: (a) First, we observe that

\[
S \oplus I_{\mathrm{ENF}(p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) \bullet \mathrm{LOF}(p_{ij} \rightarrow p_{ij})} \\
\equiv ((p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) \oplus (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1})) \vec{p}_{i1} \cdots \vec{p}_{i1-1} \vec{p}_{h1+1} \cdots \vec{p}_{m} \\
\equiv ((p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) \vec{p}_{i1} \cdots \vec{p}_{i1-1} \vec{p}_{h1+1} \cdots \vec{p}_{m} \\
\equiv ((p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) \vec{p}_{i1} \cdots \vec{p}_{i1-1} \vec{p}_{h1+1} \cdots \vec{p}_{m} \\
\quad \text{(By making use of } (AB)\text{(A + C)} \equiv AB + C \text{ and rewriting) } \\
\equiv (p_{i1} + \cdots + p_{ij} + \cdots + p_{ih_1}) \vec{p}_{i1} \cdots \vec{p}_{i1-1} \vec{p}_{h1+1} \cdots \vec{p}_{m} + \vec{p}_{ij} \overline{S} \\
\]"
Now, $S(\vec{t}) \neq I_{\text{ENF}(S = \bar{S}) \text{LOF}(p_{i_2}, \neg p_{i_2, j_2})}(\vec{t})$
if and only if $S(\vec{t}) \oplus I_{\text{ENF}(S = \bar{S}) \text{LOF}(p_{i_2}, \neg p_{i_2, j_2})}(\vec{t}) = 1$
if and only if $S + \bar{p}_{i_2, j_2} \bar{S}$ evaluates to 1 on $\vec{t}$
if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in TP(S)$, or
2. $\vec{t} \in FP(S)$ and $p_{i_2, j_2} = 0$ on $\vec{t}$.

Hence, the result follows. $\Box$

Similar to Case 2 of ENF $\cong$ LNF, when $m = 1$, the detection condition 1 of Theorem 9.1.4(a) degenerates to $\vec{t} \in TP(S)$ which is the detection condition 1 of Theorem 9.1.4(b). Moreover, the detection condition 2 of both parts in Theorem 9.1.4 are the same. Hence, without loss of generality, we can still use detection conditions 1 and 2 in Theorem 9.1.4(a) to represent the detection conditions of Expression (8.4) in Table 8.12 for $m \geq 1$.

**Theorem 9.1.5 (ENF $\cong$ LIF - Case 1)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $p_{i_1} + \cdots + p_{i_1}$ in $S$ is negated and the literal $x_{i_2}$ is inserted in $i_2$-th term, $p_{i_2}$, in $S$ where $1 \leq i_1 < h_1 < i_2 \leq m$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting implementation denoted as $I_{\text{ENF}(p_1 + \cdots + p_{i_1} \neg p_{i_1} + \cdots + p_{i_2}) \text{LIF}(p_{i_2} \rightarrow x_{i_2})}$ is equivalent to Expression (8.5) in Table 8.12. Then, $S \neq I_{\text{ENF}(p_1 + \cdots + p_{i_1} \neg p_{i_1} + \cdots + p_{i_2}) \text{LIF}(p_{i_2} \rightarrow x_{i_2})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in \bigcup_{i = i_1}^{h_1} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_1} TP_i(S)$ on $\vec{t}$,
2. $\vec{t} \in \left( \bigcup_{i = i_1}^{h_1} TP_i(S) \cap TP_{i_2}(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1, i_2} TP_i(S) \right)$ such that $x_{i_2} = 0$ on $\vec{t}$, or
3. $\vec{t} \in FP(S)$.

**Proof:**

First, we observe that $S \oplus I_{\text{ENF}(p_1 + \cdots + p_{i_1} \neg p_{i_1} + \cdots + p_{i_2}) \text{LIF}(p_{i_2} \rightarrow x_{i_2})}$

$\equiv ((p_1 + \cdots + p_{i_1} + p_{i_2}) \oplus (p_{i_1} + \cdots + p_{i_1} + p_{i_2}x_{i_2})) \bar{p}_1 \cdots \bar{p} - \bar{p}_{h_1 - 1} \bar{p} + \bar{p}_{i_2 - 1} \bar{p} + \bar{p}_m$

$\equiv ((p_1 + \cdots + p_{i_1} + p_{i_2} \bar{p}_{i_1} + \cdots + p_{i_1} + p_{i_2}x_{i_2}) \bar{p}_1 \cdots \bar{p} - \bar{p}_{h_1 - 1} \bar{p} + \bar{p}_{i_2 - 1} \bar{p} + \bar{p}_m$

$\equiv ((p_1 + \cdots + p_{i_1} + p_{i_2}) (p_1 + \cdots + p_{i_1} + p_{i_2}x_{i_2}) \bar{p}_1 \cdots \bar{p} - \bar{p}_{h_1 - 1} \bar{p} + \bar{p}_{i_2 - 1} \bar{p} + \bar{p}_m$

$\equiv ((p_1 + \cdots + p_{i_1} + p_{i_2}) \bar{p}_1 \cdots \bar{p} - \bar{p}_{h_1 - 1} \bar{p} + \bar{p}_{i_2 - 1} \bar{p} + \bar{p}_m$

$\equiv S(\vec{t}) \neq I_{\text{ENF}(S = \bar{S}) \text{LOF}(p_{i_2}, \neg p_{i_2, j_2})}(\vec{t})$. 

186
Theorem 9.1.6 (ENF \& LIF - Case 2)

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( p_1 + \cdots + p_{i_2} + \cdots + p_{h_1} \) in \( S \) is negated and the literal \( x_{i_2} \) is inserted in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) where \( 1 \leq i_1 \leq i_2 \leq h_1 \leq m \), \( i_1 \neq h_1 \) and \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the resulting implementation denoted as \( I_{\text{ENF}(p_1 + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_2} x_{i_2})} \) \& \( I_{\text{LIF}(p_{i_2} - p_{i_2} x_{i_2})} \) is equivalent to Expression (8.6) in Table 8.12. Then, \( S \neq I_{\text{ENF}(p_1 + \cdots + p_{i_2} + \cdots + p_{h_1} - p_{i_2} + \cdots + p_{h_1})} \) \& \( I_{\text{LIF}(p_{i_2} - p_{i_2} x_{i_2})} \) if and only if there is a test case \( \bar{\iota} \) that satisfies any of the following conditions:

1. \( \bar{\iota} \in \bigcup_{i=i_1}^{h_1} TP_i(S) \setminus \bigcup_{i=1}^{m} TP_i(S) \) on \( \bar{\iota} \),

2. \( \bar{\iota} \in \bigcup_{i=i_1}^{h_1} TP_i(S) \cap TP_{i_2}(S) \setminus \bigcup_{i=1}^{m} TP_i(S) \) such that \( x_{i_2} = 0 \) on \( \bar{\iota} \), or

3. \( \bar{\iota} \in FP(S) \).

Hence, the result follows. \( \square \)
Proof: First, we observe that $S \oplus I_{ENF}(p_1 + \cdots + p_i + \cdots + p_m) \equiv (p_i + \cdots + p_{i-1} + p_{i+1} + \cdots + p_m)\bar{p}_1 \cdots \bar{p}_{i-1}p_{i+1} \cdots \bar{p}_m$.

\begin{align*}
(\text{By rewriting } (A + B)(AC + B) \text{ as } ACB + B \text{ because they are equivalent; and making use of } (A + B)(AC + B) \equiv (A + B)) \end{align*}

$S(\bar{t}) \equiv I_{ENF}(p_1 + \cdots + p_{i-1} + p_{i+1} + \cdots + p_m)(\bar{t})$

if and only if

$S(\bar{t}) \equiv I_{ENF}(p_1 + \cdots + p_{i-1} + p_{i+1} + \cdots + p_m)(\bar{t}) = 1$

if and only if

$p_{i-1}x_{i-1}p_1 \cdots \bar{p}_i + (p_{i+1} + \cdots + p_{i-1} + p_{i+1} + \cdots + p_m)p_1 \cdots \bar{p}_i + \bar{S}$ evaluates to 1 on $\bar{t}$

if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i-1}(S)$ such that $x_{i-1} = 1$ on $\bar{t}$,

2. $\bar{t} \in \left( \bigcup_{i=1}^{k_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$, or

3. $\bar{t} \in FP(S)$.

Hence, the result follows. \qed

Unlike Case 2 of ENF $\propto$ LOF, we do not need to consider the situation of $m = 1$ in Case 2 of ENF $\propto$ LIF. It is because when $m = 1$, $S = p_1$ and hence there is no missing literal to be inserted into $p_i$.

Theorem 9.1.7 (ENF $\propto$ LRF - Case 1)

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $p_i + \cdots + p_{i-1} - p_{i+1} + \cdots + p_m$ in $S$ is negated and the $j_2$-th literal $x_{j_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by the literal $x_{i_2}$ where $1 \leq i_2 < h_1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2}$ is the number of literals in $p_{i_2}$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting implementation denoted as $I_{ENF}(p_1 + \cdots + p_i + \cdots + p_m) \propto LRF(p_{i_2} - p_{i_2}, x_{i_2})$ is equivalent to Expression (8.7).
in Table 8.12. Then, $S \not\equiv I_{\text{ENF}(p_1+\cdots+p_{h_1}-p_1+\cdots+p_{h_1})} \times \text{LRF}(p_2-p_{12},x_{i_2})$ if and only if there is a test case $\tilde{t}$ that satisfies any of the following conditions:

1. $\tilde{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1,\ldots,h_1}^{m} TP_i(S) \right)$ such that $p_{12,j_2} = 0$ on $\tilde{t}$.

2. $\tilde{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1,\ldots,h_1,i_2}^{m} TP_i(S) \right)$ such that $x_{i_2} = 0$ on $\tilde{t}$, or

3. $\tilde{t} \in FP(S)$.

Proof:

First, we observe that $S \oplus I_{\text{ENF}(p_1+\cdots+p_{h_1}-p_1+\cdots+p_{h_1})} \times \text{LRF}(p_2-p_{12},x_{i_2})$

$\equiv ((p_1+\cdots+p_{h_1}+p_{12}) \oplus (p_1+\cdots+p_{h_1}+p_{12}) \oplus (p_1+\cdots+p_{h_1}+p_{12})) \oplus (p_1+\cdots+p_{h_1}+p_{12})$

$\equiv (p_1+\cdots+p_{h_1}+p_{12})(p_1+\cdots+p_{h_1}+p_{12}) \oplus (p_1+\cdots+p_{h_1}+p_{12})$

$\equiv (p_1+\cdots+p_{h_1}+p_{12})(p_1+\cdots+p_{h_1}+p_{12}) \oplus (p_1+\cdots+p_{h_1}+p_{12})$

$\equiv (p_1+\cdots+p_{h_1})(p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\approx (p_1+\cdots+p_{h_1})(p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1})(p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$

$\equiv (p_1+\cdots+p_{h_1}) \oplus (p_1+\cdots+p_{h_1})$
if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in \bigcup_{i=1}^{m} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, i_{k_2}}^{m} TP_i(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$,

2. $\vec{t} \in \bigcup_{i=1}^{m} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_{k_2}}^{m} TP_i(S)$ such that $x_{i_2} = 0$ on $\vec{t}$, or

3. $\vec{t} \in FP(S)$.

Hence, the result follows. $\square$

Similar to Case 2 of ENF $\not\ltimes$ LIF, we do not need to consider the situation of $m = 1$ in Case 2 of ENF $\not\ltimes$ LRF because when $m = 1$, $S = p_1$ and there is no missing literal to replace any literal in $p_1$.

**Theorem 9.1.8 (ENF $\not\ltimes$ LRF - Case 2)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}}$ in $S$ is negated and the $j_2$-th literal $x_{j_2}$ in the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by the literal $x_{i_2}$ where $1 \leq i_1 \leq i_2 \leq h_{k_2}$, $i_1 \neq h_{k_2}$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2}$ is the number of literals in $p_{i_2}$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting implementation denoted as $I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}} \not\ltimes \text{LRF}(p_{i_2} \not\ltimes p_{i_2,j_2} x_{i_2}))$ is equivalent to Expression (8.8) in Table 8.12.

Then, $S \not\equiv I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}} \not\ltimes \text{LRF}(p_{i_2} \not\ltimes p_{i_2,j_2} x_{i_2}))$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_2}(S)$ such that $x_{i_2} = 1$ on $\vec{t}$,

2. $\vec{t} \in \bigcup_{i=1}^{m} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_{k_2}}^{m} TP_i(S)$,

3. $\vec{t} \in FP(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$, or

4. $\vec{t} \in FP(S)$ such that $x_{i_2} = 0$ on $\vec{t}$.

**Proof:** First, we observe that $S \oplus I_{\text{ENF}}(p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}} \not\ltimes \text{LRF}(p_{i_2} \not\ltimes p_{i_2,j_2} x_{i_2}))$

\[
\equiv ((p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}}) \oplus (p_1 + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_{h_{k_2}})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_{k_1}} \cdots \bar{p}_m \\
\equiv ((p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}})(p_1 + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_{h_{k_2}}) + (p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}})) \\
\quad (\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_{k_1}} \cdots \bar{p}_m) \\
\equiv ((p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}})(p_1 + \cdots + p_{i_2,j_2} x_{i_2} + \cdots + p_{h_{k_2}}) + (p_1 + \cdots + p_{i_2} + \cdots + p_{h_{k_2}})) \\
\quad (\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{h_{k_1}} \cdots \bar{p}_m) \\
\equiv (p_{i_2} x_{i_2} (p_1 + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_{k_2}}) + (p_1 + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_{k_2}}))
\]

190
ab

if there is a test case $\vec{t}$

S

(S

$\equiv p_{i_2}x_{i_2}(p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\bar{p}_1 \cdots \bar{p}_{h_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m$

$+ (p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\bar{p}_1 \cdots \bar{p}_{h_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m$

$+ (p_{i_1} + \cdots + p_{h_1})p_{i_2,j_2}\overline{x_{i_2}}\bar{p}_1 \cdots \bar{p}_{h_1-1} \bar{p}_{h_1+1} \cdots \bar{p}_m$

$\equiv p_{i_2}x_{i_2}\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m$

$+ p_{i_2,j_2}\overline{x_{i_2}} \cdot \overline{S}$

Now, $S(\vec{t}) \neq I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\overline{\text{LRF}}(p_{i_2} - p_{i_2,j_2}x_{i_2})(\vec{t})$

if and only if $S(\vec{t}) \oplus I_{\text{ENF}}(p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\overline{\text{LRF}}(p_{i_2} - p_{i_2,j_2}x_{i_2})(\vec{t}) = 1$

if and only if $p_{i_2}x_{i_2}\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{i_1} + \cdots + p_{i_2-1} + p_{i_2+1} + \cdots + p_{h_1})\bar{p}_1 \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2,j_2} \cdot \overline{S} + \overline{x_{i_2}} \cdot \overline{S}$ evaluates to 1 on $\vec{t}$

if and only if $\vec{t}$ satisfies any of the following conditions:

1. $\vec{t} \in UTP_{i_2}(S)$ such that $x_{i_2} = 1$ on $\vec{t}$,

2. $\vec{t} \in \left( \bigcup_{i \neq i_2}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{h_1} TP_i(S) \right)$,

3. $\vec{t} \in FP(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$, or

4. $\vec{t} \in FP(S)$ such that $x_{i_2} = 0$ on $\vec{t}$.

Hence, the result follows.

\begin{flushright} \Box \end{flushright}

\section{TNF with Other Literal Faults}

In this section, we study the detection conditions of double faults in which one of the single faults is an TNF.

\begin{theorem}[TNF with LNF - Case 1]
Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i_1$-th term, $p_{i_1}$, in $S$ and the $j_2$-th literal $x_{i_2}^{\overline{k_2}}$ in the $i_2$-th term, $p_{i_2}$, in $S$ are negated where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2} > 1$ and $k_{i_2}$ is the number of literals in $p_{i_2}$, the resulting implementation denoted as $I_{\text{TNF}}(p_{i_1} \bar{p}_{i_1})\overline{\text{LNF}}(p_{i_2} - p_{i_2,j_2})$ is equivalent to Expression (8.9) in Table 8.12. Then, $S \neq I_{\text{TNF}}(p_{i_1} \bar{p}_{i_1})\overline{\text{LNF}}(p_{i_2} - p_{i_2,j_2})$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

\end{theorem}
1. \( \bar{t} \in TP_{t_i}(S) \setminus \bigcup_{i=1}^{m} TP_i(S) \) such that \( p_{i_2,j_2} = 0 \), or

2. \( \bar{t} \in FP(S) \).

**Proof:** First, we observe that \( S \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} \)
\( \equiv \left( (p_{i_1} + p_{i_2}) \oplus (\bar{p}_{i_1} + p_{i_2,j_2}) \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
\( \equiv ((p_{i_1} + p_{i_2})(\bar{p}_{i_1} + p_{i_2,j_2}) + (p_{i_1} + p_{i_2})(\bar{p}_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
\( \equiv ((p_{i_1} + p_{i_2})p_{i_2,j_2} + \bar{p}_{i_1} \bar{p}_{i_2}(p_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \)
\( \equiv (p_{i_1} \bar{p}_{i_2,j_2} + \bar{p}_{i_1} \bar{p}_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + S \)

Now, \( S(\bar{t}) \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})}(\bar{t}) \)
if and only if \( S(\bar{t}) \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})}(\bar{t}) = 1 \)
if and only if \( p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + S \) evaluates to 1 on \( \bar{t} \)
if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in TP_{t_i}(S) \setminus \bigcup_{i=1}^{m} TP_i(S) \) such that \( p_{i_2,j_2} = 0 \), or

2. \( \bar{t} \in FP(S) \).

Hence, the result follows. \( \square \)

**Theorem 9.1.10 (TNF \( \otimes \) LNF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) and the \( j_2 \)-th literal \( x_{j_2}^{1_i} \) in \( p_{i_1} \) are negated where \( 1 \leq i_1 \leq m, 1 \leq j_2 \leq k_{i_1} \), \( k_{i_1} > 1 \) and \( k_{i_1} \) is the number of literals in \( p_{i_1} \), the resulting implementation denoted as \( I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_1} \rightarrow \bar{p}_{i_1,j_2})} \) is equivalent to Expression (8.10) in Table 8.12. Then, \( S \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_1} \rightarrow \bar{p}_{i_1,j_2})} \) if and only if there is a test case \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_2} = 0 \) on \( \bar{t} \).

**Proof:** First, we observe that \( S \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes LNF(p_{i_1} \rightarrow \bar{p}_{i_1,j_2})} \)
\( \equiv (p_{i_1} + \bar{p}_{i_1,j_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
\( \equiv (p_{i_1} p_{i_2,j_2} + \bar{p}_{i_1} \bar{p}_{i_2}(p_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
\( \equiv (0 + \bar{p}_{i_1} \bar{p}_{i_2,j_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
\( \text{(By making use of } AB(AB) \equiv 0 \text{ and } (AB)(AB) \equiv (AB)A) \)
\( \equiv \bar{p}_{i_1,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)

192
≡ \bar{p}_{t_1,j_2} \overline{S}

Now,\quad S(\bar{t}) \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} (\bar{t})

if and only if\quad S(\bar{t}) \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} (\bar{t}) = 1

if and only if\quad \bar{p}_{t_1,j_2} \overline{S} evaluates to 1 on \bar{t}

if and only if\quad \bar{t} \in FP(S) such that \bar{p}_{t_1,j_2} = 0 on \bar{t}.

Hence, the result follows. □

Theorem 9.1.11 (TNF with LOF - Case 1)
Let \( S=p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), is negated and the \( j_2 \)-th literal \( x_{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 \leq i_1 < i_2 \leq m \), \( 1 \leq j_2 \leq k_{i_2} \), \( k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \). The resulting implementation denoted as \( I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} \) is equivalent to Expression (8.11) in Table 8.12. Then, \( S \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{t_1}(S) \) such that \( p_{t_2,j_2} = 0 \) on \( \bar{t} \), or

2. \( \bar{t} \in FP(S) \).

Proof: First, we observe that \( S \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} \)
\[ \equiv ((p_{i_1} + p_{i_2}) \oplus (\bar{p}_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv ((p_{i_1} + p_{i_2})(p_{i_1} + p_{i_2,j_2}) + (\bar{p}_{i_1} + p_{i_2})(\bar{p}_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv ((p_{i_1} + p_{i_2})p_{i_1} \bar{p}_{i_2,j_2} + \bar{p}_{i_1} p_{i_2} + (\bar{p}_{i_1} + p_{i_2,j_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[ \equiv (p_{i_1} \bar{p}_{i_2,j_2} + \bar{p}_{i_1} p_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By making use of \( (A + B) \bar{A} \equiv A \) and \( A(A + B) \equiv A \))
\[ \equiv (p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By rewriting \( \bar{A} \) as \( \overline{AB} \) because they are equivalent)
\[ \equiv p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + \overline{S} \]

Now,\quad S(\bar{t}) \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} (\bar{t})

if and only if\quad S(\bar{t}) \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1})} \otimes_{LOF(p_{i_2} \rightarrow \bar{p}_{i_2,j_2})} (\bar{t}) = 1

if and only if\quad p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + \overline{S} \) evaluates to 1 on \( \bar{t} \).

if and only if\quad \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{t_1}(S) \) such that \( p_{t_2,j_2} = 0 \) on \( \bar{t} \), or

2. \( \bar{t} \in FP(S) \).

Hence, the result follows. □
Theorem 9.1.12 (TNF \& LOF - Case 2)
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the \( j_2 \)-th literal \( x_{j_2}^{i_1} \) in \( p_{i_1} \) is omitted where \( 1 \leq i_1 \leq m \), \( 1 \leq j_2 \leq k_{i_1} \), \( k_{i_1} > 1 \) and \( k_{i_1} \) is the number of literals in \( p_{i_1} \), the resulting implementation denoted as \( I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LOF(p_{i_1} \rightarrow \overline{p}_{i_1}, j_2)} \) is equivalent to Expression (8.12) in Table 8.12. Then, \( S \not\equiv I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LOF(p_{i_1} \rightarrow \overline{p}_{i_1}, j_2)} \) if and only if there is a test case \( \overline{t} \) that satisfies any of the following conditions:

1. \( \overline{t} \in UTP_{i_1}(S) \), or
2. \( \overline{t} \in FP(S) \) such that \( p_{i_1, j_2} = 0 \) on \( \overline{t} \).

Proof: First, we observe that \( S \equiv I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LOF(p_{i_1} \rightarrow \overline{p}_{i_1}, j_2)} \)

\[
\equiv (p_{i_1} \oplus \overline{p}_{i_1}, j_2)\overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1+1} \cdots \overline{p}_m \\
\equiv (p_{i_1}p_{i_1, j_2} + \overline{p}_{i_1} \overline{p}_{i_1, j_2})\overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1+1} \cdots \overline{p}_m \\
\equiv p_{i_1} \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1+1} \cdots \overline{p}_m + \overline{p}_{i_1, j_2} S
\]

Now,
\[
S(\overline{t}) \neq I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LOF(p_{i_1} \rightarrow \overline{p}_{i_1}, j_2)}(\overline{t})
\]
if and only if \( S(\overline{t}) \oplus I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LOF(p_{i_1} \rightarrow \overline{p}_{i_1}, j_2)}(\overline{t}) = 1 \)
if and only if \( p_{i_1} \overline{p}_{i_1} \cdots \overline{p}_{i_1-1} \overline{p}_{i_1+1} \cdots \overline{p}_m + \overline{p}_{i_1, j_2} S \) evaluates to 1 on \( \overline{t} \).
if and only if \( \overline{t} \) satisfies any of the following conditions:

1. \( \overline{t} \in UTP_{i_1}(S) \), or
2. \( \overline{t} \in FP(S) \) such that \( p_{i_1, j_2} = 0 \) on \( \overline{t} \).

Hence, the result follows.

Theorem 9.1.13 (TNF with LIF - Case 1)
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the literal \( x_{i_2} \) is inserted into the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) where \( 1 \leq i_1 < i_2 \leq m \) and \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the resulting implementation denoted as \( I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LIF(p_{i_2} \rightarrow p_{i_2}, x_{i_2})} \) is equivalent to Expression (8.13) in Table 8.12. Then, \( S \not\equiv I_{TNF(p_{i_1} \rightarrow \overline{p}_{i_1}) \& LIF(p_{i_2} \rightarrow p_{i_2}, x_{i_2})} \) if and only if there is a test case \( \overline{t} \) that satisfies any of the following conditions:

1. \( \overline{t} \in UTP_{i_1}(S) \),
2. \( \overline{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \backslash \left( \bigcup_{i=1}^{m} TP_i(S) \right) \) such that \( x_{i_2} = 0 \) on \( \overline{t} \), or
3. \( \overline{t} \in FP(S) \).

194
Proof: First, we observe that

\[ S \oplus I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_2 \rightarrow \bar{p}_2 x_{i_2})} \]

\[ \equiv (p_1 + p_2) \oplus (\bar{p}_1 + p_2 x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]

\[ \equiv (p_1 + p_2)(\bar{p}_1 + p_2 x_{i_2} + (\bar{p}_1 + p_2 x_{i_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]

\[ \equiv (p_1 + p_2)(\bar{p}_1 x_{i_2} + \bar{p}_1 p_2 (\bar{p}_1 + p_2 x_{i_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]

\[ \equiv (p_1 \bar{p}_2 x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]

(By making use of \((A + B)A \equiv A\) and \(A(A + B) \equiv A\))

\[ \equiv p_1 \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_1 p_2 \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + S \]

Now, \(S(\vec{t}) \neq I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_2 \rightarrow \bar{p}_2 x_{i_2})}(\vec{t})\)

if and only if \(S(\vec{t}) \oplus I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_2 \rightarrow \bar{p}_2 x_{i_2})}(\vec{t}) = 1\)

if and only if \(p_1 \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + p_1 p_2 \bar{x}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m + S\) evaluates to 1 on \(\vec{t}\)

if and only if \(\vec{t}\) satisfies any of the following conditions:

1. \(\vec{t} \in UTP_{i_1}(S)\),

2. \(\vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right)\) such that \(x_{i_2} = 0\) on \(\vec{t}\), or

3. \(\vec{t} \in FP(S)\).

Hence, the result follows. \(\square\)

Theorem 9.1.14 (\(\text{TNF} \& \text{LIF} - \text{Case 2}\))

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term, \(p_{i_1}\), in \(S\) is negated and the literal \(x_{i_2}\) is inserted in \(p_{i_1}\) where \(1 \leq i_1 \leq m\) and \(x_{i_2}\) is a missing literal of \(p_{i_1}\), the resulting implementation denoted as \(I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_1 \rightarrow \bar{p}_1 x_{i_2})}\) is equivalent to Expression (8.14) in Table 8.12. Then, \(S \neq I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_1 \rightarrow \bar{p}_1 x_{i_2})}\)

if and only if there is a test case \(\vec{t}\) that satisfies any of the following conditions:

1. \(\vec{t} \in UTP_{i_1}(S)\) such that \(x_{i_2} = 1\) on \(\vec{t}\), or

2. \(\vec{t} \in FP(S)\).

Proof: First, we observe that \(S \oplus I_{\text{TNF}(p_1 \rightarrow \bar{p}_1) \& \text{LIF}(p_1 \rightarrow \bar{p}_1 x_{i_2})}\)

\[ \equiv (p_1 \oplus \bar{p}_1 x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \]
≡ (p_{i_1}p_{i_2}x_{t_2} + \bar{p}_{i_1}(\bar{p}_{i_1} + \bar{x}_{t_2}))\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\equiv (p_{i_1}x_{t_2}\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\quad \text{(By making use of } A(A + B) \equiv A)
\equiv p_{i_1}x_{t_2}\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m + \bar{S}

Now,
S(\bar{t}) \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes \text{LIF}(p_{i_1} \rightarrow p_{i_2}x_{t_2})}(\bar{t}) \text{ if and only if }
S(\bar{t}) \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes \text{LIF}(p_{i_1} \rightarrow p_{i_2}x_{t_2})}(\bar{t}) = 1
\text{ if and only if } p_{i_1}x_{t_2}\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m + \bar{S} \text{ evaluates to 1 on } \bar{t}
\text{ if and only if } \bar{t} \text{ satisfies any of the following conditions:}

1. \bar{t} \in \text{UTP}_{i_1}(S) \text{ such that } x_{t_2} = 1 \text{ on } \bar{t}, \text{ or}
2. \bar{t} \in \text{FP}(S).

Hence, the result follows. \square

**Theorem 9.1.15 (TNF with LRF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated and the \( j_2 \)-th literal \( x_{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is replaced by the literal \( x_{t_2} \) where \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_2 \leq k_{i_2}, k_{i_2} \) is the number of literals in \( p_{i_2} \) and \( x_{t_2} \) is a missing literal of \( p_{i_2} \), the resulting implementation denoted as \( I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2}x_{j_2})} \) is equivalent to Expression (8.15) in Table 8.12.

Then, \( S \neq I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2}x_{j_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in \text{UTP}_{i_1}(S) \text{ such that } x_{t_2} = 0 \text{ on } \bar{t}, \)
2. \( \bar{t} \in \text{TP}_{i_1}(S) \setminus \left( \bigcup_{i = 1}^{m} \text{TP}_{i}(S) \right) \text{ such that } x_{t_2} = 0 \text{ on } \bar{t}, \text{ or} \)
3. \( \bar{t} \in \text{FP}(S). \)

**Proof:** First, we observe that \( S \oplus I_{TNF(p_{i_1} \rightarrow \bar{p}_{i_1}) \otimes \text{LRF}(p_{i_2} \rightarrow p_{i_2,j_2}x_{j_2})} \)
\equiv \left( (p_{i_1} + p_{i_2}) \oplus (\bar{p}_{i_1} + p_{i_2,j_2}x_{j_2}) \right)\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\equiv \left( (p_{i_1} + p_{i_2})\bar{p}_{i_1} + p_{i_2,j_2}x_{j_2} + (p_{i_1} + p_{i_2,j_2}x_{j_2})\bar{p}_{i_1} \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\equiv \left( (p_{i_1} + p_{i_2})\bar{p}_{i_1} + p_{i_2,j_2}x_{j_2} + \bar{p}_{i_1}p_{i_2,j_2}x_{j_2} + \bar{p}_{i_1}p_{i_2} + p_{i_2,j_2}x_{j_2}) \right)\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\equiv \left( p_{i_1}p_{i_2,j_2}x_{j_2} + \bar{p}_{i_1}p_{i_2} \right)\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\quad \text{(By making use of } (A + B)A \equiv A \text{ and } A(A + B) \equiv A)
\equiv p_{i_1}(\bar{p}_{i_2,j_2}\bar{p}_{i_2} + \bar{x}_{j_2})\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\quad + \bar{p}_{i_1}p_{i_2}\bar{p}_1 \cdots \bar{p}_{i_1 - 1}\bar{p}_{i_1 + 1} \cdots \bar{p}_m
\quad \text{(By rewriting } (AC) \text{ as } (A)(\overline{AB}) + \overline{C} \text{ because they are equivalent)}
\[
\equiv \bar{p}_1 \bar{p}_2 \bar{p}_3 \cdots \bar{p}_{i_1} - 1 \bar{p}_{i_1} + \cdots \bar{p}_m + p_1 \bar{x}_1 \bar{p}_1 \cdots \bar{p}_{i_1} - 1 \bar{p}_{i_1} + \cdots \bar{p}_{i_2} - 1 \bar{p}_{i_2} + \cdots \bar{p}_m + \bar{S}
\]

Now, \[S(\bar{t}) \neq I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2}) \]
if and only if \[S(\bar{t}) \oplus I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2}) = 1 \]
if and only if \[p_1 \bar{p}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1} - 1 \bar{p}_{i_1} + \cdots \bar{p}_{i_2} - 1 \bar{p}_{i_2} + \cdots \bar{p}_m + \bar{S} \]
evaluates to 1 on \(\bar{t}\).

if and only if \(\bar{t}\) satisfies any of the following conditions:

1. \(\bar{t} \in UTP_{i_1}(S)\) such that \(p_{i_2,j_2} = 0\) on \(\bar{t}\),
2. \(\bar{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i = 1}^{m} TP_x(S) \right)\) such that \(x_{i_2} = 0\) on \(\bar{t}\), or
3. \(\bar{t} \in FP(S)\).

Hence, the result follows.

\textbf{Theorem 9.1.16 (TNF \texttimes LRF - Case 2)}

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term, \(p_{i_1}\), in \(S\) is negated and the \(j_2\)-th literal \(x_{i_2}^{i_1}\) in \(p_{i_1}\) is replaced by the literal \(x_{i_2}\) where \(1 \leq i \leq m, 1 \leq j_2 \leq k_{i_1}, k_{i_1}\) is the number of literals in \(p_{i_1}\) and \(x_{i_2}\) is a missing literal of \(p_{i_1}\).

(a) When \(k_{i_1} > 1\), the resulting implementation denoted as \(I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2})\)

is equivalent to Expression (8.16) in Table 8.12. Then, we have \(S \neq I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2})\)

if and only if there is a test case \(\bar{t}\) that satisfies any of the following conditions:

1. \(\bar{t} \in UTP_{i_1}(S)\) such that \(x_{i_2} = 1\) on \(\bar{t}\),
2. \(\bar{t} \in FP(S)\) such that \(p_{i_1,j_2} = 0\) on \(\bar{t}\), or
3. \(\bar{t} \in FP(S)\) such that \(x_{i_2} = 0\) on \(\bar{t}\).

(b) When \(k_{i_1} = 1\), the resulting implementation denoted as \(I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2})\)

is equivalent to that given by Expression (8.16) in Table 8.12 without \(p_{i_1,j_2}\) because \(p_{i_1}\) contains only one literal. Then, we have \(S \neq I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2})\)

if and only if there is a test case \(\bar{t}\) that satisfies any of the following conditions:

1. \(\bar{t} \in UTP_{i_1}(S)\) such that \(x_{i_2} = 1\) on \(\bar{t}\),
2. \(\bar{t} \in FP(S)\) such that \(x_{i_2} = 0\) on \(\bar{t}\).

\textbf{Proof :} (a) First, we observe that \(S \oplus I_{\text{TNF}}(p_{i_1} \rightarrow \bar{p}_{i_1}, x_{i_2})\)

\(\equiv (p_{i_1} \oplus \bar{p}_{i_1,j_2} x_{i_2}) \bar{p}_1 \cdots \bar{p}_{i_1} - 1 \bar{p}_{i_1} + \cdots \bar{p}_m\)
bearing in mind that the detection condition 2 of Theorem 9.1.16(a) does not exist.

First, we observe that

\[
\begin{align*}
& p_i x_{l_2} \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m \\
& \equiv (p_i x_{l_2} + \tilde{p}_i \overline{x_{l_2}}) \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m
\end{align*}
\]

Now, \( S(\tilde{t}) \neq I_{\text{TNF}(p_i \rightarrow \tilde{p}_i)} \ast L_{\text{LRF}(p_i \rightarrow x_{l_2})}(\tilde{t}) \)

if and only if \( S(\tilde{t}) \oplus I_{\text{TNF}(p_i \rightarrow \tilde{p}_i)} \ast L_{\text{LRF}(p_i \rightarrow x_{l_2})}(\tilde{t}) = 1 \)

if and only if \( p_i x_{l_2} \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m + \tilde{p}_i \overline{x_{l_2}} S + \overline{x_{l_2}} \overline{S} \) evaluates to 1 on \( \tilde{t} \)

if and only if \( \tilde{t} \) satisfies any of the following conditions:

1. \( \tilde{t} \in U_{\text{TP}_{i_2}}(S) \) such that \( x_{l_2} = 1 \) on \( \tilde{t} \),

2. \( \tilde{t} \in F_P(S) \) such that \( p_{i_1, j_2} = 0 \) on \( \tilde{t} \), or

3. \( \tilde{t} \in F_P(S) \) such that \( x_{l_2} = 0 \) on \( \tilde{t} \).

Hence, the result follows.

(b) The proof of this part is very similar to (a) except that the term \( p_{i_1, j_2} \) does not appear in the proof. We proceed the proof as follows.

First, we observe that \( S \oplus I_{\text{TNF}(p_i \rightarrow \tilde{p}_i)} \ast L_{\text{LRF}(p_i \rightarrow x_{l_2})} \)

\[
\begin{align*}
& \equiv (p_i \oplus x_{l_2}) \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m \\
& \equiv (p_i x_{l_2} + \tilde{p}_i x_{l_2}) \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m \\
& \equiv p_i x_{l_2} \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m + x_{l_2} S
\end{align*}
\]

Now, \( S(\tilde{t}) \neq I_{\text{TNF}(p_i \rightarrow \tilde{p}_i)} \ast L_{\text{LRF}(p_i \rightarrow x_{l_2})}(\tilde{t}) \)

if and only if \( S(\tilde{t}) \oplus I_{\text{TNF}(p_i \rightarrow \tilde{p}_i)} \ast L_{\text{LRF}(p_i \rightarrow x_{l_2})}(\tilde{t}) = 1 \)

if and only if \( p_i x_{l_2} \tilde{p}_1 \cdots \tilde{p}_{i-1} \tilde{p}_{i+1} \cdots \tilde{p}_m + \tilde{p}_i \overline{x_{l_2}} S \) evaluates to 1 on \( \tilde{t} \)

if and only if \( \tilde{t} \) satisfies any of the following conditions:

1. \( \tilde{t} \in U_{\text{TP}_{i_2}}(S) \) such that \( x_{l_2} = 1 \) on \( \tilde{t} \),

2. \( \tilde{t} \in F_P(S) \) such that \( x_{l_2} = 0 \) on \( \tilde{t} \).

Hence, the result follows. \( \square \)

It should be noted that the only difference between the detection conditions of Theorem 9.1.16(a) and (b) lies in the detection condition 2 of Theorem 9.1.16(a). In fact, by just looking at the syntax, when \( k_{i_1} = 1 \), \( p_{i_1, j_2} \) does not exist. Hence, without loss of generality, we can still use detection conditions 1, 2 and 3 in Theorem 9.1.16(a) to represent the detection conditions of Expression (8.16) in Table 8.12 for \( k_{i_1} \geq 1 \), bearing in mind that the detection condition 2 of Theorem 9.1.16(a) does not exist when \( k_{i_1} = 1 \).
9.1.3 TOF with Other Literal Faults

In this section, we study the detection conditions of double faults in which one of the single fault is an TOF.

Theorem 9.1.17 (TOF with LNF)

Let \( S=p_1 + \cdots + p_m \) be a Boolean specification in \( IDNF \). Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted and the \( j_2 \)-th literal \( x_{j_2}^{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is negated where \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_2 \leq k_{i_2} \), \( k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the implementation denoted as \( I_{TOF(p_{i_1} \rightarrow LNF(p_{i_2} \rightarrow p_{j_2}))} \) is equivalent to Expression (8.17) in Table 8.12. Then, \( S \neq I_{TOF(p_{i_1} \rightarrow LNF(p_{i_2} \rightarrow p_{j_2}))} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in TP_{i_1}(S) \backslash \left( \bigcup_{i=1 \atop i \neq i_1, j_2}^{m} TP_i(S) \right) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \),

2. \( \vec{t} \in TP_{i_2}(S) \backslash \left( \bigcup_{i=1 \atop i \neq i_1, j_2}^{m} TP_i(S) \right) \), or

3. \( \vec{t} \in NFP_{i_2,j_2}(S) \).

Proof : First, we observe that

\[
S \oplus I_{TOF(p_{i_1} \rightarrow LNF(p_{i_2} \rightarrow p_{j_2}))} \\
≡ \left( (p_{i_1} + p_{i_2}) \oplus p_{i_2,j_2} \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
≡ \left( (p_{i_1} + p_{i_2}) \bar{p}_{i_2,j_2} + \overline{(p_{i_1} + p_{i_2})p_{i_2,j_2}} \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
≡ \left( p_{i_1} \bar{p}_{i_2,j_2} + p_{i_2} + \bar{p}_1 \bar{p}_2 p_{i_2,j_2} \right) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
\text{(By making use of (AB)(\overline{AB}) \equiv (AB))} \\
≡ p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
+ p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
≡ p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
+ p_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2,j_2} \overline{S}
\]

199
Now, \( S(\vec{t}) \neq I_{TOF(p_{i_1} \rightarrow ) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})}(\vec{t}) \)
if and only if
\begin{align*}
S(\vec{t}) \oplus I_{TOF(p_{i_1} \rightarrow ) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})}(\vec{t}) &= 1 \\
\text{if and only if } \\
p_{i_1} \bar{p}_{i_2} \bar{p}_1 \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
+ \\
p_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \cdots \bar{p}_{i_2-1} \\
\bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2,j_2} S \text{ evaluates to 1 on } \vec{t}
\end{align*}
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i_{1,i_2} \neq 1} TP_i(S) \right) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \),

2. \( \vec{t} \in TP_{i_2}(S) \setminus \left( \bigcup_{i_{1,i_2} \neq 1} TP_i(S) \right) \), or

3. \( \vec{t} \in NFP_{i_2,j_2}(S) \).

Hence, the result follows. \( \square \)

**Theorem 9.1.18 (TOF with LOF)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted and the \( j_2 \)-th literal \( x_{i_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 \leq i_1 < i_2 \leq m \), \( 1 \leq j_2 \leq k_{i_2} \), \( k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the implementation denoted as \( I_{TOF(p_{i_1} \rightarrow ) \times LOF(p_{i_2} \rightarrow -p_{i_2,j_2})} \) is equivalent to Expression (8.18) in Table 8.12. Then, \( S \neq I_{TOF(p_{i_1} \rightarrow ) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \), or

2. \( \vec{t} \in NFP_{i_2,j_2}(S) \).

**Proof:** First, we observe that \( S \oplus I_{TOF(p_{i_1} \rightarrow ) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})} \)
\begin{align*}
&= (p_{i_1} + p_{i_2} \oplus p_{i_2,j_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
&= (p_{i_1} + p_{i_2}) \bar{p}_{i_2,j_2} + (\bar{p}_{i_1} + p_{i_2,j_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
&= (p_{i_1} \bar{p}_{i_2,j_2} + 0 + \bar{p}_{i_1} p_{i_2,j_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
& \quad \text{(By making use of } AB(A) = 0) \\
&= (p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
& \quad \text{(By rewriting } A \text{ as } (A)(AB) \text{ because they are equivalent, and } (AB) A \text{ as } (AB)(AB) \text{ because they are equivalent)} \\
&= p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
& \quad + p_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \\
&= p_{i_1} \bar{p}_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \bar{p}_m + p_{i_2,j_2} S
\end{align*}
Now, \[ S(\bar{i}) \neq I_{TOF(p_1 \rightarrow \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \·
Now, \( S(\vec{t}) \neq I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} (\vec{t}) \)

if and only if

\[ S(\vec{t}) \oplus I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} (\vec{t}) = 1 \]

if and only if

\[ p_i \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1}+1 \cdots \vec{p}_m + p_i \vec{x}_{i_2} \vec{p}_1 \cdots \vec{p}_{i_2-1} \vec{p}_{i_2}+1 \cdots \vec{p}_m \]

\[ + p_i p_{i_2} \vec{x}_{i_2} \vec{p}_1 \cdots \vec{p}_{i_1-1} \vec{p}_{i_1}+1 \cdots \vec{p}_{i_2-1} \vec{p}_{i_2}+1 \cdots \vec{p}_m \]

evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \),
2. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \), or
3. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

**Theorem 9.1.20 (TOF with LRF)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted and the \( j_2 \)-th literal \( x_{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is replaced by the literal \( x_{i_2} \) where \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_2 \leq k_{i_2}, k_{i_2} \) is the number of literals in \( p_{i_2} \) and \( x_{i_2} \) is a missing literal of \( p_{i_2} \).

(a) When \( k_{i_2} > 1 \), the resulting implementation denoted as \( I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} \)

is equivalent to Expression (8.20) in Table 8.12. Then, \( S \neq I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \),
2. \( \vec{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} = 0 \) on \( \vec{t} \),
3. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \),
4. \( \vec{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \) on \( \vec{t} \), or
5. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).

(b) When \( k_{i_2} = 1 \), the resulting implementation denoted as \( I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} \)

is equivalent to Expression (8.20) in Table 8.12 without \( p_{i_2,j_2} \) because \( p_{i_2} \) contains just one literal. Then, \( S \neq I_{TOF(p_1 \rightarrow \LRF(p_{i_2} \rightarrow x_{i_2})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \),
2. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).
3. \( \vec{t} \in FP(S) \) such that \( x_{i_2} = 1 \) on \( \vec{t} \), or

4. \( \vec{t} \in \left( TP_{i_1}(S) \cap TP_{i_2}(S) \right) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).

**Proof:** (a) First, we observe that \( S \oplus I_{\text{TOF}(p_1 \to \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \
Now, \( S(\tilde{t}) \neq I_{TOF(p_{i_1} \rightarrow ) \times LRF(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2})}(\tilde{t}) \)
if and only if
\[ S(\tilde{t}) \oplus I_{TOF(p_{i_1} \rightarrow ) \times LRF(p_{i_2} \rightarrow p_{i_2,j_2} x_{i_2})}(\tilde{t}) = 1 \]
if and only if
\[ p_{i_1} x_{i_2} \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m + p_{i_1} \bar{p}_{i_2} \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ + p_{i_1} \bar{p}_{i_2} \bar{x}_{i_2} \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
evaluates to 1 on \( \tilde{t} \)
if and only if \( \tilde{t} \) satisfies any of the following conditions:
1. \( \tilde{t} \in UTP_{t_1}(S) \) such that \( x_{i_2} = 0 \) on \( \tilde{t} \),
2. \( \tilde{t} \in UTP_{t_1}(S) \) such that \( p_{i_2,j_2} = 0 \) on \( \tilde{t} \),
3. \( \tilde{t} \in UTP_{t_2}(S) \) such that \( x_{i_2} = 0 \) on \( \tilde{t} \),
4. \( \tilde{t} \in NFP_{t_2,j_2}(S) \) such that \( x_{i_2} = 1 \) on \( \tilde{t} \), or
5. \( \tilde{t} \in (TP_{t_1}(S) \cap TP_{t_2}(S)) \setminus \left( \bigcup_{i \neq i_1,j_2} TP_i(S) \right) \) such that
\( x_{i_2} = 0 \) on \( \tilde{t} \).

Hence, the result follows.

(b) The proof of this part is similar to (a) above except that the term \( p_{i_2,j_2} \) does not appear in the proof. We proceed the proof as follows.

First, we observe that
\[ S \oplus I_{TOF(p_{i_1} \rightarrow ) \times LRF(p_{i_2} \rightarrow x_{i_2})} \]
\[ \equiv ((p_{i_1} + p_{i_2}) \oplus x_{i_2}) \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ \equiv ((p_{i_1} + p_{i_2}) \bar{x}_{i_2} + (p_{i_1} + p_{i_2}) x_{i_2}) \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ \equiv ((p_{i_1} \bar{p}_{i_2} + \bar{p}_{i_2} p_{i_2} + p_{i_1} p_{i_2}) \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2} x_{i_2}) \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ (\text{By rewriting } A + B \text{ as } AB + AB + AB \text{ because they are equivalent}) \]
\[ \equiv (p_{i_1} \bar{p}_{i_2} \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2} \bar{x}_{i_2} + p_{i_1} p_{i_2} \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2} x_{i_2}) \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ \equiv (p_{i_1} \bar{p}_{i_2} \bar{x}_{i_2} + \bar{p}_{i_1} p_{i_2} \bar{x}_{i_2} + \bar{p}_{i_1} \bar{p}_{i_2} x_{i_2} + p_{i_1} p_{i_2} \bar{x}_{i_2}) \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ \equiv p_{i_1} \bar{x}_{i_2} \bar{p}_1 + \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m + p_{i_1} \bar{x}_{i_2} \bar{p}_1 + \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
\[ + p_{i_1} \bar{p}_{i_2} \bar{x}_{i_2} \bar{p}_1 \ldots \bar{p}_{i_1-1,1} \bar{p}_{i_1+1} \bar{p}_{i_2-1,1} \bar{p}_{i_2+1} \ldots \bar{p}_m \]
Now, \( S(\vec{t}) \neq I_{TOF(p_{i_1} \rightarrow ) \cap LRF(p_{i_2} \rightarrow x_{l_{2}})}(\vec{t}) \)
if and only if
\( S(\vec{t}) \oplus I_{TOF(p_{i_1} \rightarrow ) \cap LRF(p_{i_2} \rightarrow x_{l_{2}})}(\vec{t}) = 1 \)
if and only if
\[
p_{i_1} x_{l_{2}} \vec{p} \cdots \vec{p}_{l_{1}+1} \cdots \vec{p}_{m} + p_{i_2} \vec{x}_{l_{2}} \vec{p} \cdots \vec{p}_{l_{2}-1} \vec{p}_{l_{2}+1} \cdots \vec{p}_{m} + \]
\[
x_{l_{2}} \vec{p} + p_{i_1} p_{i_2} \vec{x}_{l_{2}} \vec{p} \cdots \vec{p}_{l_{1}-1} \vec{p}_{l_{1}+1} \cdots \vec{p}_{m} + \]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{l_{1}}(S) \) such that \( x_{l_{2}} = 0 \) on \( \vec{t} \),

2. \( \vec{t} \in UTP_{l_{2}}(S) \) such that \( x_{l_{2}} = 0 \) on \( \vec{t} \),

3. \( \vec{t} \in FP(S) \) such that \( x_{l_{2}} = 1 \) on \( \vec{t} \), or

4. \( \vec{t} \in (TP_{l_{1}}(S) \cap TP_{l_{2}}(S)) \setminus (\bigcup_{i=1}^{m} TP_{i}(S)) \) such that \( x_{l_{2}} = 0 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

It should be noted that there are two differences between the detection conditions of Theorem 9.1.20(a) and (b). First, the detection condition 2 of Theorem 9.1.20(a) is related to the term \( p_{i_2,j_2} \) which does not exist when \( k_{i_2} = 1 \) in Theorem 9.1.20(b).

Second, the detection condition 4 of Theorem 9.1.20(a) (that is, “\( \vec{t} \in NFP_{i_2,j_2}(S) \)” such that \( x_{l_{2}} = 1 \)” differs from the detection condition 3 of Theorem 9.1.20(b) (that is, “\( \vec{t} \in FP(S) \)” such that \( x_{l_{2}} = 1 \)” syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when \( k_{i_2} = 1 \) (that is, when \( p_{i_2} \) contains just one literal). It is because of the following reasons

\[
\vec{t} \in FP(S) \text{ such that } x_{l_{2}} = 0
\equiv \vec{t} \in FP(S) \text{ such that } p_{i_2} = x_{l_{2}}^{i_2} = 0 \text{ and } x_{l_{2}} = 1
\equiv \vec{t} \in FP(S) \text{ such that } p_{i_2,\bar{1}} = \bar{x}_{l_{2}}^{i_2} = 1 \text{ and } x_{l_{2}} = 1
\equiv \vec{t} \in NFP_{i_2,\bar{1}}(S) \text{ such that } x_{l_{2}} = 1
\]

(Please be noted that \( j_2 = 1 \) when \( k_{i_2} = 1 \).)

Hence, without loss of generality, we can still use the five detection conditions in Theorem 9.1.20(a) to represent the detection conditions of Expression (8.20) in Table 8.12 for \( k_{i_2} \geq 1 \), bearing in mind the non-existence of the term \( p_{i_2,j_2} \) and the equivalence between “\( \vec{t} \in FP(S) \)” such that \( x_{l_{2}} = 1 \)” and “\( \vec{t} \in NFP_{i_2,j_2}(S) \)” such that \( x_{l_{2}} = 1 \)” when \( k_{i_2} = 1 \).
9.1.4 DORF with Other Literal Faults

In this section, we study the detection conditions of double faults in which one of the single fault is an DORF.

**Theorem 9.1.21 (DORF \( \bowtie \) LNF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \((p_i + p_{i+1})\) in \( S \) is implemented as \( p_ip_{i+1} \), and the \( j_2 \)-th literal, \( x_{j_2}^p \), of the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is negated where \( 1 < i_1 + 1 < i_2 \leq m \), \( 1 \leq j_2 \leq k_{i_2} \) and \( k_{i_2} > 1 \) if \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the resulting implementation denoted as \( I_{DORF(p_1 + p_{i+1} \rightarrow p_1 p_{i+1}) \bowtie LNF(p_2 \rightarrow p_{i_2})} \) is equivalent to Expression (8.21) in Table 8.12. Then, \( S \not\equiv I_{DORF(p_1 + p_{i+1} \rightarrow p_1 p_{i+1}) \bowtie LNF(p_2 \rightarrow p_{i_2})} \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_{i}(S) \right) \) such that \( p_{i_2 \vec{t}} = 0 \) on \( \vec{t} \),

2. \( \vec{t} \in TP_{i_1+1}(S) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_{i}(S) \right) \) such that \( p_{i_2 \vec{t}} = 0 \) on \( \vec{t} \),

3. \( \vec{t} \in UTP_{i_2}(S) \) on \( \vec{t} \),

4. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_{i}(S) \right) \) on \( \vec{t} \),

5. \( \vec{t} \in (TP_{i_1+1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_{i}(S) \right) \) on \( \vec{t} \), or

6. \( \vec{t} \in NFP_{i_2 \vec{t}_2}(S) \).

**Proof:** First, we observe that \( S \oplus I_{DORF(p_1 + p_{i+1} \rightarrow p_1 p_{i+1}) \bowtie LNF(p_2 \rightarrow p_{i_2})} \)

\[
\equiv ( (p_i + p_{i_1+1} + p_{i_2}) \oplus (p_ip_{i+1} + p_{i_2})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv ( (p_i + p_{i_1+1} + p_{i_2})p_ip_{i+1} + p_{i_2}) + (p_i + p_{i_1+1} + p_{i_2})(p_ip_{i+1} + p_{i_2})
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[
\equiv (p_i \bar{p}_{i_1+1} \bar{p}_{i_2} + p_{i_1+1} \bar{p}_i \bar{p}_{i_2} + p_{i_2} \bar{p}_i \bar{p}_{i_1+1} + 0 + \bar{p}_i \bar{p}_{i_1+1} \bar{p}_i \bar{p}_{i_2})
\]

\[
\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

(\text{By making use } A(\overline{AB}) \equiv \overline{AB}, \overline{AB(\overline{AB})} \equiv AB \text{ and } (A)A = \equiv 0)

\[
\equiv p_i \bar{p}_{i_1+1} \bar{p}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[+ p_{i_1+1} \bar{p}_i \bar{p}_{i_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\[+ p_{i_2} (p_i \bar{p}_{i_1+1} + p_{i_1+1} \bar{p}_i \bar{p}_{i_1+1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]

\]

206
\[ \begin{align*}
+ & p_{i_1} \bar{p}_{i_2} \bar{p}_{i_3} \bar{p}_{i_4} \cdots \bar{p}_{i_m} + \bar{p}_{i_1-1} \bar{p}_{i_1+1} + \bar{p}_{i_2-1} \bar{p}_{i_2+1} + \cdots + \bar{p}_m \\
(\text{By rewriting } (AB) \text{ as } (A)(B) + A(B) + (A)B \text{ because they are equivalent})
\end{align*} \]

\[ \equiv p_{i_1} \bar{p}_{i_2} \bar{p}_{i_3} \bar{p}_{i_4} \cdots \bar{p}_{i_m} + \bar{p}_{i_1-1} \bar{p}_{i_1+1} + \bar{p}_{i_2-1} \bar{p}_{i_2+1} + \cdots + \bar{p}_m \\
+ p_{i_1+1} \bar{p}_{i_2} \bar{p}_{i_3} \bar{p}_{i_4} \cdots \bar{p}_{i_m} + \bar{p}_{i_1} \bar{p}_{i_1+2} + \bar{p}_{i_2-1} \bar{p}_{i_2+1} + \cdots + \bar{p}_m \\
+ p_{i_1} p_{i_2} \bar{p}_{i_3} \bar{p}_{i_4} \cdots \bar{p}_{i_m} + \bar{p}_{i_1} \bar{p}_{i_1+2} + \bar{p}_{i_2-1} \bar{p}_{i_2+1} + \cdots + \bar{p}_m + p_{i_2} S \\
\]

Now, \[ S(\bar{t}) \neq I_{DORF}(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})^{NLF}(p_{i_2}-p_{i_2})^{NLF}(\bar{t}) \]
if and only if \[ S(\bar{t}) \oplus I_{DORF}(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})^{NLF}(p_{i_2}-p_{i_2})^{NLF}(\bar{t}) = 1 \]
if and only if \[ p_{i_1} \bar{p}_{i_2} \bar{p}_{i_3} \bar{p}_{i_4} \cdots \bar{p}_{i_m} + \bar{p}_{i_1-1} \bar{p}_{i_1+1} + \bar{p}_{i_2-1} \bar{p}_{i_2+1} + \cdots + \bar{p}_m + p_{i_2} S \]
evaluates to 1 on \( \bar{t} \)
if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in TP_{i_1}(S) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_i(S) \right) \) such that \( p_{i_2,j_2} = 0 \) on \( \bar{t} \),

2. \( \bar{t} \in TP_{i_1+1}(S) \setminus \left( \bigcup_{i \neq i_1+1,i_2}^{m} TP_i(S) \right) \) such that \( p_{i_2,j_2} = 0 \) on \( \bar{t} \),

3. \( \bar{t} \in UTP_{i_2}(S) \) on \( \bar{t} \),

4. \( \bar{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1,i_2}^{m} TP_i(S) \right) \) on \( \bar{t} \),

5. \( \bar{t} \in (TP_{i_1+1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1+1,i_2}^{m} TP_i(S) \right) \) on \( \bar{t} \), or

6. \( \bar{t} \in NFP_{i_2,j_2}(S) \).

Hence, the result follows. \( \square \)

**Theorem 9.1.22** (DORF with LNF - Case 2)
Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \((p_{i_1} + p_{i_1+1})\) in \( S \) is implemented as \( p_{i_1}p_{i_1+1} \), and the \( j_2 \)-th literal \( x_{j_2}^{i_1} \) in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is negated where \( 1 \leq i_1 < m \), \( 1 \leq j_2 \leq k_{i_1} \), \( k_{i_1} > 1 \) and \( k_{i_1} \) is the number of literals in \( p_{i_1} \), the resulting implementation denoted
as \( I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LNF(p_{i_1} \rightarrow p_{i_1 + 1})} \) is equivalent to Expression (8.22) in Table 8.12. Then, \( S \neq I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LNF(p_{i_1} \rightarrow p_{i_1 + 1})} \), if and only if there is a test case \( \tilde{t} \) that satisfies any of the following conditions:

1. \( \tilde{t} \in TP_{i_1}(S) \setminus \bigcup_{i = 1}^{m} TP_i(S) \), or
2. \( \tilde{t} \in TP_{i_1+1}(S) \setminus \bigcup_{i = 1}^{m} TP_i(S) \) such that \( p_{i_1,j_2} = 0 \) on \( \tilde{t} \).

Proof: First, we observe that \( S \oplus I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LNF(p_{i_1} \rightarrow p_{i_1 + 1})} \)

\[
\equiv (p_{i_1} + p_{i_1 + 1}) (p_{i_1,j_2} p_{i_1+1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_m \\
\equiv (p_{i_1} + p_{i_1 + 1}) (p_{i_1,j_2} p_{i_1+1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} + \bar{p}_{i_1+2} \cdots \bar{p}_m \\
\equiv (p_{i_1} + p_{i_1 + 1} p_{i_1,j_2} + 0) \bar{p}_1 \cdots \bar{p}_{i_1-1} + \bar{p}_{i_1+2} \cdots \bar{p}_m \\
\text{(By making use of } A(BC) = AB, A(BA) = AB \text{ and } (A + B)A = 0) \\
\equiv p_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1+1} \bar{p}_{i_1,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} + \bar{p}_{i_1+2} \cdots \bar{p}_m \\
\text{Now,}\]

\[
S(\tilde{t}) \neq I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LNF(p_{i_1} \rightarrow p_{i_1 + 1})}(\tilde{t}) \\
\text{if and only if } S(\tilde{t}) \oplus I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LNF(p_{i_1} \rightarrow p_{i_1 + 1})}(\tilde{t}) = 1 \\
\text{if and only if } p_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1+1} \bar{p}_{i_1,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} + \bar{p}_{i_1+2} \cdots \bar{p}_m \\
\text{evaluates to } 1 \text{ on } \tilde{t}. \\
\text{if and only if } \tilde{t} \text{ satisfies any of the following conditions:}

1. \( \tilde{t} \in TP_{i_1}(S) \setminus \bigcup_{i = 1}^{m} TP_i(S) \), or
2. \( \tilde{t} \in TP_{i_1+1}(S) \setminus \bigcup_{i = 1}^{m} TP_i(S) \) such that \( p_{i_1,j_2} = 0 \) on \( \tilde{t} \).

Hence, the result follows. \( \square \)

Theorem 9.1.23 (DORF with LOF - Case 1)

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_{i_1} + p_{i_1 + 1}) \) is implemented as \( p_{i_1} p_{i_1 + 1} \), and the \( j_2 \)-th literal \( x_{j_2} \) in the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is omitted where \( 1 < i_1 + 1 < i_2 \leq m \), \( 1 \leq j_2 \leq k_{i_2} \), \( k_{i_2} > 1 \) and \( k_{i_2} \) is the number of literals in \( p_{i_2} \), the resulting implementation denoted as \( I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})} \) is equivalent to Expression (8.23) in Table 8.12. Then, \( S \neq I_{DORF(p_{i_1} + p_{i_1 + 1} \rightarrow p_{i_1} p_{i_1 + 1}) \times LOF(p_{i_2} \rightarrow p_{i_2,j_2})} \), if and only if there is a test case \( \tilde{t} \) that satisfies any of the following conditions:

1. \( \tilde{t} \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} = 0 \) on \( \tilde{t} \).
2. \( \bar{t} \in UTP_{\bar{t}+1}(S) \) such that \( p_{i_2j_2} = 0 \) on \( \bar{t} \), or
3. \( \bar{t} \in NFP_{i_2j_2}(S) \).

**Proof:** First, we observe that \( S \oplus I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1p_{i_1+1}}) \times LOF(p_{i_2} - p_{i_2j_2}) \)
\( \equiv ((p_{i_1} + p_{i_1+1} + p_{i_2}) \oplus (p_{i_1p_{i_1}+1} + p_{i_2j_2})) \bar{p}_{t_1} \cdots \bar{p}_{t_1+1} \bar{p}_{t_2} \cdots \bar{p}_{t_2+1} \cdots \bar{p}_{m} \)
\( \equiv ((p_{i_1} + p_{i_1+1} + p_{i_2}) (p_{i_1p_{i_1}+1} + p_{i_2j_2}) + (p_{i_1} + p_{i_1+1} + p_{i_2}) (p_{i_1p_{i_1}+1} + p_{i_2j_2})) \)
\( \bar{p}_{t_1} \cdots \bar{p}_{t_1+1} \bar{p}_{t_2} \cdots \bar{p}_{t_2+1} \cdots \bar{p}_{m} \)
\( \equiv ((p_{i_1} + p_{i_1+1} + p_{i_2}) (p_{i_1p_{i_1}+1} + p_{i_2j_2} + \bar{p}_{i_1p_{i_1}+1} + \bar{p}_{i_2j_2} + 0 + \bar{p}_{i_1} \bar{p}_{i_2j_2})) \)
\( \bar{p}_{t_1} \cdots \bar{p}_{t_1+1} \bar{p}_{t_2} \cdots \bar{p}_{t_2+1} \cdots \bar{p}_{m} \)

(By making use of \( A(AB) = AB \), \( AB(\bar{A}) = 0 \) and \( A(\bar{A}) = 0 \))
\( \equiv (p_{i_1} \bar{p}_{i_2j_2} p_{i_1+1} \bar{p}_{i_2j_2} + p_{i_1+1} p_{i_2} \bar{p}_{i_2j_2} \bar{p}_{t_2} + \bar{p}_{i_1} p_{i_2} \bar{p}_{i_2j_2} + p_{i_1} p_{i_2} \bar{p}_{i_2j_2} \bar{p}_{t_2}) \)
\( \bar{p}_{t_1} \cdots \bar{p}_{t_1+1} \bar{p}_{t_2} \cdots \bar{p}_{t_2+1} \bar{p}_{m} \)

(By rewriting \( \bar{A} \) as \( (\bar{A})(AB) \) because they are equivalent; and \( (AB)A \) as \( (AB)AB \) because they are equivalent)
\( \equiv p_{i_1} \bar{p}_{i_2j_2} \bar{p}_{t_1+1} \cdots \bar{p}_{m} + p_{i_1} \bar{p}_{i_2j_2} \bar{p}_{t_1+1} \cdots \bar{p}_{m} + p_{i_2j_2} S \)

Now,
\[ S(\bar{t}) \neq I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1p_{i_1+1}}) \times LOF(p_{i_2} - p_{i_2j_2})(\bar{t}) \]
if and only if
\[ S(\bar{t}) \oplus I_{DORF}(p_{i_1} + p_{i_1+1} - p_{i_1p_{i_1+1}}) \times LOF(p_{i_2} - p_{i_2j_2})(\bar{t}) = 1 \]
if and only if
\[ p_{i_1} \bar{p}_{i_2j_2} \bar{p}_{t_1+1} \cdots \bar{p}_{m} + p_{i_1} \bar{p}_{i_2j_2} \bar{p}_{t_1+1} \cdots \bar{p}_{m} + p_{i_2j_2} S \] evaluates to 1 on \( \bar{t} \).

if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in UTP_{\bar{t}+1}(S) \) such that \( p_{i_2j_2} = 0 \) on \( \bar{t} \),
2. \( \bar{t} \in UTP_{\bar{t}+1}(S) \) such that \( p_{i_2j_2} = 0 \) on \( \bar{t} \), or
3. \( \bar{t} \in NFP_{i_2j_2}(S) \).

Hence, the result follows. \( \square \)

**Theorem 9.1.24 (DORF with LOF - Case 2)**
Let \( S = p_{i_1} + \cdots + p_{m} \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_{i_1} + p_{i_1+1}) \) in \( S \) is implemented as \( p_{i_1}p_{i_1+1} \), and the \( j_2 \)-th literal \( x_{i_2}^1 \) in the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is omitted where \( 1 \leq i_1 < m, 1 \leq j_2 \leq k_{i_1}, k_{i_1} > 1 \) and \( k_{i_1} \) is the number of literals in \( p_{i_1} \), the resulting implementation denoted
as $I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LOF(p_i - p_i, i_2)$ is equivalent to Expression (8.24) in Table 8.12. Then, $S \neq I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LOF(p_i - p_i, i_2)$ if and only if there is a test case $\breve{i}$ that satisfies any of the following conditions:

1. $\breve{i} \in UTP_{i_1}(S)$, or
2. $\breve{i} \in UTP_{i_1 + 1}(S)$ such that $p_{i_1, i_2} = 0$ on $\breve{i}$.

**Proof:** First, we observe that $S \oplus I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LOF(p_i - p_i, i_2)$

$\equiv ((p_i + p_i + 1) \oplus (p_{i_1, i_2}\bar{p}_{i_1 + 1})) \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

$\equiv ((p_i + p_i + 1)(\bar{p}_{i_1, i_2} p_{i_1 + 1}) + (p_i + p_i + 1)p_{i_1, i_2} p_{i_1 + 1}) \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

$\equiv (p_i \bar{p}_{i_1 + 1} + p_i + 1 p_{i_1, i_2} p_{i_1}) \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

(By making use of $AB(AC) \equiv ABC$, $A(BA) \equiv AB$ and $(A + C)BC \equiv 0$)

$\equiv (p_i \bar{p}_1 p_{i_1 + 1} p_{i_1, i_2} p_{i_1}) \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

(By rewriting $(A)$ as $(A)(AB)$ because they are equivalent)

$\equiv p_i \bar{p}_1 \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 1} \cdots \bar{p}_m + p_i + 1 \bar{p}_{i_1, i_2} \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

Now,

$S(\breve{i}) \neq I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LOF(p_i - p_i, i_2)(\breve{i})$

if and only if

$S(\breve{i}) \oplus I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LOF(p_i - p_i, i_2)(\breve{i}) = 1$

if and only if

$p_i \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 1} \cdots \bar{p}_m + p_i + 1 \bar{p}_{i_1, i_2} \bar{p}_1 \cdots \bar{p}_{i_1 - 1} \bar{p}_{i_1 + 2} \cdots \bar{p}_m$

evaluates to 1 on $\breve{i}$

if and only if $\breve{i}$ satisfies any of the following conditions:

1. $\breve{i} \in UTP_{i_1}(S)$, or
2. $\breve{i} \in UTP_{i_1 + 1}(S)$ such that $p_{i_1, i_2} = 0$ on $\breve{i}$.

Hence, the result follows.

**Theorem 9.1.25 (DORF with LIF - Case 1)**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the subexpression $(p_i + p_i + 1)$ in $S$ is implemented as $p_i p_i + 1$, and the literal $x_{i_2}$ is inserted into the $i_2$-th term, $p_{i_2}$, in $S$ where $1 < i_1 + 1 < i_2 \leq m$ and $x_{i_2}$ is a missing literal of $p_{i_2}$, the resulting implementation denoted as $I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LIF(p_i - p_i, x_{i_2})$ is equivalent to Expression (8.25) in Table 8.12. Then, $S \neq I_{DORF}(p_i + p_i + 1 - p_i + p_i + 1) \times LIF(p_i - p_i, x_{i_2})$

if and only if there is a test case $\breve{i}$ that satisfies any of the following conditions:

1. $\breve{i} \in UTP_{i_1}(S)$,
2. $\breve{i} \in UTP_{i_1 + 1}(S)$,
3. $\breve{i} \in UTP_{i_2}(S)$ such that $x_{i_2} = 0$ on $\breve{i}$.
4. \( i \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1}^{m} TP_i(S) \) such that \( x_{i_2} = 0 \) on \( i \), or

5. \( i \in (TP_{i_1+1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1}^{m} TP_i(S) \) such that \( x_{i_2} = 0 \) on \( i \).

**Proof:** First, we observe that

\[
\begin{align*}
\equiv & \left((p_{i_1} + p_{i_1+1} + p_{i_2}) \oplus (p_{i_1}p_{i_1+1} + p_{i_2}x_{i_2})\right)
\equiv & \left((p_{i_1} + p_{i_1+1} + p_{i_2})(\bar{p}_1p_{i_1+1} + p_{i_2}x_{i_2}) + (\bar{p}_1 + p_{i_1+1} + p_{i_2})(p_{i_1}p_{i_1+1} + p_{i_2}x_{i_2})\right)
\equiv & \left((p_{i_1} + p_{i_1+1} + p_{i_2})\bar{p}_1p_{i_1+1} + \bar{p}_1p_{i_1+1}p_{i_2}x_{i_2} \right)
\end{align*}
\]

(By making use of \( A(AB) \equiv AB \) and \( (A)A \equiv 0 \))

\[
\begin{align*}
\equiv & \left((p_{i_1}p_{i_1+1}p_{i_2}x_{i_2}) + p_{i_1}p_{i_1+1}(\bar{p}_{i_2} + p_{i_2}x_{i_2}) + (\bar{p}_1p_{i_1+1} + p_{i_1}p_{i_1+1} + \bar{p}_1p_{i_1+1})p_{i_2}x_{i_2}\right)
\end{align*}
\]

(By rewriting \( AB \) as \( \bar{A} + \bar{B} \) because they are equivalent, and \( \bar{A} \bar{B} \) as \( (\bar{A})(\bar{B}) + A\bar{B} + \bar{A}\bar{B} \) because they are equivalent)

\[
\begin{align*}
\equiv & \left(p_{i_1}\bar{p}_{i_2} + p_{i_1}\bar{p}_{i_2}p_{i_2}x_{i_2} + \bar{p}_1p_{i_1+1}\bar{p}_{i_2}x_{i_2} + \bar{p}_1p_{i_1+1}p_{i_2}x_{i_2} + \bar{p}_1p_{i_1+1}p_{i_2}x_{i_2}\right)
\end{align*}
\]
Now, \( S(\vec{t}) \neq I_{DORF(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) LIF(p_{i_2} - p_{i_2} x_{i_2})}(\vec{t}) \)

if and only if

\[ S(\vec{t}) \oplus I_{DORF(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) LIF(p_{i_2} - p_{i_2} x_{i_2})}(\vec{t}) \]

if and only if

\[ p_{i_1} p_1 \cdots p_{i_1-1} p_{i_1+1} \cdots \tilde{p}_m + p_{i_1+1} \tilde{p}_1 \cdots \tilde{p}_{i_1} \tilde{p}_{i_1+2} \cdots \tilde{p}_m + \]

\[ p_{i_1} p_{i_2} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2+1} \cdots \tilde{p}_m + p_{i_1+1} p_{i_2} x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2+1} \cdots \tilde{p}_m \] evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \),
2. \( \vec{t} \in UTP_{i_1+1}(S) \),
3. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \),
4. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus ( \bigcup \limits_{j=1}^{m} TP_i(S) ) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \), or
5. \( \vec{t} \in (TP_{i_1+1}(S) \cap TP_{i_2}(S)) \setminus ( \bigcup \limits_{j=1}^{m} TP_i(S) ) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

**Theorem 9.1.26 (DORF with LIF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( (p_{i_1} + p_{i_1+1}) \) in \( S \) is implemented as \( p_{i_1} p_{i_1+1} \), and the literal \( x_{i_2} \) is inserted into the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) where \( 1 \leq i_1 < m \) and \( x_{i_2} \) is a missing literal of \( p_{i_1} \), the resulting implementation denoted as \( I_{DORF(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) LIF(p_{i_1} - p_{i_1} x_{i_2})} \) is equivalent to Expression (8.26) in Table 8.12. Then, \( S \neq I_{DORF(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) LIF(p_{i_1} - p_{i_1} x_{i_2})} \)

if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \),
2. \( \vec{t} \in UTP_{i_1+1}(S) \),
3. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus ( \bigcup \limits_{i=1}^{m} TP_i(S) ) \) such that \( x_{i_2} = 0 \) on \( \vec{t} \).

**Proof :** First, we observe that

\[
S \oplus I_{DORF(p_{i_1} + p_{i_1+1} - p_{i_1} p_{i_1+1}) LIF(p_{i_2} - p_{i_2} x_{i_2})}
\]

\( \equiv \((p_{i_1} + p_{i_1+1}) \oplus (p_{i_1} x_{i_2} p_{i_1+1})\) \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_m + \)

\( (p_{i_1} + p_{i_1+1})(p_{i_1} x_{i_2} p_{i_1+1}) + (p_{i_1} + p_{i_1+1})(p_{i_1} x_{i_2} p_{i_1+1})\) \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_m \]

212
\[\equiv (p_{i_1}x_{i_2}p_{i_{1+1}} + p_{i_{1+1}}p_{i_1}x_{i_2} + 0)\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+2} \cdots \bar{p}_m\]

(By making use of \(A(AB) \equiv AB\) and \((A+B)A \equiv 0\))

\[\equiv (p_{i_1}(\bar{x}_{i_2}p_{i_{1+1}} + p_{i_{1+1}}(p_{i_1} + \bar{x}_{i_2}p_{i_1})))\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+2} \cdots \bar{p}_m\]

(By rewriting \(\bar{A}B\) as \(AB + \bar{B}\) because they are equivalent)

\[\equiv (p_{i_1}p_{i_{1+1}} + p_{i_1}\bar{p}_{i_{1+1}} + p_{i_1}p_{i_{1+1}}\bar{x}_{i_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+2} \cdots \bar{p}_m\]

\[\equiv p_{i_1}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1+1}\bar{p}_1 \cdots \bar{p}_{i_1}\bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1+1}\bar{x}_{i_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+2} \cdots \bar{p}_m\]

Now, \(S(\bar{i}) \neq I_{DORF(p_{i_1}+p_{i_{1+1}})}\) \(\bar{LIF}(p_1 \rightarrow p_{i_1}\bar{x}_{i_2})(\bar{i})\)

if and only if \(S(\bar{i}) \oplus I_{DORF(p_{i_1}+p_{i_{1+1}})}\) \(\bar{LIF}(p_1 \rightarrow p_{i_1}\bar{x}_{i_2})(\bar{i}) = 1\)

if and only if \(p_{i_1}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + p_{i_1+1}\bar{p}_1 \cdots \bar{p}_{i_1}\bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1+1}\bar{x}_{i_2}\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+2} \cdots \bar{p}_m\)

\[\text{evaluates to 1 on } \bar{i}\]

if and only if \(\bar{i}\) satisfies any of the following conditions:

1. \(\bar{i} \in UTP_{i_1}(S)\),
2. \(\bar{i} \in UTP_{i_1+1}(S)\), or
3. \(\bar{i} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1, i_1+1} TP_i(S) \right)\)

such that \(x_{i_2} = 0\) on \(\bar{i}\).

Hence, the result follows. \(\square\)

**Theorem 9.1.27 (DORG with LRF - Case 1)**

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the subexpression \((p_{i_1} + p_{i_{1+1}})\) in \(S\) is implemented as \(p_{i_1}p_{i_{1+1}}\), and the \(j_2\)-th literal \(x_{j_2}\) in the \(i_2\)-th term, \(p_{i_2}\), in \(S\) is replaced by the literal \(x_{i_2}\) where \(1 < i_1 + 1 < i_2 \leq m\), \(1 \leq j_2 \leq k_{i_2}\), \(k_{i_2}\) is the number of literals in \(p_{i_2}\) and \(x_{i_2}\) is a missing literal of \(p_{i_2}\).

(a) When \(k_{i_2} > 1\), the resulting implementation denoted as \(I_{DORF(p_{i_1}+p_{i_{1+1}} \rightarrow p_{i_1}p_{i_{1+1}} \oplus LRF(p_2 \rightarrow p_{i_2}x_{i_2})\)}\) is equivalent to Expression (8.27) in Table 8.12. Then, we have \(S \neq I_{DORF(p_{i_1}+p_{i_{1+1}} \rightarrow p_{i_1}p_{i_{1+1}} \oplus LRF(p_2 \rightarrow p_{i_2}x_{i_2})\)}\) if and only if there is a test case \(\bar{i}\) that satisfies any of the following conditions:

1. \(\bar{i} \in UTP_{i_1}(S)\) such that \(x_{i_2} = 0\) on \(\bar{i}\),
2. \(\bar{i} \in UTP_{i_1}(S)\) such that \(p_{i_2,j_2} = 0\) on \(\bar{i}\),
3. \(\bar{i} \in UTP_{i_{1+1}}(S)\) such that \(x_{i_2} = 0\) on \(\bar{i}\),
4. \(\bar{i} \in UTP_{i_{1+1}}(S)\) such that \(p_{i_2,j_2} = 0\) on \(\bar{i}\),
5. \(\bar{i} \in UTP_{i_2}(S)\) such that \(x_{i_2} = 0\) on \(\bar{i}\),
6. \(\bar{i} \in NFP_{i_2,j_2}(S)\) such that \(x_{i_2} = 1\) on \(\bar{i}\),
7. \(\bar{i} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right)\) such that \(x_{i_2} = 0\) on \(\bar{i}\), or
8. $\vec{t} \in (TP_{i+1}(S) \cap TP_{i+2}(S)) \setminus \left( \bigcup_{i \neq i+1, i+2} TP_i(S) \right)$ such that $x_{i+2} = 0$ on $\vec{t}$.

(b) When $k_{i+2} = 1$, the resulting implementation denoted as $I_{DORF(p_{i+1}+p_{i+1}+p_{i+1}+1) \times LRF(p_{i+2} \rightarrow x_{i+2})}$ is equivalent to Expression (8.27) in Table 8.12 without $p_{i+2}$ because $p_{i+2}$ contains just one literal. Then, $S \not \equiv I_{DORF(p_{i+1}+p_{i+1}+p_{i+1}+1) \times LRF(p_{i+2} \rightarrow x_{i+2})}$ if and only if there is a test case $\vec{t}$ that satisfies any of the following conditions:

1. $\vec{t} \in UTP_i(S)$ such that $x_{i+2} = 0$ on $\vec{t}$,
2. $\vec{t} \in UTP_{i+1}(S)$ such that $x_{i+2} = 0$ on $\vec{t}$,
3. $\vec{t} \in UTP_{i+2}(S)$ such that $x_{i+2} = 0$ on $\vec{t}$,
4. $\vec{t} \in FP(S)$ such that $x_{i+2} = 1$ on $\vec{t}$,
5. $\vec{t} \in (TP_{i+1}(S) \cap TP_{i+2}(S)) \setminus \left( \bigcup_{i \neq i+1, i+2} TP_i(S) \right)$ such that $x_{i+2} = 0$ on $\vec{t}$, or
6. $\vec{t} \in (TP_{i+1}(S) \cap TP_{i+2}(S)) \setminus \left( \bigcup_{i \neq i+1, i+2} TP_i(S) \right)$ such that $x_{i+2} = 0$ on $\vec{t}$.

**Proof:** (a) First, we observe that $S \not \equiv I_{DORF(p_{i+1}+p_{i+1}+p_{i+1}+1) \times LRF(p_{i+2} \rightarrow x_{i+2})}$

$\equiv ((p_{i+1} + p_{i+1} + p_{i+2}) \oplus (p_{i+1}p_{i+1} + p_{i+2}x_{i+2}))\tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

$\equiv ((p_{i+1} + p_{i+1} + p_{i+2})(p_{i+1}p_{i+1} + p_{i+2}x_{i+2}) + (p_{i+1} + p_{i+1} + p_{i+2})(p_{i+1}p_{i+1} + p_{i+2}x_{i+2}))$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

$\equiv ((p_{i+1} + p_{i+1} + p_{i+2})(p_{i+1}p_{i+1} + p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+1}p_{i+1} + p_{i+2}x_{i+2}))$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

$\equiv (p_{i+1}p_{i+1}+p_{i+2})(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+2}x_{i+2}) + (p_{i+1}p_{i+1})p_{i+2}x_{i+2} + 0 + \tilde{p}_1p_{i+1}p_{i+2}p_{i+2}x_{i+2})$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

(By making use of $A(\overline{AB}) \equiv \overline{A}B$ and $(\overline{A})A \equiv 0$)

$\equiv (p_{i+1}p_{i+1}+p_{i+2})(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+2}x_{i+2}) + (p_{i+1}p_{i+1})p_{i+2}x_{i+2} + \tilde{p}_1p_{i+1}p_{i+2}p_{i+2}x_{i+2})$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

(By rewriting $(\overline{AB})A$ as $(\overline{AB})A$ because they are equivalent)

$\equiv (p_{i+1}p_{i+1}+p_{i+2})(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+2}x_{i+2}) + (p_{i+1}p_{i+1})p_{i+2}x_{i+2} + \tilde{p}_1p_{i+1}p_{i+2}p_{i+2}x_{i+2})$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

(By rewriting $\overline{AC}$ as $\overline{A}(\overline{AB}) + (\overline{AB})\overline{C} + AB\overline{C}$ because they are equivalent to $\overline{A} + \overline{C}$, and $\overline{AB}$ as $(\overline{A})(\overline{B}) = AB + \overline{AB}$ because they are equivalent)

$\equiv (p_{i+1}p_{i+1}+p_{i+2})(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}x_{i+2})$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$

$\equiv (p_{i+1}p_{i+1}+p_{i+2})(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}(p_{i+2}x_{i+2}) + \tilde{p}_1p_{i+1}p_{i+2}x_{i+2})$

$\cdot \tilde{p}_1 \cdots \tilde{p}_{i+1}p_{i+2}1 \cdots \tilde{p}_{i+2}1 \cdots \tilde{p}_m$
\[ + \bar{p}_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_i + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} \]
\[ \equiv (p_i, \bar{p}_i + p_i \bar{p}_{i+1} + p_{i+2}) \bar{p}_{i+1} + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+1} \bar{p}_{i+2} \]
\[ + p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} \]
\[ \equiv p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} + p_i \bar{p}_{i+1} + p_i \bar{p}_{i+1} \bar{p}_{i+2} \]
\[ + p_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} \]
\[ + p_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_i \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} \]
\[ \text{Now,} \quad S(t) \neq I_{Dorf}(p_i, p_{i+1}, p_{i+1}, p_i) \cdot LRF(p_{i+2}, p_{i+2}, p_{i+2}, p_{i+2}) \]
\[ \text{if and only if} \quad S(t) \oplus I_{Dorf}(p_i, p_{i+1}, p_{i+1}, p_i) \cdot LRF(p_{i+2}, p_{i+2}, p_{i+2}, p_{i+2}) \]
\[ \equiv 1 \]
\[ \text{if and only if} \quad p_{i+1} \bar{p}_{i+1} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} + p_{i+1} \bar{p}_{i+1} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} \]
\[ + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} \]
\[ + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} \bar{p}_{i+3} \bar{p}_{i+4} \]
\[ \text{such that} \quad x_{i+2} = 0 \text{ on } t \]
\[ 1. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 0 \text{ on } t \]
\[ 2. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 0 \text{ on } t \]
\[ 3. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 0 \text{ on } t \]
\[ 4. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 0 \text{ on } t \]
\[ 5. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 0 \text{ on } t \]
\[ 6. \quad t \in UTP_{i+1}(S) \text{ such that } x_{i+2} = 1 \text{ on } t \]
\[ 7. \quad t \in \left( (TP_{i+1}(S) \cap TP_{i+2}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \text{ such that} \]
\[ x_{i+2} = 0 \text{ on } t, \] or
\[ 8. \quad t \in \left( (TP_{i+1}(S) \cap TP_{i+2}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \text{ such that} \]
\[ x_{i+2} = 0 \text{ on } t. \]

Hence, the result follows.

(b) The proof of this part is similar to (a) above except that the term \( p_{i+2,j_2} \) does not appear in the proof. We proceed the proof as follows.

First, we observe that \( S \oplus I_{Dorf}(p_i, p_{i+1}, p_{i+1}, p_i) \cdot LRF(p_{i+2}, p_{i+2}, p_{i+2}, p_{i+2}) \]
\[ \equiv \left( (p_i + p_{i+1} + p_{i+2}) \oplus (p_i p_{i+1} + x_{i+2}) \right) \bar{p}_{i+1} + p_{i+1} \bar{p}_{i+1} + p_{i+2} \bar{p}_{i+2} + p_{i+1} \bar{p}_{i+1} \bar{p}_{i+2} + \cdots + \bar{p}_{i+2} \bar{p}_{i+2} + \cdots + \bar{p}_m \]
\( \equiv (p_1 + p_{i_1+1} + p_{i_2})(\tilde{p}_1 p_{i_1+1} + x_{i_2}) + (p_1 + p_{i_1+1} + p_{i_2})(p_1 p_{i_1+1} + x_{i_2}) \)
\[
\cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]
\( \equiv (p_1 + p_{i_1+1} + p_{i_2})(\tilde{p}_1 p_{i_1+1} + x_{i_2}) + \tilde{p}_1 \tilde{p}_{i_1+1} + \tilde{p}_2 (p_1 p_{i_1+1} + x_{i_2}) \)
\[
\cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]
\( \equiv (p_1 \tilde{p}_{i_1+1} + \tilde{p}_1 p_{i_1+1} + x_{i_2}) + (\tilde{p}_1 p_{i_1+1} + \tilde{p}_2 x_{i_2} + 0 + \tilde{p}_1 \tilde{p}_{i_1+1} x_{i_2}) \)
\[
\cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]

(By making use of \(A(\tilde{A}B) \equiv \tilde{A}B\) and \((\tilde{A})A \equiv 0)\)
\( \equiv (p_1 \tilde{p}_{i_1+1} + \tilde{p}_1 p_{i_1+1} + x_{i_2}) + (\tilde{p}_1 p_{i_1+1} + \tilde{p}_2 x_{i_2} + \tilde{p}_1 p_{i_1+1} x_{i_2}) \)
\[
\cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]
\( \equiv (p_1 \tilde{p}_{i_1+1} + p_{i_2} \tilde{x}_{i_2} + p_{i_1+1} x_{i_2} + \tilde{p}_1 p_{i_1+1} + \tilde{p}_2 x_{i_2} + \tilde{p}_1 p_{i_1+1} x_{i_2}) \)
\[
\cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+2} \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]
\( \equiv p_1 \tilde{x}_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1+1} \tilde{x}_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_2} \tilde{x}_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_2-1} \tilde{p}_{i_2+1} \cdots \tilde{p}_m
\]
\[
+ x_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m + p_{i_1+1} \tilde{x}_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m
\]
\[
+ p_{i_1+1} p_{i_2} \tilde{x}_{i_2} \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_m
\]
Now, if and only if
\[ S(\vec{t}) \neq I_{DORF(p_{i_1}+p_{i_1+1} \rightarrow p_{i_1},p_{i_1+1}) \times LRF(p_{i_2} \rightarrow x_{l_2})} (\vec{t}) \]
and if and only if
\[ S(\vec{t}) \oplus I_{DORF(p_{i_1}+p_{i_1+1} \rightarrow p_{i_1},p_{i_1+1}) \times LRF(p_{i_2} \rightarrow x_{l_2})} (\vec{t}) = 1 \]
if and only if
\[ p_{i_1} \bar{x}_{l_2} \bar{p}_1 \cdots \bar{p}_{l_1+1} \cdots \bar{p}_m + p_{i_1+1} \bar{x}_{l_2} \bar{p}_1 \cdots \bar{p}_{l_1+1} \cdots \bar{p}_m + x_{l_2} \bar{S} \]
+ \[ p_{i_1} x_{l_2} \bar{p}_1 \cdots \bar{p}_{l_1+1} \cdots \bar{p}_m + x_{l_2} \bar{S} \]
+ \[ p_{i_1+1} x_{l_2} \bar{p}_1 \cdots \bar{p}_{l_1+1} \cdots \bar{p}_m \]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:
1. \( \vec{t} \in UTP_{i_1}(S) \) such that \( x_{l_2} = 0 \) on \( \vec{t} \),
2. \( \vec{t} \in UTP_{i_1+1}(S) \) such that \( x_{l_2} = 0 \) on \( \vec{t} \),
3. \( \vec{t} \in UTP_{i_2}(S) \) such that \( x_{l_2} = 0 \) on \( \vec{t} \),
4. \( \vec{t} \in FP(S) \) such that \( x_{l_2} = 1 \) on \( \vec{t} \),
5. \( \vec{t} \in \left( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1, \ i \neq i_1, i_2} TP_i(S) \right) \) such that
   \[ x_{l_2} = 0 \] on \( \vec{t} \), or
6. \( \vec{t} \in \left( (TP_{i_1+1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i=1, \ i \neq i_1+1, i_2} TP_i(S) \right) \) such that
   \[ x_{l_2} = 0 \] on \( \vec{t} \).

Hence, the result follows. \( \square \)

It should be noted that there are two differences between the detection conditions of Theorem 9.1.27(a) and (b). First, the detection conditions 2 and 4 of Theorem 9.1.27(a) are related to the term \( p_{i_2,j_2} \) and this term does not exist in Theorem 9.1.27(b) when \( k_{l_2} = 1 \). Second, the detection condition 6 of Theorem 9.1.27(a) (that is, \( "\vec{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{l_2} = 1" \)) differs from the detection condition 4 of Theorem 9.1.27(b) (that is, \( "\vec{t} \in FP(S) \) such that \( x_{l_2} = 1" \)) syntactically. We say that these two conditions are syntactically different because they are actually equivalent to each other when \( k_{l_2} = 1 \) (that is, when \( p_{i_2} \) contains just one literal). The reason is similar to those as explained in the paragraph after Theorem 9.1.20.

Hence, without loss of generality, we can still use the eight detection conditions in Theorem 9.1.27(a) to represent the detection conditions of Expression (8.27) in Table 8.12 for \( k_{l_2} \geq 1 \), bearing in mind the non-existence of the term \( p_{i_2,j_2} \) and the equivalence between \( "\vec{t} \in FP(S) \) such that \( x_{l_2} = 1" \) and \( "\vec{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{l_2} = 1" \) when \( k_{l_2} = 1 \).

**Theorem 9.1.28** (DORF with LRF - Case 2)
Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the sub-expression $(p_i + p_{i+1})$ in $S$ is implemented as $p_i p_{i+1}$, and the $j_2$-th literal $x_{j_2}^1$ in the $i_1$-th term, $p_i$, in $S$ is replaced by $x_{j_2}$ where $1 \leq i < m$, $1 \leq j_2 \leq k_i$, $k_i$ is the number of literals in $p_i$, and $x_{j_2}$ is a missing literal of $p_i$.

(a) When $k_i > 1$, the resulting implementation denoted as $I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ is equivalent to Expression (8.28) in Table 8.12. Then, we have $S \neq I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$,
2. $\bar{t} \in UTP_{i_1+1}(S)$ such that $p_{i_1} = 0$ on $\bar{t}$,
3. $\bar{t} \in UTP_{i_1+1}(S)$ such that $x_{j_2} = 0$ on $\bar{t}$, or
4. $\bar{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$ such that $x_{j_2} = 0$ on $\bar{t}$.

(b) When $k_i = 1$, the resulting implementation denoted as $I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ is equivalent to Expression (8.28) in Table 8.12 without $p_{i+2}$ because $p_i$ contains just one literal. Then, $S \neq I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$,
2. $\bar{t} \in UTP_{i_1+1}(S)$ such that $x_{j_2} = 0$ on $\bar{t}$, or
3. $\bar{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$ such that $x_{j_2} = 0$ on $\bar{t}$.

Proof: (a) First, we observe that $S \oplus I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ is equivalent to Expression (8.28) in Table 8.12 because $p_i$ contains just one literal. Then, $S \neq I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$,
2. $\bar{t} \in UTP_{i_1+1}(S)$ such that $x_{j_2} = 0$ on $\bar{t}$, or
3. $\bar{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$ such that $x_{j_2} = 0$ on $\bar{t}$.

Proof: (a) First, we observe that $S \oplus I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ is equivalent to Expression (8.28) in Table 8.12 because $p_i$ contains just one literal. Then, $S \neq I_{DORF(p_1 + p_{i+1} \rightarrow p_i p_{i+1})}$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in UTP_{i_1}(S)$,
2. $\bar{t} \in UTP_{i_1+1}(S)$ such that $x_{j_2} = 0$ on $\bar{t}$, or
3. $\bar{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right)$ such that $x_{j_2} = 0$ on $\bar{t}$.
Now, \[ S(\vec{t}) \neq I_{DORF(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})} \ast LRF(p_{i_1}-p_{i_1,j_2}x_{l_2})(\vec{t}) \]
if and only if \[ S(\vec{t}) \oplus I_{DORF(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})} \ast LRF(p_{i_1}-p_{i_1,j_2}x_{l_2})(\vec{t}) = 1 \]
if and only if \[
p_i(p_{i_1} \cdot \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m) + p_{i_1+1}p_{l_1,j_2}p_{i_1}p_{i_1+2} \cdots \bar{p}_m
\]
\[
+p_i p_{i_1+1} x_{l_2} p_{i_1} p_{i_1+1} \cdots \bar{p}_m
\]
evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_1}(S) \),
2. \( \vec{t} \in UTP_{i_1+1}(S) \) such that \( p_{i_1,j_2} = 0 \) on \( \vec{t} \),
3. \( \vec{t} \in UTP_{i_1+1}(S) \) such that \( x_{l_2} = 0 \) on \( \vec{t} \), or
4. \( \vec{t} \in (TP_{i_1}(S) \cap TP_{i_1+1}(S)) \setminus \left( \bigcup_{i \neq i_1,j_1+1} TP_i(S) \right) \) such that \( x_{l_2} = 0 \) on \( \vec{t} \).

Hence, the result follows. (b) The proof of this part is similar to that in (a) above except that the term \( p_{i_1,j_2} \) does not appear in the proof. We proceed the proof as follows.

First, we observe that \[
S \oplus I_{DORF(p_{i_1}+p_{i_1+1}-p_{i_1}p_{i_1+1})} \ast LRF(p_{i_1}-x_{l_2})
\equiv ((p_{i_1} + p_{i_1+1}) \oplus (x_{l_2}p_{i_1+1})) \bar{p}_1 \cdot \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m
\equiv ((p_{i_1} + p_{i_1+1})(x_{l_2}p_{i_1+1}) + (\bar{p}_{i_1} + p_{i_1+1})(x_{l_2}p_{i_1+1})) \bar{p}_1 \cdot \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m
\equiv (p_{i_1}(x_{l_2}p_{i_1+1}) + p_{i_1+1}x_{l_2} + 0) \bar{p}_1 \cdot \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m
\]
(By making use of \( A(BA) = A\bar{B} \) and \( A(BA) \equiv 0 \))
\[ \equiv (p_{i_1}p_{i_1+1}x_{l_2} + p_{i_1+1}p_{i_1} + p_{i_1}x_{l_2} + p_{i_1}x_{l_2}) \bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m \]
(By using \( AB \equiv BA \bar{B} \) and \( B \equiv \bar{A}B + AB \))
\[ \equiv (p_{i_1}x_{l_2}p_{i_1+1} + p_{i_1+1}x_{l_2}p_{i_1} + p_{i_1}x_{l_2}p_{i_1+1}) \bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m \]
\[ \equiv p_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+1} \cdots \bar{p}_m + p_{i_1+1}x_{l_2} \bar{p}_1 \cdots \bar{p}_{i_1+2} \cdots \bar{p}_m + p_{i_1}p_{i_1+1}x_{l_2} \bar{p}_1 \cdots \bar{p}_{i_1-1}p_{i_1+2} \cdots \bar{p}_m \]

219
Now, \( S(\vec{t}) \neq I_{DORF}(p_1 + p_{i_1+1} - p_{i_1} p_{i_1+1}) \times LNF(p_{i_1} - x_{t_2})(\vec{t}) \)
if and only if \( S(\vec{t}) \oplus I_{DORF}(p_1 + p_{i_1+1} - p_{i_1} p_{i_1+1}) \times LNF(p_{i_1} - x_{t_2})(\vec{t}) = 1 \)
if and only if
\[
p_{i_1} p_{i_1+1} \cdots p_{i_1+1} \cdot \bar{p}_m + p_{i_1+1} \bar{x}_{t_2} p_1 \cdots p_{i_1} p_{i_1+2} \cdots \bar{p}_m
\]
+ \( p_{i_1+1} \bar{x}_{t_2} p_1 \cdots \bar{p}_{i_1+2} \cdots \bar{p}_m \) evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{t_2}(S) \),
2. \( \vec{t} \in UTP_{t_2+1}(S) \) such that \( x_{t_2} = 0 \) on \( \vec{t} \), or
3. \( \vec{t} \in \left( T_{P_{t_2}}(S) \cap T_{P_{t_2+1}}(S) \right) \setminus \left( \bigcup_{i \neq i_1, i_1+1} T_{P_i}(S) \right) \) such that \( x_{t_2} = 0 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

The only difference between the detection conditions of Theorem 9.1.28(a) and (b) is that the detection condition 2 Theorem 9.1.28(a) is related to the term \( p_{i_1,j_2} \) which does not exist in Theorem 9.1.28(b) when \( k_{i_1} = 1 \). Hence, without loss of generality, we can still use the four detection conditions in Theorem 9.1.28(a) to represent the detection conditions of Expression (8.28) in Table 8.12 for \( k_{i_1} \geq 1 \), bearing in mind the non-existence of the term \( p_{i_1,j_2} \) when \( k_{i_1} = 1 \).

### 9.1.5 CORF with Other Literal Faults

In this section, we study the detection conditions of double faults in which one of the single faults is an CORF.

**Theorem 9.1.29 (CORF with LNF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term \( p_{i_1} \) in \( S \) is implemented as \( p_{i_1,i_1,j_1} + p_{i_1,j_1+1,k_1} \) and the \( j_2 \)-th literal \( x_{t_2}^{i_2} \) in the \( i_2 \)-th term \( p_{i_2} \) in \( S \) is negated where \( p_{i_1} = p_{i_1,i_1,j_1} \cdot p_{i_1,j_1+1,k_1} \), \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_1 < k_1, 1 \leq j_2 \leq k_{i_2}, k_{i_2} > 1 \) and \( k_{i_1} \) and \( k_{i_2} \) are the numbers of literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, the resulting implementation denoted as \( I_{CORF}(p_1 - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \times LNF(p_{i_2} - p_{i_2,j_2}) \) is equivalent to Expression (8.29) in Table 8.12. Then, \( S \neq I_{CORF}(p_1 - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \times LNF(p_{i_2} - p_{i_2,j_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{t_2}(S) \) such that \( p_{i_1,i_1,j_1} + p_{i_1,j_1+1,k_1} = 0 \) on \( \vec{t} \),
2. \( \vec{t} \in NFP_{t_2,j_2}(S) \),
3. \( \vec{t} \in FP(S) \) such that \( p_{i_1,i_1,j_1} = 1 \) on \( \vec{t} \), or
4. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_1} = 1 \) on \( \vec{t} \).
Proof: First, we observe that $S \equiv I_{\text{CORF}}(p_{1_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \mathcal{L}_{\text{NLF}}(p_{i_2} \rightarrow p_{i_2,j_2})$
\[\equiv ((p_{1_1} + p_{i_2}) \oplus (p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2})) \bar{p}_{i_1-i} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[= (p_{1_1} + p_{i_2})(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2}) + (p_{1_1} + p_{i_2})(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2})) \]
\[\bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[= (p_{1_1} + p_{i_2})(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \bar{p}_{i_1} \bar{p}_{i_2} + \bar{p}_{i_1} \bar{p}_{i_2} (p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2})) \]
\[\bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By making use of $AB(A+B) \equiv 0$ and $AB(AB) \equiv AB$)
\[\equiv (p_{1_2}(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \bar{p}_{i_1} + p_{1_1} \bar{p}_{i_2}(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2})) \]
\[\bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]

(By rewriting $(A+B)$ as $(A+B)(AB)$ because they are equivalent)
\[\equiv p_{1_2}(p_{1_1,j_1} + p_{1_1,j_2+1,k_{i_1}}) \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[+(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2}) \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m \]
\[\equiv p_{1_2}(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + (p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} + p_{i_2,j_2}) \bar{S} \]
\[\equiv p_{1_2}(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2,j_2} \bar{S} + p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} \bar{S} \]

Now, \[S(\bar{t}) \neq I_{\text{CORF}}(p_{1_1} \rightarrow p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \mathcal{L}_{\text{NLF}}(p_{i_2} \rightarrow p_{i_2,j_2})(\bar{t})\]
if and only if \[S(\bar{t}) \lor I_{\text{CORF}}(p_{1_1} \rightarrow p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \mathcal{L}_{\text{NLF}}(p_{i_2} \rightarrow p_{i_2,j_2})(\bar{t}) = 1\]
if and only if \[p_{1_2}(p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \bar{p}_{i_1} \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m + p_{i_2,j_2} \bar{S} + p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} \bar{S} \]
\[\text{evaluates to 1 on } \bar{t}.\]

if and only if \[\bar{t} \text{ satisfies any of the following conditions:}\]

1. \[\bar{t} \in UTP_{\bar{t}_1} (S) \text{ such that } p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}} = 0 \text{ on } \bar{t},\]
2. \[\bar{t} \in NFP_{\bar{t}_2,j_2} (S),\]
3. \[\bar{t} \in FP(S) \text{ such that } p_{1_1,j_1} = 1 \text{ on } \bar{t}, \text{ or}\]
4. \[\bar{t} \in FP(S) \text{ such that } p_{1_1,j_1+1,k_{i_1}} = 1 \text{ on } \bar{t}.\]

Hence, the result follows. \qed

Theorem 9.1.30 (CORF with LNF - Case 2)
Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i$-th term $p_i$ in $S$ is implemented as $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and the $j_1$-th literal $x_{j_1}^i$ in $p_i$ is negated where $p_i = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_{i_1}}$, $1 \leq i_1 \leq m$, $1 < j_1 < k_{i_1}$ and $k_{i_1}$ is the number of literals of $p_i$, the resulting implementation denoted as $I_{\text{CORF}}(p_{1_1} \rightarrow p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \mathcal{L}_{\text{NLF}}(p_{i_2} \rightarrow p_{i_2,j_2})$ is equivalent to Expression (8.30) in Table 8.12. Then, $S \neq I_{\text{CORF}}(p_{1_1} \rightarrow p_{1_1,j_1} + p_{1_1,j_1+1,k_{i_1}}) \mathcal{L}_{\text{NLF}}(p_{i_2} \rightarrow p_{i_2,j_2})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. \[\bar{t} \in FP(S) \text{ such that } p_{1_1,j_1} = 1 \text{ on } \bar{t}, \text{ or}\]
2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_{i_1}} = 1 \) on \( \bar{t} \).

**Proof:** First, we observe that \( S \oplus I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_1 \rightarrow p_{i_1,j_1})} \)

\[
\equiv (p_{i_1} \oplus (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}))(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\equiv (p_{i_1} \bar{p}_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}})(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\equiv (0 + \bar{p}_1 (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}})(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\]

(By making use of \( ABC(AB + C) \equiv 0 \))

\[
\equiv (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}})(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\equiv (p_{i_1,j_1} \bar{p}_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}})(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\equiv p_{i_1,j_1} \bar{p}_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\]

Now, \( S(\bar{t}) \neq I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_1 \rightarrow p_{i_1,j_1})} \) (\( \bar{t} \))

if and only if \( S(\bar{t}) \oplus I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_1 \rightarrow p_{i_1,j_1})} \) (\( \bar{t} \)) = 1

if and only if \( p_{i_1,j_1} \bar{p}_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m) \) evaluates to 1 on \( \bar{t} \)

if and only if \( \bar{t} \) satisfies any of the following conditions:

1. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1} = 1 \) on \( \bar{t} \), or
2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_{i_1}} = 1 \) on \( \bar{t} \).

Hence, the result follows. \( \Box \)

**Theorem 9.1.31 (CORF with LOF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term \( p_{i_1} \) in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} \) and the \( j_2 \)-th literal \( x_{j_2} \) in the \( i_2 \)-th term \( p_{i_2} \) in \( S \) is omitted where \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_{i_1}} \), \( 1 \leq i_1 < i_2 \leq m \), \( 1 \leq j_1 < k_{i_1} \), \( 1 \leq j_2 \leq k_{i_2} \), \( k_{i_2} > 1 \) and \( k_{i_1} \) and \( k_{i_2} \) are the numbers of literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, the resulting implementation denoted as \( I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_2 \rightarrow p_{i_2,j_2})} \) is equivalent to Expression (8.31) in Table 8.12. Then, \( S \neq I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_2 \rightarrow p_{i_2,j_2})} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in NFP_{i_2,j_2}(S) \),
2. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1} = 1 \) on \( \bar{t} \), or
3. \( \bar{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_{i_1}} = 1 \) on \( \bar{t} \).

**Proof:** First, we observe that \( S \oplus I_{CORF(p_1 \rightarrow p_{i_1,j_1},p_{i_1,j_1+1,k_{i_1}})} \) \( \chi_{LOF(p_2 \rightarrow p_{i_2,j_2})} \)

\[
\equiv ((p_{i_1} + p_{i_2}) \oplus (p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2}))(\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m)
\equiv ((p_{i_1} + p_{i_2})p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2} + (p_{i_1} + p_{i_2})(p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2})
\]

\( \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \)
\[
\equiv ((p_{i_1} + p_{i_2})(p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})\bar{p}_{i_2,j_2} + \bar{p}_{i_1} \bar{p}_{i_2}(p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}} + p_{i_2,j_2}))
\]
\[
\cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]
\[
\equiv (0 + \bar{p}_{i_1} \bar{p}_{i_2}(p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}) + \bar{p}_{i_1} \bar{p}_{i_2} p_{i_2,j_2})\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]
(By making use of \(AB(A+B) = 0\) and \(AB(A) = 0\))
\[
\equiv (\bar{p}_{i_1} \bar{p}_{i_2}(p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}) + \bar{p}_{i_1} \bar{p}_{i_2} p_{i_2,j_2})\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]
(By rewriting \((AB)(A)\) as \((AB)(A)\) because they are equivalent)
\[
\equiv (p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]
\[
+ p_{i_2,j_2} \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_{i_2-1} \bar{p}_{i_2+1} \cdots \bar{p}_m
\]
\[
\equiv (p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})\bar{S} + p_{i_2,j_2}\bar{S}
\]
\[
\equiv p_{i_2,j_2}\bar{S} + p_{i_1,1,j_1}\bar{S} + p_{i_1,j_1+1,k_{i_1}}\bar{S}
\]

Now,
\[
S(\vec{t}) \neq I_{\text{CORF}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})}(\text{LOF}(p_{i_2} - p_{i_2,j_2}))\bar{t}
\]
if and only if
\[
S(\vec{t}) \oplus I_{\text{CORF}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})}(\text{LOF}(p_{i_2} - p_{i_2,j_2}))\bar{t}) = 1
\]
if and only if
\[
p_{i_2,j_2}, S + p_{i_1,1,j_1}\bar{S} + p_{i_1,j_1+1,k_{i_1}}\bar{S}
\]
evaluates to 1 on \(\vec{t} \).

if and only if \(\vec{t} \) satisfies any of the following conditions:

1. \(\vec{t} \in NFP_{i_2,j_2}(S)\),

2. \(\vec{t} \in FP(S)\) such that \(p_{i_1,1,j_1} = 1\) on \(\vec{t}\), or

3. \(\vec{t} \in FP(S)\) such that \(p_{i_1,j_1+1,k_{i_1}} = 1\) on \(\vec{t}\).

Hence, the result follows. \(\square\)

**Theorem 9.1.32** \(\text{(CORF with LOF - Case 2)}\)

Let \(S = p_1 + \cdots + p_m\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term \(p_{i_1}\) in \(S\) is implemented as \(p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}}\) and the \(j_1\)-th literal \(x_{j_1}^1\) in the \(i_1\)-th term \(p_{i_1}\) in \(S\) is omitted where \(p_{i_1} = p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}},\ 1 \leq i_1 \leq m, \ 1 < j_1 < k_{i_1}\), and \(k_{i_1}\) is the number of literals of \(p_{i_1}\), the resulting implementation denoted as \(I_{\text{CORF}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})}(\text{LOF}(p_{i_1} - p_{i_1,j_1}))\) is equivalent to Expression (8.32) in Table 8.12. Then, \(S \neq I_{\text{CORF}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})}(\text{LOF}(p_{i_1} - p_{i_1,j_1}))\) if and only if there is a test case \(\vec{t} \) that satisfies any of the following conditions:

1. \(\vec{t} \in FP(S)\) such that \(p_{i_1,1,j_1-1} = 1\) on \(\vec{t}\), or

2. \(\vec{t} \in FP(S)\) such that \(p_{i_1,j_1+1,k_{i_1}} = 1\) on \(\vec{t}\).

**Proof:** First, we observe that \(S \oplus I_{\text{CORF}(p_{i_1} - p_{i_1,1,j_1} + p_{i_1,j_1+1,k_{i_1}})}(\text{LOF}(p_{i_1} - p_{i_1,j_1}))\)
\[
\equiv (p_{i_1} \oplus (p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_{i_1}}))\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv (p_{i_1}(p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_{i_1}}) + \bar{p}_{i_1}(p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_{i_1}}))\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
\[
\equiv (0 + \bar{p}_{i_1}(p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_{i_1}}))\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]
(By making use of \(ABC(A+C) = 0\))
\[
\equiv (p_{i_1,1,j_1-1} + p_{i_1,j_1+1,k_{i_1}})\bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m
\]

223
\[ \equiv (p_{i_1,j_1-1} + p_{i_1,j_1+1,k_1}) \overline{S} \]
\[ \equiv p_{i_1,j_1-1} \overline{S} + p_{i_1,j_1+1,k_1} \overline{S} \]

Now, \[ S(\vec{t}) \neq I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{\text{LOF}}(p_{i_1} - p_{i_1,j_2})(\vec{t}) \]
if and only if \[ S(\vec{t}) \oplus I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{\text{LOF}}(p_{i_1} - p_{i_1,j_2})(\vec{t}) = 1 \]
if and only if \[ \exists \vec{t}' \text{ such that } p_{i_1,j_1-1} \overline{S} + p_{i_1,j_1+1,k_1} \overline{S} \text{ evaluates to 1 on } \vec{t}'. \]
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1-1} = 1 \) on \( \vec{t} \), or

2. \( \vec{t} \in \text{FP}(S) \) such that \( p_{i_1,j_1+1,k_1} = 1 \) on \( \vec{t} \).

Hence, the result follows. \[ \square \]

**Theorem 9.1.33** (CORF with LIF - Case 1)

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_1} \) and the literal \( x_{i_2} \) is inserted into the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) where \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_1}, 1 \leq i_1 < i_2 \leq m, 1 \leq j_1 < k_1, k_1 \)
is the number of literals of \( p_{i_1} \) and \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the resulting implementation denoted as \( I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{\text{LOF}}(p_{i_1} - p_{i_1,j_2}) \overline{x_{i_2}} \) is equivalent to Expression (8.33) in Table 8.12. Then, \( S \neq I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{\text{LOF}}(p_{i_1} - p_{i_1,j_2}) \overline{x_{i_2}} \)
if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \text{UTP}_{x_{i_2}}(S) \) such that \( p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + x_{i_2} = 0 \) on \( \vec{t} \),

2. \( \vec{t} \in \text{FP}(S) \), such that \( p_{i_1,j_1} = 1 \) on \( \vec{t} \), or

3. \( \vec{t} \in \text{FP}(S) \), such that \( p_{i_1,j_1+1,k_1} = 1 \) on \( \vec{t} \).

**Proof:** First, we observe that \( S \oplus I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{\text{LOF}}(p_{i_1} - p_{i_1,j_2}) \overline{x_{i_2}} \)
\[ \equiv ((p_{i_1} + p_{i_2}) \oplus (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2})) \overline{p_{i_1+1}} \cdots \overline{p_{i_1+1}} \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ \equiv (p_{i_1} + p_{i_2}) \overline{(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2})} + (p_{i_1} + p_{i_2}) (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2})) \]
\[ \cdot \overline{p_{i_1+1}} \cdots \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ \equiv (p_{i_1} + p_{i_2}) \overline{(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2})} + p_{i_1} \overline{p_{i_2}} (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2}) \]
\[ \cdot \overline{p_{i_1+1}} \cdots \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ \equiv (0 + (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2}) + p_{i_1} \overline{p_{i_2}} (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2} x_{i_2}) \]
\[ \cdot \overline{p_{i_1+1}} \cdots \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ \text{(By making use of } AB(A + B) \equiv 0, A(AB) \equiv AB \text{ and } (A)(AC) \equiv 0) \]
\[ \equiv ((p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{p_{i_1} p_{i_2} x_{i_2}} + p_{i_1} \overline{p_{i_2}} (p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \]
\[ \cdot \overline{p_{i_1+1}} \cdots \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ (\text{By rewriting } A + B \text{ as } (A + B)(AB) \text{ because they are equivalent}) \]
\[ \equiv (p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{p_{i_2} x_{i_2}} \overline{p_{i_1+1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ + (p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \overline{p_{i_1} p_{i_2} x_{i_2}} \overline{p_{i_1+1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
\[ + \overline{p_{i_1+1}} \overline{p_{i_1+1}} \overline{p_{i_2+1}} \cdots \overline{p_{m}} \]
Theorem 9.1.34 (CORF with LIF - Case 2)

Let $S=p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that $p_{i_1}$ in $S$ is implemented as $p_{i_1} = p_{i_1,j_1} + p_{i_1,j_1+1,k_1}$ and the literal $x_{i_2}$ is inserted into $p_{i_1}$, where $p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_1}$, $1 \leq i_1 \leq m$, $1 \leq j_1 < k_1$, $k_1$ is the number of literals of $p_{i_1}$ and $x_{i_2}$ is a missing literal of $p_{i_1}$, the resulting implementation denoted as $I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1})_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1} x_{i_2})$ is equivalent to Expression (8.34) in Table 8.12. Then, $S \not\equiv I_{\text{CORF}}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1})_{\text{LIF}}(p_{i_1} \rightarrow p_{i_1} x_{i_2})$, if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in FP(S)$ such that $p_{i_1,1,j_1} \cdot x_{i_2} = 1$ on $\bar{t}$,

2. $\bar{t} \in FP(S)$ such that $p_{i_1,1,j_1+1,k_1} = 1$ on $\bar{t}$.

Hence, the result follows.

Proof: First, we observe that $S \equiv (p_{i_1} \oplus (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m$

$\equiv (p_{i_1} (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1}) + \bar{p}_{i_1} (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m$

$\equiv (0 + \bar{p}_{i_1} (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1})) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m$

(By making use of $AB(AC+B) \equiv 0$)

$\equiv (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1}) \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m$

$\equiv (p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1}) S$

$\equiv p_{i_1,1,j_1} x_{i_2} + p_{i_1,j_1+1,k_1} S$

$\equiv p_{i_1,1,j_1} x_{i_2} \overline{S} + p_{i_1,j_1+1,k_1} \overline{S}$
Now, \[ S(\vec{t}) \neq I_{CORF}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \land LRF(p_{i_1} \rightarrow p_{i_1,x_2})(\vec{t}) \]
if and only if
\[ S(\vec{t}) \oplus I_{CORF}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \land LRF(p_{i_1} \rightarrow p_{i_1,x_2})(\vec{t}) = 1 \]
if and only if
\[ p_{i_1,j_1,x_2} \overline{S} + p_{i_1,j_1+1,k_1} \overline{S} \] evaluates to 1 on \( \vec{t} \)
if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1,x_2} = 1 \) on \( \vec{t} \), or
2. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_1} = 1 \) on \( \vec{t} \).

Hence, the result follows. \( \square \)

**Theorem 9.1.35 (CORF with LRF - Case 1)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term \( p_{i_1} \) in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_1} \) and the \( j_2 \)-th literal \( x_{i_2} \) in the \( i_2 \)-th term \( p_{i_2} \) in \( S \) is replaced by the literal \( x_{i_2} \) where \( p_{i_1} = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_1} \), \( 1 \leq i_1 < i_2 \leq m \), \( 1 \leq j_1 < k_1 \), \( 1 \leq j_2 \leq k_2 \), \( k_1 \) and \( k_2 \) are the numbers of literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, and \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the resulting implementation denoted as \( I_{CORF}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \land LRF(p_{i_2} \rightarrow p_{i_2,j_2}x_{i_2}) \) is equivalent to Expression (8.35) in Table 8.12. Then, \( S \neq I_{CORF}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \land LRF(p_{i_2} \rightarrow p_{i_2,j_2}x_{i_2}) \), if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in UTP_{i_2}(S) \) such that \( p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + x_{i_2} = 0 \) on \( \vec{t} \), or
2. \( \vec{t} \in NFP_{i_2,j_2}(S) \) such that \( x_{i_2} = 1 \) on \( \vec{t} \),
3. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1} = 1 \) on \( \vec{t} \), or
4. \( \vec{t} \in FP(S) \) such that \( p_{i_1,j_1+1,k_1} = 1 \) on \( \vec{t} \).

**Proof:** First, we observe that \( S \oplus I_{CORF}(p_{i_1} \rightarrow p_{i_1,j_1} + p_{i_1,j_1+1,k_1}) \land LRF(p_{i_2} \rightarrow p_{i_2,j_2}x_{i_2}) \)
\[ \equiv ((p_{i_1} + p_{i_2}) \oplus (p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2,j_2}x_{i_2})) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2-1}} \overline{p_{i_2+1}} \cdots \overline{p_m} \]
\[ \equiv ((p_{i_1} + p_{i_2})(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2,j_2}x_{i_2}) \oplus (p_{i_1} + p_{i_2})(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_2,j_2}x_{i_2})) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2-1}} \overline{p_{i_2+1}} \cdots \overline{p_m} \]
\[ \equiv (0 + (p_{i_1,j_1} + p_{i_1,j_1+1,k_1})p_{i_2,j_2}x_{i_2} + \overline{p_1} \overline{p_2}(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + \overline{p_1} \overline{p_2}p_{i_2,j_2}x_{i_2})) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2-1}} \overline{p_{i_2+1}} \cdots \overline{p_m} \]
(By making use of \( AB(A + B) \equiv 0 \) and \( AB(\overline{AC}) \equiv AB\overline{C} \))
\[ \equiv ((p_{i_1,j_1} + p_{i_1,j_1+1,k_1})p_{i_2,j_2}x_{i_2} + \overline{p_1} \overline{p_2}(p_{i_1,j_1} + p_{i_1,j_1+1,k_1} + p_{i_1}p_{i_2}p_{i_2,j_2}x_{i_2})) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2-1}} \overline{p_{i_2+1}} \cdots \overline{p_m} \]
(By rewriting \( A + B \) as \( (A + B)(\overline{AB}) \) because they are equivalent; and \( AB)(A) \) as \( (AB)(AB) \) because they are equivalent)
\[ \equiv (p_{i_1,j_1} + p_{i_1,j_1+1,k_1})p_{i_2,j_2}x_{i_2} \cdot \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_{i_2-1}} \overline{p_{i_2+1}} \cdots \overline{p_m} \]

226
First, we observe that

\[
\begin{align*}
\text{Proof :} \\
S\text{pression (8.36) in Table 8.12. Then,} \\
\text{the number of literals of} \\
\equiv p_2(p_{1,1,j_1} + p_{1,j_1+1,k_1} + x_{i_2}) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_{m} + p_{i_2,j_2} x_{i_2} \mathcal{S} \\
\equiv p_2(\bar{p}_{1,1,j_1} + p_{1,j_1+1,k_1} + x_{i_2}) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_{m} + p_{i_2,j_2} x_{i_2} \mathcal{S} + p_{1,1,j_1} \mathcal{S} \\
\text{Now,} \\
S(\bar{t}) \notin \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \\
\text{if and only if} \\
S(\bar{t}) \oplus \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \bar{t} = 1 \\
\text{if and only if} \\
p_2(p_{1,1,j_1} + p_{1,j_1+1,k_1} + x_{i_2}) \cdot \tilde{p}_1 \cdots \tilde{p}_{i_1-1} \tilde{p}_{i_1+1} \cdots \tilde{p}_{m} + p_{i_2,j_2} x_{i_2} \mathcal{S} + p_{1,1,j_1} \mathcal{S} \\
\text{evaluates to 1 on} \bar{t} \\
\text{if and only if} \bar{t} \text{satisfies any of the following conditions:} \\
1. \bar{t} \in \text{UTP}_{i_2}(S) \text{ such that} \ p_{1,1,j_1} + p_{1,j_1+1,k_1} + x_{i_2} = 0 \\
\text{on} \bar{t}, \text{ or} \\
2. \bar{t} \in \text{NFP}_{i_2,j_2}(S) \text{ such that} \ x_{i_2} = 1 \text{ on} \bar{t}, \\
3. \bar{t} \in \text{FP}(S) \text{ such that} \ p_{1,1,j_1} = 1 \text{ on} \bar{t}, \text{ or} \\
4. \bar{t} \in \text{FP}(S) \text{ such that} \ p_{1,j_1+1,k_1} = 1 \text{ on} \bar{t}.
\]

Hence, the result follows. \( \square \)

**Theorem 9.1.36 (CORF with LRF - Case 2)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i_1 \)-th term \( p_{i_1} \) in \( S \) is implemented as \( p_{1,1,j_1} + p_{1,j_1+1,k_1} \) and the \( j_1 \)-th literal \( x^{i_1}_{j_1} \) in the \( i_1 \)-th term \( p_{i_1} \) in \( S \) is replaced by \( x_{i_2} \) where \( p_{i_1} = p_{1,1,j_1} \cdot p_{1,j_1+1,k_1} \), \( 1 \leq i_1 \leq m, 1 \leq j_1 < k_1 \), \( k_1 \) is the number of literals of \( p_{i_1} \) and \( x_{i_2} \) is a missing literal of \( p_{i_1} \), the resulting implementation denoted as \( I_{\text{CORF}}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) x_{i_2} \) is equivalent to Expression (8.36) in Table 8.12. Then, \( S \neq I_{\text{CORF}}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) x_{i_2} \) if and only if there is a test case \( \bar{t} \) that satisfies any of the following conditions:

1. \( \bar{t} \in \text{FP}(S) \) such that \( p_{1,1,j_1-1} x_{i_2} = 1 \) on \( \bar{t} \), or
2. \( \bar{t} \in \text{FP}(S) \) such that \( p_{1,j_1+1,k_1} = 1 \) on \( \bar{t} \).

**Proof :** First, we observe that \( S \oplus I_{\text{CORF}}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) \mathcal{X} \mathcal{LRF}(p_{1,1,j_1} + p_{1,j_1+1,k_1}) x_{i_2} \)

\[
\equiv (p_1 \oplus (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1})) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \\
\equiv (p_1 (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1}) + \bar{p}_1 (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1})) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \\
\equiv (0 + \bar{p}_1 (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1})) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \\
\text{(By making use of } ABC(AD + C) \equiv 0) \\
\equiv (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1}) \cdot \bar{p}_1 \cdots \bar{p}_{i_1-1} \bar{p}_{i_1+1} \cdots \bar{p}_m \\
\equiv (p_{1,1,j_1-1} x_{i_2} + p_{1,j_1+1,k_1}) \mathcal{S} \\
\equiv p_{1,1,j_1-1} x_{i_2} \mathcal{S} + p_{1,j_1+1,k_1} \mathcal{S}
\]
Now, \( S(\vec{t}) \neq I_{\text{CORF}}(p_{i_1,1,j_1} + p_{i_2,1,j_2} + \lambda_{i_1}) \cdot L_{\text{RF}}(p_{i_1,1,j_1} x_{i_2})(\vec{t}) \)
if and only if
\[ S(\vec{t}) \oplus I_{\text{CORF}}(p_{i_1,1,j_1} + p_{i_2,1,j_2} + \lambda_{i_1}) \cdot L_{\text{RF}}(p_{i_1,1,j_1} x_{i_2})(\vec{t}) = 1 \]
if and only if
\[ p_{i_1,1,j_1-1}x_{i_2} + p_{i_2,1,j_2+1} \bar{S} \] evaluates to 1 on \( \vec{t} \)
if and only if
\( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in FP(S) \) such that \( p_{i_1,1,j_1-1}x_{i_2} = 1 \) on \( \vec{t} \), or
2. \( \vec{t} \in FP(S) \) such that \( p_{i_2,1,j_2+1} = 1 \) on \( \vec{t} \).

Hence, the result follows.

It should be noted that, when \( j_1 = 1 \), the term \( p_{i_1,1,j_1-1}x_{i_2} \) in the first detection condition in Theorem 9.1.36 degenerates to \( x_{i_2} \).

### 9.2 Detection conditions of 6 Remaining Double-fault Expressions with Ordering

In this section, we study the detection conditions of 6 double-fault expressions in double faults with ordering which do not have their equivalent counterparts in double faults without ordering.

**Theorem 9.2.1 (ENF \( \times \) LIF - Case 3)**

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the subexpression \( p_1 + \cdots + p_{h_1} \) in \( S \) is negated and then the literal \( x_{i_2} \) is inserted into the newly created term, \( \bar{p_1} + \cdots + \bar{p}_{h_1} \), where \( 1 \leq i_1 < h_1 \leq m \) and \( x_{i_2} \) is a literal of \( S \), the resulting implementation denoted as \( I_{\text{ENF}}(p_1 + p_{h_1} + \cdots + p_m) \cdot L_{\text{RF}}(p_{i_1} + \cdots + p_{h_1} - p_{i_1} + \cdots + p_{h_1} x_{i_2}) \)

is equivalent to Expression (8.43) in Table 8.12. Then, \( S \neq I_{\text{ENF}}(p_1 + p_{h_1} + \cdots + p_m) \cdot L_{\text{RF}}(p_{i_1} + p_{i_2} - p_{i_2} x_{i_2}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in \bigcup_{i_1} TP_i(S) \setminus \bigcup_{i \neq i_1} TP_i(S) \), or
2. \( \vec{t} \in FP(S) \) such that \( x_{i_2} = 1 \) on \( \vec{t} \).

**Proof:** First, we observe that \( S \oplus I_{\text{ENF}}(p_1 + p_{h_1} + \cdots + p_{h_1} x_{i_2}) \)
\[ \equiv (p_1 + \cdots + p_{h_1}) \oplus \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} \]
\[ \equiv (p_1 + \cdots + p_{h_1}) (p_1 + \cdots + p_{h_1} x_{i_2}) + (p_1 + \cdots + p_{h_1}) (1 + \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2}) \]
\[ \\
\equiv (p_1 + \cdots + p_{h_1}) \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} \]
\[ \\
\equiv (p_1 + \cdots + p_{h_1}) \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} + \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} \]

(By making use of \( A \cdot \bar{A} = A \))
\[ \equiv (p_1 + \cdots + p_{h_1}) \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} + \bar{p}_1 + \cdots + \bar{p}_{h_1} x_{i_2} \]

228
\(\equiv (p_{i_1} + \cdots + p_{i_m})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + x_{l_2} S\)

Now, \(S(\bar{t}) \neq I_{\text{EF}}(p_{i_1} + \cdots + p_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2}) (\bar{t})\)

if and only if \(S(\bar{t}) \oplus I_{\text{EF}}(p_{i_1} + \cdots + p_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2}) (\bar{t}) = 1\)

if and only if \((p_{i_1} + \cdots + p_{i_1})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + x_{l_2} \bar{S}\) evaluates to 1 on \(\bar{t}\)

if and only if \(\bar{t}\) satisfies any of the following conditions:

1. \(\bar{t} \in \left( \bigcup_{i=1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq 1, \ldots, h_1} TP_i(S) \right),\) or

2. \(\bar{t} \in FP(S)\) such that \(x_{l_2} = 1\) on \(\bar{t}\).

Hence, the result follows. \(\square\)

**Theorem 9.2.2 (TNNF \(\otimes\) LIF - Case 3)**

Let \(S=p_{i_1} + \cdots + p_{i_m}\) be a Boolean specification in IDNF. Suppose that the \(i_1\)-th term, \(p_{i_1}\), in \(S\) is negated and then the literal \(x_{l_2}\) is inserted into the newly created term \(\bar{p}_{i_1}\), where \(1 \leq i_1 \leq m\) and \(x_{l_2}\) is a literal of \(S\), the resulting implementation denoted as \(I_{\text{EF}}(p_{i_1} - \bar{p}_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2})\) is equivalent to Expression (8.52) in Table 8.12. Then, \(S \neq I_{\text{EF}}(p_{i_1} - \bar{p}_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2})\) if and only if there is a test case \(\bar{t}\) that satisfies any of the following conditions:

1. \(\bar{t} \in UTP_{i_1}(S),\) or

2. \(\bar{t} \in FP(S)\) such that \(x_{l_2} = 1\) on \(\bar{t}\).

**Proof:** First, we observe that \(S \oplus I_{\text{EF}}(p_{i_1} - \bar{p}_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2})\)

\(\equiv (p_{i_1} + \bar{p}_{i_1} x_{l_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m\)

\(\equiv (p_{i_1} \bar{p}_{i_1} x_{l_2} + \bar{p}_{i_1} x_{l_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m\)

\(\equiv (p_{i_1} + \bar{p}_{i_1} x_{l_2})\bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m\)

\(\quad (By \ making \ use \ of \ A \cdot \overline{AB} \equiv A)\)

\(\equiv p_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + x_{l_2} \bar{p}_1 \cdots \bar{p}_i \cdots \bar{p}_m\)

\(\equiv p_{i_1} \bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + x_{l_2} \bar{S}\)

Now, \(S(\bar{t}) \neq I_{\text{EF}}(p_{i_1} - \bar{p}_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2}) (\bar{t})\)

if and only if \(S(\bar{t}) \oplus I_{\text{EF}}(p_{i_1} - \bar{p}_{i_1})\wedge L(\bar{p}_{i_1}-\bar{p}_{i_1} x_{l_2}) (\bar{t}) = 1\)

if and only if \(p_{i_1} x_{l_2} \bar{p}_1 \cdots \bar{p}_{i_1-1}\bar{p}_{i_1+1} \cdots \bar{p}_m + x_{l_2} \bar{S}\) evaluates to 1 on \(\bar{t}\)

if and only if \(\bar{t}\) satisfies any of the following conditions:

1. \(\bar{t} \in UTP_{i_1}(S),\) or

2. \(\bar{t} \in FP(S)\) such that \(x_{l_2} = 1\) on \(\bar{t}\).
Hence, the result follows. \qed

**Theorem 9.2.3 (CORF with LIF - Case 2(b))**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i_1$-th term $p_{i_1}$ in $S$ is implemented as $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and then the $(j_1 + 1)$-th literal $x_{i_1,j_1+1}^j$ of $p_{i_1}$ is inserted into the newly created term $p_{i_1,j_1}$ where $p_{i_1} = p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$, $1 \leq i_1 \leq m$, $1 \leq j_1 < k_{i_1}$ and $k_{i_1}$ is the number of literals of $p_{i_1}$, the resulting implementation denoted as $I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{CORF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ is equivalent to Expression (8.73) in Table 8.12. Then, $S \notin I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{CORF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:

1. $\bar{t} \in FP(S)$, such that $p_{i_1,j_1+1} = 1$ on $\bar{t}$, or
2. $\bar{t} \in FP(S)$, such that $p_{i_1,j_1,k_{i_1}} = 1$ on $\bar{t}$.

**Proof:** First, we observe that $S \oplus I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{LIF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ is equivalent to Expression (8.73) in Table 8.12. Then, $S \notin I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{LIF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ if and only if $\bar{t}$ satisfies any of the following conditions:

1. $\bar{t} \in FP(S)$ such that $p_{i_1,j_1+1} = 1$ on $\bar{t}$, or
2. $\bar{t} \in FP(S)$ such that $p_{i_1,j_1,k_{i_1}} = 1$ on $\bar{t}$.

Hence, the result follows. \qed

**Theorem 9.2.4 (CORF with LIF - Case 2(c))**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i_1$-th term $p_{i_1}$ in $S$ is implemented as $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$ and then the negation of the $(j_1 + 1)$-th literal $\bar{x}_{i_1,j_1+1}^j$ of $p_{i_1}$ is inserted into the newly created term $p_{i_1,j_1}$ where $p_{i_1} = p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}$, $1 \leq i_1 \leq m$, $1 \leq j_1 < k_{i_1}$ and $k_{i_1}$ is the number of literals of $p_{i_1}$, the resulting implementation denoted as $I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{CORF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ is equivalent to Expression (8.74) in Table 8.12. Then, $S \notin I_{\text{CORF}}(p_{i_1} - p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}}) \text{LIF}(p_{i_1,j_1} - p_{i_1,j_1+1,k_{i_1}})$ if and only if there is a test case $\bar{t}$ that satisfies any of the following conditions:
1. \( \vec{t} \in FP(S) \), such that \( p_{i_1,j_1} \vec{x}_{j_1+1} = 1 \) on \( \vec{t} \), or

2. \( \vec{t} \in FP(S) \), such that \( p_{i_1,j_1,k_i} = 1 \) on \( \vec{t} \).

Proof: First, we observe that \( S \in I_{\text{CORF}}(p_{i_1} + p_{i_1,j_1} + p_{i_1,j_1+1}) \) if and only if \( p_{i_1,j_1} \vec{x}_{j_1+1} = 1 \) on \( \vec{t} \).

Now, \( S(\vec{t}) \neq I_{\text{CORF}}(p_{i_1} + p_{i_1,j_1} + p_{i_1,j_1+1}) \) if and only if \( p_{i_1,j_1} \vec{x}_{j_1+1} = 1 \) on \( \vec{t} \).

Hence, the result follows.

\[ \square \]

Theorem 9.2.5 (CORF with LRF - Case 2(b))

Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( i \)-th term, \( p_i \), in \( S \) is implemented as \( p_{i_1,j_1} + p_{i_1,j_1+1,k_i} \) and then the literal \( \vec{x}_{j_1} \) in the newly created term \( p_{i_1,j_1} \) is replaced by \( x_{j_1+1} \) where \( p_i = p_{i_1,j_1} \cdot p_{i_1,j_1+1,k_i} \), \( 1 \leq i \leq m \), \( 1 \leq j_1 < k_i \) and \( k_i \) is the number of literals of \( p_i \), the resulting implementation denoted as \( I_{\text{CORF}}(p_{i_1,j_1} - p_{i_1,j_1+1}, x_{j_1+1}) \) is equivalent to Expression (8.77) in Table 8.12. Then, \( S \neq I_{\text{CORF}}(p_{i_1,j_1} - p_{i_1,j_1+1}, x_{j_1+1}) \) if and only if there is a test case \( \vec{t} \) that satisfies any of the following conditions:

1. \( \vec{t} \in FP(S) \), such that \( p_{i_1,j_1} \vec{x}_{j_1+1} = 1 \) on \( \vec{t} \), or

2. \( \vec{t} \in FP(S) \), such that \( p_{i_1,j_1,k_i} = 1 \) on \( \vec{t} \).

Proof: First, we observe that \( S \in I_{\text{CORF}}(p_{i_1,j_1} + p_{i_1,j_1+1,k_i}) \) if and only if \( p_{i_1,j_1} \vec{x}_{j_1+1} = 1 \) on \( \vec{t} \).
Proof: First, we observe that $S \oplus I_{\text{CORF}(p_1 \cdot p_{1,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \equiv 0$.

\[
\begin{align*}
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_m} \\
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{S} \\
&= p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} \overline{S} + p_{i_1,j_1+1,k_{i_1}} \overline{S}
\end{align*}
\]

Now, $S(\bar{i}) \neq I_{\text{CORF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \oplus I_{\text{LRF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \oplus I_{\text{LRF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})}(\bar{i}) = 1$

if and only if $p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} \overline{S} + p_{i_1,j_1+1,k_{i_1}} \overline{S}$ evaluates to 1 on $\bar{i}$

if and only if $\bar{i}$ satisfies any of the following conditions:

1. $\bar{i} \in \text{FP}(S)$, such that $p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} = 1$ on $\bar{i}$, or

2. $\bar{i} \in \text{FP}(S)$, such that $p_{i_1,j_1+1,k_{i_1}} = 1$ on $\bar{i}$.

Hence, the result follows. \qed

**Theorem 9.2.6 (CORF with LRF - Case 2(c))**

Let $S = p_1 + \cdots + p_m$ be a Boolean specification in irredundant disjunctive normal form. Suppose that the $i$-th term $p_{i_1}$ in $S$ is implemented as $p_{1,i,j_1} + p_{1,i,j_1+1,k_{i_1}}$ and the $j_1$-th literal $x_{j_1}^{i_1}$ in the newly created term $p_{1,i,j_1}$ is replaced by $\bar{x}_{j_1}^{i_1}$ where $p_{i_1} = p_{1,i,j_1} \cdot p_{1,i,j_1+1,k_{i_1}}$, $1 \leq i_1 \leq m$, $1 \leq j_1 < k_{i_1}$ and $k_{i_1}$ is the number of literals of $p_{i_1}$, the resulting implementation denoted as $I_{\text{CORF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \oplus I_{\text{LRF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \oplus I_{\text{LRF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})}$ is equivalent to Expression (8.78) in Table 8.12. Then, $S \neq 0$ if and only if there is a test case $\bar{i}$ that satisfies any of the following conditions:

1. $\bar{i} \in \text{FP}(S)$, such that $p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} = 1$ on $\bar{i}$, or

2. $\bar{i} \in \text{FP}(S)$, such that $p_{i_1,j_1+1,k_{i_1}} = 1$ on $\bar{i}$.

**Proof:** First, we observe that $S \oplus I_{\text{CORF}(p_1 \cdot p_{1,i,j_1-1+i_{i_1}} + p_{i_1,j_1+1,k_{i_1}})} \equiv 0$.

\[
\begin{align*}
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_m} \\
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{S} \\
&= p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} \overline{S} + p_{i_1,j_1+1,k_{i_1}} \overline{S}
\end{align*}
\]

(BY making use of $ABCD(AC + CD) \equiv 0$)

\[
\begin{align*}
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{p_1} \cdots \overline{p_{i_1-1}} \overline{p_{i_1+1}} \cdots \overline{p_m} \\
&= \left( p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} + p_{i_1,j_1+1,k_{i_1}} \right) \overline{S} \\
&= p_{1,i,j_1-1} \cdot x_{j_1+1}^{i_{i_1}} \overline{S} + p_{i_1,j_1+1,k_{i_1}} \overline{S}
\end{align*}
\]
Now, \( S(\vec{t}) \neq I_{\text{CORF}}(p_{i1} \rightarrow p_{i1,j_1} + p_{i1,j_1+1,k_1}) \otimes \text{LRF}(p_{i1,j_1} \rightarrow p_{i1,j_1-1} \bar{x}_{j_1+1}) \)(\(\vec{t}\)) if and only if \( S(\vec{t}) \oplus I_{\text{CORF}}(p_{i1} \rightarrow p_{i1,j_1} + p_{i1,j_1+1,k_1}) \otimes \text{LRF}(p_{i1,j_1} \rightarrow p_{i1,j_1-1} \bar{x}_{j_1+1}) \)(\(\vec{t}\)) = 1 \)

if and only if \( (p_{i1,j_1-1} \cdot \bar{x}_{j_1+1} + p_{i1,j_1+1,k_1})^S \) evaluates to 1 on \( \vec{t} \)

if and only if \( \vec{t} \) satisfies any of the following conditions:

1. \( \vec{t} \in FP(S) \), such that \( p_{i1,j_1-1} \cdot \bar{x}_{j_1+1} = 1 \) on \( \vec{t} \), or
2. \( \vec{t} \in FP(S) \), such that \( p_{i1,j_1+1,k_1} = 1 \) on \( \vec{t} \).

Hence, the result follows.

\[\square\]

9.3 Summary

For ease of reading, Table 9.2 lists all double fault classes studied in this chapter, the corresponding double-fault expression numbers and their corresponding fault detection conditions. For example, the second row of Table 9.2, which presents the detection conditions of two double-fault expressions of \( \text{ENF} \otimes \text{LOF} \). For double-fault expression (8.3) (please refer to Table 8.12 for the actual double-fault expression), the detection condition shows that any true point of \( S \) in \( \bigcup_{i=1}^{h_1} TP_i(S) \) \( \setminus \bigcup_{i \neq i_1,...,h_1} TP_i(S) \) such that \( p_{i_2,j_2} = 0 \) or any false point of \( S \) can distinguish \( S \) and the expression. While for double-fault expression (6) in Table 8.12, the detection condition shows that any true point of \( S \) in \( \bigcup_{i=1}^{h_1} TP_i(S) \) \( \setminus \bigcup_{i \neq i_1,...,h_1} TP_i(S) \) or any false point of \( S \) such that \( p_{i_2,j_2} = 0 \) can distinguish \( S \) and the expression.
Table 9.2: Double fault, double-fault expression and detection condition

(a) Double-fault expressions due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.)</th>
<th>Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF ∞ LNF</td>
<td>(8.1): (C1) any point in ( \bigcup_{i=1}^{h_1} T\gamma_i(S) \backslash \bigcup_{i\neq i_1,\ldots,i_2}^{m} T\gamma_i(S) ) such that ( p_{i_2,j_2} = 0 ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8.2): (C1) any point in ( \bigcup_{i=1}^{h_1} T\gamma_i(S) \backslash \bigcup_{i\neq i_1,\ldots,h_1}^{m} T\gamma_i(S) ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) ) such that ( p_{i_2,j_2} = 0 )</td>
<td></td>
</tr>
<tr>
<td>ENF ∞ LOF</td>
<td>(8.3): (C1) any point in ( \bigcup_{i=1}^{h_1} T\gamma_i(S) \backslash \bigcup_{i\neq i_1,\ldots,h_1}^{m} T\gamma_i(S) ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) )</td>
<td></td>
</tr>
<tr>
<td>ENF ∞ LIF</td>
<td>(8.5): (C1) any point in ( \bigcup_{i=1}^{h_1} T\gamma_i(S) \backslash \bigcup_{i\neq i_1,\ldots,h_1}^{m} T\gamma_i(S) ),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) ) such that ( x_{i_2} = 0 ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C3) any point in ( FP(S) )</td>
<td></td>
</tr>
<tr>
<td>ENF ∞ LRF</td>
<td>(8.7): (C1) any point in ( \bigcup_{i=1}^{h_1} T\gamma_i(S) \backslash \bigcup_{i\neq i_1,\ldots,h_1}^{m} T\gamma_i(S) ) such that ( p_{i_2,j_2} = 0 ),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) ) such that ( x_{i_2} = 0 ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C3) any point in ( FP(S) )</td>
<td></td>
</tr>
<tr>
<td>TNE ∞ LNF</td>
<td>(8.9): (C1) any point in ( T\eta_{i_1}(S) \backslash \bigcup_{i\neq i_1,i_2}^{m} T\gamma_i(S) ) such that ( p_{i_2,j_2} = 0 ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) )</td>
<td></td>
</tr>
<tr>
<td>TNE ∞ LOF</td>
<td>(8.11): (C1) any point in ( UT\eta_{i_1}(S) ) such that ( p_{i_2,j_2} = 0 ), or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C2) any point in ( FP(S) ) such that ( p_{i_1,j_2} = 0 )</td>
<td></td>
</tr>
</tbody>
</table>
Table 9.2 (cont'd) Double fault, double-fault expression and detection condition

(a) (cont’d) Double-fault expressions due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( TNF \ltimes LIF )</td>
<td>(8.13): (C1) any point in ( UTP_{i_1}(S) ), (C2) any point in ((TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)) such that ( x_{i_2} = 0 ), or (C3) any point in ( FP(S) )</td>
</tr>
<tr>
<td>( TNF \ltimes LRF )</td>
<td>(8.14): (C1) any point in ( UTP_{i_1}(S) ) such that ( x_{i_2} = 1 ), or (C2) any point in ( FP(S) )</td>
</tr>
<tr>
<td>( TOF \ltimes LNF )</td>
<td>(8.15): (C1) any point in ( UTP_{i_1}(S) ) such that ( p_{i_2, j_2} = 0 ), (C2) any point in ( TP_{i_2}(S) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S) ) such that ( x_{i_2} = 0 ), or (C3) any point in ( FP(S) ) such that ( p_{i_1, j_2} = 0 )</td>
</tr>
<tr>
<td>( TOF \ltimes LOF )</td>
<td>(8.16): (C1) any point in ( UTP_{i_1}(S) ) such that ( x_{i_2} = 1 ), (C2) any point in ( FP(S) ) such that ( x_{i_2} = 0 ), or (C3) any point in ( FP(S) ) such that ( p_{i_1, j_2} = 0 )</td>
</tr>
<tr>
<td>( TOF \ltimes LIF )</td>
<td>(8.17): (C1) any point in ( TP_{i_1}(S) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S) ) such that ( p_{i_2, j_2} = 0 ), (C2) any point in ( TP_{i_2}(S) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S) ), or (C3) any point in ( NFP_{i_1,j_1}(S) )</td>
</tr>
<tr>
<td>( TOF \ltimes LRF )</td>
<td>(8.18): (C1) any point in ( UTP_{i_1}(S) ) such that ( p_{i_2, j_2} = 0 ), or (C2) any point in ( NFP(S)_{i_2, j_2} )</td>
</tr>
<tr>
<td>( TOF \ltimes LNF )</td>
<td>(8.19): (C1) any point in ( UTP_{i_1}(S) ), (C2) any point in ( UTP_{i_2}(S) ) such that ( x_{i_2} = 0 ), or (C3) any point in ((TP_{i_2}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)) such that ( x_{i_2} = 0 )</td>
</tr>
<tr>
<td>( TOF \ltimes LOF )</td>
<td>(8.20): (C1) any point in ( UTP_{i_1}(S) ) such that ( x_{i_2} = 0 ), (C2) any point in ( UTP_{i_2}(S) ) such that ( p_{i_2, j_2} = 0 ), (C3) any point in ( UTP_{i_2}(S) ) such that ( x_{i_2} = 0 ), (C4) any point in ( NFP_{i_1,j_1}(S) ) such that ( x_{i_2} = 1 ), or (C5) any point in ((TP_{i_2}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)) such that ( x_{i_2} = 0 )</td>
</tr>
</tbody>
</table>
Table 9.2 (cont’d) Double fault, double-fault expression and detection condition
(a) (cont’d) Double-fault expressions due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNF × DORF</td>
<td>(8.21) (C1) any point in $TP_{i_1}(S) \setminus \bigcup_{i \neq i_1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $p_{i_2, j_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $TP_{i_1 +1}(S) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $p_{i_2, j_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $UTP_{i_1}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $(TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C5) any point in $(TP_{i_1 +1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$, or</td>
</tr>
<tr>
<td></td>
<td>(C6) any point in $NFP_{i_2, j_2}(S)$</td>
</tr>
<tr>
<td>LOF × DORF</td>
<td>(8.22) (C1) any point in $TP_{i_1}(S) \setminus \bigcup_{i \neq i_1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $TP_{i_1 +1}(S) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $p_{i_1, j_2} = 0$</td>
</tr>
<tr>
<td>LOF × DORF</td>
<td>(8.23) (C1) any point in $UTP_{i_1}(S)$ such that $p_{i_2, j_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_1 +1}(S)$ such that $p_{i_2, j_2} = 0$, or</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $NFP_{i_2, j_2}(S)$</td>
</tr>
<tr>
<td>DORF × LNF</td>
<td>(8.24) (C1) any point in $UTP_{i_1}(S)$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_1 +1}(S)$ such that $p_{i_1, j_2} = 0$</td>
</tr>
<tr>
<td>LNF × DORF</td>
<td>(8.25) (C1) any point in $UTP_{i_1}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_1 +1}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $UTP_{i_2}(S)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $(TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $x_{i_2} = 0$, or</td>
</tr>
<tr>
<td></td>
<td>(C5) any point in $(TP_{i_1 +1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $x_{i_2} = 0$</td>
</tr>
<tr>
<td>LNF × DORF</td>
<td>(8.26) (C1) any point in $UTP_{i_1}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_1 +1}(S)$, or</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $(TP_{i_1}(S) \cap TP_{i_1 +1}(S)) \setminus \bigcup_{i \neq i_1 +1, i_2} \bigcup_{m} \cap_{i_1 +1, i_2} \cap_{m} TP_i(S)$ such that $x_{i_2} = 0$</td>
</tr>
</tbody>
</table>
Table 9.2 (cont’d) Double fault, double-fault expression and detection condition

(a) (cont’d) Double-fault expressions due to double faults without ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DORF LRF</td>
<td>(8.27):(C1) any point in $UTP_{i_k}(S)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_k}(S)$ such that $p_{i_2,j_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $UTP_{i_k+i}(S)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $UTP_{i_k+i}(S)$ such that $p_{i_2,j_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C5) any point in $UTP_{i_k}(S)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C6) any point in $NFP_{i_{j_k}}(S)$ such that $x_{i_2} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(C7) any point in $\left( \left( TP_{i_k}(S) \cap TP_{i_k}(S) \right) \setminus \bigcup_{i \neq i_1,i_2}^{m} TP_i(S) \right)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C8) any point in $\left( \left( TP_{i_k}(S) \cap TP_{i_k+i}(S) \right) \setminus \bigcup_{i = 1}^{m} TP_i(S) \right)$ such that $x_{i_2} = 0$</td>
</tr>
<tr>
<td></td>
<td>(8.28):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $UTP_{i_k}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $UTP_{i_k+i}(S)$ such that $x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $UTP_{i_k+i}(S)$ such that $p_{i_1,j_2} = 0$, or</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $\left( TP_{i_k}(S) \cap TP_{i_k+i}(S) \right) \setminus \bigcup_{i \neq i_1+1} TP_i(S)$ such that $x_{i_2} = 0$</td>
</tr>
<tr>
<td>CORF LNF</td>
<td>(8.29):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $UTP_{i_k}(S)$ such that $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $NFP_{i_{j_k}}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $FP(S)$ such that $p_{i_1,j_1} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(8.30):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $FP(S)$ such that $p_{i_1,j_1} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td>CORF LOF</td>
<td>(8.31):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $NFP_{i_{j_k}}(S)$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(8.32):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $FP(S)$ such that $p_{i_1,j_1-1,k_{i_1}} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td>CORF LIF</td>
<td>(8.33):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $UTP_{i_k}(S)$ such that $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(8.34):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $FP(S)$ such that $p_{i_1,j_1} + x_{i_2} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td>CORF LRF</td>
<td>(8.35):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $UTP_{i_k}(S)$ such that $p_{i_1,j_1} + p_{i_1,j_1+1,k_{i_1}} + x_{i_2} = 0$,</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $NFP_{i_{j_k}}(S)$ such that $x_{i_2} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(C3) any point in $FP(S)$ such that $p_{i_1,j_1} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C4) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$,</td>
</tr>
<tr>
<td></td>
<td>(8.36):</td>
</tr>
<tr>
<td></td>
<td>(C1) any point in $FP(S)$ such that $p_{i_1,j_1-1,x_{i_2}} = 1$, or</td>
</tr>
<tr>
<td></td>
<td>(C2) any point in $FP(S)$ such that $p_{i_1,j_1+1,k_{i_1}} = 1$</td>
</tr>
</tbody>
</table>
Table 9.2 (cont’d) Double fault, double-fault expression and detection condition

\[ S = p_1 + \ldots + p_m \]

(b) Six remaining double-fault expressions due to double faults with ordering

<table>
<thead>
<tr>
<th>Fault Class</th>
<th>(Expression No.): Detection Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENF ⋊ LIF</td>
<td>( (8.43): (C1) ) any point in ( \bigcup_{i=1}^{h_1} TP_i(S) \setminus \bigcup_{i \neq i_1, \ldots, h_1}^{m} TP_i(S) ), or ( (C2) ) any point in ( FP(S) ) such that ( x_{i_2} = 1 )</td>
</tr>
<tr>
<td>TNF ⋊ LIF</td>
<td>( (8.52): (C1) ) any point in ( UTP_{i_1}(S) ), or ( (C2) ) any point in ( FP(S) ) such that ( x_{i_2} = 1 )</td>
</tr>
<tr>
<td>CORF ⋊ LIF</td>
<td>( (8.73): (C1) ) any point in ( FP(S) ), such that ( p_{i_1, j_1, k_1} = 1 ), or ( (C2) ) any point in ( FP(S) ), such that ( p_{i_1, j_1+1, k_1} = 1 )</td>
</tr>
<tr>
<td></td>
<td>( (8.74): (C1) ) any point in ( FP(S) ), such that ( p_{i_1, j_1} x_{j_1+1} = 1 ), or ( (C2) ) any point in ( FP(S) ), such that ( p_{i_1, j_1+1, k_1} = 1 )</td>
</tr>
<tr>
<td>CORF ⋊ LRF</td>
<td>( (8.77): (C1) ) any point in ( FP(S) ), such that ( p_{i_1, j_1, k_1} - x_{j_1+1} = 1 ), or ( (C2) ) any point in ( FP(S) ), such that ( p_{i_1, j_1+1, k_1} = 1 )</td>
</tr>
<tr>
<td></td>
<td>( (8.78): (C1) ) any point in ( FP(S) ), such that ( p_{i_1, j_1, k_1} - x_{j_1+1} = 1 ), or ( (C2) ) any point in ( FP(S) ), such that ( p_{i_1, j_1+1, k_1} = 1 )</td>
</tr>
</tbody>
</table>
Chapter 10

Existing Strategies in Detecting Double Faults

Instead of concentrating on specifying test sets for detecting double fault classes obtained from Chapters 4, 6 and 8, this thesis focuses on finding test case selection strategies which can guarantee to detect these double fault classes. Therefore, the following questions are considered in this chapter:

Many test case selection strategies have been developed to detect all single fault classes described in Chapter 2. Can these strategies also detect all double fault classes considered in this study? If yes, which one can be used. If no, which double fault classes can be detected and which cannot?

Several test case selection strategies that can detect certain types of single faults are reviewed in Section 2.3 including the BASIC, MAX-A and MAX-B strategies [56] and the MUMCUT strategy [10]. As a reminder, the MAX-B strategy subsumes the MAX-A strategy, which in turn subsumes the MUMCUT strategy, which in turn subsumes the BASIC strategy. In this chapter, we study the fault detection capability of these four test case selection strategies with respect to all double-fault expressions in this thesis.

Unfortunately, none of these four strategies can guarantee to detect all double faults. Figure 10.1 shows the relationship of double-fault expressions detected by each of these strategies. The figure indicates that:

(1) The BASIC strategy can detect 83 out of 92 (=83+1+3+5) double-fault expressions. The corresponding proofs are discussed in Section 10.1. Moreover, an example is given in Section 10.1 to illustrate that the BASIC strategy cannot detect the double-fault expression (6.15).

(2) Since the MUMCUT strategy subsumes the BASIC strategy, it can detect all double-fault expressions that can be detected by the BASIC strategy. Moreover, we prove in Section 10.2 that it can detect double-fault expression (6.15). As a result,
the MUMCUT strategy can detect 84 (=83+1) double fault expressions. Three examples are given in Section 10.2 to illustrate that the MUMCUT strategy cannot detect double-fault expressions (6.13), (6.17) and (6.19).

(3) Similarly, the MAX-A strategy can detect 87 double-fault expressions including 84 double-fault expressions that can be detected by the MUMCUT strategy and double-fault expressions (6.13), (6.17) and (6.19). The corresponding proofs are discussed in Section 10.3. As a result the MAX-B strategy can also detect these 87 faulty expressions.

(4) There are five double-fault expressions that cannot be detected by the MAX-B strategy and corresponding examples are given in Section 10.4.

10.1 The BASIC strategy

This section identifies which double-fault expression can be detected by the BASIC strategy. Theorems 10.1.1 to 10.1.4 show that the test set selected by the BASIC strategy satisfies the detection conditions of 83 double-fault expressions considered in this study. Moreover, Theorem 10.1.5 shows the detection conditions of all such 83 double-fault expressions can be satisfied by any test case selection strategy that subsumes the BASIC strategy.

We now prove Theorems 10.1.1 to 10.1.5.

**Theorem 10.1.1** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose
that \( T \) is the set of near false points formed by selecting a near false point from every NFP\(_{i,j}\)(\( S \)). Then \( T \) satisfies the following conditions:

1. There exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \).
2. For every possible NFP\(_{i,j}\)(\( S \)), there exists \( \vec{t} \in T \) such that \( \vec{t} \in NFP_{i,j}(S) \).
3. For every possible \( i \) and \( j \) pair where \( 1 \leq i \leq m, 1 \leq j \leq k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j} = 1 \).
4. For every possible \( i, j_1 \) and \( j_2 \) where \( 1 \leq i \leq m, 1 \leq j_1 < j_2 \leq k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j_1,j_2} = 1 \).
5. For every possible \( i, j_1 \) and \( j_2 \) where \( 1 \leq i \leq m, 1 \leq j_1 < j_2 \leq k_i, k_i > 2 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j_1,j_2} = 1 \).
6. For every possible \( i \) and \( j \) pair where \( 1 \leq i \leq m, 1 \leq j < k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j} = 1 \) where \( p_i = p_{i,1,j} \cdot p_{i,j+1,k_i} \).
7. For every possible \( i \) and \( j \) pair where \( 1 \leq i \leq m, 1 \leq j < k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j+1,k_i} = 0 \) where \( p_i = p_{i,1,j} \cdot p_{i,j+1,k_i} \).
8. For every possible \( i, j_1 \) and \( j_2 \) where \( 1 \leq i \leq m, 1 \leq j_1 < j_2 < k_i, k_i > 2 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j_1+1,j_2} = 1 \) where \( p_i = p_{i,1,j_1} \cdot p_{i,j_1+1,j_2} \cdot p_{i,j_2+1,k_i} \).
9. For every possible \( i \) and \( j \) pair where \( 1 \leq i \leq m, 1 \leq j < k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j} = 0 \) where \( p_i = p_{i,1,j} \cdot p_{i,j+1,k_i} \).
10. For every possible \( i \) and \( j \) pair where \( 1 \leq i \leq m, 1 \leq j < k_i, k_i > 1 \) and \( k_i \) is the number of literals of \( p_i \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in FP(S) \) and \( p_{i,j} = 1 \) where \( p_i = p_{i,1,j} \cdot p_{i,j+1,k_i} \) and \( p_{i,j} \) is obtained from \( p_{i,1,j} \) by negating its literal \( x_i^j \).

---

1As a reminder, \( p_{i,j_1,j_2} = x_i^1 \ldots x_{j_1}^1 \ldots x_j^j \ldots x_{j_2-1}^j \ldots x_{j_2}^{j_2+1} \ldots x_{k_i}^i \) are the terms obtained from \( p_i \) by negating and omitting its \( j_1 \)-th and \( j_2 \)-th literals, respectively.
2As a reminder, \( p_{i,j_1,j_2} = x_i^1 \ldots x_{j_1-1}^j \ldots x_j^j \ldots x_{j_2-1}^j \ldots x_{j_2}^{j_2+1} \ldots x_{k_i}^i \) are the terms obtained from \( p_i \) by omitting its \( j_1 \)-th and \( j_2 \)-th literals.
3As a reminder, \( p_{i,1,j} = x_i^1 \ldots x_j^j \) and \( p_{i,j+1,k_i} = x_j^j+1 \ldots x_{k_i}^i \) are the terms obtained from \( p_i \) by keeping its first \( j \) literals and last \( k_i - j \) literals, respectively. Furthermore, both \( p_{i,1,0} \) and \( p_{i,k_i+1,k_i} \) are null terms for notational purposes.
11. For every possible $i$ and $j$ pair where $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{FP}(S)$ and $p_{i,1,j-1} = 1$ where $p_i = p_{i,1,j} \cdot p_{i,j+1,k_i}$ and $p_{i,1,j-1}$ is obtained from $p_{i,1,j}$ by omitting its literal $x^i_j$.

12. For every possible $i$ and $j$ pair where $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{FP}(S)$ and $p_{i,1,j} \cdot x^i_{j+1} = 1$ where $p_i = p_{i,1,j} \cdot p_{i,j+1,k_i}$.

13. For every possible $i$ and $j$ pair where $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{FP}(S)$ and $p_{i,1,j-1} \cdot x^i_{j+1} = 1$ where $p_i = p_{i,1,j} \cdot p_{i,j+1,k_i}$.

14. For every possible $i$ and $j$ pair where $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{FP}(S)$ and $p_{i,1,j-1} \cdot x^i_{j+1} = 1$ where $p_i = p_{i,1,j} \cdot p_{i,j+1,k_i}$.

**Proof:** Since $S$ is in IDNF, $\mathit{NFP}_{i,j}(S) \neq \emptyset$ for every possible $i$ and $j$ pair where $1 \leq i \leq m$, $1 \leq j \leq k_i$ and $k_i$ is the number of literals of $p_i$ by Lemma 2.1.1.

1. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{NFP}_{1,1}(S) \subseteq \mathit{FP}(S)$. Hence, the result follows.

2. For any possible $\mathit{NFP}_{i,j}(S)$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{NFP}_{i,j}(S)$ by definition of $T$. Hence, the result follows.

3. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j \leq k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. We have the following two cases

(a) When $j > 1$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{NFP}_{i,1}(S) \subseteq \mathit{FP}(S)$. Now, $x^i_1$ evaluates to 0 on $\vec{t}$. Hence, $p_{i,j} = x^i_1 \cdot \ldots \cdot x^i_{j-1} \cdot x^i_{j+1} \cdot \ldots \cdot x^i_{k_i} = 0$ on $\vec{t}$.

(b) When $j = 1$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{NFP}_{i,2}(S) \subseteq \mathit{FP}(S)$ because $k_i > 1$. Now, $x^i_2$ evaluates to 0 on $\vec{t}$. Hence, $p_{i,j} = p_{i,1} = x^i_2 \cdot \ldots \cdot x^i_{k_i} = 0$ on $\vec{t}$.

Hence, the result follows.

4. Let $i$, $j_1$ and $j_2$ be such that $1 \leq i \leq m$, $1 \leq j_1 < j_2 \leq k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \mathit{NFP}_{i,j_1}(S) \subseteq \mathit{FP}(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x^i_{j_1}$ which evaluates to 0. Therefore, $p_{i,j_1,j_2} = x^i_1 \cdot \ldots \cdot x^i_{j_1} \cdot \ldots \cdot x^i_{j_2-1} \cdot x^i_{j_2+1} \cdot \ldots \cdot x^i_{k_i} = 1$. Hence, the result follows.
5. Let $i$, $j_1$ and $j_2$ be such that $1 \leq i \leq m$, $1 \leq j_1 < j_2 \leq k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,j_1}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j_1}^i$ which evaluates to 0. Therefore, $p_{i,j_1,j_2} = x_1^i \cdots x_{j_1-1}^i x_{j_1+1}^i \cdots x_{j_2-1}^i x_{j_2+1}^i \cdots x_{k_i}^i = 1$. Hence, the result follows.

6. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,k_i}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j}^i$ which evaluates to 0. Since $j < k_i$, $p_{i,j} = x_1^i \cdots x_j^i = 1$. Hence, the result follows.

7. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,j}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j}^i$ which evaluates to 0. Since $j < k_i$, $p_{i,j} = x_1^i \cdots x_j^i = 1$. Hence, the result follows.

8. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j_1 < j_2 < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,j_1}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j_1}^i$ which evaluates to 0. Since $j_1 < j_2$, $p_{i,j_1,j_2} = x_{j_1+1}^i \cdots x_{j_2}^i = 1$. Hence, the result follows.

9. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,k_i}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{k_i}^i$ which evaluates to 0. Since $j < k_i$, $p_{i} = x_1^i \cdots x_{k_i}^i = 0$. Hence, the result follows.

10. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,j}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j}^i$ which evaluates to 0. Therefore, $p_{i,j} = x_1^i \cdots x_{j}^i = 1$. Hence, the result follows.

11. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in NFP_{i,j}(S) \subseteq FP(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_{j}^i$ which evaluates to 0. Therefore, $p_{i,j} = x_1^i \cdots x_{j-1}^i = 1$. Hence, the result follows.
12. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 \leq j < k_i$, $k_i > 1$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{NFP}_{i,j+1}(S) \subseteq \text{FP}(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_j^{i+1}$ which evaluates to 0. Therefore, $p_{i,1,j} \vec{t}_{j+1}^i = x_1^i \cdots x_j^i \vec{t}_{j+1}^i = 1$. Hence, the result follows.

13. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{NFP}_{i,j-1}(S) \subseteq \text{FP}(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_j^i$ which evaluates to 0. Therefore, $p_{i,1,j-1} \vec{t}_{j+1}^i = x_1^i \cdots x_{j-1}^i \vec{t}_{j+1}^i = 1$. Hence, the result follows.

14. Let $i$ and $j$ be such that $1 \leq i \leq m$, $1 < j < k_i$, $k_i > 2$ and $k_i$ is the number of literals of $p_i$ in $S$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{NFP}_{i,j-1}(S) \subseteq \text{FP}(S)$. Therefore, all literals of $p_i$ evaluate to 1 on $\vec{t}$ except $x_j^i \vec{t}_{j+1}$ which evaluates to 0. Therefore, $p_{i,1,j-1} \vec{t}_{j+1}^i = x_1^i \cdots x_{j-1}^i \vec{t}_{j+1}^i = 1$. Hence, the result follows.

\[ \square \]

**Theorem 10.1.2** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that $T$ is the set of unique true points formed by selecting a unique true point from every $\text{UTP}_i(S)$. Then $T$ satisfies the following conditions:

1. For every possible $\text{UTP}_i(S)$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{UTP}_i(S)$.

2. For every possible $i_1$ and $i_2$ pair where $1 \leq i_1 < i_2 \leq m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{TP}_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m \setminus i \neq i_1,i_2} \text{TP}_i(S) \right)$.

3. For every possible $i_1$ and $i_2$ pair where $1 \leq i_1 < i_2 \leq m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \text{TP}_{i_2}(S) \setminus \left( \bigcup_{i=1}^{m \setminus i \neq i_1,i_2} \text{TP}_i(S) \right)$.

4. For every possible $i_1$ and $i_2$ pair where $1 \leq i_1 < i_2 \leq m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \text{TP}_{i_1}(S) \cup \text{TP}_{i_2}(S) \right) \setminus \left( \bigcup_{i=1}^{m \setminus i \neq i_1,i_2 \setminus i_1} \text{TP}_i(S) \right)$.

5. For every possible $i_1$ and $h_1$ pair where $1 \leq i_1 < h_1 \leq m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=i_1}^{h_1} \text{TP}_i(S) \right) \setminus \left( \bigcup_{i=1}^{m \setminus i \neq i_1,...,h_1} \text{TP}_i(S) \right)$.
6. For every possible $i_1$, $h_1$, $i_2$ and $h_2$ where $1 \leq i_1 \leq i_2 < h_1 \leq h_2 \leq m$, $i_1 \neq i_2$ and $h_1 \neq h_2$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=1}^{m} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{i_1} TP_i(S) \right)$.

7. For every possible $i_1$, $h_1$ and $i_2$ where $1 \leq i_1 \leq i_2 \leq h_1 \leq m$ and $i_1 \neq h_1$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1}^{i_1} TP_i(S) \right)$.

8. For every possible $i_1$, $h_1$ and $i_2$ where $1 \leq i_1 < h_1 < i_2 \leq m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1, \ldots, h_1, i_2}^{i_1} TP_i(S) \right)$.

9. For every possible $i_1$ and $i_2$ where $1 < i_1 + 1 < i_2 < m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=i_1,i_1+1,i_1+2} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1,i_1+1,i_1+2} TP_i(S) \right)$.

10. For every possible $i_1$ where $1 < i_1 + 1 < m$, there exists $\vec{t} \in T$ such that $\vec{t} \in \left( \bigcup_{i=i_1,i_1+1,i_1+2} TP_i(S) \right) \setminus \left( \bigcup_{i \neq i_1,i_1+1,i_1+2} TP_i(S) \right)$.

**Proof:** Since $S$ is in IDNF, $UTP_i(S) \neq \emptyset$ for every $i$ by Lemma 2.1.1.

1. For any possible $UTP_i(S)$, there exists $\vec{t} \in T$ such that $\vec{t} \in UTP_i(S)$ by definition of $T$. Hence, the result follows.

2. Let $i_1$ and $i_2$ be such that $1 \leq i_1 < i_2 \leq m$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \bigcup_{i \neq i_1}^{m} TP_i(S) \right) \subseteq \left( TP_{i_1}(S) \setminus \bigcup_{i \neq i_1,i_2}^{m} TP_i(S) \right)$ because $(A \setminus (B \cup C)) \subseteq (A \setminus B)$. Hence, the result follows.

3. Let $i_1$ and $i_2$ be such that $1 \leq i_1 < i_2 \leq m$. By definition of $T$, there exists $\vec{t} \in T$ such that $\vec{t} \in UTP_{i_2}(S) = \left( TP_{i_2}(S) \setminus \bigcup_{i \neq i_2}^{m} TP_i(S) \right) \subseteq \left( TP_{i_2}(S) \setminus \bigcup_{i \neq i_1,i_2}^{m} TP_i(S) \right)$ because $(A \setminus (B \cup C)) \subseteq (A \setminus B)$. Hence, the result follows.
4. Let \( i_1 \) and \( i_2 \) be such that \( 1 \leq i_1 < i_2 \leq m \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( \left( TP_{i_1}(S) \cup TP_{i_2}(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1, i_2}^{m} TP_i(S) \right) \right) \) because \((A \setminus (B \cup C)) \subseteq ((A \cup C) \setminus B)\). Hence, the result follows.

5. Let \( i_1 \) and \( h_1 \) be such that \( 1 \leq i_1 < h_1 \leq m \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \right) \subseteq \left( \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1, i_2}^{m} TP_i(S) \right) \right) \) because \((A \setminus (B \cup C)) \subseteq ((A \cup D) \setminus B)\). Hence, the result follows.

6. Let \( i_1, i_2, h_1, h_2 \) be such that \( 1 \leq i_1 \leq i_2 < h_2 \leq h_1 \leq m \) and \( i_1 \neq i_2 \) and \( h_2 \neq h_1 \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1, i_2}^{m} TP_i(S) \right) \right) \) by using similar arguments as in (5) above. Hence, the result follows.

7. Let \( i_1, i_2 \) and \( h_1 \) be such that \( 1 \leq i_1 \leq i_2 \leq h_1 \leq m \) and \( i_1 \neq h_1 \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1}^{m} TP_i(S) \right) \right) \) by using similar arguments as in (5) above. Hence, the result follows.

8. Let \( i_1, i_2 \) and \( h_1 \) be such that \( 1 \leq i_1 < h_1 < i_2 \leq m \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1, i_2}^{m} TP_i(S) \right) \right) \) by using similar arguments as in (5) above. Hence, the result follows.

9. Let \( i_1, i_2 \) be such that \( 1 < i_1 + 1 < i_2 < m \). By definition of \( T \), there exists \( \vec{t} \in T \) such that \( \vec{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) \right) \subseteq \left( \left( \bigcup_{i=1, i \neq i_1}^{h_1} TP_i(S) \right) \setminus \left( \bigcup_{i=1, i \neq i_1, i_2}^{m} TP_i(S) \right) \right) \) by using similar arguments as in (5) above. Hence, the result follows.
\[
\left( \bigcup_{i=1}^{m} TP_i(S) \right) \cup \left( TP_{i_1}(S) \cap TP_{i_1+1}(S) \right) \subseteq \left( \bigcup_{i \neq i_1, i_1+1, i_2, i_2+1} TP_i(S) \right) \cup \left( \bigcup_{i \neq i_1, i_1+1, i_2, i_2+1} TP_i(S) \right) \cup \left( \bigcup_{i \neq i_1, i_1+1, i_1+2} TP_i(S) \right) \cup \left( \bigcup_{i \neq i_1, i_1+1, i_1+2} TP_i(S) \right) \]

Hence, the result follows.

10. Let \( i_1 \) be such that \( 1 < i_1 + 1 < m \). By definition of \( T \), there exists \( \tilde{t} \in T \) such that \( \tilde{t} \in UTP_{i_1}(S) = \left( TP_{i_1}(S) \setminus \bigcup_{i=1}^{m} TP_i(S) \right) \subseteq \left( \bigcup_{i=1}^{m} TP_i(S) \right) \setminus \left( \bigcup_{i=i_1+1, i_1+2} TP_{i_1}(S) \setminus TP_{i_1+1}(S) \cap TP_{i_1+2}(S) \right) \) by using similar arguments as in (9) above. Hence, the result follows.

\[\Box\]

**Theorem 10.1.3** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. The BASIC meaningful impact strategy can detect all double-fault expressions in Tables 10.1 to 10.3.

**Proof:** Tables 10.1 to 10.3 indicate that the conditions listed in Theorems 10.1.1 and 10.1.2 can satisfy detection conditions of some double-fault expressions in Chapters 5, 7 and 9. The BASIC strategy requires to select a test case from every \( UTP_i(S) \) and a test case from every \( NFP_{i,j}(S) \). Let \( T \) be a test set generated by the BASIC strategy. By Theorems 10.1.1 and 10.1.2, \( T \) satisfies those conditions listed in the two theorems. Tables 10.1 to 10.3 list the detection conditions of some double-fault expressions in Tables 5.2, 7.2 and 9.2 and their corresponding conditions in Theorems 10.1.1 and 10.1.2. As a result, \( T \) can detect all these double fault expressions. Hence, the result follows. \[\Box\]

It has been shown in Table 10.1 that the detection conditions of all 31 double-fault expressions shown in Table 5.2 can be satisfied by any test set selected by the BASIC strategy. Therefore, all faulty expressions of double-fault related to terms only can be detected by the BASIC strategy. More precisely, the BASIC strategy guarantees to detect double faults related to terms only. While Tables 10.2 and 10.3 show that of 12 out of 19 double-fault expressions shown in Table 7.2 and 39 out of 42 double-fault expressions shown in Table 9.2 can be satisfied by any test set selected by the BASIC strategy. More clearly, 12 faulty expressions of double faults related to literals only and 39 faulty expressions of double faults related to a term and a literal can be detected by the BASIC strategy.
Table 10.1: Conclusions in Theorems 10.1.1 and 10.1.2 satisfying detection conditions in Table 5.2

(a) Double-fault expressions based on double fault without ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4.1)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td></td>
</tr>
<tr>
<td>(4.2)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(6)</td>
</tr>
<tr>
<td>(4.3)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.4)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(7)</td>
</tr>
<tr>
<td>(4.5)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(8)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td></td>
</tr>
<tr>
<td>(4.6)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(7)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td></td>
</tr>
<tr>
<td>(4.7)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.8)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(8)</td>
</tr>
<tr>
<td>(4.9)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.10)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.11)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.12)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(5)</td>
</tr>
<tr>
<td>(4.13)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.14)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(2)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.15)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.16)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(2)</td>
</tr>
<tr>
<td>(4.17)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(4.18)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(4.19)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(4)</td>
</tr>
<tr>
<td>(4.20)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(4)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(4)</td>
</tr>
<tr>
<td>(4.21)</td>
<td>(C2)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(4.22)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(9)</td>
</tr>
<tr>
<td>(4.23)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(10)</td>
</tr>
</tbody>
</table>

Table 10.1 (cont’d) Conclusions in Theorems 10.1.1 and 10.1.2 satisfying detection conditions in Table 5.2

(a) Double-fault expressions based on double fault without ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4.24)</td>
<td>(C3)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(4.25)</td>
<td>(C2)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td>(4.26)</td>
<td>(C1)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(4.27)</td>
<td>(C1)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(8)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
</tbody>
</table>
(b) Four remaining double-fault expressions based on double fault with ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Conclusions in Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4.53)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(4)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(4)</td>
</tr>
<tr>
<td>(4.70)</td>
<td>(C2)</td>
<td>10.1.1(9)</td>
<td>-</td>
</tr>
<tr>
<td>(4.73)</td>
<td>(C1)</td>
<td>10.1.1(9)</td>
<td>-</td>
</tr>
<tr>
<td>(4.76)</td>
<td>(C1)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 10.2: Conclusions in Theorems 10.1.1 and 10.1.2 satisfying detection conditions in Table 7.2

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Conclusions in Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(6.1)</td>
<td>(C3)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.2)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(6.3)</td>
<td>(C2)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.4)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(4)</td>
<td>-</td>
</tr>
<tr>
<td>(6.5)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.6)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(6.7)</td>
<td>(C4)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.8)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(6.9)</td>
<td>(C1)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.10)</td>
<td>(C1)</td>
<td>10.1.1(5)</td>
<td>-</td>
</tr>
<tr>
<td>(6.11)</td>
<td>(C2)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(6.12)</td>
<td>(C2)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 10.3: Conclusions in Theorems 10.1.1 and 10.1.2 satisfying detection conditions in Table 9.2

(a) Double-fault expressions based on double fault without ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Conclusions in Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8.1)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.2)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(7)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(3)</td>
<td>-</td>
</tr>
<tr>
<td>(8.3)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(3)</td>
<td>-</td>
</tr>
<tr>
<td>(8.4)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(5)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(3)</td>
<td>-</td>
</tr>
<tr>
<td>(8.5)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(5)</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.6)</td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(7)</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.7)</td>
<td>(C3)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.8)</td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(7)</td>
</tr>
<tr>
<td>(8.9)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.10)</td>
<td>(C1)</td>
<td>10.1.1(3)</td>
<td>-</td>
</tr>
<tr>
<td>(8.11)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.12)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(3)</td>
<td>-</td>
</tr>
<tr>
<td>(8.13)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.14)</td>
<td>(C2)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.15)</td>
<td>(C3)</td>
<td>10.1.1(1)</td>
<td>-</td>
</tr>
<tr>
<td>(8.17)</td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(3)</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(8.18)</td>
<td>(C2)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(8.19)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.21)</td>
<td>(C3)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C6)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(8.22)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(2)</td>
</tr>
<tr>
<td>(8.23)</td>
<td>(C3)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td>(8.24)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.25)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.26)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.28)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.29)</td>
<td>(C2)</td>
<td>10.1.2(2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 10.3 (cont’d) Conclusions in Theorems 10.1.1 and 10.1.2 satisfying detection conditions in Table 9.2

(a) Double-fault expressions based on double fault without ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Conclusions in Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8.30)</td>
<td>(C1)</td>
<td>10.1.1(10)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.31)</td>
<td>(C1)</td>
<td>10.1.1(2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.32)</td>
<td>(C1)</td>
<td>10.1.1(11)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.33)</td>
<td>(C2)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C3)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.34)</td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.35)</td>
<td>(C3)</td>
<td>10.1.1(6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C4)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.36)</td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
</tbody>
</table>

(b) Six remaining double-fault expresssions based on double fault with ordering

<table>
<thead>
<tr>
<th>Double-fault Expression</th>
<th>Detection Conditions</th>
<th>Conclusions in Theorem 10.1.1</th>
<th>Conclusions in Theorem 10.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8.43)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(5)</td>
</tr>
<tr>
<td>(8.52)</td>
<td>(C1)</td>
<td>-</td>
<td>10.1.2(1)</td>
</tr>
<tr>
<td>(8.73)</td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.74)</td>
<td>(C1)</td>
<td>10.1.1(12)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.77)</td>
<td>(C1)</td>
<td>10.1.1(13)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
<tr>
<td>(8.78)</td>
<td>(C1)</td>
<td>10.1.1(14)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(C2)</td>
<td>10.1.1(7)</td>
<td>-</td>
</tr>
</tbody>
</table>
Theorem 10.1.4 proves the BASIC strategy can detect double-fault expression (8.16). As a result, there are total 83 (=31+12+39+1) double-fault expressions can be detected by the BASIC strategy. The following lemma is needed before proving Theorem 10.1.4.

**Lemma 10.1.1** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose $x_{i_1}^1$, the only literal of the $i_1$-th term, $p_{i_1}$, in $S$ is negated and then replaced by $x_1$ where $x_1$ is a missing literal of $p_{i_1}$ and $1 \leq i_1 \leq i_2 \leq m$ and $m > 1$. Then the resulting implementation denoted by $I$ will be equivalent to $p_1 + \cdots + x_1 + \cdots + p_m$.

**Proof**: Let $p_{i_2} (i_1 \neq i_2)$ be the term containing either $x_1$ or $\bar{x}_1$. We proceed our proof using the following two cases:

1. The literal $x_1$ is in $p_{i_2}$. There exists $j_2$ such that $x_j$ is the $j_2$-th literal in $p_{i_2}$, that is $x_{j_2}^1 = x_j$. As a result, for any $\vec{t} \in NFP_{i_2,j_2} (S)$, $\vec{t} \in FP(S)$ and $x_{j_2}^2 = x_j = 0$.

2. The literal $\bar{x}_1$ is in $p_{i_2}$. There are two subcases:

   (a) There is more than one literal in $p_{i_2}$. Suppose that $\bar{x}_1$ is the $j_2$-th literal. That is, $p_{i_2} = x_{i_2}^1 \cdots \bar{x}_1 \cdots x_{k_2}^2$. Since $k_2 > 1$, there exists $j$ such that $1 \leq j \leq k_2$ and $j \neq j_2$. For any point $\vec{t} \in NFP_{i_2,j_2} (S)$,

   

   \[ p_{i_2,j_2}(\vec{t}) = x_{i_2}^1 \cdots \bar{x}_1 \cdots x_{k_2}^2 = 1. \]

   Hence, $x_j = 0$ on $\vec{t}$. As a result, for any $\vec{t} \in NFP_{i_2,j_2} (S)$, $\vec{t} \in FP(S)$ and $x_j = 0$.

   (b) There is only one literal in $p_{i_2}$ that is $p_{i_2} = \bar{x}_1$. For any unique true point $\vec{t}$ in $UTP_{i_1,j_1} (S)$ ($i_1 \neq i_2$), $p_{i_2} = \bar{x}_1 = 0$. Therefore, $x_1 = 1$ on $\vec{t}$. As a result, for any $\vec{t} \in UTP_{i_1,j_1} (S)$ and $x_1 = 1$.

Hence, the result follows. $\square$

**Theorem 10.1.4** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i_1$-th term, $p_{i_1}$, in $S$ is negated and $j_1$-th literal, $x_{j_1}^1$, of $p_{i_1}$ in $S$ is replaced by $x_{j_2}$ where $1 \leq i_1 \leq m$, $1 \leq j_2 \leq k_{i_1}$, $m > 1$ and $x_{j_2}$ is a missing literal of $p_{i_1}$. Then the resulting implementation denoted by $I$ will be equivalent to that given by Expression (8.16) in Table 8.12. The BASIC strategy can guarantee the detection of Expression (8.16) provided that $S \neq I$.  

252
Proof: Since \( S \not\equiv I \), there is at least one test case \( \vec{t} \) such that \( \vec{t} \) satisfies any one of the following (C1)–(C3) given in Table 8.12.

(C1) any point in \( UTP_{i_1}(S) \) such that \( x_{t_2} = 1 \),
(C2) any point in \( FP(S) \) such that \( x_{t_2} = 0 \), or
(C3) any point in \( FP(S) \) such that \( p_{i_1,j_2} = 0 \).

We proceed our proof using the following two cases:

Case 1. The \( p_{i_1} \) has more than one literal. That is, \( k_{i_1} > 1 \).

By Theorem 10.1.1(3), the BASIC strategy can select test cases that satisfy condition (C3).

Case 2. The \( p_{i_1} \) has one literal. That is \( p_{i_1} = x_{i_1}^1 \).

By Lemma 10.1.1, the BASIC strategy can select test cases to satisfy either (C1) or (C2).

Hence, The result follows.

\[ \Box \]

Theorem 10.1.5 Let \( S = p_1 + \cdots + p_m \) be a Boolean expression in IDNF. Any test case selection strategy that subsumes the BASIC meaningful impact strategy can detect all double-fault expressions listed in Tables 10.1 to 10.3 and Expression (8.16) in Table 8.12.

Proof: By Theorem 10.1.3 and the definition of subsumption of test case selection strategy.

\[ \Box \]

The following example shows that the BASIC strategy cannot detect the double-fault expression (6.15) that can be detected by the MUMCUT strategy.

Example 10.1.1 Let \( S = abc + bcd + bc\bar{e} + \bar{b}e\bar{f} + \bar{c}e\bar{f} + bcf \). Table 10.4 lists the sets \( UTP_i(S) \) of all unique true points for every possible \( i \), the sets \( NFP_{i,j}(S) \) of all near false points for every possible \( i \) and \( j \) pair, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \). Suppose the literals \( a \) and \( c \) are inserted into the fourth term \( \bar{b}e\bar{f} \) of \( S \). The resulting double-fault expression is equivalent to \( I = abc + bcd + bc\bar{e} + \bar{a}bc\bar{e}\bar{f} + \bar{c}e\bar{f} + bcf \).\(^4\) Note that, \( S \) and \( I \) are not equivalent because \( S \) and \( I \) evaluate to 1 and 0 on 001110, respectively.

Now, if we select \( T = \{111110, 110111, 110110, 110011, 110001, 101111, 101110, 101001, 100100, 011111, 011110, 011100, 011010 \} \)\(^5\) to satisfy the BASIC strategy, \( S \) and \( I \) agree on all points in \( T \). Hence, the BASIC strategy cannot guarantee to distinguish \( S \) and \( I \).

\[ \diamond \]

\(^4\) It is a special instance of Expression 6.15.
\(^5\) The selected test cases are underlined in Table 10.6.
Table 10.4: All points of $S$ where $S = abc + bcd + b\bar{e} + \bar{b}\bar{f} + \bar{c}e\bar{f} + bc\bar{f}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$1$</th>
<th>$2$</th>
<th>$3$</th>
<th>$4$</th>
<th>$5$</th>
<th>$6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$UTP_i(S)$</td>
<td>111110</td>
<td>011010</td>
<td>011100</td>
<td>101110, 101010, 001110, 001010</td>
<td>110110, 110010, 011100, 010110, 010100</td>
<td>011111</td>
</tr>
<tr>
<td>$NFP_{i,1}(S)$</td>
<td>011110</td>
<td>101011, 101001, 101000, 001011, 001001, 001000, 001000</td>
<td>101011, 101100, 101000, 001101, 001100, 001001, 001000</td>
<td>011110</td>
<td>011110</td>
<td>101111, 101101, 101101, 101001, 001111, 001101, 001011, 001001</td>
</tr>
<tr>
<td>$NFP_{i,2}(S)$</td>
<td>101111, 101101, 101100, 101011, 101001, 101000, 110000, 010111, 010001, 010000, 010000</td>
<td>110011, 110001, 110000, 010101, 010010, 010001, 010000</td>
<td>101100, 101000, 100100, 000100, 000100, 000000, 000000</td>
<td>110100, 110000, 110000, 100110, 100100, 100000, 100000</td>
<td>110111, 110101, 110101, 110001, 110001, 110001, 110001, 110001</td>
<td></td>
</tr>
<tr>
<td>$NFP_{i,3}(S)$</td>
<td>110111, 110101, 110100, 110011, 110001, 110000</td>
<td>011110</td>
<td>011110</td>
<td>101111, 101011, 100111, 100011, 001111, 001011, 000111, 000011</td>
<td>110111, 110011, 110011, 110011, 110011, 110011, 110011, 110011</td>
<td>011110</td>
</tr>
</tbody>
</table>

$OTP(S) = \{ 111111, 011101, 100110, 000010, 011011, 111100, 111100, 011001, 000110, 111001, 111111, 011000, 111101, 111100, 100010 \}$ and $RFP(S) = \{ 000001, 100001, 100101, 000101 \}$.

10.2 The MUMCUT strategy

Since the MUMCUT strategy subsumes the BASIC strategy, by Theorem 10.1.5, the MUMCUT strategy can also detect 83 double-fault expressions including double-fault expressions shown in Tables 10.1 to 10.3 and Expression (8.16). Theorem 10.2.1 shows that any test set selected by the MUMCUT strategy satisfies the detection condition of double-fault expression (6.15). As a result, 84 expressions can be detected by the MUMCUT strategy. Moreover, Theorem 10.2.2 shows that all these double-fault expressions can be detected by any test case selection strategies that subsumes the MUMCUT strategy. Examples 10.2.1, 10.2.2 and 10.2.3 illustrate that the MUMCUT strategy cannot detect Expressions (6.13), (6.17) and (6.19), respectively.

**Theorem 10.2.1** Let $S = p_1 + \cdots + p_m$ be a Boolean expression in IDNF. Suppose
that two literals \( x_{l_1} \) and \( x_{l_2} \) are inserted into \( i \)-th term, \( p_{i_1} \), in \( S \) where \( x_{l_1} \) and \( x_{l_2} \) are two different missing literals of \( p_{i_1} \). Then the resulting implementation denoted by \( I \) is equivalent to Expression (6.15) in Table 6.7 and its detection conditions are given by (C1) and (C2) in Table 7.2. The MUTP strategy can guarantee the detection of Expression (6.15) provided that \( S \not\equiv I \).

**Proof :**

Since \( S \not\equiv I \), there is at least one test case \( \vec{t} \) such that \( \vec{t} \) satisfies any one of the following two conditions given in Table 7.2.

(C1) any point in \( UTP_{i_1}(S) \) such that \( x_{l_1} = 0 \), or

(C2) any point in \( UTP_{i_1}(S) \) such that \( x_{l_2} = 0 \).

Now \( x_{l_1} \) and \( x_{l_2} \) are missing literals of \( p_{i_1} \), and the MUTP strategy requires test cases from \( UTP_{i_1}(S) \) such that all possible truth values (that is, 0 and 1) of every missing literal of \( p_{i_1} \) are covered. Thus, the MUTP strategy can detect double-fault expression (6.15).

\( \square \)

As mentioned before, the MUMCUT strategy actually comprises of the MUTP, MNFP and CUTPNFP strategies, thus, the MUMCUT strategy can also guarantee the detection of double-fault expression (6.15).

**Theorem 10.2.2** Let \( S = p_1 + \cdots + p_m \) be a Boolean expression in IDNF. Any test case selection strategy that subsumes the MUMCUT meaningful impact strategy can detect double-fault expressions in Tables 10.1 to 10.3, (8.16) in Table 8.12 and (6.15) in Table 6.7.

**Proof :** By Theorems 10.1.5 and 10.2.1, and the definitions of the MUMCUT strategy and subsumption of test case selection strategy, respectively. \( \square \)

**Example 10.2.1** Let \( S = abcd + abc\bar{e} + \bar{a}be + a\bar{b}e \). Table 10.5 lists the sets \( UTP_{i_1}(S) \) of all unique true points, the sets \( NFP_{i,j}(S) \) of all near false points, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \). Suppose the literal \( a \) of the first term \( abcd \) is omitted and the literal \( b \) of the first term \( abcd \) in \( S \) is replaced by the literal \( e \). The resulting double-fault expression is equivalent to \( cde + abc\bar{e} + \bar{a}be + a\bar{b}e \).\(^6\) Note that, \( S \) and \( I \) are not equivalent because \( S \) and \( I \) evaluate to 0 and 1 on 00111, respectively.

Now, if we select \( T = \{ \text{11111, 11101, 11100, 11011, 11010, 11000, 10110, 10101, 10100, 10011, 10010, 01110, 01110, 01101, 01100, 01011, 01010, 00101, 00011} \} \)
7 to satisfy the MUMCUT strategy, \( S \) and \( I \) agree on all points in \( T \). Hence, the MUMCUT strategy cannot guarantee to distinguish \( S \) and \( I \).

\( \diamond \)

\(^6\)It is a special instance of Expression 6.13.

\(^7\)The selected test cases are underlined in Table 10.5.
Example 10.2.2 Let \( S = \overline{abc} + bcd + bc\overline{e} + \overline{be}\overline{f} + \overline{ce}\overline{f} + bcf \). Table 10.6 lists the sets \( UTP_i(S) \) of all unique true points for every possible \( i \), the sets \( NFP_{i,j}(S) \) of all near false points for every possible \( i \) and \( j \) pair, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \).

Suppose the literal \( d \) is inserted into the first term \( abc \) and the literal \( c \) of the first term \( abc \) is replaced by the literal \( e \). The resulting double-fault expression is equivalent to \( abde + bcd + bc\overline{e} + \overline{be}\overline{f} + \overline{ce}\overline{f} + bcf \). Note that, \( S \) and \( I \) are not equivalent because \( S \) and \( I \) evaluate to 0 and 1 on 110111, respectively.

Now, if we select \( T = \{ 111110, 110100, 110011, 110010, 110001, 110000, 101100, 101011, 101001, 101000, 011111, 011110, 011101, 011100, 011010, 011011, 010100, 010000, 001111, 001110, 010110, 001100, 001000, 000111, 000110, 000100 \} \) to satisfy the MUMCUT strategy, \( S \) and \( I \) agree on all points in \( T \). Hence, the MUMCUT strategy cannot guarantee to distinguish \( S \) and \( I \).

\( \diamond \)

Example 10.2.3 Let \( S = \overline{abc} + bcd + bc\overline{e} + \overline{be}\overline{f} + \overline{ce}\overline{f} + bcf \). Please noted that we use the same \( S \) here as in Example 10.1.1. Hence, Table 10.6 lists the sets \( UTP_i(S) \) of all unique true points for every possible \( i \), the sets \( NFP_{i,j}(S) \) of all near false points for every possible \( i \) and \( j \) pair, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \).

Suppose the literal \( b \) and the literal \( c \) of the first term \( abc \) in \( S \) are replaced by the literal \( d \) and the literal \( e \), respectively. The resulting double-fault expression is equivalent to \( ade + bcd + bc\overline{e} + \overline{be}\overline{f} + \overline{ce}\overline{f} + bcf \). Note that, \( S \) and \( I \) are not equivalent because \( S \) and \( I \) evaluate to 0 and 1 on 110111, respectively.

Now, if we select \( T = \{ 111110, 110100, 110011, 110010, 110001, 110000, 101100, 101011, 101001, 101000, 011111, 011110, 011101, 011100, 011010, 011011, 010100, 010000, 001111, 001110, 010110, 001100, 001000, 000111, 000110, 000100 \} \) to satisfy the MUMCUT strategy, \( S \) and \( I \) agree on all points in \( T \). Hence, the MUMCUT strategy cannot guarantee to distinguish \( S \) and \( I \).

\( \diamond \)

---

Table 10.5: All points of \( S \) where \( S = \overline{abcd} + abce + \overline{abe} + abe \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( UTP_i(S) )</th>
<th>( NFP_{i,1}(S) )</th>
<th>( NFP_{i,2}(S) )</th>
<th>( NFP_{i,3}(S) )</th>
<th>( NFP_{i,4}(S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111</td>
<td>0111</td>
<td>1011</td>
<td>1111</td>
<td>1111</td>
</tr>
<tr>
<td>2</td>
<td>1100</td>
<td>0110</td>
<td>1010</td>
<td>1100</td>
<td>1100</td>
</tr>
<tr>
<td>3</td>
<td>0111, 0110, 0101, 0100</td>
<td>1101, 1100, 1100, 1100</td>
<td>0111, 0111, 0110, 0110</td>
<td>0111, 0111, 0110, 0110</td>
<td>--</td>
</tr>
<tr>
<td>4</td>
<td>1011, 1010, 1010, 1010</td>
<td>1101, 1100, 1100, 1100</td>
<td>0110, 0110, 0110, 0110</td>
<td>0110, 0110, 0110, 0110</td>
<td>--</td>
</tr>
</tbody>
</table>

\( OTP(S) = \{ 11110 \} \) and \( RFP(S) = \{ 00000, 00010, 00100, 00110 \} \).

---

\( \)
Table 10.6: All points of $S$ where $S = abc + bcd + bce + bef + cef + bcf$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$UTP_i(S)$</th>
<th>$NFP_{i,1}(S)$</th>
<th>$NFP_{i,2}(S)$</th>
<th>$NFP_{i,3}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>111110</td>
<td>011110</td>
<td>101111, 101101, 101000, 001011, 001001, 001000</td>
<td>110111, 110101, 110100, 100101, 100100, 010101, 100000, 001000</td>
</tr>
<tr>
<td>2</td>
<td>011010</td>
<td>101011, 101000, 001011, 001001, 001000</td>
<td>101001, 101000, 001011, 001001, 001000</td>
<td>001001, 001000</td>
</tr>
<tr>
<td>3</td>
<td>011100</td>
<td>101010, 101000, 001011, 001001, 001000</td>
<td>101001, 101000, 001011, 001001, 001000</td>
<td>001001, 001000</td>
</tr>
<tr>
<td>4</td>
<td>101101, 101010, 001011, 001001, 001000</td>
<td>011110</td>
<td>011110</td>
<td>101111, 101011, 101001, 101000, 100111, 100110, 100101, 100001</td>
</tr>
<tr>
<td>5</td>
<td>110110, 110100, 001011, 001001, 001000</td>
<td>011001, 011000, 001011, 001001, 001000</td>
<td>101000, 101000, 001010, 001000, 001000, 000100, 000000</td>
<td>110111, 110101, 110100, 110000, 100100, 100000, 101000, 010000, 010000, 010000, 010000, 010000, 010000, 010000, 010000</td>
</tr>
<tr>
<td>6</td>
<td>011111</td>
<td>011111</td>
<td>110000, 100000, 001001, 001000</td>
<td>101111, 110111, 100111, 100110, 100101, 100011, 100010, 010011, 010010, 010000, 010000, 010000, 010000, 010000, 010000, 010000</td>
</tr>
</tbody>
</table>

$OTP(S) = \{111111, 011101, 100110, 000010, 011011, 111010, 111100, 011001, 001110, 111001, 111011, 011000, 111101, 111000, 100010, 011000, 011000, 001111, 001100, 001000, 000110, 000100, 000011, 000000, 001000, 000100, 000011, 000000\}$

and $RFP(S) = \{000001, 100001, 100101, 000101\}$

Therefore, $010100, 010000, 001111, 001110, 010110, 001100, 001000, 001111, 000100 \}^{11}$ to satisfy the MUMCUT strategy, $S$ and $I$ agree on all points in $T$. Hence, the MUMCUT strategy cannot guarantee to distinguish $S$ and $I$.

\[\Diamond\]

## 10.3 The MAX-A Strategy

Since the MAX-A strategy subsumes the MUMCUT strategy, it can also detect 84 double-fault expressions that can be detected by the MUMCUT strategy. Theorems 10.3.1, 10.3.2 and 10.3.3 show that test set generated by the MAX-A strategy can detect double-fault expressions (6.13), (6.17) and (6.19), respectively. Therefore, the MAX-A strategy can detect 87 out of 92 double-fault expressions can be detected by the MAX-A strategy. We need the following lemma before proving

\[\text{11The selected test cases are underlined in Table 10.6.}\]
Theorem 10.3.1.

Lemma 10.3.1 Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $j_1$-th literal, $x_{j_1}^{i_1}$, of the $i_1$-th term, $p_{i_1}$, in $S$ is omitted and the $j_2$-th literal, $x_{j_2}^{i_1}$, in $p_{i_1}$ is replaced by $x_{l_2}$ where $1 \leq i_1 \leq m$, $m > 1$, $1 \leq j_1 < j_2 \leq k_{i_1}$ and $x_{l_2}$ is a missing literal of $p_{i_1}$, the resulting implementation denoted by $I$ is equivalent to that given by Expression (6.13) in Table 6.7 and its detection conditions are given by (C1) to (C4) in Table 7.2. If (C1), (C2) and (C3) cannot be satisfied, any test case that satisfies (C4) is a near false point of $S$ provided that $S \neq I$.

Proof: Since $S \neq I$, there exists a test case that can satisfy any one of the following detection conditions of $I$ as given in Table 7.2 for Expression (6.13)

(C1) any point in $UTP_{i_1}(S)$ such that $x_{l_2} = 0$,
(C2) any point in $\in NFP_{i_1,j_1}(S)$ such that $x_{l_2} = 1$,
(C3) any point in $\in NFP_{i_1,j_2}(S)$ such that $x_{l_2} = 1$, or
(C4) any point in $\in FP(S)$ such that $p_{i_1,j_1,j_2}x_{l_2} = 1$.

Based on the assumptions of the lemma, (C1)–(C3) cannot be satisfied. There exists a test case that satisfies (C4). We now prove that any test case satisfying (C4) is a near false point of $S$.

Let $\vec{t}$ be a test case that satisfies (C4). Then, $p_{i_1,j_1,j_2}x_{l_2}(\vec{t}) = x_{i_1}^{j_1} \cdots x_{j_1}^{i_1} \cdots x_{j_2}^{i_1} x_{l_2}(\vec{t}) = 1$. Let $\vec{t}_1$ be such that $\vec{t}_1$ and $\vec{t}$ only differ at $x_{j_2}^{i_1}$.

Then, $p_{i_1,j_1}x_{l_2}(\vec{t}_1) = x_{i_1}^{j_1} \cdots x_{j_1}^{i_1} \cdots x_{j_2}^{i_1} \cdots x_{k_{i_1}} x_{l_2}(\vec{t}_1) = 1$. Note that $\vec{t}_1$ must be a true point of $S$. Otherwise, $S(\vec{t}_1) = 0$. In addition, $p_{i_1,j_1}(\vec{t}_1) = 1$ because $p_{i_1,j_1}x_{l_2}(\vec{t}_1) = 1$. Hence, $\vec{t}_1 \in NFP_{i_1,j_1}(S)$ such that $x_{l_2} = 1$. This violates the assumption that (C2) cannot be satisfied.

Now $\vec{t}_1$ is a true point of $S$. Since $p_{i_1,j_1}(\vec{t}_1) = 1$, $p_{i_1}(\vec{t}_1) = 0$. There exists $i_3(\neq i_1)$ such that $p_{i_3}(\vec{t}_1) = 1$.

Note that $\vec{x}_{l_2}$ does not appear in $p_{i_3}$. Otherwise, $p_{i_3}(\vec{t}_1) = \cdots \vec{x}_{l_2} \cdots = 0$ because $p_{i_1,j_1}x_{l_2}(\vec{t}_1) = 1$ implies $x_{l_2}(\vec{t}_1) = 1$.

We are going to prove that $x_{j_2}^{i_1}$ exists in $p_{i_3}$. If both $x_{j_2}^{i_1}$ and $\vec{x}_{j_2}^{i_1}$ do not appear in $p_{i_3}$, then $p_{i_3}(\vec{t}) = p_{i_3}(\vec{t}_1) = 1$ because $\vec{t}$ and $\vec{t}_1$ differ only at $x_{j_2}^{i_1}$. This contradicts to the fact that $\vec{t}$ is a false point of $S$. If $\vec{x}_{j_2}^{i_1}$ appears in $p_{i_3}$, then $p_{i_3}(\vec{t}_1) = \cdots \vec{x}_{j_2}^{i_1} \cdots = 0$ because $p_{i_1,j_1}(\vec{t}_1) = 1$ implies $x_{j_2}^{i_1}(\vec{t}_1) = 1$. This contradicts to the fact that $p_{i_3}(\vec{t}_1) = 1$.

Therefore $x_{j_2}^{i_1}$ must appear in $p_{i_3}$. Since $x_{j_2}^{i_1}$ is a literal of $p_{i_3}$, there exists $j_3$ such that $x_{j_2}^{i_1} = x_{j_3}^{i_3}$.

Now, since $p_{i_3}(\vec{t}_1) = 1$ and $\vec{t}_1$ and $\vec{t}$ only differ at $x_{j_3}^{i_3}(= x_{j_2}^{i_1})$, we have $p_{i_3,j_3}(\vec{t}) = x_{i_3}^{j_3} \cdots \vec{x}_{j_3}^{i_3} \cdots x_{k_{i_3}}(\vec{t}) = 1$. Since $\vec{t}$ is a false point of $S$, $\vec{t} \in NFP_{i_3,j_3}(S)$.

Hence the result follows.

We now prove Theorem 10.3.1.
Theorem 10.3.1 Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $j_1$-th literal, $x_{i_1}^{j_1}$, of the $i_1$-th term, $p_{i_1}$, in $S$ is omitted and the $j_2$-th literal, $x_{i_2}^{j_2}$, in $p_{i_1}$ is replaced by $x_{i_2}$, where $1 \leq i_1 \leq m$, $m > 1$, $1 \leq j_1 < j_2 \leq k_{i_1}$, and $x_{i_2}$ is a missing literal of $p_{i_1}$, the resulting implementation denoted by $I$ is equivalent to that given by Expression (6.13) in Table 6.7 and its detection conditions are given by (C1) to (C4) in Table 7.2. The MAX-A strategy can guarantee the detection of Expression (6.13) provided that $S \not\equiv I$.

Proof: Since $S \not\equiv I$, there is at least one test case $\bar{t}$ such that $\bar{t}$ satisfies any one of the following conditions (C1) to (C4) given in Table 7.2:

- (C1) any point in $\text{UTP}_{i_1}(S)$ such that $x_{i_2} = 0$,
- (C2) any point in $\text{NFP}_{i_1,j_1}(S)$ such that $x_{i_2} = 1$,
- (C3) any point in $\text{NFP}_{i_1,j_2}(S)$ such that $x_{i_2} = 1$, or
- (C4) any point in $\text{FP}(S)$ such that $p_{i_1,j_1,j_2} x_{i_2} = 1$.

We proceed our proof using the following two cases:

Case 1. Conditions (C1) – (C3) cannot be satisfied.

Then $\bar{t}$ must satisfy (C4). By Lemma 10.3.1, $\bar{t}$ is a near false point of $S$. Hence, the MAX-A strategy can select $\bar{t}$.

Case 2. The test case $\bar{t}$ satisfies any of (C1), (C2) or (C3).

(a) If $\bar{t}$ satisfies (C1), $\bar{t}$ must be a unique true point.
(b) If $\bar{t}$ satisfies (C2), $\bar{t}$ must be a near false point.
(c) If $\bar{t}$ satisfies (C3), $\bar{t}$ must be a near false point.

As a result, the MAX-A strategy can select $\bar{t}$.

Hence, the result follows. $\square$

We now prove Theorem 10.3.2.

Theorem 10.3.2 Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{i_1}$ is inserted in the $i_1$-th term, $p_{i_1}$, in $S$ and the $j_2$-th literal, $x_{i_2}^{j_2}$, in $p_{i_1}$ is replaced by $x_{i_2}$ where $1 \leq i_1 \leq m$, $m > 1$, $1 \leq j_1 < j_2 \leq k_{i_1}$, and $x_{i_2}$ and $x_{i_1}$ are two different missing literals of $p_{i_1}$. Then the resulting implementation denoted by $I$ is equivalent to that given by Expression (6.17) in Table 6.7 and its detection conditions are given by (C1) to (C3) in Table 7.2. The MAX-A strategy can guarantee the detection of Expression (6.17) provided that $S \not\equiv I$.

Proof: Since $S \not\equiv I$, there is at least one test case $\bar{t}$ such that $\bar{t}$ satisfies any one of the following conditions (C1) to (C3) given in Table 7.2:

- (C1) any point in $\text{UTP}_{i_1}(S)$ such that $x_{i_1} = 0$,
- (C2) any point in $\text{UTP}_{i_1}(S)$ such that $x_{i_2} = 0$,
- (C3) any point in $\in \text{NFP}_{i_1,j_1,j_2}(S)$ such that $x_{i_1} x_{i_2} = 1$,
We prove that the MAX-A strategy can select \( \tilde{t} \) in the following three cases:

1. If \( \tilde{t} \) satisfies (C1), \( \tilde{t} \) must be a unique true point.
2. If \( \tilde{t} \) satisfies (C2), \( \tilde{t} \) must be a unique true point.
3. If \( \tilde{t} \) satisfies (C3), \( \tilde{t} \) must be a near false point.

As a result, the MAX-A strategy can select \( \tilde{t} \). Hence, the result follows. \( \square \)

Theorem 10.3.3 proves that the MAX-A strategy can detect double-fault expression (6.19). The following lemma is needed before the proving Theorem 10.3.3.

**Lemma 10.3.2** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( j_1 \)-th literal, \( x_{j_1}^{i_1} \), and the \( j_2 \)-th literal, \( x_{j_2}^{i_2} \), of the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) are replaced by \( x_{i_1} \) and \( x_{i_2} \), respectively, where \( 1 \leq i_1 \leq m, m > 1, 1 \leq j_1 < j_2 \leq k_{i_1}, \) and \( x_{i_1} \) and \( x_{i_2} \) are two different missing literals of \( p_{i_1} \), the resulting implementation denoted by \( I \) is equivalent to that given by Expression (6.19) in Table 7.2 and its detection conditions are given by (C1) to (C5) in Table 7.2. If (C1), (C2), (C3) and (C4) cannot be satisfied, any test case that satisfies (C5) is a near false point of \( S \) provided that \( S \neq I \).

**Proof**: Since \( S \neq I \), there exists a test case that can satisfy any one of the following detection conditions of \( I \) as given in Table 7.2 for Expression (6.19)

- (C1) any point in \( UTP_{p_{i_1}}(S) \) such that \( x_{i_1} = 0 \),
- (C2) any point in \( UTP_{p_{i_1}}(S) \) such that \( x_{i_2} = 0 \),
- (C3) any point in \( \in NFP_{i_1,j_1}(S) \) such that \( x_{i_1}x_{i_2} = 1 \),
- (C4) any point in \( \in NFP_{i_1,j_2}(S) \) such that \( x_{i_1}x_{i_2} = 1 \), or
- (C5) any point in \( \in FP(S) \) such that \( p_{i_1,j_1,j_2}x_{i_1}x_{i_2} = 1 \).

Based on the assumptions of the lemma, (C1)–(C4) cannot be satisfied. There exists a test case that satisfies (C5). We now prove that any test case satisfying (C5) is a near false point of \( S \).

Let \( \tilde{t} \) be a test case that satisfies (C5). Then, \( p_{i_1,j_1,j_2}x_{i_1}x_{i_2}(\tilde{t}) = x_{i_1}^{i_1} \cdots \tilde{x}_{j_1}^{i_1} \cdots \tilde{x}_{j_2}^{i_1} \cdots x_{k_{i_1}}^{i_1} \cdot x_{i_1}x_{i_2}(\tilde{t}) = 1 \). Let \( \tilde{t}_1 \) be such that \( \tilde{t}_1 \) and \( \tilde{t} \) only differ at \( x_{j_2}^{i_1} \). Then, \( p_{i_1,j_1,j_2}x_{i_1}x_{i_2}(\tilde{t}_1) = x_{i_1}^{i_1} \cdots \tilde{x}_{j_1}^{i_1} \cdots \tilde{x}_{j_2}^{i_1} \cdots x_{k_{i_1}}^{i_1} \cdot x_{i_1}x_{i_2}(\tilde{t}_1) = 1 \). Note that \( \tilde{t}_1 \) must be a true point of \( S \). Otherwise, \( S(\tilde{t}_1) = 0 \). In addition, \( p_{i_1,j_1}(\tilde{t}_1) = 1 \) because \( p_{i_1,j_1}x_{i_1}x_{i_2}(\tilde{t}_1) = 1 \). Hence, \( \tilde{t}_1 \in NFP_{i_1,j_1}(S) \) such that \( x_{i_1}x_{i_2} = 1 \). This violates the assumption that (C3) cannot be satisfied.

Now \( \tilde{t}_1 \) is a true point of \( S \). Since \( p_{i_1,j_1}(\tilde{t}_1) = 1 \), \( p_{i_1}(\tilde{t}_1) = 0 \). There exists \( i_3(\neq i_1) \) such that \( p_{i_3}(\tilde{t}_1) = 1 \).

Note that \( \tilde{x}_{i_1} \) does not appear in \( p_{i_3} \). Otherwise, \( p_{i_3}(\tilde{t}_1) = \cdots \tilde{x}_{i_1} \cdots = 0 \) because \( p_{i_1,j_1}x_{i_1}x_{i_2}(\tilde{t}_1) = 1 \) implies \( x_{i_1}(\tilde{t}_1) = 1 \). Similarly, \( \tilde{x}_{i_2} \) does not appear in \( p_{i_3} \).

We are going to prove that \( x_{j_2}^{i_1} \) exists in \( p_{i_3} \). If both \( x_{j_2}^{i_1} \) and \( \tilde{x}_{j_2}^{i_1} \) do not appear in \( p_3 \), then \( p_{i_3}(\tilde{t}) = p_{i_3}(\tilde{t}_1) = 1 \) because \( \tilde{t} \) and \( \tilde{t}_1 \) differ only at \( x_{j_2}^{i_1} \). This contradicts to the fact that \( \tilde{t} \) is a false point of \( S \). If \( \tilde{x}_{j_2}^{i_1} \) appears in \( p_{i_3} \), then \( p_{i_3}(\tilde{t}_1) = \cdots \tilde{x}_{j_2}^{i_1} \cdots = 0 \).
because \( p_{i_1,j_1} x_{i_1} x_{i_2}(\bar{t}_1) = 1 \) implies \( x_{i_2}^{i_1}(\bar{t}_1) = 1 \). This contradicts to the fact that \( p_{i_3}(\bar{t}_1) = 1 \). Therefore \( x_{i_2}^{i_1} \) must appear in \( p_{i_3} \). Since \( x_{j_2}^{i_1} \) is a literal of \( p_{i_3} \), there exists \( j_3 \) such that \( x_{j_2}^{i_1} = x_{j_2}^{i_3} \). Now, since \( p_{i_3}(\bar{t}_1) = 1 \) and \( \bar{t} \) and \( \bar{t}_1 \) only differ at \( x_{j_3}^{i_3}(= x_{j_2}^{i_1}) \), we have, \( p_{i_3,j_3}(\bar{t}) = x_{i_3}^{i_3} \bar{x}_{j_3}^{i_3} \cdots x_{k_3}^{i_3}(\bar{t}) = 1 \). Since \( \bar{t} \) is a false point of \( S \), \( \bar{t} \in \text{NFP}_{i_3,j_3}(S) \).

Hence the result follows.

\[ \Box \]

**Theorem 10.3.3** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( j_1 \)-th literal, \( x_{i_1}^{i_1} \), and the \( j_2 \)-th literal, \( x_{i_2}^{i_2} \), of the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) are replaced by \( x_{i_1} \) and \( x_{i_2} \), respectively, where \( 1 \leq i_1 \leq m \), \( m > 1 \), \( 1 \leq j_1 < j_2 \leq k_{i_1} \), and \( x_{i_1} \) and \( x_{i_2} \) are two missing literals of \( p_{i_1} \), the resulting implementation denoted by \( I \) is equivalent to that given by Expression (6.19) in Table 6.7 and its detection conditions are given by (C1) to (C5) in Table 7.2. The MAX-A strategy can guarantee the detection of Expression (6.19) in Table 6.7.

**Proof**: Since \( S \neq I \), there is at least one test case \( \bar{t} \) such that \( \bar{t} \) satisfies any one of the following conditions (C1) to (C5) given in Table 7.2:

- (C1) any point in \( \text{UTP}_{i_1}(S) \) such that \( x_{i_1} = 0 \),
- (C2) any point in \( \text{UTP}_{i_1}(S) \) such that \( x_{i_2} = 0 \),
- (C3) any point in \( \text{NFP}_{i_1,j_1}(S) \) such that \( x_{i_1} x_{i_2} = 1 \),
- (C4) any point in \( \text{NFP}_{i_1,j_3}(S) \) such that \( x_{i_1} x_{i_2} = 1 \), or
- (C5) any point in \( \text{FP}(S) \) such that \( p_{i_1,j_1,j_2} x_{i_1} x_{i_2} = 1 \).

We proceed our proof using the following two cases:

Case 1. Conditions (C1) - (C4) cannot be satisfied.

Then, \( \bar{t} \) must satisfy condition (C5). By Lemma 10.3.2, \( \bar{t} \) is a near false point of \( S \). Hence, the MAX-A strategy can select it.

Case 2. The test case \( \bar{t} \) satisfies any of (C1), (C2), (C3) or (C4).

- (a) If \( \bar{t} \) satisfies (C1), \( \bar{t} \) must be a unique true point.
- (b) If \( \bar{t} \) satisfies (C2), \( \bar{t} \) must be a unique true point.
- (c) If \( \bar{t} \) satisfies (C3), \( \bar{t} \) must be a near false point.
- (d) If \( \bar{t} \) satisfies (C4), \( \bar{t} \) must be a near false point.

As a result, the MAX-A strategy can select \( \bar{t} \).

Hence, the result follows. \[ \Box \]

**Theorem 10.3.4** Let \( S = p_1 + \cdots + p_m \) be a Boolean expression in irredundant disjunctive normal form. Any test case selection strategy that subsumes the MAX-A meaningful impact strategy can detect double-fault expressions listed in Tables 10.1 to 10.3, (8.16) in Table 8.12, (6.15), (6.17), (6.13) and (6.19) in Table 6.7.
Proof: By Theorems 10.2.2, 10.3.1, 10.3.2 and 10.3.3 and the definition of subsumption of test case selection strategy.

\[ \square \]

10.4 The MAX-B strategy

Since the MAX-B strategy subsumes the MAX-A strategy, it also can detect 87 double-fault expressions that MAX-A strategy can detect by Theorem 10.3.4. For the remaining 5 double-fault expressions, the MAX-B strategy cannot guarantee to detect them. Five examples are given to illustrate that the MAX-B strategy cannot detect the double-fault expressions (6.14), (6.16), (6.18), (8.20) and (8.27).

Table 10.7: All points of \( S \) where \( S = ab + ac + bd \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( UTP_i(S) )</th>
<th>( NFP_{i,1}(S) )</th>
<th>( NFP_{i,2}(S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1100</td>
<td>0110,0100</td>
<td>1001,1000</td>
</tr>
<tr>
<td>2</td>
<td>1011,1010</td>
<td>0110,0011,0010</td>
<td>1001,0011,0001</td>
</tr>
<tr>
<td>3</td>
<td>0111,0101</td>
<td></td>
<td>0110,0100</td>
</tr>
</tbody>
</table>

\[ OTP(S) = \{ 1101, 1111, 1110 \} \text{ and } RFP(S) = \{ 0000 \}. \]

Example 10.4.1 Let \( S = ab + ac + bd \). Table 10.7 lists the sets \( UTP_i(S) \) of all unique true points, the sets \( NFP_{i,j}(S) \) of all near false points, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \).

Suppose the literals \( \bar{c} \) and \( \bar{b} \) are inserted into the first and second terms of \( S \) respectively. The resulting double-fault expression is equivalent to \( I = ab\bar{c} + a\bar{b}c + bd \).\(^{12}\) Note that, \( S \) and \( I \) are not equivalent because \( S \) and \( I \) evaluate to 1 and 0 on 1110, respectively.

Now, let \( T \) be the set that includes (1) all unique true points in \( UTP_i(S) \) for every \( i \), (2) all near false points in \( NFP_{i,j}(S) \) for every \( i \) and \( j \), (3) 2 overlapping true points 1111 and 1101, and (4) 1 remaining false 0000. It should be noted that \( T \) satisfies the MAX-B strategy because it contains all unique true points of \( S \), all near false points of \( S \), 2 out of 3 overlapping true points of \( S \) and 1 remaining false point of \( S \). Since \( S \) and \( I \) agree on all points in \( T \). Hence, the MAX-B strategy cannot guarantee to distinguish \( S \) and \( I \).

\[ \diamond \]

Example 10.4.2 Let \( S = ab + ac + \bar{b}d + \bar{a}d + \bar{b}\bar{e} \). Table 10.8 lists the sets \( UTP_i(S) \) of all unique true points, the sets \( NFP_{i,j}(S) \) of all near false points, the set \( OTP(S) \) of all overlapping true points and the set \( RFP(S) \) of all remaining false points of \( S \).

Suppose the literal \( e \) is inserted into the first term \( ab \) and the literal \( a \) of the second term \( ac \) of \( S \) is replaced by the literal \( d \). The resulting double-fault expression is

\[^{12}\text{It is a special instance of Expression 6.14.} \]
Table 10.8: All points of $S$ where $S = ab + ac + \overline{bd} + \overline{ad} + b\overline{ce}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$UTP_i(S)$</th>
<th>$NFP_{i,1}(S)$</th>
<th>$NFP_{i,2}(S)$</th>
<th>$NFP_{i,3}(S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11011, 11001</td>
<td>01101, 01100, 01001</td>
<td>10011, 10010, 10010</td>
<td>--</td>
</tr>
<tr>
<td>2</td>
<td>10111, 10110</td>
<td>01101, 01100, 01001</td>
<td>01101, 01100, 01001</td>
<td>--</td>
</tr>
<tr>
<td>3</td>
<td>10001, 10000, 00101, 00100, 00001, 00000</td>
<td>10111, 01110, 01011, 00111, 00011, 00010</td>
<td>10011, 01010, 01001</td>
<td>--</td>
</tr>
<tr>
<td>4</td>
<td>01101, 01100, 01001</td>
<td>10011, 10010, 10001</td>
<td>01101, 01100, 01001</td>
<td>--</td>
</tr>
<tr>
<td>5</td>
<td>01000</td>
<td>10010</td>
<td>01100</td>
<td>01001</td>
</tr>
</tbody>
</table>

$OTP(S) = \{ 11100, 10101, 01010, 11101, 11100, 11110, 11101, 10100, 11111 \}$ and $RFP(S) = \emptyset$.

equivalent to $I = abe + cd + \overline{bd} + \overline{ad} + b\overline{ce}$. Note that, $S$ and $I$ are not equivalent because $S$ and $I$ evaluate to 1 and 0 on 11100, respectively. Now, let $T$ be the set that includes (1) all unique true points in $UTP_i(S)$ for every $i$, (2) all near false points in $NFP_{i,j}(S)$ for every $i$ and $j$, and (3) 4 overlapping true points 10101, 11000, 11110 and 11111. It should be noted that $T$ satisfies the MAX-B strategy because it contains all unique true points of $S$, all near false points of $S$, 4 out of the 9 overlapping true points of $S$, and there are no remaining false points of $S$. Since $S$ and $I$ agree on all points in $T$. Hence, the MAX-B strategy cannot guarantee to distinguish $S$ and $I$.

Example 10.4.3 Let $S = abc + abd + \overline{cd}\overline{f} + \overline{cde} + \overline{cde} + \overline{cde} + \overline{cdf}$. Table 10.9 lists the sets $UTP_i(S)$ of all unique true points, the sets $NFP_{i,j}(S)$ of all near false points, the set $OTP(S)$ of all overlapping true points and the set $RFP(S)$ of all remaining false points of $S$. Suppose the literal $c$ in the first term $abc$ and the literal $d$ in second term $abd$ of $S$ are replaced by literals $e$ and $f$, respectively. The resulting double-fault expression is equivalent to $I = abe + abf + \overline{cdf} + \overline{cde} + \overline{cde} + \overline{cdf}$. Note that, $S$ and $I$ are not equivalent because $S$ and $I$ evaluate to 1 and 0 on 111100, respectively. Now, let $T$ be the set that includes (1) all unique true points in $UTP_i(S)$ for every $i$, (2) all near false points in $NFP_{i,j}(S)$ for every $i$ and $j$, (3) 4 overlapping true points 110100, 110101, 111110 and 111111, and (4) 1 remaining false points 001111. It should be noted that $T$ satisfies the MAX-B strategy because it contains all unique true points of $S$, all near false points of $S$, 4 out of the 12 overlapping true points of $S$, and 1 remaining false point of $S$. Since $S$ and $I$ agree on all points in $T$. Hence, the MAX-B strategy cannot guarantee to distinguish $S$ and $I$.

---

\[^{13}\text{It is a special instance of Expression 6.16.}\]
\[^{14}\text{It is a special instance of Expression 6.18.}\]
Table 10.9: All points of $S$ where $S = abc + abd + c\bar{d}f + \bar{c}de + \bar{c}\bar{d}e$

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$UTP_i(S)$</td>
<td>111011, 111001</td>
<td>110111, 110110</td>
<td>101010, 101000, 011010, 011000, 001010, 001000</td>
<td>100101, 100100, 010101, 010100, 000101, 000100</td>
<td>110010, 100010, 010010, 000010</td>
<td>110001, 100001, 010001, 000001</td>
</tr>
<tr>
<td>$NFP_{i,1}(S)$</td>
<td>011111, 011110, 011101, 011100, 011011, 011001</td>
<td>011111, 011110, 011101, 011100, 011011, 011010</td>
<td>101100, 100000, 010000, 000000</td>
<td>101101, 101100, 011101, 011100, 011101, 011100</td>
<td>101011, 011110, 110110, 001001, 101001</td>
<td>101011, 101010, 101101, 001001, 101001</td>
</tr>
<tr>
<td>$NFP_{i,2}(S)$</td>
<td>101111, 101110, 101101, 101100, 101011, 101001</td>
<td>101111, 101110, 101101, 101100, 101011, 101010</td>
<td>110000, 100000, 100000, 000000</td>
<td>100111, 100110, 010111, 010110, 000111, 000110</td>
<td>100111, 010111, 010110, 000111, 000110</td>
<td>000110, 000110, 000000, 000000, 000000</td>
</tr>
<tr>
<td>$NFP_{i,3}(S)$</td>
<td>110000</td>
<td>110000</td>
<td>101011, 101001, 011011, 011010, 001011, 001001, 001000, 000101</td>
<td>100111, 100110, 010111, 010110, 000111, 000110, 000010, 000001</td>
<td>110000, 100000, 100000, 000000</td>
<td>110000, 100000, 100000, 000000</td>
</tr>
</tbody>
</table>

$OTP(S) = \{ 000011, 111010, 010011, 111000, 100011, 110011, 110100, 110101, 111110, 111111, 111100, 111101 \}$ and $RFP(S) = \emptyset$.

Table 10.10: All points of $S$ where $S = ab + ac + ad + b\bar{d}e + \bar{b}cde + \bar{b}\bar{c}de$

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$UTP_i(S)$</td>
<td>11001</td>
<td>10101</td>
<td>10011</td>
<td>01000</td>
<td>00100</td>
<td>00010</td>
<td>10001, 00001</td>
</tr>
<tr>
<td>$NFP_{i,1}(S)$</td>
<td>01111, 01110, 01101, 01011, 01010, 01001</td>
<td>01111, 01110, 01101, 01011, 01010, 00111, 00110, 00010</td>
<td>01111, 01110, 01101, 01011, 01010, 00111, 00110, 00010</td>
<td>10000, 00000, 00000, 00000</td>
<td>01010, 01001</td>
<td>01000, 00000</td>
<td></td>
</tr>
<tr>
<td>$NFP_{i,2}(S)$</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>01110, 01101, 01010, 00110</td>
<td>01110, 01101, 01010, 00110</td>
<td>00110, 00110</td>
<td>00011</td>
</tr>
<tr>
<td>$NFP_{i,3}(S)$</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>01101, 01001</td>
<td>01101, 00101</td>
<td>10000, 00000</td>
<td>00011</td>
</tr>
<tr>
<td>$NFP_{i,4}(S)$</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00011</td>
<td>10000, 00000</td>
</tr>
</tbody>
</table>

$OTP(S) = \{ 11100, 11101, 11000, 11110, 10010, 11010, 10110, 10100, 11011, 10111, 11111, 01100 \}$ and $RFP(S) = \emptyset$.

**Example 10.4.4** Let $S = ab + ac + ad + b\bar{d}e + \bar{b}cde + \bar{b}\bar{c}de$. Table 10.10 lists the sets $UTP_i(S)$ of all unique true points, the sets $NFP_{i,j}(S)$ of all near false points,
the set $OTP(S)$ of all overlapping true points and the set $RFP(S)$ of all remaining false points of $S$. Suppose the first term $ab$ of $S$ is omitted and the literal $d$ in the third term $ad$ of $S$ is replaced by the literal $e$. The resulting double-fault expression is equivalent to $I = ac + ae + bd\overline{e} + cd\overline{e} + b\overline{cd} + b\overline{c}d\overline{e}$.\textsuperscript{15} Note that, $S$ and $I$ are not equivalent because $S$ and $I$ evaluate to 1 and 0 on 11010, respectively.

Now, let $T$ be the set that includes (1) all unique true points in $UTP_i(S)$ for every $i$, (2) all near false points in $NFP_{i,j}(S)$ for every $i$ and $j$, and (3) 4 overlapping true points, 11100, 11101, 11000, and 11110. It should be noted that $T$ satisfies the MAX-B strategy because it contains all unique true points of $S$, all near false points of $S$, 4 out of the 12 overlapping true points of $S$, and there are no remaining false points of $S$. Since $S$ and $I$ agree on all points in $T$. Hence, the MAX-B strategy cannot guarantee to distinguish $S$ and $I$. \hfill\textdegree

**Example 10.4.5** Please note that we use the same $S$ here as in Example 10.4.4. Table 10.10 lists the sets $UTP_i(S)$ of all unique true points, the sets $NFP_{i,j}(S)$ of all near false points, the set $OTP(S)$ of all overlapping true points and the set $RFP(S)$ of all remaining false points of $S$.

Suppose the first two terms $ab$ and $ac$ of $S$ are wrongly implemented as $ab \cdot ac$ and the literal $d$ in the third term $ad$ of $S$ is replaced by the literal $e$. The resulting double-fault expression is equivalent to $I = abac + ae + bd\overline{e} + cd\overline{e} + b\overline{cd} + b\overline{c}d\overline{e}$.\textsuperscript{16} Note that, $S$ and $I$ are not equivalent because $S$ and $I$ evaluate to 1 and 0 on 11010, respectively.

Now, let $T$ be the set that includes (1) all unique true points in $UTP_i(S)$ for every $i$, (2) all near false points in $NFP_{i,j}(S)$ for every $i$ and $j$, and (3) 4 overlapping true points, 11100, 11101, 11000, and 11110. It should be noted that $T$ satisfies the MAX-B strategy because it contains all unique true points of $S$, all near false points of $S$, 4 out of the 12 overlapping true points of $S$, and there are no remaining false points of $S$. Since $S$ and $I$ agree on all points in $T$. Hence, the MAX-B strategy cannot guarantee to distinguish $S$ and $I$. \hfill\textdegree

**10.5 Summary**

Table 10.11 summarizes fault detection capability of the four test case selection strategies considered in Section 2.3. The ‘√’ is used to denote that the strategy can detect the corresponding double-fault expressions while ‘×’ is used to denote that the strategy cannot detect the double-fault expressions. For example, the seventh row of Table 10.11 shows that, for double-fault expression (6.13), it cannot be detected by the BASIC and MUMCUT strategies, but it can be detected by either the MAX-A or the MAX-B strategy.

\textsuperscript{15}It is a special instance of Expression 8.20.
\textsuperscript{16}It is a special instance of Expression 8.27.
Table 10.11: Double-Fault Expressions Detected by Each Test Strategy

<table>
<thead>
<tr>
<th>Expr No.</th>
<th>Total</th>
<th>BASIC</th>
<th>MUMCUT</th>
<th>MAX-A</th>
<th>MAX-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4.1) – (4.27)</td>
<td>27</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(4.53)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(4.70)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(4.73)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(4.76)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(6.1) – (6.12)</td>
<td>12</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(6.13)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(6.14)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(6.15)</td>
<td>1</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(6.16)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(6.17)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(6.18)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(6.19)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.1) – (8.19)</td>
<td>19</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.20)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(8.21) – (8.26)</td>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.27)</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>(8.28) – (8.36)</td>
<td>9</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.43)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.52)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.73)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.74)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.77)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(8.78)</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Total</td>
<td>92</td>
<td>83</td>
<td>84</td>
<td>87</td>
<td>87</td>
</tr>
</tbody>
</table>

Since any test case selection strategy that subsumes BASIC strategy can detect all double fault related to terms, Table 10.11 shows the BASIC, MUMCUT, MAX-A and MAX-B strategies can detect all 31 double-fault expressions related to terms. For double fault related to literal, the BASIC and MUMCUT strategies can detect 12 and 13 double-fault expressions, respectively and the MAX-A and MAX-B can detect 16 faulty expressions, respectively. These four strategies can detect 40 out of 42 faulty expression related to a term and a literal. The last row shows the total number of faulty expressions detected by these strategies. There are 83 (=31+12+40) and 84 (=31+13+40) double-fault expressions can be detected by the BASIC and MUMCUT strategies, respectively. While both MAX-A and MAX-B strategies can detect 87(=31+16+40) out of 92 double-fault expressions. It is noted that most of the double-fault expressions can be detected by the existing test case selection strategies which originally developed to detect single faults.

Moreover, it is easy to see that the BASIC and the MUMCUT strategies cannot detect 9 and 8 double-fault expressions, respectively, while the MAX-A and MAX-B strategies cannot detect 5 double-fault expressions.
Chapter 11

New Testing Strategies for Double Fault Detection

As discussed in the previous chapter, existing test case selection strategies for single fault detection cannot guarantee to detect double faults. Therefore, new fault-based test case selection strategies are needed for the detection of double faults. Six test case selection strategies are proposed in this chapter to supplement the MUMCUT strategy to detect double faults.

11.1 Approach for developing new strategies

There are two approaches in developing new test case selection strategy. The first approach is to propose new test case selection strategies from scratch, that is without making use of any existing strategies for single fault detection. However, test case selection strategies developed via this approach may not guarantee to detect all single faults. This is illustrated in Example 11.1.1

Example 11.1.1 Let $S = ab + cd + ef$. Table 11.1 lists the sets $UTP_i(S)$ of all unique true points, the sets $NFP_{ij}(S)$ of all near false points, the set $OTP(S)$ of all overlapping true points and the set $RFP(S)$ of all remaining false points of $S$. Let $T$ be the set $\{110010, 110101, 001110, 011101, 010111, 101011, 010101, 011010, 100110, 101001, 100101, 101010, 011001, 010110\}$. Test cases in $T$ have been underlined in Table 11.1 for ease of reference.

Now, as detailed below, test cases in $T$ collectively satisfy all detection conditions of all 92 double-fault expressions listed in Tables 4.8, 6.7 and 8.12. Hence, $T$ can detect all double faults studied in this thesis.

(A) For all double-fault expressions in Table 4.8, double-fault expressions (6.1) to (6.12) in Table 6.7, and double-fault expressions (8.1) to (8.19), (8.21) to (8.26), (8.28) to (8.36), (8.43), (8.52), (8.73) (8.74), (8.77) and (8.78) in Table 8.12: Since $S$ is in irredundant disjunctive normal form, the sets $UTP_i(S)$
and $NFP_{i,j}(S)$ are non-empty for every $i$ and $j$. Now, the set $T$ contains at least one element in $UTP_i(S)$ for every $i$ and at least one element in $NFP_{i,j}(S)$ for every $i$ and $j$. Thus, $T$ satisfies the detection conditions of these 83 double-fault expressions. As a result, $T$ detects all these double-fault expressions. More precisely, $T$ can distinguish $S$ from all these 83 double-fault expressions.

(B) For double-fault expressions (6.13), (6.16) and (6.18) in Table 6.7 and double-fault expressions (8.20) and (8.27) in Table 8.12: Conditions (C3), (C4) and (C6) of double-fault expressions (6.13), (6.16) and (6.18), respectively, as shown in Table 7.2, and conditions (C4) and (C6) of double-fault expressions (8.20) and (8.27), respectively, as shown in Table 9.2 are exactly the same: “any point in $NFP_{i,j}(S)$ such that $x_{t_2} = 1$” where $x_{t_2}$ is a missing literal of $p_{t_2}$. Test cases in $T$ are selected from $NFP_{i,j}(S)$ for every $i$ and $j$ and they collectively satisfy the following condition: “those points from $NFP_{i,j}(S)$ make every missing literal $x_l$ of $p_i$ evaluates to 1 for every $i$ and $j$”. Hence, $T$ can detect double-fault expressions (6.13), (6.16) and (6.18) in Table 6.7, and double-fault expressions (8.20) and (8.27) in Table 8.12.

(C) For double-fault expression (6.14) in Table 6.7: The detection condition is “(C1) any point in $UTP_{i_1}(S)$ such that $x_{t_1} = 0$ or (C2) any point in $UTP_{i_2}(S)$ such that $x_{t_2} = 0$ or (C3) any point in $(TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1, i_2} TP_i(S)$ such that $x_{t_1} + x_{t_2} = 0$”. Those test cases in $T$, selected from $UTP_i(S)$ for every $i$, collectively satisfy the condition: “those points from $UTP_i(S)$ make every missing literal $x_l$ of $p_i$ evaluate to 0 for every $i$”, except for the missing literal
\( \bar{c} \) of the first term and \( \bar{a} \) of the second term in \( S \), respectively. Therefore, all selected points from \( UTP(S) \) can detect all double-fault expressions of type (6.14), except \( I = ab\bar{c} + \bar{a}cd + ef \) in which the literal \( \bar{c} \) is inserted into \( ab \) and \( \bar{a} \) is inserted into \( cd \) of \( S \). However, \( I \) can be detected by the test case 111100 in \( T \) which is selected from \( OTP(S) \). Hence, \( T \) can detect double-fault expression (6.14).

(D) For double-fault expressions (6.15), (6.17) and (6.19) in Table 6.7: The first two detection conditions for these double-fault expressions is: “(C1) any point in \( UTP_{i_1}(S) \) such that \( x_{l_1} = 0 \), or (C2) any point in \( UTP_{i_1}(S) \) such that \( x_{l_2} = 0 \)” where both \( x_{l_1} \) and \( x_{l_2} \) are two different missing literals of \( p_{i_1} \). Those test cases in \( T \), selected from \( UTP_i(S) \) for every \( i \), collectively satisfy the condition: “the points in \( UTP_{i}(S) \) that make \( x_{l_1} \) evaluate to 0, or the points in \( UTP_{i}(S) \) that make \( x_{l_2} \) evaluate to 0 where \( x_{l_1} \) and \( x_{l_2} \) are two different missing literals of \( p_i \), for every \( i \)” Hence, \( T \) can detect double-fault expressions (6.15), (6.17) and (6.19) in Table 6.7.

However, \( T \) cannot detect the literal insertion fault in the single-fault expression \( I_1 = ab\bar{c} + cd + ef \) (where the literal \( \bar{c} \) has been inserted into the first term \( ab \) of \( S \)) because \( S \) and \( I_1 \) agree on all points in \( T \). \( S \) and \( I_1 \) are not equivalent because \( S \) and \( I_1 \) evaluate to 1 and 0 on 111010, respectively.

The second approach is to propose new test case selection strategies that can supplement those existing single fault detection strategies for double fault detection. Hence, when all these strategies are applied, both single and double faults are detected. Moreover, this approach is more practical in testing than the first approach because of the following two reasons:

1. Since time and resources allocated for testing is limited, testing practitioners may opt to select test cases using existing test case selection strategies to first assure that the program is free from single faults. When there is still room for further testing, they may then opt to use supplementary strategies that can help to increase the chance of detecting other fault classes, such as double faults.

2. As shown in Chapter 10, some of the existing test case selection strategies for single fault detection can guarantee to detect certain double faults within Boolean expressions. It would be simpler to develop new strategies that focus only on the undetected double faults.

Therefore, we adopt the second approach. The next question is to choose one to be the base strategy.
11.2 The base test case selection strategy

Before proposing any new supplementary strategy, we need to identify one existing strategy as the basis for detecting double fault classes.

Table 11.2: Comparison of Different Test Strategies

<table>
<thead>
<tr>
<th>Fault Type</th>
<th>Faulty Expressions</th>
<th>BASIC</th>
<th>MUMCUT</th>
<th>MAX-A</th>
<th>MAX-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Fault (Detected)</td>
<td>9(^a)</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Double Fault (Detected)</td>
<td>92(^b)</td>
<td>83</td>
<td>84</td>
<td>87</td>
<td>87</td>
</tr>
<tr>
<td>Size of Test Set(^c)</td>
<td>9.8%</td>
<td>11.9%</td>
<td>40.6%</td>
<td>48.2%</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\)For single fault classes considered in thesis, there are 9 single-fault expressions as shown in Table 2.1.

\(^b\)For double fault classes studied in thesis, there are 92 double-fault expressions as shown in Table 10.11. The numbers of detected expressions for each of the four strategies can also be found in Table 10.11.

\(^c\)Mean size of test set as percentage of exhaustive test set for TCAS II expressions from [56, 60]

Table 11.2 summarizes the numbers of single-fault and double-fault expressions that can be detected by the BASIC, MUMCUT, MAX-A and MAX-B strategies and their corresponding test set sizes required. It has been shown that the BASIC strategy cannot guarantee to detect LIF and LRF [9], while the MUMCUT strategy (and hence the MAX-A and MAX-B strategies) guarantees to detect all nine single fault classes [9, 33]. Moreover, empirical studies have shown that the MUMCUT, MAX-A and MAX-B strategies require, on average, 11.9%, 40.6% and 48.2% of the entire input domain, respectively [56, 60]. Thus, the MUMCUT strategy is more cost-effective than any of the MAX-A and MAX-B strategies in detecting single faults. Furthermore, as shown in Table 11.2, the MUMCUT, MAX-A and MAX-B strategies cannot detect 8, 5, and 5 out of the 92 double-fault expressions, respectively. Thus, the MUMCUT strategy is almost as effective as the MAX-A and MAX-B strategies in detecting the double fault expressions considered in this study, but at a much smaller sized test set. Hence, the MUMCUT strategy is chosen to be supplemented by other strategies for double fault detection. Note that, although this chapter only discuss the case for MUMCUT strategy, the idea of supplementing other strategies is similar.

Next, as shown in Table 10.11, the MUMCUT strategy cannot guarantee to detect 8 out of the 92 double-fault expressions, namely, Expressions (6.13), (6.14), (6.16)–(6.19) in Table 6.7 and Expressions (8.20) and (8.27) in Table 8.12. The first 6 double-fault expressions are from double fault related to literals only and the remaining 2 expressions are from double fault related to term and literal. In Section 11.3, new test case selection strategies are developed to supplement the MUMCUT strategy to detect the 6 faulty expressions related to double fault. We
Table 11.3: Organization of supplementary strategies for double literal faults

<table>
<thead>
<tr>
<th>New Strategy</th>
<th>Detailed Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplementary Multiple False Point (SMFP) strategy</td>
<td>Section 11.3.1</td>
</tr>
<tr>
<td>Supplementary Multiple Overlapping True Point (SMOTP) strategy</td>
<td>Section 11.3.2</td>
</tr>
<tr>
<td>Supplementary Multiple Unique True Point (SMUTP) strategy</td>
<td>Section 11.3.3</td>
</tr>
<tr>
<td>Pairwise Multiple Unique True Point (PMUTP) strategy</td>
<td>Section 11.3.4</td>
</tr>
<tr>
<td>Pairwise Multiple Near False Point (PMNFP) strategy</td>
<td>Section 11.3.5</td>
</tr>
<tr>
<td>Supplementary Pairwise Multiple False Point (SPMFP) strategy</td>
<td>Section 11.3.6</td>
</tr>
</tbody>
</table>

Table 11.4: Suggested test case selection strategies for 6 double-fault expression

<table>
<thead>
<tr>
<th>Expr No.</th>
<th>Test Case Selection Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(6.13)</td>
<td>MUMCUT strategy and then SMFP strategy</td>
</tr>
<tr>
<td>(6.14)</td>
<td>MUMCUT strategy and then SMOTP strategy</td>
</tr>
<tr>
<td>(6.16)</td>
<td>MUMCUT strategy and then PMUTP, SMUTP, and SMOTP strategies</td>
</tr>
<tr>
<td>(6.17)</td>
<td>MUMCUT strategy and then PMNFP strategy</td>
</tr>
<tr>
<td>(6.18)</td>
<td>MUMCUT strategy and then PMUTP, SMUTP, and SMOTP strategies</td>
</tr>
<tr>
<td>(6.19)</td>
<td>MUMCUT strategy and then PMNFP and SPMFP strategies</td>
</tr>
</tbody>
</table>

also prove that one of the new strategies together with the MUMCUT strategy can detect Expressions (8.20) and (8.27) in Section 11.4.

11.3 New Test Case Selection Strategies For Detecting Double Faults Related to Literals Only

In this section, six test case selection strategies are developed to supplement the MUMCUT strategy to detect six undetected double-fault expressions related to literals only. The discussion on these newly proposed supplementary strategies are organized into 6 subsections as shown in Table 11.3. For each of the six double-fault expressions, Table 11.4 summarizes the corresponding test cases selection strategies that can detect it. For example, the MUMCUT strategy together with the SMFP strategy can guarantee to detect double-fault expression (6.13).

11.3.1 The Supplementary Multiple False Point (SMFP) Strategy

We propose the *Supplementary Multiple False Point (SMFP)* strategy to detect double-fault expression (6.13) in Table 6.7. It aims at selecting test cases satisfying the detection condition (C4) of double-fault expression (6.13), that is “any point in $FP(S)$ such that $p_{i_1,j_1,j_2}x_{l_2} = 1$ where $x_{l_2}$ is a missing literal of $p_i$”, as listed in Table 7.2.
Table 11.5: All false points of $S$ where $S = ab + cd + ef$

| 000000, 000010, 000100, 000101, 000110, 001000, 001001, 001010, 010000, 010001, 010010, 010100, 010101, 010110, 011000, 011001, 011010, 100000, 100001, 100010, 100100, 100101, 100110, 101000, 101001, 101010 |

The SMFP strategy requires to select test cases from $FP(S)$ to form a set $T$ such that, for every term $p_i$ of $S$ and for every pair of literals $x^i_{j_1}$ and $x^i_{j_2}$ in $p_i$ and for every missing literal $x_l$ of $p_i$,

1. there is a test case $\vec{t}_1 \in T$ such that $p_{i,j_1,j_2} = 1$ and $x_l = 1$ if possible; and
2. there is a test case $\vec{t}_2 \in T$ such that $p_{i,j_1,j_2} = 1$ and $x_l = 0$ if possible.

In other words, points in $T$ are from $FP(S)$ such that (1) $p_{i,j_1,j_2} = 1$ for every pair of literals $x^i_{j_1}$ and $x^i_{j_2}$ in every term $p_i$, and (2) they cover all possible truth values of every missing literal $x_l$ of every term $p_i$. Example 11.3.1 illustrates how to select test cases satisfying the SMFP strategy.

**Example 11.3.1** Let $S = ab + cd + ef$. Table 11.5 lists the set of all false points of $S$. Now, let $T = \{000101, 001010, 010001, 010100, 100010, 101000\}$. Test cases in $T$ are underlined in Table 11.5 for ease of reference. The set $T$ satisfies the SMFP strategy because of the following reasons

1. For the first term $ab$ of $S$, the test cases 000101 and 001010 can make $\overline{a} \overline{b} = 1$ and they cover all possible truth values of every missing literal ($c$, $d$, $e$ and $f$) of $ab$. Hence, they satisfy the requirements of the SMFP strategy for the first term of $S$.

2. Similarly, test cases 010001 and 100010 satisfy the requirements of the SMFP strategy for the second term of $S$, test cases 010100 and 101000 for the third term of $S$.

We prove in Theorem 11.3.1 that the SMFP and MUMCUT strategies can select test cases to detect double-fault expression (6.13).

**Theorem 11.3.1** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $j_1$-th literal, $x_{j_1}^i$, of the $i$-th term, $p_i$, in $S$ is omitted and the $j_2$-th literal, $x_{j_2}^i$, of $p_i$ is replaced by $x_{i_2}$, where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_i$, $k_i$ is the number of literals in $p_i$ and $x_{i_2}$ is a missing literal of $p_i$, the resulting implementation denoted as $I$ will be equivalent to that given by double-fault expression (6.13) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C4) of double-fault expression (6.13) in Table 7.2. The SMFP strategy can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.13) provided that $S \not\equiv I$. 

272
Proof:

Since $S$ and $I$ are not equivalent, by Theorem 7.2.5, there is a point $\vec{t}_1$ that satisfies any of the following conditions:

(C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{l_2} = 0$

(C2) $\vec{t}_1 \in NFP_{i_1,j_1}(S)$ such that $x_{l_2} = 1$

(C3) $\vec{t}_1 \in NFP_{i_1,j_2}(S)$ such that $x_{l_2} = 1$, or

(C4) $\vec{t}_1 \in FP(S)$ such that $p_{i_1,j_1,j_2}x_{l_2} = 1$.

We then have the following cases:

Case 1 $\vec{t}_1$ satisfies (C1). Then, $\{\vec{t} \in UTP_{i_1}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset$. Since $x_{l_2}$ is a missing literal of $p_{i_1}$, the MUTP strategy can select a point that satisfies (C1).

Case 2 $\vec{t}_1$ satisfies (C2). Then, $\{\vec{t} \in NFP_{i_1,j_1}(S) : x_{l_2}(\vec{t}) = 1\} \neq \emptyset$. Since $x_{l_2}$ is a missing literal of $p_{i_1}$, the MNFP strategy can select a point that satisfies (C2).

Case 3 $\vec{t}_1$ satisfies (C3). Then, $\{\vec{t} \in NFP_{i_1,j_2}(S) : x_{l_2}(\vec{t}) = 1\} \neq \emptyset$. Since $x_{l_2}$ is a missing literal of $p_{i_1}$, the MNFP strategy can select a point that satisfies (C3).

Case 4 $\vec{t}_1$ satisfies (C4). Then, $\{\vec{t} \in FP(S) : p_{i_1,j_1,j_2}x_{l_2}(\vec{t}) = 1\} = \{\vec{t} \in FP(S) : p_{i_1,j_1,j_2}(\vec{t}) = 1 \text{ and } x_{l_2}(\vec{t}) = 1\} \neq \emptyset$. The SMFP strategy can select a point that satisfies (C4).

Hence, the result follows. □

11.3.2 The Supplementary Multiple Overlapping True Point (SMOTP) Strategy

We propose the Supplementary Multiple Overlapping True Point (SMOTP) strategy to detect double-fault expression (6.14) in Table 6.7. In fact, it aims at selecting test cases that satisfy the following detection conditions

1. (C3) of double-fault expression (6.14) (that is, “any point in $\left((TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left(\bigcup_{i=1 \atop i \neq i_1,i_2}^{m} TP_{i}(S)\right)\right)$ such that $x_{l_1} + x_{l_2} = 0$"),

273
2. (C5) of double-fault expression (6.16) (that is, “any point in \(\left( TP_{i_1}(S) \cap TP_{i_2}(S)\right) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S)\right)\) such that \(x_{i_1} + x_{i_2} = 0\), and

3. (C7) of double-fault expression (6.18) (that is, “any point in \(\left( TP_{i_1}(S) \cap TP_{i_2}(S)\right) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S)\right)\) such that \(x_{i_1} + x_{i_2} = 0\)."

The SMOTP strategy requires to select test cases from \(\left( TP_{i_1}(S) \cap TP_{i_2}(S)\right) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S)\right)\), for every two different terms \(p_{i_1}\) and \(p_{i_2}\) of \(S\), to form a set \(T\) such that, for every missing literal \(x_{i_1}\) of \(p_{i_1}\) and every missing literal \(x_{i_2}\) of \(p_{i_2}\),

1. there is a test case \(\vec{t}_1 \in T\) such that \(x_{i_1} = 0\) and \(x_{i_2} = 0\), if possible;
2. there is a test case \(\vec{t}_2 \in T\) such that \(x_{i_1} = 0\) and \(x_{i_2} = 1\), if possible;
3. there is a test case \(\vec{t}_3 \in T\) such that \(x_{i_1} = 1\) and \(x_{i_2} = 0\), if possible; and
4. there is a test case \(\vec{t}_4 \in T\) such that \(x_{i_1} = 1\) and \(x_{i_2} = 1\), if possible.

In other words, points in \(T\) are from \(\left( TP_{i_1}(S) \cap TP_{i_2}(S)\right) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S)\right)\) for every pair of different terms \(p_{i_1}\) and \(p_{i_2}\) of \(S\) such that they can cover all possible truth value combinations of \(x_{i_1}\) and \(x_{i_2}\) (that is \(00, 01, 10\) and \(11\)) for every missing literal \(x_{i_1}\) of \(p_{i_1}\) and every missing literal \(x_{i_2}\) of \(p_{i_2}\). Example 11.3.2 illustrates how to select test cases satisfying the SMOTP strategy.

**Example 11.3.2** Let \(S = ab + cd + e\). The set of overlapping true points of \(S\) is \{00111, 01111, 10111, 11001, 11011, 11101, 11110, 11111\}. Now, let \(T = \{00111, 01111, 10111, 11001, 11011, 11101, 11110\}\). The set \(T\) satisfies the SMOTP strategy because of the following reasons

(1) For the first two terms \(ab\) and \(cd\) of \(S\), \(\left( TP_1(S) \cap TP_2(S)\right) \setminus TP_3(S)\) = \{11110\}. This test case, 11110, covers all possible truth value combinations of every possible pair of missing literals \(x_{i_1}\) of \(ab\) and \(x_{i_2}\) of \(cd\). Hence, 11110 satisfies the requirements of the SMOTP strategy on \(\left( TP_1(S) \cap TP_2(S)\right) \setminus TP_3(S)\).

(2) For the first term \(ab\) and third term \(e\) of \(S\), \(\left( TP_1(S) \cap TP_3(S)\right) \setminus TP_2(S)\) = \{11001, 11011, 11101\}. The test cases 11001, 11011, and 11101 can cover all possible truth value combinations of every possible pair of missing literals \(x_{i_1}\) of
ab and $x_{l_2}$ of $e$. Thus, they satisfy the requirements of the SMOTP strategy on
$$\left( (TP_1(S) \cap TP_3(S)) \setminus TP_2(S) \right).$$

(3) Similarly, test cases 00111, 01111 and 10111 cover all possible truth value combinations of every possible pair of missing literals $x_{l_1}$ of $cd$ and $x_{l_2}$ of $e$. Thus, they satisfy the requirements of the SMOTP strategy on
$$\left( (TP_2(S) \cap TP_3(S)) \setminus TP_1(S) \right).$$

We now prove that the SMOTP strategy supplements the MUMCUT strategy to guarantee the detection of double-fault expression (6.14). Since extra strategies are required to detect double-fault expressions (6.16) and (6.18), we will postpone the discussion to later sections.

**Theorem 11.3.2** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two literals $x_{l_1}$ and $x_{l_2}$ are inserted into the $i_1$-th term, $p_{i_1}$, and the $i_2$-th term, $p_{i_2}$, in $S$, respectively, where $1 \leq i_1 < i_2 \leq m$, $x_{l_1}$ and $x_{l_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting implementation denoted as $I$ will be equivalent to that given by double-fault expression (6.14) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C3) of double-fault expression (6.14) in Table 7.2. The SMOTP strategy can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.14) provided that $S \not\equiv I$.

**Proof:**

Since $S$ and $I$ are not equivalent, by Theorem 7.3.1, there is a point $\vec{t}_1$ that satisfies any of the following conditions

(C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{l_1} = 0$

(C2) $\vec{t}_1 \in UTP_{i_2}(S)$ such that $x_{l_2} = 0$, or

(C3) $\vec{t}_1 \in \left( TP_{i_1}(S) \cap TP_{i_2}(S) \right) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right)$ such that $x_{l_1} + x_{l_2} = 0$.

We then have the following cases:

Case 1 $\vec{t}_1$ satisfies (C1). Then, $\{\vec{t} \in UTP_{i_1}(S) : x_{l_1}(\vec{t}) = 0\} \neq \emptyset$. Since $x_{l_1}$ is a missing literal of $p_{i_1}$, the MUTP strategy can select a point that satisfies (C1).

Case 2 $\vec{t}_1$ satisfies (C2). Then, $\{\vec{t} \in UTP_{i_2}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset$. Since $x_{l_2}$ is a missing literal of $p_{i_2}$, the MUTP strategy can select a point that satisfies (C2).
Table 11.6: All unique true points of $S$ where $S = abc + def$

| $UTP_1(S)$ | 111000, 111001, 111010, 111011, 111100, 111101, 111110 |
| $UTP_2(S)$ | 000111, 001111, 010111, 011111, 100111, 101111, 110111 |

Case 3 $\vec{t}_1$ satisfies (C3). Then, $\{\vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i=1}^{m} TP_i(S) \right) : x_{l_1}(\vec{t}) + x_{l_2}(\vec{t}) = 0 \}$ is not an empty set. The SMOTP strategy can select a point that satisfies (C3).

Hence, the result follows. □

11.3.3 The Supplementary Multiple Unique True Point (SMUTP) Strategy

We propose the Supplementary Multiple Unique True Point (SMUTP) strategy to select test cases that satisfy the following detection conditions

1. (C1) of double-fault expression (6.16) (that is, “any point in $UTP_{i_1}(S)$ such that $p_{i_2,j} + x_{i_1} = 0$” where $x_{i_1}$ is a missing literal of $p_{i_1}$ and $i_1 \neq i_2$),

2. (C1) of double-fault expression (6.18) (that is, “any point in $UTP_{i_1}(S)$ such that $p_{i_2,j} + x_{i_1} = 0$” where $x_{i_1}$ is a missing literal of $p_{i_1}$ and $i_1 \neq i_2$), and

3. (C3) of double-fault expression (6.18) (that is, “any point in $UTP_{i_2}(S)$ such that $p_{i_1,j} + x_{i_2} = 0$” where $x_{i_2}$ is a missing literal of $p_{i_2}$ and $i_1 \neq i_2$).

The SMUTP strategy requires to select test cases from $UTP_{i_1}(S)$, for every term $p_{i_1}$ of $S$, to form a set $T$ such that, for every term $p_{i_2}$ ($i_1 \neq i_2$), every literal $x_{j_2}^{i_2}$ in $p_{i_2}$ and every missing literal $x_{j_1}^{i_1}$ of $p_{i_1}$,

1. there is a test case $\vec{t}_1 \in T$ such that $p_{i_2,j_2} = 0$ and $x_{i_1} = 0$, if possible; and

2. there is a test case $\vec{t}_2 \in T$ such that $p_{i_2,j_2} = 0$ and $x_{i_1} = 1$, if possible.

In other words, points in $T$ are from $UTP_{i_1}(S)$ for every $p_{i_1}$ such that (1) $p_{i_2,j_2} = 0$ for every term $p_{i_2}$ different from $p_{i_1}$ and every literal $x_{j_2}^{i_2}$ in $p_{i_2}$, and (2) they cover all possible truth values of $x_{i_1}$ for every missing literal $x_{i_1}$ of $p_{i_1}$. Example 11.3.3 illustrates how to select test cases satisfying the SMUTP strategy.
Example 11.3.3 Let $S = abc + def$. Table 11.6 lists the sets $UTP_i(S)$ of unique true points of $S$. Now, let $T = \{111010, 111111, 111100, 111101, 010111, 011111, 100111, 101111\}$. Test cases in $T$ are underlined in Table 11.6 for ease of reference. Test cases in $T$ satisfy the SMUTP strategy because of the following reasons

(1) For the first term $abc$ of $S$, we need to select test cases from $UTP_1(S)$. In $UTP_1(S)$, the test cases $111010$ and $111101$ are such that $p_{2,1} = ef = 0$ and they cover all possible truth values of every missing literal ($d$, $e$ and $f$) of $p_1 = abc$.

Similarly, the test cases $111011$ and $111100$ in $UTP_1(S)$ are such that $p_{2,2} = df = 0$ and they cover all possible truth values of every missing literal ($d$, $e$ and $f$) of $abc$.

Finally, the test cases $111010$ and $111101$, previously selected for $p_{2,1} = ef = 0$, are such that $p_{2,3} = de = 0$ and they cover all possible truth values of every missing literal ($d$, $e$ and $f$) of $abc$. Incidentally, the test cases $111011$ and $111100$, previously selected for $p_{2,2} = df = 0$, also cause $p_{2,3} = de = 0$ and cover all possible truth values of every missing literal ($d$, $e$ and $f$) of $abc$. Hence, these four test cases satisfy the requirements of the SMUTP strategy on $UTP_1(S)$.

(2) Similarly, test cases $010111$, $011111$, $100111$ and $101111$ satisfy the requirements of the SMUTP strategy on $UTP_2(S)$.

To guarantee the detection of double-fault expressions (6.16) and (6.18), we need an extra strategy which is discussed in the next section.

11.3.4 The Pairwise Multiple Unique True Point (PMUTP) Strategy

We propose the Pairwise Multiple Unique True Point (PMUTP) strategy to select test cases that satisfy the following detection conditions

1. (C2) of double-fault expression (6.16) (that is, “any point in $UTP_{i_1}(S)$ such that $x_{i_1} + x_{i_2} = 0$” where $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively),

2. (C2) of double-fault expression (6.18) (that is “any point in $UTP_{i_1}(S)$ such that $x_{i_1} + x_{i_2} = 0$” where $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively), and

3. (C4) of double-fault expression (6.18) (that is, “any point in $UTP_{i_2}(S)$ such that $x_{i_1} + x_{i_2} = 0$” where $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively).
Table 11.7: All unique true points of $S$ where $S = ab + cd + ef$

| $UTP_1(S)$ | 110000, 110001, 110010, 110100, 110101, 110110, 111000, 111001, 111010 |
| $UTP_2(S)$ | 001100, 001101, 001110, 011100, 011101, 011110, 101100, 101101, 101110 |
| $UTP_3(S)$ | 000011, 000111, 001011, 010011, 010111, 011011, 100011, 100111, 101011 |

The PMUTP strategy requires to select test cases from every $UTP_{i_1}(S)$ for every $p_{i_1}$ of $S$ to form a set $T$ such that, for every term $p_{i_2}$ different from $p_{i_1}$ (that is, $i_2 \neq i_1$), for every missing literal $x_{i_1}$ of $p_{i_1}$, and every missing literal $x_{i_2}$ of $p_{i_2}$,

1. there is a test case $\bar{t}_1 \in T$ such that $x_{i_1} = 0$ and $x_{i_2} = 0$, if possible;
2. there is a test case $\bar{t}_2 \in T$ such that $x_{i_1} = 0$ and $x_{i_2} = 1$, if possible;
3. there is a test case $\bar{t}_3 \in T$ such that $x_{i_1} = 1$ and $x_{i_2} = 0$, if possible; and
4. there is a test case $\bar{t}_4 \in T$ such that $x_{i_1} = 1$ and $x_{i_2} = 1$, if possible.

In other words, points in $T$ are from $UTP_{i_1}(S)$ for every $p_{i_1}$ such that, for every term $p_{i_2}$ different from $p_{i_1}$, they can cover all possible truth value combinations of $x_{i_1}$ and $x_{i_2}$ (that is 00, 01, 10 and 11) for every missing literal $x_{i_1}$ of $p_{i_1}$ and every missing literal $x_{i_2}$ of $p_{i_2}$. Example 11.3.4 illustrates how to select test cases satisfying the PMUTP strategy.

**Example 11.3.4** Let $S = ab + cd + ef$. Table 11.7 lists the sets $UTP_i(S)$ of all unique true points of $S$. Now, let $T = \{110000, 110101, 111010, 111001, 111010, 001100, 011101, 011110, 101101, 101110, 000011, 010111, 011011, 100111, 101011\}$. Test cases in $T$ are underlined in Table 11.7 for ease of reference. Test cases in $T$ satisfy the PMUTP strategy because of the following reasons

1. For the first term $ab$, the test cases 110000, 110101, 111010, 111001 and 111010 cover all possible truth value combinations of every possible pair of missing literals $x_{i_1}$ of $ab$ ($c$, $d$, $e$ and $f$) and $x_{i_2}$ of $cd$ ($a$, $b$, $e$ and $f$). It should also be noted that it is impossible to select points from $UTP_i(S)$ such that both $e$ and $f$ evaluate to 1 because such points are true points of $p_3 = ef$ and, hence, are not unique true points of $p_1$.

Furthermore, the previously selected 5 test cases also cover all possible truth value combinations of every possible pair of missing literals $x_{i_1}$ of $ab$ and $x_{i_2}$ of $ef$.

As a result, these 5 test cases satisfy the requirements of PMUTP strategy on $UTP_1(S)$. 

278
Similarly, the test cases 001100, 011101, 011110, 101101 and 101110 from $UTP_2(S)$ cover all possible truth value combinations of every possible pair of missing literals $x_{l_1}$ of $cd$ and $x_{l_2}$ of $ab$.

Furthermore, they cover all possible truth value combinations of every possible pair of missing literals $x_{l_1}$ of $cd$ and $x_{l_2}$ of $ef$.

Hence, these 5 test cases satisfy the requirements of PMUTP strategy on $UTP_2(S)$.

Finally, the test cases 000011, 010111, 011011, 100111 and 101011 satisfy the requirements of PMUTP strategy on $UTP_3(S)$.

We now prove that the SMUTP, PMUTP, SMOTP and MUMCUT strategies together can select test cases to detect double-fault expressions (6.16) and (6.18) in Theorems 11.3.3 and 11.3.4, respectively.

**Theorem 11.3.3** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{l_1}$ is inserted into the $i_1$-th term, $p_{i_1}$, in $S$ and the $j_2$-th literal, $x_{l_2}^{i_2}$, of the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by $x_{l_2}$, where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, $k_{i_2}$ is the number of literals in $p_{i_2}$, and $x_{l_1}$ and $x_{l_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the resulting implementation denoted by $I$ will be equivalent to that given by double-fault expression (6.16) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C5) of double-fault expression (6.16) in Table 7.2. The SMUTP, PMUTP and SMOTP strategies can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.16) provided that $S \not\equiv I$.

**Proof** : Since $S$ and $I$ are not equivalent, by Theorem 7.3.3, there is a point $\bar{t}_1$ that satisfies any of the following conditions

(C1) $\bar{t}_1 \in UTP_{i_1}(S)$ such that $p_{i_2,j_2} + x_{l_1} = 0$,
(C2) $\bar{t}_1 \in UTP_{i_1}(S)$ such that $x_{l_1} + x_{l_2} = 0$,
(C3) $\bar{t}_1 \in UTP_{i_2}(S)$ such that $x_{l_2} = 0$,
(C4) $\bar{t}_1 \in NFP_{i_2,j_2}(S)$ such that $x_{l_2} = 1$, or
(C5) $\bar{t}_1 \in \left( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1,j_2} TP_i(S) \right)$ such that $x_{l_1} + x_{l_2} = 0$.

We then have the following cases:

Case 1 $\bar{t}_1$ satisfies (C1). Then, $\{ \bar{t} \in UTP_{i_1}(S) : p_{i_2,j_2}(\bar{t}) + x_{l_1}(\bar{t}) = 0 \} = \{ \bar{t} \in UTP_{i_1}(S) : p_{i_2,j_2}(\bar{t}) = 0 \text{ and } x_{l_1}(\bar{t}) = 0 \} \neq \emptyset$. The SMUTP strategy can select a point that satisfies (C1).
Case 2 \( \vec{t}_1 \) satisfies (C2). Then, \( \{ \vec{t} \in UTP_{i_1}(S) : x_{i_1}(\vec{t}) + x_{i_2}(\vec{t}) = 0 \} = \{ \vec{t} \in UTP_{i_1}(S) : x_{i_1}(\vec{t}) = 0 \text{ and } x_{i_2}(\vec{t}) = 0 \} \neq \emptyset. \) Since \( x_{i_1} \) and \( x_{i_2} \) are missing literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, the PMUTP strategy can select a point that satisfies (C2).

Case 3 \( \vec{t}_1 \) satisfies (C3). Then, \( \{ \vec{t} \in UTP_{i_2}(S) : x_{i_2}(\vec{t}) = 0 \} \neq \emptyset. \) Since \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the MUTP strategy can select a point that satisfies (C3).

Case 4 \( \vec{t}_1 \) satisfies (C4). Then, \( \{ \vec{t} \in NFP_{i_2,j_1}(S) : x_{i_2}(\vec{t}) = 1 \} \neq \emptyset. \) Since \( x_{i_2} \) is a missing literal of \( p_{i_2} \), the MNFP strategy can select a point that satisfies (C4).

Case 5 \( \vec{t}_1 \) satisfies (C5). Then, \( \{ \vec{t} \in \left( \left( TP_{i_1}(S) \cap TP_{i_2}(S) \right) \setminus \bigcup_{i \neq i_1,i_2}^m TP_i(S) \right) \) : \( x_{i_1}(\vec{t}) + x_{i_2}(\vec{t}) = 0 \} = \{ \vec{t} \in \left( \left( TP_{i_1}(S) \cap TP_{i_2}(S) \right) \setminus \bigcup_{i \neq i_1,i_2}^m TP_i(S) \right) : \( x_{i_1}(\vec{t}) = 0 \text{ and } x_{i_2}(\vec{t}) = 0 \} \neq \emptyset. \) Since \( x_{i_1} \) and \( x_{i_2} \) are missing literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, the SMOTP strategy can select a point that satisfies (C5).

Hence, the result follows.

\[ \square \]

**Theorem 11.3.4** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF. Suppose that the \( j_1 \)-th literal, \( x_{i_1}^{j_1} \), of the \( i_1 \)-th term, \( p_{i_1} \), in \( S \) is replaced by \( x_{i_1} \) and the \( j_2 \)-th literal, \( x_{i_2}^{j_2} \), of the \( i_2 \)-th term, \( p_{i_2} \), in \( S \) is replaced by \( x_{i_2} \) where \( 1 \leq i_1 < i_2 \leq m, 1 \leq j_1 \leq k_{i_1}, 1 \leq j_2 \leq k_{i_2}, k_{i_1} \) and \( k_{i_2} \) are the numbers of literals in \( p_{i_1} \) and \( p_{i_2} \), respectively, and \( x_{i_1} \) and \( x_{i_2} \) are missing literals of \( p_{i_1} \) and \( p_{i_2} \), respectively, the resulting implementation denoted by \( I \) will be equivalent to that given by double-fault expression (6.18) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C7) of double-fault expression (6.18) in Table 7.2. The SMUTP, PMUTP and SMOTP strategies can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.18) provided that \( S \neq I \).

**Proof:** Since \( S \) and \( I \) are not equivalent, by Theorem 7.4.1, there is a point \( \vec{t}_1 \) that satisfies any of the following conditions

(C1) \( \vec{t}_1 \in UTP_{i_1}(S) \) such that \( p_{i_2,j_2} + x_{i_1} = 0, \)

(C2) \( \vec{t}_1 \in UTP_{i_1}(S) \) such that \( x_{i_1} + x_{i_2} = 0, \)

(C3) \( \vec{t}_1 \in UTP_{i_2}(S) \) such that \( p_{i_1,j_1} + x_{i_2} = 0, \)

280
We then have the following cases:

Case 1 $\bar{t}_1$ satisfies (C1). Then, $\{\bar{t} \in UTP_{i_1}(S) : p_{i_2,j_2}(\bar{t}) + x_{i_1}(\bar{t}) = 0\} \neq \emptyset$. The SMUTP strategy can select a point that satisfies (C1).

Case 2 $\bar{t}_1$ satisfies (C2). Then, $\{\bar{t} \in UTP_{i_1}(S) : x_{i_1}(\bar{t}) + x_{i_2}(\bar{t}) = 0\} \neq \emptyset$. Since $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the PMUTP strategy can select a point that satisfies (C2).

Case 3 $\bar{t}_1$ satisfies (C3). Then, $\{\bar{t} \in UTP_{i_2}(S) : p_{i_1,j_1}(\bar{t}) + x_{i_2}(\bar{t}) = 0\} \neq \emptyset$. The SMUTP strategy can select a point that satisfies (C3).

Case 4 $\bar{t}_1$ satisfies (C4). Then, $\{\bar{t} \in UTP_{i_2}(S) : x_{i_1}(\bar{t}) + x_{i_2}(\bar{t}) = 0\} \neq \emptyset$. Since $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the PMUTP strategy can select a point that satisfies (C4).

Case 5 $\bar{t}_1$ satisfies (C5). Then, $\{\bar{t} \in NFP_{i_1,j_1}(S) : x_{i_1}(\bar{t}) = 1\} \neq \emptyset$. Since $x_{i_1}$ is a missing literal of $p_{i_1}$, the MNFP strategy can select a point that satisfies (C5).

Case 6 $\bar{t}_1$ satisfies (C6). Then, $\{\bar{t} \in NFP_{i_2,j_2}(S) : x_{i_2}(\bar{t}) = 1\} \neq \emptyset$. Since $x_{i_2}$ is a missing literal of $p_{i_2}$, the MNFP strategy can select a point that satisfies (C6).

Case 7 $\bar{t}_1$ satisfies (C7). Then, $\{\bar{t} \in \big((TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1,i_2} TP_i(S)\big) : x_{i_1}(\bar{t}) + x_{i_2}(\bar{t}) = 0\} \neq \emptyset$. Since $x_{i_1}$ and $x_{i_2}$ are missing literals of $p_{i_1}$ and $p_{i_2}$, respectively, the SMOTP strategy can select a point that satisfies (C7).
Hence, the result follows.

\[ \square \]

### 11.3.5 The Pairwise Multiple Near False Point (PMNFP) Strategy

We propose the *Pairwise Multiple Near False Point (PMNFP)* strategy to detect double-fault expression (6.17) in Table 6.7. It aims at selecting test cases that satisfy the following detection conditions:

1. (C2) of double-fault expression (6.17) (that is, “any point in $NFP_{i,j_1}(S)$ such that $x_{l_1}x_{l_2}=1$” where $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$),

2. (C2) of double-fault expression (6.19) (that is, “any point in $NFP_{i,j_1}(S)$ such that $x_{l_1}x_{l_2}=1$” where $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$), and

3. (C3) of double-fault expression (6.19) (that is, “any point in $NFP_{i,j_2}(S)$ such that $x_{l_1}x_{l_2}=1$” where $x_{l_1}$ and $x_{l_2}$ are two different missing literals of $p_{i_1}$).

The PMNFP strategy requires to select test cases from every possible $NFP_{i,j}(S)$ of $S$ to form a set $T$ such that, for every pair of missing literals $x_{l_1}$ and $x_{l_2}$ of $p_{i_1}$ where $x_{l_1} \neq x_{l_2}$ and $x_{l_1} \neq \bar{x}_{l_2}$,

1. there is a test case $\vec{t}_1 \in T$ such that $x_{l_1} = 0$ and $x_{l_2} = 0$, if possible;

2. there is a test case $\vec{t}_2 \in T$ such that $x_{l_1} = 0$ and $x_{l_2} = 1$, if possible;

3. there is a test case $\vec{t}_3 \in T$ such that $x_{l_1} = 1$ and $x_{l_2} = 0$, if possible; and

4. there is a test case $\vec{t}_4 \in T$ such that $x_{l_1} = 1$ and $x_{l_2} = 1$, if possible.

In other words, points in $T$ are from $NFP_{i,j}(S)$ for every $p_{i_1}$ of $S$ such that, they can cover all possible truth value combinations of $x_{l_1}$ and $x_{l_2}$ (that is 00, 01, 10 and 11) for every pair of two different missing literals $x_{l_1}$ and $x_{l_2}$ of $p_{i_1}$. Example 11.3.5 illustrates how to select test cases satisfying the PMNFP strategy.

**Example 11.3.5** Let $S = ab+cd+ef$. Table 11.8 lists the sets $NFP_{i,j}(S)$ of all near false points of $S$. Now, let $T = \{010000, 010101, 010110, 011001, 011010, 000100, 010101, 010110, 100110, 000001, 010101, 011001, 010101, 100101, 010010, 100000, 100101, 100110, 101010, 001000, 011001, 011010, 101010, 000010, 010110, 011010, 010101, 101010\}$. Test cases in $T$ are underlined in Table 11.8 for ease of references. Test cases in $T$ satisfy the PMNFP strategy because of the following reasons...
Table 11.8: All near false points of $S$ where $S = ab + cd + ef$

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NFP_{1,1}(S)$</td>
<td>010000, 010001, 000100, 010100, 010001, 011001, 000110, 100100, 100001, 011010</td>
<td>000100, 000101, 010100, 010110, 010101, 010110, 010010, 010001, 011001, 011010</td>
<td>000001, 000101, 000110, 100110, 100001, 100101, 100110, 100010, 101001, 101010</td>
</tr>
<tr>
<td>$NFP_{1,2}(S)$</td>
<td>100000, 100001, 100100, 000100, 010101, 001010, 011000, 011001, 100010, 100110</td>
<td>100100, 100101, 100110, 011010, 010010, 010011, 011011, 011010, 100001, 100110</td>
<td>100101, 101001, 101010, 101010, 101010, 101010</td>
</tr>
</tbody>
</table>

(1) For $NFP_{1,1}(S)$, the test cases 010000, 010101, 011010, 011011 and 011010 cover all possible truth value combinations of every possible pair of missing literals ($c, d, e$ and $f$) of $p_1 = ab$. It should be noted that it is impossible to select points from $NFP_{1,1}(S)$ such that both $c$ and $d$ evaluate to 1 because such points are true points of $p_2 = cd$ and, hence, are true points of $S$. Hence, these five test cases satisfy the requirements of the PMNFP strategy on $NFP_{1,1}(S)$.

(2) Similarly, test cases underlined in each $NFP_{i,j}(S)$ in Table 11.8 cover all possible truth value combinations of every possible pair of missing literals of every term $p_i$.

We now prove that the PMNFP and MUMCUT strategies can select test cases to detect double-fault expression (6.17) in Theorem 11.3.5.

**Theorem 11.3.5** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the literal $x_{i_1}$ is inserted into the $i_1$-th term, $p_{i_1}$, in $S$ and the $j_2$-th literal, $x_{j_2}$, of $p_{i_1}$, in $S$ is replaced by $x_{i_2}$ where $1 \leq i_1 \leq m$, $1 \leq j_2 \leq k_{i_1}$, $k_{i_1}$ is the number of literals in $p_{i_1}$, and $x_{i_1}$ and $x_{i_2}$ are two different missing literals of $p_{i_1}$, the resulting implementation denoted by $I$ will be equivalent to that given by double-fault expression (6.17) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C3) of double-fault expression (6.17) in Table 7.2. The PMNFP strategy can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.17) provided that $S \not\equiv I$.

**Proof**: Since $S$ and $I$ are not equivalent, by Theorem 7.3.4, there is a point $\vec{t}_1$ that satisfies any of the following conditions

(C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{i_1} = 0$,

(C2) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{i_2} = 0$, or

(C3) $\vec{t}_1 \in NFP_{i_1,j_2}(S)$ such that $x_{i_1}x_{i_2} = 1$.

283
We then have the following cases:

Case 1 \( \vec{t}_1 \) satisfies (C1). Then, \( \{ \vec{t} \in UTP_{i_1}(S) : x_{l_1}(\vec{t}) = 0 \} \neq \emptyset \). Since \( x_{l_1} \) is a missing literal of \( p_{i_1} \), the MUTP strategy can select a point that satisfies (C1).

Case 2 \( \vec{t}_1 \) satisfies (C2). Then, \( \{ \vec{t} \in UTP_{i_1}(S) : x_{l_2}(\vec{t}) = 0 \} \neq \emptyset \). Since \( x_{l_2} \) is a missing literal of \( p_{i_1} \), the MUTP strategy can select a point that satisfies (C2).

Case 3 \( \vec{t}_1 \) satisfies (C3). Then, \( \{ \vec{t} \in NFP_{i_1,j_2}(S) : x_{l_1}(\vec{t}) = 0 \} \neq \emptyset \). Since \( x_{l_1} \) and \( x_{l_2} \) are two different missing literals of \( p_{i_1} \), the PMNFP strategy can select a point that satisfies (C3).

Hence, the result follows.

\[ \square \]

11.3.6 The Supplementary Pairwise Multiple False Point (SPMFP) Strategy

We propose the Supplementary Pairwise Multiple False Point (SPMFP) strategy to select test cases satisfying the detection condition (C4) of double-fault expression (6.19) (that is, “any point in \( FP(S) \) such that \( p_{i_1,j_1,j_2} = 1 \) and \( x_{l_1}x_{l_2} = 1 \)” where \( x_{l_1} \) and \( x_{l_2} \) are two different missing literals of \( p_{i_1} \)).

The SPMFP strategy requires to select test cases from \( FP(S) \) to form a set \( T \) such that, for every term \( p_i \), every pair of literals \( x_{j_1}^i \) and \( x_{j_2}^i \) in \( p_i \), and every pair of missing literals \( x_{l_1} \) and \( x_{l_2} \) of \( p_i \),

1. there is a test case \( \vec{t}_1 \in T \) such that, \( p_{i,j_1,j_2} = 1, x_{l_1} = 0 \) and \( x_{l_2} = 0 \), if possible;
2. there is a test case \( \vec{t}_2 \in T \) such that, \( p_{i,j_1,j_2} = 1, x_{l_1} = 0 \) and \( x_{l_2} = 1 \), if possible;
3. there is a test case \( \vec{t}_3 \in T \) such that, \( p_{i,j_1,j_2} = 1, x_{l_1} = 1 \) and \( x_{l_2} = 0 \), if possible; and
4. there is a test case \( \vec{t}_4 \in T \) such that, \( p_{i,j_1,j_2} = 1, x_{l_1} = 1 \) and \( x_{l_2} = 1 \), if possible.

In other words, points in \( T \) are from \( FP(S) \) such that, for every term \( p_i \), (1) \( p_{i,j_1,j_2} = 1 \) for every pair of literals \( x_{j_1}^i \) and \( x_{j_2}^i \) in \( p_i \) and (2) they cover all possible truth value combinations (that is \( 00, 01, 10 \) and \( 11 \)) of every pair of two different missing literals \( x_{l_1} \) and \( x_{l_2} \) of \( p_i \). Example 11.3.6 illustrates how to select test cases satisfying the SPMFP strategy.
Table 11.9: All false points of $S$ where $S = ab + cd + ef$

000000, 000010, 000100, 001001, 001010, 010000, 010010, 010100, 010110, 011000, 011010, 100000, 100010, 100100, 100110, 101000, 101010, 101100, 101110, 110000, 110010, 110100, 110110, 111000, 111010, 101000, 110100, 101001, 101010, 101101, 101111

Example 11.3.6 Let $S = ab + cd + ef$. Table 11.9 lists the set $FP(S)$ of all false points of $S$. Now, let $T = \{000000, 000101, 001001, 001010, 010001, 010010, 010100, 010110, 011000, 011010, 100001, 100010, 100100, 100101, 001100, 011001, 101000\}$. Test cases in $T$ are underlined in Table 11.9 for ease of reference. Test cases in $T$ satisfy the SPMFP strategy because of the following reasons

1. For the first term $ab$, the test cases 000000, 000101, 001001 and 001010 in $FP(S)$ are such that $\bar{a} \bar{b} = 1$ and they cover all possible truth value combinations of every possible pair of missing literals ($c$, $d$, $e$ and $f$) of $ab$. It should be noted that it is impossible to select points from $FP(S)$ such that both $c$ and $d$ evaluate to 1 because such points are true points of $p_2 = cd$ and, hence, are true points of $S$. Similarly, it is impossible to select points from $FP(S)$ such that both $e$ and $f$ evaluate to 1 because such points are true points of $p_3 = ef$. Hence, these 5 test cases satisfy the requirements of the SPMFP strategy for $ab$ of $S$.

2. Similarly, the test cases 000000, 010001, 010010, 100001 and 100010 in $FP(S)$ are such that $\bar{c} \bar{d} = 1$ and they cover all possible truth value combinations of every possible pair of missing literals ($a$, $b$, $e$ and $f$) of $cd$. It should be noted that it is impossible to select points from $FP(S)$ such that both $a$ and $b$ evaluate to 1 because such points are true points of $p_1 = ab$ and, hence, are true points of $S$. Similarly, it is impossible to select points from $FP(S)$ such that both $e$ and $f$ evaluate to 1 because such points are true points of $p_3 = ef$. Hence, these 5 test cases satisfy the requirements of the SPMFP strategy for $cd$ of $S$.

3. Finally, the test cases 000000, 010100, 011000, 100100 and 101000 in $FP(S)$ are such that $\bar{e} f = 1$ and they cover all possible truth value combinations of every possible pair of missing literals ($a$, $b$, $c$, $d$) of $ef$. It should be noted that it is impossible to select points from $FP(S)$ such that both $a$ and $b$ evaluate to 1 because such points are true points of $p_1 = ab$ and, hence, are true points of $S$. Similarly, it is impossible to select points from $FP(S)$ such that both $c$ and $d$ evaluate to 1 because such points are true points of $p_2 = cd$. Hence, these 5 test cases satisfy the requirements of the SPMFP strategy for $ef$ of $S$.

We now prove that the SPMFP, PMNFP and MUMCUT strategies select test cases to detect double-fault expression (6.19) in Theorem 11.3.6.
Theorem 11.3.6 Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $j_1$-th and $j_2$-th literals, $x_{j_1}^{i_1}$ and $x_{j_2}^{i_2}$, of the $i_1$-th term, $p_{i_1}$, in $S$ are replaced by $x_{t_1}$ and $x_{t_2}$, respectively, where $1 \leq i_1 \leq m$, $1 \leq j_1 < j_2 \leq k_{i_1}$, $k_{i_1}$ is the number of literals in $p_{i_1}$, and $x_{t_1}$ and $x_{t_2}$ are two different missing literals of $p_{i_1}$, the resulting implementation denoted by $I$ will be equivalent to that given by double-fault expression (6.19) in Table 6.7 and its detection conditions are given by the corresponding conditions (C1) to (C5) of double-fault expression (6.19) in Table 7.2. The SPMFP and PMNFP strategies can supplement the MUMCUT strategy to guarantee the detection of double-fault expression (6.19) provided that $S \neq I$.

Proof: Since $S$ and $I$ are not equivalent, by Theorem 7.4.2, there is a point $\vec{t}_1$ that satisfies any of the following conditions

(C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{t_1} = 0$,

(C2) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{t_2} = 0$,

(C3) $\vec{t}_1 \in NFP_{i_1,j_1}(S)$ such that $x_{t_1}x_{t_2} = 1$,

(C4) $\vec{t}_1 \in NFP_{i_1,j_2}(S)$ such that $x_{t_1}x_{t_2} = 1$, or

(C5) $\vec{t}_1 \in FP(S)$ such that $p_{i_1,j_1,j_2} = 1$ and $x_{t_1}x_{t_2} = 1$.

We then have the following cases:

Case 1 $\vec{t}_1$ satisfies (C1). Then, \( \{ \vec{t} \in UTP_{i_1}(S) : x_{t_1}(\vec{t}) = 0 \} \neq \emptyset \). Since $x_{t_1}$ is a missing literal of $p_{i_1}$, the MUTP strategy can select a point that satisfies (C1).

Case 2 $\vec{t}_1$ satisfies (C2). Then, \( \{ \vec{t} \in UTP_{i_1}(S) : x_{t_2}(\vec{t}) = 0 \} \neq \emptyset \). Since $x_{t_2}$ is a missing literal of $p_{i_1}$, the MUTP strategy can select a point that satisfies (C2).

Case 3 $\vec{t}_1$ satisfies (C3). Then, \( \{ \vec{t} \in NFP_{i_1,j_1}(S) : x_{t_1}x_{t_2}(\vec{t}) = 1 \} \neq \emptyset \). The PMNFP strategy can select a point that satisfies (C3).

Case 4 $\vec{t}_1$ satisfies (C4). Then, \( \{ \vec{t} \in NFP_{i_1,j_2}(S) : x_{t_1}x_{t_2}(\vec{t}) = 1 \} \neq \emptyset \). The PMNFP strategy can select a point that satisfies (C4).

Case 5 $\vec{t}_1$ satisfies (C5). Then, \( \{ \vec{t} \in FP(S) : p_{i_1,j_1,j_2}x_{t_1}x_{t_2}(\vec{t}) = 1 \} \neq \emptyset \). The SPMFP strategy can select a point that satisfies (C5).
Table 11.10: Detection conditions aimed by the test case selection strategies

<table>
<thead>
<tr>
<th>Test Case Selection Strategy</th>
<th>Detection Condition as in Table 7.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMNFP</td>
<td>(C3) of (6.17); (C3) and (C4) of (6.19)</td>
</tr>
<tr>
<td>SMFP</td>
<td>(C4) of (6.13)</td>
</tr>
<tr>
<td>SPMFP</td>
<td>(C4) of (6.19)</td>
</tr>
<tr>
<td>PMUTP</td>
<td>(C2) of (6.16); (C2) and (C4) of (6.18)</td>
</tr>
<tr>
<td>SMUTP</td>
<td>(C1) of (6.16); (C1) and (C3) of (6.18)</td>
</tr>
<tr>
<td>SMOTP</td>
<td>(C3) of (6.14); (C5) of (6.16); (C7) of (6.18)</td>
</tr>
</tbody>
</table>

Hence, the result follows.

In summary, Table 11.10 lists all six new strategies and the corresponding detection conditions at which they aim to satisfy. For example, the SMOTP strategy aims at satisfying “any point in \((TP_{i_1}(S) \cap TP_{i_2}(S))\) \(\setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right)\)” such that \(x_{l_1} + x_{l_2} = 0\), which corresponds to (C3), (C5) and (C7) of double-fault expressions (6.14), (6.16) and (6.18), respectively.

11.4 Test Case Selection Strategies For Detecting Double Faults Related to Term and Literal Only

As discussed in previous chapter, any test case selection strategy that subsumes the BASIC strategy can detect 40 out of 42 double-fault expressions in Table 8.12. For the remaining double-fault expressions (8.20) and (8.27), none of the test case selection strategies discussed in Chapter 10 can detect them. After a thorough analysis on the detection conditions of double-fault expressions (8.20) and (8.27), we find that only one out of the six test case selection strategies, the SMOTP strategy is needed to supplement the MUMCUT strategy in detecting these two faulty expressions.

The SMOTP strategy was originally developed to supplement the MUMCUT strategy to guarantee the detection of the double literal fault \(LIF \times \LIF\). It aims to select test cases that satisfy the detection condition “\(\left( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2} TP_i(S) \right) \right)\)” where \(x_{l_1}\) and \(x_{l_2}\) are missing literals for terms \(p_{i_1}\) and \(p_{i_2}\) of \(S\), respectively.
In the rest of this section, we will show that the MUMCUT and SMOTP strategies together guarantees to select test cases that satisfy the detection conditions (C1)–(C5) of Expression (8.20) and (C1)–(C8) of Expression (8.27). Furthermore, since the MUMCUT strategy subsumes the BASIC strategy, the MUMCUT and SMOTP strategies together also guarantee to detect all 42 double-fault expressions related to term and literal. The main results are documented as Theorems 11.4.1 and 11.4.2. However, we need the following 5 lemmas before discussing these two main theorems.

**Lemma 11.4.1** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF and \( p_{i_1} \) be a minterm of \( S \). Then, for any term \( p_{i_2} \) \( (i_2 = 1, \ldots, m \) and \( i_2 \neq i_1) \), there are at least 2 literals in \( p_{i_2} \) whose negations can be found in \( p_{i_1} \).

**Proof:** Since \( S \) is an IDNF, by Lemma 2.1.1, \( \text{UTP}_{i_1}(S) \neq \emptyset \) and \( \text{NFP}_{i_1, j}(S) \neq \emptyset \) for \( j = 1, \ldots, k_{i_1} \) where \( k_{i_1} \) denotes the number of literals in \( p_{i_1} \). Since \( p_{i_1} \) is a minterm, it contains all variables in \( S \). Suppose, on the contrary, that there are at most 1 literal in \( p_{i_2} \) whose negation can be found in \( p_{i_1} \). We have the following two cases:

1. No literal in \( p_{i_2} \) whose negation can be found in \( p_{i_1} \). Thus, all literals in \( p_{i_2} \) can be found in \( p_{i_1} \). Then, for any point \( \vec{t} \in \mathbb{B}^n \) such that \( p_{i_1}(\vec{t}) = 1, p_{i_2}(\vec{t}) = 1 \). As a result, \( \text{UTP}_{i_1}(S) = \emptyset \). This contradicts to the assumption that \( S \) is IDNF.

2. There is exactly one literal in \( p_{i_2} \) whose negation can be found in \( p_{i_1} \). Without loss of generality, we may assume that this literal is the negation of the first literal \( x_{i_1}^1 \) in the \( p_{i_1} \). Otherwise, we can always rearrange the literal so that it can become the first literal of \( p_{i_1} \). Since all other literals except \( x_{i_1}^1 \) in \( p_{i_2} \) can be found in \( p_{i_1} \), for any \( \vec{t} \in \mathbb{B}^n \) such that \( p_{i_1}(\vec{t}) = 1, p_{i_2}(\vec{t}) = 1 \). As a result, \( \text{NFP}_{i_1, \vec{t}}(S) = \emptyset \). This contradicts to assumption that \( S \) is IDNF.

Hence, the result follows. \( \square \)

Lemma 11.4.1 implies that whenever a Boolean expression \( S \) is in IDNF and one of its term is a minterm, every remaining term must have at least two literals and their negations can be found in the minterm.

**Lemma 11.4.2** Let \( S = p_1 + \cdots + p_m \) be a Boolean specification in IDNF and \( p_i \) be a minterm of \( S \). Then,

1. \( \text{UTP}_i(S) = \{\vec{u}\} \)
2. \( \text{NFP}_{i, j}(S) = \{\vec{n}_j\} \) for all \( j = 1, \ldots, k_i \), where \( k_i \) denotes the number of literals in \( p_i \)
3. \( \vec{u} \) and \( \vec{n}_j \) only differ on the literal \( x_j^i \) for all \( j = 1, \ldots, k_i \), where \( k_i \) denotes the number of literals in \( p_i \).
**Proof:** Note that, $p_i$ contains all variables in $S$ because it is a minterm.

1. Therefore, $TP_i(S)$ is a singleton. Since $S$ is an IDNF, by Lemma 2.1.1, $UTP_i(S) \neq \emptyset$. Therefore, $UTP_i(S) = TP_i(S)$ because $UTP_i(S) \subseteq TP_i(S)$.

2. For each $j = 1, \ldots, k_i$, let $X_j = \{ \bar{t} \in B^n : p_i \bar{t}(\bar{t}) = 1 \}$ where $p_i \bar{t}$ denotes the term obtained by negating the $j$-th literal of $p_i$. By definition, elements in $NFP_{i,j}(S)$ are those in $X_j$ such that $S$ evaluates to 0. Hence, $NFP_{i,j}(S) \subseteq X_j$. Since $p_i$ contains all variables in $S$, so does $p_{i,j}$. Therefore, $X_j$ is singleton.

3. By (1) and (2), $UTP_i(S) = \{ \bar{u} \}$ and $NFP_{i,j}(S) = \{ \bar{n}_j \}$ for all $j = 1, \ldots, k_i$. By definitions of $\bar{u}$ and $\bar{n}_j$, $p_i(\bar{u}) = 1$ and $p_{i,j}(\bar{n}_j) = 1$. Therefore, $\bar{u}$ and $\bar{n}_j$ differ at $x_j^i$ for all $j = 1, \ldots, k_i$.

Hence, the result follows. \qed

The implication of Lemma 11.4.2 is that, if $p_i$ is a minterm, for every possible $UTP_i(S)$ and $NFP_{i,j}(S)$ pair, the only point from $UTP_i(S)$ and the only point from $NFP_{i,j}(S)$ differ only at the corresponding truth value of the literal $x_j^i$. Hence, the CUTPNFP strategy can always select these points.

**Lemma 11.4.3** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF and $p_{i_1}$ be a minterm of $S$. Then, $UTP_{i}(S) = \{ \bar{u} \}$ and $p_{i_2 j}(\bar{u}) = 0$ for all $1 \leq i_2 \neq i_1 \leq m$ and for all $1 \leq j \leq k_{i_2}$ where $k_{i_2}$ denotes the number of literals in $p_{i_2}$ and $p_{i_2 j}$ denotes the term obtained from $p_{i_2}$ by omitting its $j$-th literal $x_j^{i_2}$.

**Proof:** Since $p_{i_1}$ is a minterm, by Lemma 11.4.2, $UTP_{i_1}(S) = \{ \bar{u} \}$.

Let $p_{i_2}$ be any term in $S$ different from $p_{i_1}$ ($i_1 \neq i_2$). By Lemma 11.4.1, there are at least two literals in $p_{i_2}$ whose negations can be found in $p_{i_1}$. Without loss of generality, we may assume that these two literals are $x_1^{i_2}$ and $x_2^{i_2}$. Otherwise, we can always rearrange the literals so that they become the first two literals of $p_{i_2}$. Note that, both $x_1^{i_2}$ and $x_2^{i_2}$ evaluate to 0 on $\bar{u}$ because their negations are in $p_{i_1}$. Since $x_1^{i_2}$ or $x_2^{i_2}$ must exist in every $p_{i_2 j}$, $p_{i_2 j}(\bar{u}) = 0$ where $1 \leq j \leq k_{i_2}$. Hence, the result follows. \qed

**Lemma 11.4.4** Let $S = p_1 + \cdots + p_m$ be a Boolean expression in IDNF, and $x$ be a literal of $S$. If the $i$-th term of $S$, $p_i$, is not a minterm and there is at least one point in $UTP_i(S)$ such that $x = 0$, then the MUTP strategy will select a point from $UTP_i(S)$ such that $x = 0$.

**Proof:** Let $p_i$ be the $i$-th term in $S$. Since $S$ is irredundant, $UTP_i(S) \neq \emptyset$ by Lemma 2.1.1. Since $p_i$ is not a minterm, it has at least one missing literal. Therefore, the MUTP strategy can always be applied on $p_i$.

Since $x$ is a literal of $S$, we have three cases:
1. The literal $x$ appears in $p_i$. Then, $x = 1$ on all points in $UTP_i(S)$. This
contradicts to the assumption that there is a point in $UTP_i(S)$ such that
$x = 0$. Hence, it is impossible for $x$ to be a literal in $p_i$.

2. The literal $\overline{x}$ appears in $p_i$. Then, $x = 0$ on all points in $UTP_i(S)$. Hence, the
MUTP strategy can select a point from $UTP_i(S)$ such that $x = 0$.

3. Both literals $x$ and $\overline{x}$ do not appear in $p_i$. Then, $x$ is a missing literal of $p_i$.
Based on the given condition, there exist some points in $UTP_i(S)$ such that
$x$, as a missing literal, evaluates to 0. Since the MUTP strategy selects points
in $UTP_i(S)$ such that all possible truth values of every missing literal of $p_i$
are covered. Hence, it can select at least a point from $UTP_i(S)$ such that $x$
evaluates to 0.

Hence, the result follows. \hfill \square

Lemma 11.4.4 needs some explanation on how to apply it for the detection of Ex-
pressions (8.20) and (8.27). Let $X_1$ be the set \{\( \vec{t} \in UTP_i(S) : x \) evaluates to 0 on \( \vec{t} \)\} of
all points \( \vec{t} \) in $UTP_i(S)$ such that the literal $x$ evaluates to 0 on $\vec{t}$. Lemma 11.4.4
shows that if $p_i$ is not a minterm and $X_1$ is non-empty, the MUTP strategy will be
able to select a test case from $UTP_i(S)$ such that it is in $X_1$. As a result, test cases
selected by the MUTP strategy can satisfy (C1) of Expression (8.20) provided that
the corresponding term is not a minterm and there exists some test cases that can
satisfy the corresponding condition. Similarly, the MUTP strategy, if applicable,
can select test cases that satisfy (C1), (C3) and (C5) of Expression (8.27).

**Lemma 11.4.5** Let $S = p_1 + \cdots + p_m$ be a Boolean expression in IDNF, and $p_{i_1}$
and $p_{i_2}$ ($i_1 \neq i_2$) be two different terms of $S$. Suppose that $p_{i_1}$ is not a minterm and
that there is at least one point from $UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$ where
$p_{i_2,j_2} = x_1^{i_2} \cdots x_{j_2}^{i_2} - x_{j_2+1}^{i_2} \cdots x_{k_{i_2}}^{i_2}$ is the term obtained from $p_{i_2}$ by omitting its $j_2$-th literal,
$x_{j_2}^{i_2}$ and $k_{i_2} (> 1)$ is the number of literals in $p_{i_2}$. Then, the MUTP strategy will be
able to select a point from $UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$.

**Proof:** Since $S$ is irredundant, $UTP_{i_1}(S) \neq \emptyset$ by Lemma 2.1.1. Since $p_{i_1}$ is not
a minterm, it has at least one missing literal. Therefore, the MUTP strategy can
always be applied on $p_{i_1}$. Since $p_{i_1}$ and $p_{i_2}$ are two different terms of $S$, we have the
following two cases:

1. All literals of $p_{i_2,j_2}$ are in $p_{i_1}$. Then, $p_{i_2,j_2} = 1$ on all points in $UTP_{i_1}(S)$
because all literals in $p_{i_1}$ evaluate to 1. This contradicts to the assumption.
Hence, such a case is impossible.

2. There exists some literals of $p_{i_2,j_2}$ such that they do not appear in $p_{i_1}$. Two
subcases arise among these literals:
(a) Among these literals, there is at least one literal $x$ such that $\bar{x}$ appears in $p_i$.

Then, $x$ will evaluate to 0 on all points from $UTP_{i_1}(S)$. As a result, $p_{i_2,j_2}$ evaluates to 0 on all points from $UTP_{i_1}(S)$. Hence, the MUTP strategy can always select a point from $UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$

(b) Among these literals, all their negations do not appear in $p_i$. Hence, all these literals are missing literals of $p_i$.

Now, literals in $p_{i_2,j_2}$ can be divided into two groups. The first group contains those literals that are in $p_i$ whereas the second group contains those literals that are not in $p_i$. Based on this subcase, the negations of all literals in the second group are also not in $p_i$.

If all literals in the second group evaluate to 1 on all points of $UTP_{i_1}(S)$, $p_{i_2,j_2}$ evaluates to 1 on all points of $UTP_{i_1}$. This contradicts to the given condition that $p_{i_2,j_2}$ evaluates to 0 on some point of $UTP_{i_1}$.

Hence, there is at least one literal $y$ in the second group such that it evaluates to 0 on some points of $UTP_{i_1}(S)$.

In summary, we have (1) both $y$ and $\bar{y}$ do not appear in $p_i$; and (2) there exist some points in $UTP_{i_1}(S)$ such that $y$, as a missing literal, evaluates to 0. Since the MUTP strategy selects points from $UTP_i(S)$ such that all possible truth values of every missing literal of $p_i$ are covered, for every $p_i$ of $S$. Hence, it can select at least a point from $UTP_{i_1}(S)$ such that $y = 0$, and hence, $p_{i_2,j_2} = 0$.

The result follows. □

Similar to Lemma 11.4.4, this lemma needs some explanation on its application. Let $X_2$ be the set \{ $\bar{t} \in UTP_{i_1}(S)$ : $p_{i_2,j_2}$ evaluates to 0 on $\bar{t}$ where $i_2 \neq i_1$ \} of all points $\bar{t}$ in $UTP_{i_1}(S)$ such that $p_{i_2,j_2}$ evaluates to 0 on $\bar{t}$. Lemma 11.4.5 shows that if $p_i$ is not a minterm and $X_2$ is non-empty, the MUTP strategy will be able to select a test case from $UTP_{i_1}(S)$ such that it is in $X_2$. Hence, test cases selected by the MUTP strategy can satisfy (C2) of Expression (8.20) provided that the corresponding term is not a minterm and there are test cases that satisfy the corresponding condition. Similarly, the MUTP strategy, if applicable, can select test cases that satisfy (C2) and (C4) of Expression (8.27).

We are now ready to prove our main theorems that the MUMCUT and SMOTP strategies can detect Expressions (8.20) and (8.27).

**Theorem 11.4.1** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that the $i_1$-th term, $p_{i_1}$, in $S$ is omitted and the $j_2$-th literal, $x_{j_2}$, of the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by $x_{l_2}$, where $1 \leq i_1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, and $x_{l_2}$ is a missing literal of $p_{i_2}$. Then the resulting implementation denoted by $I$ will be equivalent
to Expression (8.20) in Table 8.12 and its detection conditions are given by (C1) to (C5) of Expression (8.20) in Table 9.2. The SMOTP strategy can supplement the MUMCUT strategy to guarantee the detection of Expression (8.20) provided that $S \not\equiv I$.

**Proof**: Since $S$ and $I$ are not equivalent, by Theorem 9.1.20, there is a point $\vec{t}_1$ that satisfies any of the following conditions:

- (C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{l_2}=0$ on $\vec{t}$,
- (C2) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $p_{i_2,j_2}=0$ on $\vec{t}$,
- (C3) $\vec{t}_1 \in UTP_{i_2}(S)$ such that $x_{l_2}=0$ on $\vec{t}$,
- (C4) $\vec{t}_1 \in NFP_{i_2,j_2}(S)$ such that $x_{l_2}=1$ on $\vec{t}$, or
- (C5) $\vec{t}_1 \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \bigcup_{i \neq i_1,i_2} TP_i(S)$ such that $x_{l_2}=0$ on $\vec{t}$.

We will prove that the SMOTP and MUMCUT strategies can select test cases to collectively satisfy conditions (C1)–(C5) of Expression (8.20). Since $x_{l_2}$ is a missing literal of $p_{i_2}$, $p_{i_2}$ is not a minterm. As a result, the SMOTP strategy can only be applied when $p_{i_1}$ is not a minterm. We have the following two cases:

1. The term $p_{i_1}$ is a minterm.

   By Lemma 11.4.1, $p_{i_2}$ has at least two literals and their negation can be found in $p_{i_1}$. By Lemma 11.4.2, $UTP_{i_1}(S) = \{\vec{u}\}$, $NFP_{i_1,j}(S) = \{\vec{n}_j\}$, and $\vec{u}$ and $\vec{n}_j$ only differ on the literal $x_{i_1}^j$ for all $j = 1, \ldots, k_{i_1}$ where $k_{i_1}$ is the number of literals in the term $p_{i_1}$. Thus, the CUTPNFP strategy will select all these points. By Lemma 11.4.3, $p_{i_2,j_2}$ evaluates to 0 on $\vec{u}$. Hence, (C2) can always be satisfied. The CUTPNFP strategy guarantees to select the required test cases.

2. The term $p_{i_1}$ is not a minterm.

   (a) If $\vec{t}_1$ satisfies (C1), then $\{\vec{t} \in UTP_{i_1}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset$. The MUTP strategy can select a point that satisfies (C1) by Lemma 11.4.4.

   (b) If $\vec{t}_1$ satisfies (C2), then $\{\vec{t} \in UTP_{i_1}(S) : p_{i_2,j_2}(\vec{t}) = 0\} \neq \emptyset$. The MUTP strategy can select a point that satisfies (C2) by Lemma 11.4.5.

   (c) If $\vec{t}_1$ satisfies (C3), then $\{\vec{t} \in UTP_{i_2}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset$. The MUTP strategy can select a point that satisfies (C3).

   (d) If $\vec{t}_1$ satisfies (C4), then $\{\vec{t} \in NFP_{i_2,j_2}(S) : x_{l_2}(\vec{t}) = 1\} \neq \emptyset$. The MNFP strategy can select a point that satisfies (C4).
(e) If $\vec{t}_1$ satisfies (C5), then $$\{\vec{t} \in (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S) \right) : x_{i_2}(\vec{t}) = 0 \} \neq \emptyset.$$ We prove that the SMOTP strategy can select points that satisfy (C5) although the points actually selected may not be $\vec{t}_1$.

Now, since $p_{i_1}$ is not a minterm, it has at least one missing literal. Suppose that $x_{l_1}$ is a missing literal of $p_{i_1}$.

The SMOTP strategy requires to select test cases that can collectively cover every possible truth value combinations of the missing literals $x_{l_1}$ and $x_{l_2}$ of $p_{i_1}$ and $p_{i_2}$, respectively, from $$(TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S) \right)$$ for every two different terms $p_{i_1}$ and $p_{i_2}$. As a result, it is straightforward to see that the SMOTP strategy can select test cases that satisfy (C5).

Hence, the SMOTP, MUTP and MNFP strategies together can always select test cases that satisfy (C1)–(C5) provided that $p_{i_1}$ is not a minterm.

The result follows. \hfill \Box

**Theorem 11.4.2** Let $S = p_1 + \cdots + p_m$ be a Boolean specification in IDNF. Suppose that two terms $p_{i_1}$ and $p_{i_1+1}$, in $S$ are implemented as $p_i p_{i_1+1}$ and the $j_2$-th literal, $x_{j_2}$, of the $i_2$-th term, $p_{i_2}$, in $S$ is replaced by $x_{l_2}$, where $1 < i_1 + 1 < i_2 \leq m$, $1 \leq j_2 \leq k_{i_2}$, and $x_{l_2}$ is a missing literal of $p_{i_2}$. Then the resulting implementation denoted by $I$ will be equivalent to Expression (8.27) in Table 8.12 and its detection conditions are given by (C1) to (C5) of Expression (8.27) in Table 9.2. The SMOTP strategy can supplement the MUMCUT strategy to guarantee the detection of Expression (8.27) provided that $S \not\equiv I$.

**Proof:** Since $S$ and $I$ are not equivalent, by Theorem 9.1.27, there is a point $\vec{t}_1$ that satisfies any of the following conditions:

(C1) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $x_{l_2} = 0$ on $\vec{t}$,

(C2) $\vec{t}_1 \in UTP_{i_1}(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$,

(C3) $\vec{t}_1 \in UTP_{i_1+1}(S)$ such that $x_{l_2} = 0$ on $\vec{t}$,

(C4) $\vec{t}_1 \in UTP_{i_1+1}(S)$ such that $p_{i_2,j_2} = 0$ on $\vec{t}$,

(C5) $\vec{t}_1 \in UTP_{i_2}(S)$ such that $x_{l_2} = 0$ on $\vec{t}$,

(C6) $\vec{t}_1 \in NFP_{i_2,j_2}(S)$ such that $x_{l_2} = 1$ on $\vec{t}$,

(C7) $\vec{t}_1 \in \left( (TP_{i_1}(S) \cap TP_{i_2}(S)) \setminus \left( \bigcup_{i \neq i_1, i_2}^m TP_i(S) \right) \right)$ such that $x_{l_2} = 0$ on $\vec{t}$, or
\((C8)\)  \(\vec{t}_1 \in \left( \left( TP_{i_1+1}(S) \cap TP_{i_2}(S) \right) \setminus \bigcup_{i \neq i_1+1,i_2}^{m} TP_i(S) \right)\) such that \(x_{l_2} = 0\) on \(\vec{t}\).

We will prove that the SMOTP and MUMCUT strategies can select test cases to collectively satisfy the conditions (C1)–(C8) of Expression (8.27) given in Table 9.2. Since the SMOTP strategy can only be applied when \(p_{i_1}\) or \(p_{i_1+1}\) is not a minterm, we have following three cases:

1. \(p_{i_1}\) is a minterm.

   Condition (C2) can always be satisfied and the CUTPNFP strategy guarantees to select the required test cases. The proof is similar to that of Case 1 in Theorem 11.4.1.

2. \(p_{i_1+1}\) is a minterm.

   Condition (C4) can always be satisfied and the CUTPNFP strategy guarantees to select the required test cases. The proof is similar to Case 1 above.

3. Both \(p_{i_1}\) and \(p_{i_1+1}\) are not minterms.

   Since \(S \neq I\), there is at least one test case \(\vec{t}\) such that \(\vec{t}\) satisfies any one of (C1)–(C8).

   (a) If \(\vec{t}_1\) satisfies (C1), then then \(\{\vec{t} \in UTP_{i_1}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset\). The MUTP strategy can select a point that satisfies (C1) by Lemma 11.4.4.

   (b) If \(\vec{t}_1\) satisfies (C2), then \(\{\vec{t} \in UTP_{i_1}(S) : p_{i_2,j_2}(\vec{t}) = 0\} \neq \emptyset\). The MUTP strategy can select a point that satisfies (C2) by Lemma 11.4.5.

   (c) If \(\vec{t}_1\) satisfies (C3), then then \(\{\vec{t} \in UTP_{i_1+1}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset\). The MUTP strategy can select a point that satisfies (C3) by Lemma 11.4.4.

   (d) If \(\vec{t}_1\) satisfies (C4), then \(\{\vec{t} \in UTP_{i_1+1}(S) : p_{i_2,j_2}(\vec{t}) = 0\} \neq \emptyset\). The MUTP strategy can select a point that satisfies (C4) by Lemma 11.4.5.

   (e) If \(\vec{t}_1\) satisfies (C5), then \(\{\vec{t} \in UTP_{i_2}(S) : x_{l_2}(\vec{t}) = 0\} \neq \emptyset\). The MUTP strategy can select a point that satisfies (C5).

   (f) If \(\vec{t}_1\) satisfies (C6), then \(\{\vec{t} \in NFP_{i_2,j_2}(S) : x_{l_2}(\vec{t}) = 1\} \neq \emptyset\). The MNFP strategy can select a point that satisfies (C6).

   (g) If \(\vec{t}_1\) satisfies (C7), the SMOTP strategy can select a point that satisfies (C7). The proof is similar to that of Case 2 (e) in Theorem 11.4.1.

   (h) If \(\vec{t}_1\) satisfies (C8), the SMOTP strategy can select a point that satisfies (C8). The proof is similar to that of Case 2 (e) in Theorem 11.4.1.

The result follows.
11.5 Summary

In this chapter, six test case selection strategies have been proposed to supplement the MUMCUT strategy to detect six double-fault expressions, namely (6.13), (6.14), (6.16), (6.18) and (6.19). They are SMFP strategy, SMOTP strategy, SMUTP strategy, PMUTP strategy, PMNFP strategy and SPMFP strategy. Further analysis shows that the SMOTP strategy together with the MUMCUT strategy can detect double-fault expressions (8.20) and (8.27). As a result, the MUMCUT strategy together with these six strategies can guarantee to detect all double faults studied in this thesis.
Chapter 12

Empirical Study on MUMCUT with New Supplementary Strategies

12.1 The Aim

In Chapter 11, six test case selection strategies were proposed to supplement the MUMCUT strategy in detecting all 92 double-fault expressions studied in this thesis. The obvious question then is whether these 7 strategies amalgamated together as one, referred as “the MUMCUT-and-Six strategy” in the sequel for ease of references, is cost effective, in terms of the size of test set satisfying the strategy, when compared with other existing strategies such as the MUMCUT, MAX-A and MAX-B strategies.

However, there is some complication here. In fact, we will show in the following paragraph that the MAX-A strategy together with the SMOTP strategy can also detect all double faults. This is out of our expectation. As a result, we also need to consider this newly formed strategy, referred to as “the MAX-A-and-SMOTP strategy” in the sequel.

Chapter 10 showed that the MAX-A strategy can detect all double-fault expressions studied in this thesis except Expressions (6.14), (6.16), (6.18), (8.20) and (8.27). As proven in previous chapter, the MUMCUT strategy together with SMOTP strategy can detect Expressions (6.14), (8.20) and (8.27), while the MUMCUT strategy together with PMUTP, SMUTP and SMOTP strategy can detect Expressions (6.16) and (6.18). Since the MAX-A strategy subsumes the MUMCUT strategy, the MAX-A strategy together with SMOTP strategy can detect Expressions (6.14), (8.20) and (8.27). Moreover, after examining the requirements of the SMUTP and PMUTP strategies, we find that the MAX-A strategy also subsumes the former two strategies. Therefore, the MAX-A strategy together with the SMOTP
strategy can detect Expressions (6.16) and (6.18). As a result, the MAX-A strategy together with SMOTP can detect all double fault expression studied in this thesis.

In summary, we need to compare the cost-effectiveness of the MUMCUT-and-Six strategy and the MAX-A-and-SMOTP strategy with other existing strategies such as MUMCUT, MAX-A and MAX-B strategies. In this chapter, an empirical study is performed to investigate this issue.

12.2 Experimental Method and Subject Expressions

The subjects of the experiments are Boolean expressions extracted from three different sources in the research literature [11, 56, 60]. We classified them into three groups based on their origins and altogether there are 80 Boolean expressions. The first group, which is referred to the LRU group, has 20 expressions. They are extracted from the software of Line Replaceable Units (LRUs), also known as black boxes, which belong to five different airborne systems across two different airplane models [11]. Figure 12.1 in Section 12.3.1 lists all expressions in the LRU group. The second group, referred to as the TCAS group, also has 20 expressions and they are extracted from the specifications for an aircraft collision avoidance system (TCAS II) and are published in [56]. Figure 12.2 in Section 12.3.2 lists all expressions in the TCAS group. The third group, referred to as the Random group, has 40 randomly generated Boolean expressions and they are extracted from [60]. Figure 12.3 in Section 12.3.3 lists all expressions in the Random group.

We now discuss the generation of the required test sets for the empirical study. First, since all seven individual test case selection strategies for the MUMCUT-and-Six strategy require the Boolean expression be in IDNF, each of these 80 Boolean expressions is first transformed into an equivalent expression in IDNF, as in [60]. Figures 12.1 to 12.3 show all resulting equivalent 80 Boolean expressions in IDNF. Then, for each of these transformed expressions, test sets that satisfy each of the studied strategies are generated. For each expression, the test set satisfying the MUMCUT-and-Six strategy is the union of the test sets satisfying each of these 7 strategies, and the test set for the MAX-A-and-SMOTP strategy is the union of test sets satisfying the MAX-A and SMOTP strategies.

Second, we make an observation that, for some expressions, the number of possible test sets generated by each of the six strategies can be quite large. We illustrate this using Example 12.2.1 for the case of the SMOTP strategy. Although we illustrate the idea using SMOTP strategy as an example, the situations for the other 5 strategies are similar.
Example 12.2.1 Let $S = ab + cd + ef + gh + ij$ be a Boolean expression in IDNF. We now apply the SMOTP strategy to the first two terms of $S$. Please note that, $(\mathcal{T}_1(S) \cap \mathcal{T}_2(S)) \setminus \bigcup_{i=3}^{5} \mathcal{T}_i(S) = \{ \overline{t} \in \mathbb{B}^{10} : ab = cd = 1 \text{ and } ef = gh = ij = 0 \} = \{ 1111000000, (that is, } a=b=c=d=1, e=f=g=h=i=j=0), 1111000001, 1111000010, 1111000100, \ldots, 1111101010 \}$. Note that, any one of the following 10 different sets satisfies the requirement of the SMOTP strategy on the first two terms of $S$. That is, every possible truth value combination of $x_{l1}$ and $x_{l2}$ can be covered for every missing literal $x_{l1}$ of the first term and every missing literal $x_{l2}$ of the second term.

1. $\mathcal{T}_{1,2}^1 = \{ 1111000010, 1111010000, 1111010110, 1111011001, 1111100101, 1111101010 \}$
2. $\mathcal{T}_{1,2}^2 = \{ 1111000010, 1111010000, 1111010110, 1111011001, 1111100101, 1111101010 \}$
3. $\mathcal{T}_{1,2}^3 = \{ 1111000010, 1111010110, 1111011001, 1111100000, 1111100101, 1111101010 \}$
4. $\mathcal{T}_{1,2}^4 = \{ 1111000110, 1111010000, 1111010110, 1111011001, 1111100101, 1111101010 \}$
5. $\mathcal{T}_{1,2}^5 = \{ 1111000001, 1111010110, 1111011001, 1111010110, 1111100000, 1111100101, 1111101010 \}$
6. $\mathcal{T}_{1,2}^6 = \{ 1111000001, 1111010110, 1111011001, 1111010110, 1111011000, 1111100101, 1111101000 \}$
7. $\mathcal{T}_{1,2}^7 = \{ 1111000001, 1111010110, 1111011001, 1111100001, 1111100100, 1111101010 \}$
8. $\mathcal{T}_{1,2}^8 = \{ 1111000001, 1111010110, 1111011001, 1111010000, 1111100101, 1111100100, 1111101010 \}$
9. $\mathcal{T}_{1,2}^9 = \{ 1111000000, 1111000110, 1111000011, 1111010101, 1111100100, 1111100100, 1111101010 \}$
10. $\mathcal{T}_{1,2}^{10} = \{ 1111000000, 1111010010, 1111011010, 1111100101, 1111101010 \}$

It should be noted that there are other sets that satisfy the SMOTP strategy for the first two terms of $S$. Due to the symmetry property of $S$, there are at least 10 different possible test sets that satisfy the SMOTP strategy for each pair of terms.

---

1In fact, this is a shorter version of the real LRU expression $L20 = (ab + cd + ef + gh + ij + kl + mn)$ with positive literals for illustrative purpose.

2The notations $\mathcal{T}_{1,2}^1, \mathcal{T}_{1,2}^2, \ldots, \mathcal{T}_{1,2}^{10}$ are used in this example to denote the test sets that satisfy the SMOTP strategy for the first term, $p_1$, and the second term, $p_2$, of $S$ for ease of reference.
Since $S$ has 5 terms, there are 10 possible such pairs of terms. A SMOTP test set can be formed by selecting one out of 10 test sets for each of these 10 possible pairs and taking their union (for example, $T_{1,2}^0 \cup T_{1,3}^a \cup T_{1,4}^b \cup \cdots \cup T_{1,5}^0$). It should be noted that any two such sets from different pairs of terms are disjoint (that is, $T_{i_1,i_2}^k \cap T_{i_3,i_4}^l = \emptyset$ for all possible $k$ and $l$ when $i_1 \neq i_3$ or $i_2 \neq i_4$). As a result, each selection will give a different SMOTP test set and there are at least $10^{10}$ SMOTP test sets for $S$. Therefore, the number of possible SMOTP test sets can be quite large.

As a result, we cannot generate all possible test sets of the MUMCUT-and-Six strategy and the MAX-A-and-SMOTP strategy for some of these expressions. In this empirical study, we randomly generate 1000 test sets for each strategy under investigation. In the case that the total number of test sets for a particular strategy is less than 1000, we generate them all. For example, there is only one test set for the MAX-A strategy because it is required to select all unique true points and all near false points.

### 12.3 Results and Observations

As explained previously, the subjects of the empirical study come from three different sources. Each individual group may have some special characteristics of their own. As a result, we divide our discussions of the results and observations in three subsections, hoping to observe some special characteristics in each group.

#### 12.3.1 Results for LRU Expressions

In this section, we analyse the results of our empirical study on 20 LRU expressions listed in 12.1. The number of variables in these 20 expressions ranges from 5 (in L01) to 14 (in L19 and L20).

We first compare the sizes of test set satisfying each of the six newly proposed strategies, which together supplements the MUMCUT strategy in detecting double faults. Table 12.1 lists the mean size of the 1000 randomly generated test sets satisfying every such strategy for each expression. For example, it indicates that, for L02, on average, we need 33.3, 39.7, 35.0, 16.0, 16.0 and 28.0 test cases to satisfy the SMFP, PMNFP, SPMFP, SMUTP, PMUTP and SMOTP strategies, respectively.

From Table 12.1, we have the following two observations

1. For L01, the mean size of test sets satisfying the PMNFP, SPMFP and SMOTP strategies is 0. Note that every term in L01 has just one missing literal. As a result, for L01, it is easy to verify that
Table 12.1: The mean size of test sets satisfying each strategy for the LRU group

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>num. of terms</th>
<th>SMFP</th>
<th>PMNFP</th>
<th>SPMFP</th>
<th>SMUTP</th>
<th>PMUTP</th>
<th>SMOTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>L01</td>
<td>5</td>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>L02</td>
<td>7</td>
<td>8</td>
<td>33.3</td>
<td>39.7</td>
<td>35.0</td>
<td>16.0</td>
<td>16.0</td>
<td>28.0</td>
</tr>
<tr>
<td>L03</td>
<td>7</td>
<td>4</td>
<td>9.1</td>
<td>21.6</td>
<td>16.9</td>
<td>24.5</td>
<td>29.0</td>
<td>28.7</td>
</tr>
<tr>
<td>L04</td>
<td>7</td>
<td>4</td>
<td>52.4</td>
<td>40.4</td>
<td>65.2</td>
<td>4.0</td>
<td>4.0</td>
<td>6.0</td>
</tr>
<tr>
<td>L05</td>
<td>7</td>
<td>6</td>
<td>1.0</td>
<td>3.0</td>
<td>1.0</td>
<td>15.0</td>
<td>16.0</td>
<td>35.0</td>
</tr>
<tr>
<td>L06</td>
<td>8</td>
<td>2</td>
<td>66.1</td>
<td>36.0</td>
<td>95.0</td>
<td>6.0</td>
<td>5.0</td>
<td>1.0</td>
</tr>
<tr>
<td>L07</td>
<td>8</td>
<td>4</td>
<td>14.7</td>
<td>47.0</td>
<td>29.9</td>
<td>30.5</td>
<td>37.7</td>
<td>39.5</td>
</tr>
<tr>
<td>L08</td>
<td>9</td>
<td>8</td>
<td>20.0</td>
<td>46.0</td>
<td>22.0</td>
<td>35.2</td>
<td>37.7</td>
<td>77.3</td>
</tr>
<tr>
<td>L09</td>
<td>9</td>
<td>6</td>
<td>23.0</td>
<td>67.1</td>
<td>35.6</td>
<td>46.6</td>
<td>55.1</td>
<td>82.6</td>
</tr>
<tr>
<td>L10</td>
<td>9</td>
<td>5</td>
<td>25.8</td>
<td>56.5</td>
<td>43.4</td>
<td>33.6</td>
<td>43.1</td>
<td>60.3</td>
</tr>
<tr>
<td>L11</td>
<td>10</td>
<td>16</td>
<td>266.0</td>
<td>237.4</td>
<td>155.9</td>
<td>16.0</td>
<td>16.0</td>
<td>48.0</td>
</tr>
<tr>
<td>L12</td>
<td>10</td>
<td>6</td>
<td>14.7</td>
<td>53.3</td>
<td>29.7</td>
<td>48.9</td>
<td>61.2</td>
<td>125.5</td>
</tr>
<tr>
<td>L13</td>
<td>11</td>
<td>3</td>
<td>120.4</td>
<td>99.0</td>
<td>221.4</td>
<td>12.7</td>
<td>22.6</td>
<td>12.0</td>
</tr>
<tr>
<td>L14</td>
<td>12</td>
<td>7</td>
<td>258.5</td>
<td>259.2</td>
<td>259.8</td>
<td>26.8</td>
<td>27.9</td>
<td>51.0</td>
</tr>
<tr>
<td>L15</td>
<td>13</td>
<td>9</td>
<td>122.5</td>
<td>276.2</td>
<td>113.5</td>
<td>81.8</td>
<td>97.1</td>
<td>228.1</td>
</tr>
<tr>
<td>L16</td>
<td>13</td>
<td>7</td>
<td>29.8</td>
<td>170.3</td>
<td>56.2</td>
<td>75.5</td>
<td>110.5</td>
<td>267.9</td>
</tr>
<tr>
<td>L17</td>
<td>13</td>
<td>4</td>
<td>158.8</td>
<td>218.2</td>
<td>291.2</td>
<td>24.0</td>
<td>46.4</td>
<td>16.0</td>
</tr>
<tr>
<td>L18</td>
<td>13</td>
<td>13</td>
<td>73.2</td>
<td>224.5</td>
<td>41.2</td>
<td>13.0</td>
<td>13.0</td>
<td>78.0</td>
</tr>
<tr>
<td>L19</td>
<td>14</td>
<td>13</td>
<td>73.2</td>
<td>224.5</td>
<td>41.2</td>
<td>13.0</td>
<td>13.0</td>
<td>78.0</td>
</tr>
<tr>
<td>L20</td>
<td>14</td>
<td>7</td>
<td>34.0</td>
<td>207.0</td>
<td>66.7</td>
<td>82.7</td>
<td>107.5</td>
<td>293.2</td>
</tr>
<tr>
<td>Average</td>
<td>10</td>
<td>7</td>
<td>67.2</td>
<td>105.2</td>
<td>79.0</td>
<td>29.8</td>
<td>38.1</td>
<td>77.8</td>
</tr>
</tbody>
</table>

(a) \( \{ \bar{r} \in NFP_{i,j} (L01) : x_{l_1} x_{l_2} = 1 \} = \emptyset \) for every term \( p_i \) in L01, every literal \( x_{l_1} \) in \( p_i \), and every pair of different missing literals \( x_{l_1} \) and \( x_{l_2} \) of \( p_i \). It is because L01 only has one missing literal for every term. Hence, there is
no test case satisfying the PMNFP strategy.

(b) \{t \in FP(L01) : p_{i,j_1,j_2}x_{i_1}x_{i_2} = 1\} = \emptyset for every term \( p_i \) in L01, every pair of different literals \( x_{j_1}^i \) and \( x_{j_2}^i \) \( (j_1 \neq j_2) \) in \( p_i \), and every pair of different missing literals \( x_{i_1} \) and \( x_{i_2} \) of \( p_i \). It is because L01 only has one missing literal for every term. Hence, there is no test case satisfying the SPMFP strategy.

(c) \( TP_{i_1}(L01) \cap TP_{i_2}(L01) \backslash \bigcup_{i \neq i_1,i_2} TP_i(L01) \) = \emptyset for every pair of different terms \( p_{i_1} \) and \( p_{i_2} \) \( (i_1 \neq i_2) \) in L01. Hence, there is no test case satisfying the SMOTP strategy.

2. For L18, the mean size of test sets satisfying the SMFP, SPMFP and SMUTP strategies are 0. Note that every term of L18 has just one literal. As a result, for L18, it is easy to verify that

(a) \{t \in FP(L18) : p_{i,j_1,j_2}x_{i_1} = 1\} = \emptyset for every term \( p_i \) in L18, every pair of different literals \( x_{j_1}^i \) and \( x_{j_2}^i \) in \( p_i \), and every missing literal \( x_{i_1} \) of \( p_i \). It is because L18 has only one literal in every term. Hence, there is no test case satisfying the SMFP strategy.

(b) \{t \in FP(L18) : p_{i,j_1,j_2}x_{i_1}x_{i_2} = 1\} = \emptyset for every term \( p_i \) of L18, every pair of different literals \( x_{j_1}^i \) and \( x_{j_2}^i \) \( (j_1 \neq j_2) \) in \( p_i \), and every pair of different missing literals \( x_{i_1} \) and \( x_{i_2} \) of \( p_i \). It is because L18 has only one literal in every term. Hence, there is no test case satisfying the SPMFP strategy.

(c) \{t \in UTP_{i_1}(L18) : p_{i_2,j_2}x_{i_1} = 0\} = \emptyset for every term \( p_{i_1} \) of L18, every missing literal \( x_{i_1} \) of \( p_{i_1} \), every term \( p_{i_2} \) different from \( p_{i_1} \) \( (i_2 \neq i_1) \) of L18, and every literal \( x_{j_2}^{i_2} \) in \( p_{i_2} \). Hence there is no test case satisfying the SMUTP strategy.

Now, we compare the mean size of the test sets satisfying the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies. Tables 12.2 lists, for each expression in the LRU group, the mean size of the test sets satisfying each strategy mentioned above. For example, it indicates that, for L01, on average, 14.0, 25.0, 14, 14.0 and 20 test cases are required to satisfy the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies, respectively. It should be noted that (1) there is only one test set for the MAX-A strategy because it requires the selection of all unique true points and all near false points; and (2) all test sets of the MAX-B strategy are of the same size because it requires us to generate a fixed number of test cases. We have the following observations based on Table 12.2.
Table 12.2: Comparing different testing strategies for LRU expressions

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>MUMCUT</th>
<th>MUMCUT-and-Six</th>
<th>MAX-A</th>
<th>MAX-A-and-SMOTP</th>
<th>MAX-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>L01</td>
<td>5</td>
<td>14.0</td>
<td>25.0</td>
<td>14</td>
<td>14.0</td>
<td>20</td>
</tr>
<tr>
<td>L02</td>
<td>7</td>
<td>30.5</td>
<td>95.6</td>
<td>66</td>
<td>94.0</td>
<td>74</td>
</tr>
<tr>
<td>L03</td>
<td>7</td>
<td>21.0</td>
<td>95.0</td>
<td>81</td>
<td>109.7</td>
<td>87</td>
</tr>
<tr>
<td>L04</td>
<td>7</td>
<td>20.0</td>
<td>97.9</td>
<td>50</td>
<td>56.0</td>
<td>61</td>
</tr>
<tr>
<td>L05</td>
<td>7</td>
<td>15.7</td>
<td>54.0</td>
<td>19</td>
<td>54.0</td>
<td>26</td>
</tr>
<tr>
<td>L06</td>
<td>8</td>
<td>25.6</td>
<td>118.0</td>
<td>42</td>
<td>43.0</td>
<td>51</td>
</tr>
<tr>
<td>L07</td>
<td>8</td>
<td>26.7</td>
<td>169.4</td>
<td>188</td>
<td>227.5</td>
<td>196</td>
</tr>
<tr>
<td>L08</td>
<td>9</td>
<td>33.8</td>
<td>189.0</td>
<td>176</td>
<td>253.3</td>
<td>185</td>
</tr>
<tr>
<td>L09</td>
<td>9</td>
<td>37.9</td>
<td>266.0</td>
<td>274</td>
<td>356.6</td>
<td>283</td>
</tr>
<tr>
<td>L10</td>
<td>9</td>
<td>32.6</td>
<td>266.0</td>
<td>332</td>
<td>392.3</td>
<td>340</td>
</tr>
<tr>
<td>L11</td>
<td>10</td>
<td>60.0</td>
<td>546.6</td>
<td>496</td>
<td>544.0</td>
<td>513</td>
</tr>
<tr>
<td>L12</td>
<td>10</td>
<td>32.9</td>
<td>301.2</td>
<td>351</td>
<td>476.5</td>
<td>361</td>
</tr>
<tr>
<td>L13</td>
<td>11</td>
<td>38.0</td>
<td>404.1</td>
<td>1058</td>
<td>1070.0</td>
<td>1074</td>
</tr>
<tr>
<td>L14</td>
<td>12</td>
<td>66.8</td>
<td>814.4</td>
<td>1228</td>
<td>1279.0</td>
<td>1248</td>
</tr>
<tr>
<td>L15</td>
<td>13</td>
<td>73.9</td>
<td>911.2</td>
<td>4792</td>
<td>5020.1</td>
<td>4814</td>
</tr>
<tr>
<td>L16</td>
<td>13</td>
<td>45.7</td>
<td>721.3</td>
<td>2916</td>
<td>3183.9</td>
<td>2929</td>
</tr>
<tr>
<td>L17</td>
<td>13</td>
<td>51.2</td>
<td>774.7</td>
<td>4348</td>
<td>4364.0</td>
<td>4367</td>
</tr>
<tr>
<td>L18</td>
<td>13</td>
<td>14.0</td>
<td>92.0</td>
<td>14</td>
<td>92.0</td>
<td>27</td>
</tr>
<tr>
<td>L19</td>
<td>14</td>
<td>28.0</td>
<td>435.7</td>
<td>8205</td>
<td>8283.0</td>
<td>8219</td>
</tr>
<tr>
<td>L20</td>
<td>14</td>
<td>49.7</td>
<td>823.1</td>
<td>7289</td>
<td>7582.2</td>
<td>7504</td>
</tr>
<tr>
<td>Avg</td>
<td>10.0</td>
<td>35.9</td>
<td>360.0</td>
<td>1597.0</td>
<td>1674.8</td>
<td>1609.0</td>
</tr>
</tbody>
</table>

- The range and the average of the mean size of the test set satisfying each individual strategy are listed as follows
  1. MUMCUT: range from 14.0 (in L01 and L18) to 73.9 (in L15) with an average of 35.9
  2. MUMCUT-and-Six: range from 25.0 (in L01) to 911.2 (in L15) with an average of 360.0
  3. MAX-A: range from 14 (in L01 and L18) to 8205 (in L19) with an average of 1597.0
  4. MAX-A-and-SMOTP: range from 14 (in L01) to 8283 (in L19) with an average of 1674.8
  5. MAX-B: range from 20 (in L01) to 8219 (in L19) with an average of 1609.0

In general, the mean size of the test sets required to satisfy each strategy can be arranged in the order of MUMCUT << MUMCUT-and-Six << MAX-A < MAX-B < MAX-A-and-SMOTP, indicating that the MUMCUT strategy requires far less test cases than the MUMCUT-and-Six strategy, which in turn requires far less test cases than the MAX-A strategy, which in turn requires...
less test cases than the MAX-B strategy, which in turn requires less test cases than the MAX-A-and-SMOTP strategy.

- As mentioned previously, only the MUMCUT-and-Six and the MAX-A-and-SMOTP strategies can guarantee the detection of all double faults and both strategies subsumes the MUMCUT strategy. Hence, besides using the MUMCUT strategy to detect all single faults, if we want to further guarantee detecting all double faults, we need to use either the MUMCUT-and-Six strategy or the MAX-A-and-SMOTP strategy. The MUMCUT-and-Six strategy requires approximately 324.1 (=360.0-35.9) extra test cases per expression, on average. Even though the jump is big, it is still small when compared with the MAX-A-and-SMOTP strategy because it requires approximately 1638.9 (=1674.8 - 35.9) more test cases. Alternatively, the MUMCUT-and-Six strategy requires approximately 10.0 (=360.0/35.9) times the resources required for the MUMCUT strategy, whereas the MAX-A-and-SMOTP strategy requires approximately 46.7 times. Hence, the MUMCUT-and-Six strategy is more cost effective than the MAX-A-and-SMOTP strategy for double fault detection, in general.

- In 13 LRU expressions out of 20, the mean size of test sets for the MUMCUT-and-Six strategy is less than that of the MAX-A-and-SMOTP strategy. For L05 and L18, both the MUMCUT-and-Six and MAX-A-and-SMOTP test sets have the same size. For L01, L02, L04, L06, and L11, the MAX-A-and-SMOTP test sets have smaller size than those of the MUMCUT-and-Six strategy.

- In general, when the number of variables in the LRU expressions is greater than or equal to 11, the size of the test set satisfying the MUMCUT-and-Six strategy is smaller than that of the MAX-A-and-SMOTP strategy, except possibly for L18.

### 12.3.2 Results for TCAS Expressions

In this section, we analyse the results of the empirical study on the TCAS expressions listed in Figure 12.2. The number of variables in these 20 expression range from 5 (in T04) to 14 (in T12).

We first compare the sizes of test set satisfying each of the six newly proposed strategies, which together supplements the MUMCUT strategy in detecting double faults. Table 12.3 lists the mean size of the test sets satisfying every such strategy for each expression. For example, it indicates that, for T01, on average, 67.40, 20, 37,
Figure 12.2: The TCAS expressions [56]

6.53, 8 and 2 test cases are required to satisfy the SMFP, PMNFP, SPMFP, SMUTP, PMUTP and SMOTP strategies, respectively. We have the following observations from Table 12.3:

1. For T08 and T09, the mean sizes of test sets satisfying each of the six strategies are 0. On closer examination, T08 and T09 are in canonical IDNF, a special type of IDNF where every term of the expression contains all variables. Therefore, every term in T08 and T09 has no missing variable. As a result, for T08, it is easy to verify that

(a) \[ \{ \vec{t} : \vec{t} \in FP(T08) \text{ such that } p_{i,j_1,j_2}x_{i_2} = 1 \} = \emptyset \text{ for every term } p_i \text{ in } T08, \]
every pair of different literals \( x_{j_1}^{i_1} \) and \( x_{j_2}^{i_2} \) in \( p_i \), and every missing literal \( x_{i_2} \) of \( p_i \). Hence, there are no test cases for the SMFP strategy because it requires some terms to have at least one missing literal.

(b) \[ \{ \vec{t} : \vec{t} \in NFP_{i,j}(T08) \text{ such that } x_{i_1}x_{i_2} = 1 \} = \emptyset \text{ for every term } p_i \text{ in } T08, \]
every literal \( x_j \) in \( p_i \), and every pair of different missing literals \( x_{i_1} \) and \( x_{i_2} \) of \( p_i \). Hence, there are no test cases for the PMNFP strategy because it requires some terms to have at least two missing literals.

(c) \[ \{ \vec{t} : \vec{t} \in FP(T08) \text{ such that } p_{i,j_1,j_2}x_{i_1}x_{i_2} = 1 \} = \emptyset \text{ for every term } p_i \text{ in } T08, \]
every pair of literals \( x_{j_1}^{i_1} \) and \( x_{j_2}^{i_2} \) in \( p_i \), and every pair of different missing
Table 12.3: The mean size of randomly generated test sets for each strategy from 20 TCAS II expressions

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>num. of terms</th>
<th>SMFP</th>
<th>PMNFP</th>
<th>SMNFP</th>
<th>SMUTP</th>
<th>PMUTP</th>
<th>SMOTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>T01</td>
<td>7</td>
<td>5</td>
<td>67.4</td>
<td>20.0</td>
<td>37.0</td>
<td>6.5</td>
<td>8.0</td>
<td>2.0</td>
</tr>
<tr>
<td>T02</td>
<td>9</td>
<td>13</td>
<td>233.0</td>
<td>0.0</td>
<td>0.0</td>
<td>15.0</td>
<td>15.0</td>
<td>3.0</td>
</tr>
<tr>
<td>T03</td>
<td>12</td>
<td>25</td>
<td>911.1</td>
<td>801.8</td>
<td>693.7</td>
<td>100.8</td>
<td>149.6</td>
<td>219.2</td>
</tr>
<tr>
<td>T04</td>
<td>5</td>
<td>3</td>
<td>9.0</td>
<td>9.5</td>
<td>9.0</td>
<td>7.1</td>
<td>8.6</td>
<td>3.0</td>
</tr>
<tr>
<td>T05</td>
<td>9</td>
<td>9</td>
<td>62.9</td>
<td>67.2</td>
<td>67.5</td>
<td>29.0</td>
<td>34.9</td>
<td>56.0</td>
</tr>
<tr>
<td>T06</td>
<td>11</td>
<td>6</td>
<td>285.2</td>
<td>100.0</td>
<td>0.0</td>
<td>8.0</td>
<td>8.0</td>
<td>4.0</td>
</tr>
<tr>
<td>T07</td>
<td>10</td>
<td>8</td>
<td>355.0</td>
<td>159.3</td>
<td>410.7</td>
<td>17.3</td>
<td>18.0</td>
<td>10.0</td>
</tr>
<tr>
<td>T08</td>
<td>8</td>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>T09</td>
<td>7</td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>T10</td>
<td>13</td>
<td>6</td>
<td>585.2</td>
<td>191.5</td>
<td>759.4</td>
<td>12.0</td>
<td>12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>T11</td>
<td>13</td>
<td>9</td>
<td>628.7</td>
<td>431.4</td>
<td>829.0</td>
<td>54.4</td>
<td>67.6</td>
<td>68.3</td>
</tr>
<tr>
<td>T12</td>
<td>14</td>
<td>8</td>
<td>545.3</td>
<td>463.0</td>
<td>641.1</td>
<td>51.6</td>
<td>73.3</td>
<td>106.3</td>
</tr>
<tr>
<td>T13</td>
<td>12</td>
<td>6</td>
<td>51.7</td>
<td>104.7</td>
<td>80.4</td>
<td>33.2</td>
<td>59.4</td>
<td>115.5</td>
</tr>
<tr>
<td>T14</td>
<td>7</td>
<td>6</td>
<td>32.8</td>
<td>39.6</td>
<td>37.6</td>
<td>20.5</td>
<td>23.7</td>
<td>19.6</td>
</tr>
<tr>
<td>T15</td>
<td>9</td>
<td>11</td>
<td>88.0</td>
<td>115.0</td>
<td>70.9</td>
<td>42.5</td>
<td>48.8</td>
<td>62.1</td>
</tr>
<tr>
<td>T16</td>
<td>12</td>
<td>23</td>
<td>375.8</td>
<td>456.5</td>
<td>293.8</td>
<td>123.8</td>
<td>152.8</td>
<td>331.2</td>
</tr>
<tr>
<td>T17</td>
<td>11</td>
<td>6</td>
<td>228.1</td>
<td>267.6</td>
<td>324.1</td>
<td>36.7</td>
<td>47.7</td>
<td>46.7</td>
</tr>
<tr>
<td>T18</td>
<td>10</td>
<td>8</td>
<td>210.0</td>
<td>187.9</td>
<td>229.1</td>
<td>40.1</td>
<td>46.2</td>
<td>50.2</td>
</tr>
<tr>
<td>T19</td>
<td>8</td>
<td>4</td>
<td>92.3</td>
<td>70.5</td>
<td>132.4</td>
<td>18.5</td>
<td>20.9</td>
<td>4.0</td>
</tr>
<tr>
<td>T20</td>
<td>7</td>
<td>2</td>
<td>50.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Average</td>
<td>10</td>
<td>8</td>
<td>240.6</td>
<td>171.3</td>
<td>250.5</td>
<td>31.0</td>
<td>39.9</td>
<td>55.7</td>
</tr>
</tbody>
</table>

literals $x_{l_1}$ and $x_{l_2}$ of $p_i$. Hence, there are no test cases for the SPMFP strategy because it requires some terms to have at least two missing literals.

(d) \( \{ \vec{t} : \vec{t} \in UTP_{i_1}(T08) \text{ such that } p_{i_2,j_2} + x_{l_1} = 0 \} = \emptyset \) for every term $p_{i_1}$ of T08, every missing literal $x_{l_1}$ of $p_{i_1}$, every term $p_{i_2}$ of T08 different from $p_{i_1}$, and every literal $x_{l_2}$ in $p_{i_2}$. Hence, there are no test cases for the SMUTP strategy because it requires some terms to have at least one missing literal.

(e) \( \{ \vec{t} : \vec{t} \in UTP_{i_1}(T08) \text{ such that } x_{l_1} + x_{l_2} = 0 \} = \emptyset \) for every term $p_{i_1}$ of T08, every missing literal $x_{l_1}$ of $p_{i_1}$, every term $p_{i_2}$ different from $p_{i_1}$ of T08, and every missing literal $x_{l_2}$ of $p_{i_2}$. Hence, there are no test cases for the PMUTP strategy because it requires some pair of terms to have at least one missing literal in every individual term in the pair.

(f) \( \{ \vec{t} : \vec{t} \in (TP_{i_1}(T08) \cap TP_{i_2}(T08)) \setminus ( \bigcup_{i \neq i_1,i_2} TP_i(T08)) \text{ such that } x_{l_1} + x_{l_2} = 0 \} = \emptyset \) for every pair of different terms $p_{i_1}$ and $p_{i_2}$ in T08, every missing literal $x_{l_1}$ of $p_{i_1}$, and every missing literal $x_{l_2}$ of $p_{i_2}$. Hence, there are no test cases for the SMOTP strategy because it requires some pair of terms
to have at least one missing literal in every individual term in the pair.

Thus, no test cases can be selected to satisfy the six new supplementary strategies. The implication is that the MUMCUT strategy can detect all double faults of a canonical IDNF. The situation for T09 is similar.

2. For T02, the mean sizes of test sets satisfying the PMNFP and SPMFP strategies are 0. Note that every term in T02 has just one missing literal, except the 7-th term $\bar{a}\bar{b}\bar{c}d\bar{e}\bar{f}\bar{g}\bar{h}\bar{i}$ which is a minterm of T02. That is, for T02, each term does not have at least two missing literals. As a result, there are no test cases satisfying the PMNFP and SPMFP strategies because any of them requires to select test cases from terms having at least two missing literals.

3. For T20, the mean sizes of test sets satisfying the PMNFP, SPMFP and SMOTP strategies for the expression are 0. Note that every term in T20 has just one missing literal. Similar to the situation in T02 above, there are no test cases satisfying the PMNFP and SPMFP strategies. For the SMOTP strategy, a closer look at T20 shows that (1) it has only two terms and (2) the literal $b$ exists in one term and its negation ($\bar{b}$) exists in the other term. As a result, $(TP_1(T20) \cap TP_2(T20)) = \emptyset$. Hence, there is no test case satisfying the SMOTP strategy.

We now compare the mean size of the test sets satisfying the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies. Tables 12.4 lists, for each expression in the TCAS group, the mean size of the test sets satisfying every strategy mentioned above. For example, it indicates that, for T01, on average, 38.48, 102, 48, 50 and 57 test cases are selected to satisfy the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies, respectively. We have the following observations based on Table 12.4

- The range and the average of the mean size of the test set satisfying each individual strategy are listed as follows

1. MUMCUT: range from 11.7 (in T09) to 276.2 (in T03) with an average of 82.2
2. MUMCUT-and-Six: range from 16 (in T09) to 2194 (in T03) with an average of 602.6
3. MAX-A: range from 16 (in T09) to 3886 (in T12) with an average of 758.8
4. MAX-A-and-SMOTP: range from 16 (in T09) to 3992.3 (in T12) with an average of 814.5
Table 12.4: Comparing different testing strategies for TCAS II expressions

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>MUMCUT</th>
<th>MUMCUT-and-Six</th>
<th>MAX-A</th>
<th>MAX-A-and-SMOTP</th>
<th>MAX-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>T01</td>
<td>7</td>
<td>38.5</td>
<td>102</td>
<td>48</td>
<td>50</td>
<td>57</td>
</tr>
<tr>
<td>T02</td>
<td>9</td>
<td>116</td>
<td>292</td>
<td>116</td>
<td>119</td>
<td>128</td>
</tr>
<tr>
<td>T03</td>
<td>12</td>
<td>276.2</td>
<td>2194</td>
<td>2570</td>
<td>2789.2</td>
<td>2590</td>
</tr>
<tr>
<td>T04</td>
<td>5</td>
<td>11.7</td>
<td>27.1</td>
<td>28</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>T05</td>
<td>9</td>
<td>43.4</td>
<td>225.0</td>
<td>324</td>
<td>380.0</td>
<td>332</td>
</tr>
<tr>
<td>T06</td>
<td>11</td>
<td>89.9</td>
<td>454</td>
<td>128</td>
<td>132</td>
<td>42</td>
</tr>
<tr>
<td>T07</td>
<td>10</td>
<td>102.5</td>
<td>531.8</td>
<td>208</td>
<td>218</td>
<td>222</td>
</tr>
<tr>
<td>T08</td>
<td>8</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>T09</td>
<td>7</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>T10</td>
<td>13</td>
<td>95.4</td>
<td>959.7</td>
<td>240</td>
<td>252</td>
<td>258</td>
</tr>
<tr>
<td>T11</td>
<td>13</td>
<td>142.3</td>
<td>1745.2</td>
<td>2006</td>
<td>2074.3</td>
<td>2027</td>
</tr>
<tr>
<td>T12</td>
<td>14</td>
<td>118.8</td>
<td>1783.5</td>
<td>3886</td>
<td>3992.3</td>
<td>3909</td>
</tr>
<tr>
<td>T13</td>
<td>12</td>
<td>37.5</td>
<td>412.7</td>
<td>1087</td>
<td>1802.5</td>
<td>1699</td>
</tr>
<tr>
<td>T14</td>
<td>7</td>
<td>35.5</td>
<td>102.6</td>
<td>86</td>
<td>105.6</td>
<td>95</td>
</tr>
<tr>
<td>T15</td>
<td>9</td>
<td>66.3</td>
<td>304.1</td>
<td>297</td>
<td>359.1</td>
<td>310</td>
</tr>
<tr>
<td>T16</td>
<td>12</td>
<td>187.5</td>
<td>1315.5</td>
<td>1924</td>
<td>2255.2</td>
<td>1941</td>
</tr>
<tr>
<td>T17</td>
<td>11</td>
<td>78.0</td>
<td>730</td>
<td>946</td>
<td>992.7</td>
<td>963</td>
</tr>
<tr>
<td>T18</td>
<td>10</td>
<td>82.8</td>
<td>560.9</td>
<td>494</td>
<td>544.2</td>
<td>510</td>
</tr>
<tr>
<td>T19</td>
<td>8</td>
<td>45.9</td>
<td>196.3</td>
<td>112</td>
<td>116</td>
<td>123</td>
</tr>
<tr>
<td>T20</td>
<td>7</td>
<td>24</td>
<td>64</td>
<td>24</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>Average</td>
<td>10</td>
<td>82.2</td>
<td>602.65</td>
<td>758.8</td>
<td>814.5</td>
<td>771.8</td>
</tr>
</tbody>
</table>

5. MAX-B: range from 23 (in T09) to 3909 (in T12) with an average of 771.8

In general, the mean size of the test sets required to satisfy each strategy can be arranged in the order of MUMCUT << MUMCUT-and-Six < MAX-A < MAX-B < MAX-A-and-SMOTP, indicating that the MUMCUT strategy requires far less test cases than the MUMCUT-and-Six strategy, which in turn requires less test cases than the MAX-A strategy, which in turn requires less test cases than the MAX-B strategy, which in turn requires less test cases than the MAX-A-and-SMOTP strategy.

- As mentioned previously, only the MUMCUT-and-Six and the MAX-A-and-SMOTP strategies can guarantee the detection of all double faults and both strategies subsumes the MUMCUT strategy. Hence, besides using the MUMCUT strategy to detect all single faults, if we want to further guarantee detecting all double faults, we need to use either the MUMCUT-and-Six strategy or the MAX-A-and-SMOTP strategy. The MUMCUT-and-Six strategy requires approximately 520.4 (=602.6 - 82.2) extra test cases per expression, on average. Even though the jump is big, it is still small when compared with the MAX-A-and-SMOTP strategy because it requires approximately 732.3 (= 814.5 - 82.2) more test cases. Alternatively, the MUMCUT-and-Six strategy
requires approximately 7.3 (=602.6/82.2) times the resources required for the MUMCUT strategy, whereas the MAX-A-and-SMOTP strategy requires approximately 9.9 times. Hence, the MUMCUT-and-Six strategy is more cost effective than the MAX-A-and-SMOTP strategy for double fault detection, in general.

- In 10 TCAS expressions out of 20, the mean size of test sets for the MUMCUT-and-Six strategy is less than that of the MAX-A-and-SMOTP strategy. For T08 and T09, both the MUMCUT-and-Six and MAX-A-and-SMOTP test sets have the same size. For T01, T02, T06, T07, T10 and T18-T20, the test sets for the MAX-A-and-SMOTP strategy have smaller size than those of the MUMCUT-and-Six strategy.

### 12.3.3 Results for Random Expressions

In this section, we analyse the results of the empirical study on the 40 random Boolean expressions listed in Figure 12.3. The number of variables in these 40 expressions ranges from 10 to 14.

We first compare the sizes of test set satisfying each of the six newly proposed strategies, which together supplements the MUMCUT strategy in detecting double faults. Table 12.5 lists the mean size of the test sets satisfying every such strategy for each expression. For example, it indicates that, for R02, on average, 56.4, 104.6, 77.7, 48.3, 66.2 and 104.8 test cases are required to satisfy the SMFP, PM-NFP, SPMFP, SMUTP, PMUTP and SMOTP strategies, respectively. There are no special observations from Table 12.5.

We now compare the mean size of the test sets satisfying the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies. Tables 12.6 lists, for each expression in the Random group, the mean size of the test sets satisfying every strategy mentioned above. For example, it indicates that, for R01, on average, 29.7, 229.3, 88, 239.4 and 98 test cases are selected to satisfy the MUMCUT, MUMCUT-and-Six, MAX-A, MAX-A-and-SMOTP and MAX-B strategies, respectively. We have the following observations based on Table 12.6.

- The range and the average of the mean size of the test set satisfying each individual strategy are listed as follows

  1. MUMCUT: range from 28.4 (in R01) to 265.5 (in R36) with an average of 86.6
  2. MUMCUT-and-Six: range from 229.3 (in R01) to 3632.7 (in R36) with an average of 874.0
Figure 12.3: The random expressions [60]

3. MAX-A: range from 88 (in R01) to 15210 (in R38) with an average of 3017.6

4. MAX-A-and-SMOTP: range from 239.4 (in R01) to 15330.9 (in R38) with 309
Table 12.5: The mean size of randomly generated test sets for each strategy from 40 random expressions

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>num. of terms</th>
<th>SMFP</th>
<th>PMNFP</th>
<th>SPMFP</th>
<th>SMUTP</th>
<th>PMUTP</th>
<th>SMOTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>10</td>
<td>8</td>
<td>7.3</td>
<td>11.9</td>
<td>8.0</td>
<td>48.1</td>
<td>52.1</td>
<td>151.4</td>
</tr>
<tr>
<td>R02</td>
<td>10</td>
<td>7</td>
<td>56.4</td>
<td>104.6</td>
<td>77.7</td>
<td>48.3</td>
<td>66.2</td>
<td>104.8</td>
</tr>
<tr>
<td>R03</td>
<td>10</td>
<td>15</td>
<td>153.6</td>
<td>167.3</td>
<td>114.6</td>
<td>33.1</td>
<td>34.5</td>
<td>84.0</td>
</tr>
<tr>
<td>R04</td>
<td>10</td>
<td>11</td>
<td>201.6</td>
<td>154.1</td>
<td>227.2</td>
<td>29.6</td>
<td>36.8</td>
<td>59.0</td>
</tr>
<tr>
<td>R05</td>
<td>10</td>
<td>7</td>
<td>173.0</td>
<td>174.3</td>
<td>237.9</td>
<td>36.1</td>
<td>51.0</td>
<td>37.2</td>
</tr>
<tr>
<td>R06</td>
<td>10</td>
<td>10</td>
<td>172.8</td>
<td>150.8</td>
<td>213.6</td>
<td>35.4</td>
<td>49.1</td>
<td>79.9</td>
</tr>
<tr>
<td>R07</td>
<td>10</td>
<td>5</td>
<td>93.6</td>
<td>93.2</td>
<td>126.8</td>
<td>18.4</td>
<td>30.3</td>
<td>31.3</td>
</tr>
<tr>
<td>R08</td>
<td>10</td>
<td>11</td>
<td>83.7</td>
<td>107.2</td>
<td>85.2</td>
<td>57.9</td>
<td>77.7</td>
<td>136.5</td>
</tr>
<tr>
<td>R09</td>
<td>11</td>
<td>8</td>
<td>10.8</td>
<td>21.3</td>
<td>10.7</td>
<td>37.4</td>
<td>54.8</td>
<td>160.5</td>
</tr>
<tr>
<td>R10</td>
<td>11</td>
<td>8</td>
<td>22.6</td>
<td>54.6</td>
<td>38.5</td>
<td>64.1</td>
<td>77.5</td>
<td>202.3</td>
</tr>
<tr>
<td>R11</td>
<td>11</td>
<td>9</td>
<td>30.0</td>
<td>59.1</td>
<td>43.8</td>
<td>64.5</td>
<td>79.1</td>
<td>206.0</td>
</tr>
<tr>
<td>R12</td>
<td>13</td>
<td>13</td>
<td>384.7</td>
<td>359.4</td>
<td>361.2</td>
<td>78.7</td>
<td>143.0</td>
<td></td>
</tr>
<tr>
<td>R13</td>
<td>11</td>
<td>11</td>
<td>88.3</td>
<td>155.1</td>
<td>67.1</td>
<td>76.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R14</td>
<td>11</td>
<td>11</td>
<td>29.9</td>
<td>39.6</td>
<td>72.9</td>
<td>82.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R15</td>
<td>11</td>
<td>12</td>
<td>77.6</td>
<td>139.2</td>
<td>81.3</td>
<td>71.1</td>
<td>79.2</td>
<td>164.6</td>
</tr>
<tr>
<td>R16</td>
<td>11</td>
<td>11</td>
<td>56.4</td>
<td>85.9</td>
<td>58.4</td>
<td>70.8</td>
<td>90.7</td>
<td></td>
</tr>
<tr>
<td>R17</td>
<td>12</td>
<td>7</td>
<td>113.4</td>
<td>166.0</td>
<td>40.6</td>
<td>64.7</td>
<td>116.7</td>
<td></td>
</tr>
<tr>
<td>R18</td>
<td>12</td>
<td>8</td>
<td>85.1</td>
<td>141.2</td>
<td>95.9</td>
<td>76.9</td>
<td>176.4</td>
<td></td>
</tr>
<tr>
<td>R19</td>
<td>12</td>
<td>10</td>
<td>18.3</td>
<td>28.6</td>
<td>62.8</td>
<td>76.5</td>
<td>259.8</td>
<td></td>
</tr>
<tr>
<td>R20</td>
<td>12</td>
<td>10</td>
<td>61.6</td>
<td>118.0</td>
<td>74.8</td>
<td>68.8</td>
<td>92.1</td>
<td></td>
</tr>
<tr>
<td>R21</td>
<td>12</td>
<td>15</td>
<td>237.3</td>
<td>294.1</td>
<td>216.4</td>
<td>66.8</td>
<td>212.0</td>
<td></td>
</tr>
<tr>
<td>R22</td>
<td>12</td>
<td>6</td>
<td>202.7</td>
<td>185.3</td>
<td>330.7</td>
<td>29.3</td>
<td>51.4</td>
<td>62.8</td>
</tr>
<tr>
<td>R23</td>
<td>12</td>
<td>9</td>
<td>101.0</td>
<td>128.4</td>
<td>58.9</td>
<td>83.1</td>
<td>174.5</td>
<td></td>
</tr>
<tr>
<td>R24</td>
<td>12</td>
<td>13</td>
<td>198.4</td>
<td>188.7</td>
<td>78.6</td>
<td>103.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R25</td>
<td>13</td>
<td>8</td>
<td>47.2</td>
<td>107.4</td>
<td>83.6</td>
<td>61.0</td>
<td>86.6</td>
<td></td>
</tr>
<tr>
<td>R26</td>
<td>13</td>
<td>4</td>
<td>184.3</td>
<td>215.7</td>
<td>348.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R27</td>
<td>13</td>
<td>8</td>
<td>679.3</td>
<td>316.6</td>
<td>764.0</td>
<td>25.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R28</td>
<td>13</td>
<td>22</td>
<td>351.8</td>
<td>510.3</td>
<td>121.7</td>
<td>156.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R29</td>
<td>13</td>
<td>7</td>
<td>64.8</td>
<td>86.4</td>
<td>38.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R30</td>
<td>13</td>
<td>9</td>
<td>172.1</td>
<td>203.0</td>
<td>60.9</td>
<td>93.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R31</td>
<td>13</td>
<td>10</td>
<td>712.4</td>
<td>721.2</td>
<td>49.6</td>
<td>59.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R32</td>
<td>13</td>
<td>13</td>
<td>311.0</td>
<td>391.8</td>
<td>426.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R33</td>
<td>13</td>
<td>8</td>
<td>50.3</td>
<td>197.2</td>
<td>76.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R34</td>
<td>13</td>
<td>12</td>
<td>299.8</td>
<td>356.2</td>
<td>85.0</td>
<td>126.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R35</td>
<td>13</td>
<td>29</td>
<td>451.0</td>
<td>246.1</td>
<td>224.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R36</td>
<td>14</td>
<td>25</td>
<td>901.4</td>
<td>1823.3</td>
<td>116.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R37</td>
<td>14</td>
<td>11</td>
<td>160.8</td>
<td>231.8</td>
<td>70.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R38</td>
<td>14</td>
<td>12</td>
<td>770.9</td>
<td>1038.6</td>
<td>58.3</td>
<td>93.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R39</td>
<td>14</td>
<td>8</td>
<td>410.4</td>
<td>670.7</td>
<td>49.9</td>
<td>86.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R40</td>
<td>14</td>
<td>13</td>
<td>140.9</td>
<td>145.8</td>
<td>101.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>12</td>
<td>11</td>
<td>209.2</td>
<td>266.5</td>
<td>63.0</td>
<td>82.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>SMFP</th>
<th>PMNFP</th>
<th>SPMFP</th>
<th>SMUTP</th>
<th>PMUTP</th>
<th>SMOTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>12</td>
<td>11</td>
<td>209.2</td>
<td>266.5</td>
<td>63.0</td>
<td>82.6</td>
<td>190.0</td>
<td></td>
</tr>
</tbody>
</table>

an average of 3207.6

5. MAX-B: range from 98 (in R01) to 15221 (in R38) with an average of 3031.0

In general, the mean size of the test sets required to satisfy each strategy can
Table 12.6: Comparing different testing strategies for Random expressions

<table>
<thead>
<tr>
<th>Expr</th>
<th>num. of variables</th>
<th>MUMCUT</th>
<th>MUMCUT-and-Six</th>
<th>MAX-A</th>
<th>MAX-A-and-SMOTP</th>
<th>MAX-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>10</td>
<td>29.7</td>
<td>229.3</td>
<td>88</td>
<td>239.4</td>
<td>98</td>
</tr>
<tr>
<td>R02</td>
<td>10</td>
<td>52.6</td>
<td>377.9</td>
<td>608</td>
<td>712.8</td>
<td>617</td>
</tr>
<tr>
<td>R03</td>
<td>10</td>
<td>68.4</td>
<td>438.1</td>
<td>581</td>
<td>665</td>
<td>596</td>
</tr>
<tr>
<td>R04</td>
<td>10</td>
<td>84.1</td>
<td>425.8</td>
<td>872</td>
<td>931</td>
<td>880</td>
</tr>
<tr>
<td>R05</td>
<td>10</td>
<td>70.5</td>
<td>516.1</td>
<td>883</td>
<td>920.2</td>
<td>896</td>
</tr>
<tr>
<td>R06</td>
<td>10</td>
<td>77.4</td>
<td>469.5</td>
<td>606</td>
<td>685.9</td>
<td>620</td>
</tr>
<tr>
<td>R07</td>
<td>10</td>
<td>36.8</td>
<td>314.1</td>
<td>729</td>
<td>760.3</td>
<td>745</td>
</tr>
<tr>
<td>R08</td>
<td>10</td>
<td>74.9</td>
<td>403</td>
<td>504</td>
<td>640.5</td>
<td>514</td>
</tr>
<tr>
<td>R09</td>
<td>11</td>
<td>28.4</td>
<td>266.3</td>
<td>169</td>
<td>329.5</td>
<td>180</td>
</tr>
<tr>
<td>R10</td>
<td>11</td>
<td>42.2</td>
<td>398.1</td>
<td>382</td>
<td>584.3</td>
<td>393</td>
</tr>
<tr>
<td>R11</td>
<td>11</td>
<td>46</td>
<td>402.4</td>
<td>381</td>
<td>587.0</td>
<td>392</td>
</tr>
<tr>
<td>R12</td>
<td>11</td>
<td>129.8</td>
<td>975.2</td>
<td>1656</td>
<td>1743.3</td>
<td>1672</td>
</tr>
<tr>
<td>R13</td>
<td>11</td>
<td>66</td>
<td>385.2</td>
<td>896</td>
<td>1039.0</td>
<td>909</td>
</tr>
<tr>
<td>R14</td>
<td>11</td>
<td>53.2</td>
<td>403.4</td>
<td>248</td>
<td>487.1</td>
<td>259</td>
</tr>
<tr>
<td>R15</td>
<td>11</td>
<td>65.6</td>
<td>527.2</td>
<td>768</td>
<td>932.6</td>
<td>781</td>
</tr>
<tr>
<td>R16</td>
<td>11</td>
<td>62.3</td>
<td>475.9</td>
<td>552</td>
<td>772.1</td>
<td>563</td>
</tr>
<tr>
<td>R17</td>
<td>12</td>
<td>58.2</td>
<td>522.8</td>
<td>2260</td>
<td>2376.7</td>
<td>2271</td>
</tr>
<tr>
<td>R18</td>
<td>12</td>
<td>57</td>
<td>556.5</td>
<td>1516</td>
<td>1692.4</td>
<td>1528</td>
</tr>
<tr>
<td>R19</td>
<td>12</td>
<td>42.9</td>
<td>407.9</td>
<td>238</td>
<td>497.8</td>
<td>250</td>
</tr>
<tr>
<td>R20</td>
<td>12</td>
<td>61.6</td>
<td>581.5</td>
<td>1161</td>
<td>1408.5</td>
<td>1173</td>
</tr>
<tr>
<td>R21</td>
<td>12</td>
<td>112.3</td>
<td>910.4</td>
<td>2634</td>
<td>2846.1</td>
<td>2645</td>
</tr>
<tr>
<td>R22</td>
<td>12</td>
<td>66.5</td>
<td>693.4</td>
<td>3140</td>
<td>3202.8</td>
<td>3158</td>
</tr>
<tr>
<td>R23</td>
<td>12</td>
<td>67</td>
<td>624.5</td>
<td>1864</td>
<td>2038.5</td>
<td>1876</td>
</tr>
<tr>
<td>R24</td>
<td>12</td>
<td>107.9</td>
<td>858.9</td>
<td>2168</td>
<td>2374.2</td>
<td>2179</td>
</tr>
<tr>
<td>R25</td>
<td>13</td>
<td>47.3</td>
<td>561.3</td>
<td>1382</td>
<td>1621.4</td>
<td>1395</td>
</tr>
<tr>
<td>R26</td>
<td>13</td>
<td>53.6</td>
<td>842.7</td>
<td>5118</td>
<td>5140.6</td>
<td>5137</td>
</tr>
<tr>
<td>R27</td>
<td>13</td>
<td>119.6</td>
<td>1370.5</td>
<td>626</td>
<td>653.8</td>
<td>646</td>
</tr>
<tr>
<td>R28</td>
<td>13</td>
<td>160.8</td>
<td>1585.3</td>
<td>2721</td>
<td>3109.0</td>
<td>2737</td>
</tr>
<tr>
<td>R29</td>
<td>13</td>
<td>39.5</td>
<td>489.9</td>
<td>2906</td>
<td>3061.2</td>
<td>2919</td>
</tr>
<tr>
<td>R30</td>
<td>13</td>
<td>79.9</td>
<td>972.0</td>
<td>5610</td>
<td>5816.9</td>
<td>5622</td>
</tr>
<tr>
<td>R31</td>
<td>13</td>
<td>147.4</td>
<td>1684.3</td>
<td>1412</td>
<td>1485.6</td>
<td>1433</td>
</tr>
<tr>
<td>R32</td>
<td>13</td>
<td>128</td>
<td>1372.9</td>
<td>4302</td>
<td>4587.6</td>
<td>4314</td>
</tr>
<tr>
<td>R33</td>
<td>14</td>
<td>53.6</td>
<td>853.7</td>
<td>4569</td>
<td>4899.6</td>
<td>4583</td>
</tr>
<tr>
<td>R34</td>
<td>14</td>
<td>119.5</td>
<td>1391.0</td>
<td>8246</td>
<td>8560.5</td>
<td>8259</td>
</tr>
<tr>
<td>R35</td>
<td>14</td>
<td>237</td>
<td>2455.3</td>
<td>5202</td>
<td>6013.8</td>
<td>5222</td>
</tr>
<tr>
<td>R36</td>
<td>14</td>
<td>265.5</td>
<td>3632.7</td>
<td>12384</td>
<td>12539.2</td>
<td>12405</td>
</tr>
<tr>
<td>R37</td>
<td>14</td>
<td>81.5</td>
<td>949.3</td>
<td>4380</td>
<td>4700.5</td>
<td>4394</td>
</tr>
<tr>
<td>R38</td>
<td>14</td>
<td>173.9</td>
<td>1995.1</td>
<td>15210</td>
<td>15330.9</td>
<td>15221</td>
</tr>
<tr>
<td>R39</td>
<td>14</td>
<td>105.5</td>
<td>1550.1</td>
<td>15109</td>
<td>15225.3</td>
<td>15120</td>
</tr>
<tr>
<td>R40</td>
<td>14</td>
<td>89.5</td>
<td>1091.7</td>
<td>6624</td>
<td>7093.0</td>
<td>6638</td>
</tr>
<tr>
<td>Average</td>
<td>12</td>
<td>86.6</td>
<td>874.0</td>
<td>3017.6</td>
<td>3207.6</td>
<td>3031.0</td>
</tr>
</tbody>
</table>

be arranged in the order of MUMCUT << MUMCUT-and-Six << MAX-A < MAX-B < MAX-A-and-SMOTP, indicating that the MUMCUT strategy requires far less test cases than the MUMCUT-and-Six strategy, which in turn requires far less test cases than the MAX-A strategy, which in turn requires less test cases than the MAX-B strategy, which in turn requires less test cases
than the MAX-A-and-SMOTP strategy.

- As mentioned previously, only the MUMCUT-and-Six and the MAX-A-and-SMOTP strategies can guarantee the detection of all double faults and both strategies subsumes the MUMCUT strategy. Hence, besides using the MUMCUT strategy to detect all single faults, if we want to further guarantee detecting all double faults, we need to use either the MUMCUT-and-Six strategy or the MAX-A-and-SMOTP strategy. The MUMCUT-and-Six strategy requires approximately 787.4 (=874.0 - 86.6) extra test cases per expression, on average. Even though the jump is big, it is still small when compared with the MAX-A-and-SMOTP strategy because it requires approximately 3121.0 (=3207.6 - 86.6) more test cases. Alternatively, the MUMCUT-and-Six strategy requires approximately 10.1 (=874.0/86.6) times the resources required for the MUMCUT strategy, whereas the MAX-A-and-SMOTP strategy requires approximately 37.0 times. Hence, the MUMCUT-and-Six strategy is more cost effective than the MAX-A-and-SMOTP strategy for double fault detection, in general.

- In 38 Random expressions out of 40, the mean size of test sets for the MUMCUT-and-Six strategy is less than that of the MAX-A-and-SMOTP strategy. For R27 and R31, the test sets for the MAX-A-and-SMOTP strategy have smaller size than those of the MUMCUT-and-Six strategy.

### 12.3.4 Summary

<table>
<thead>
<tr>
<th>Expression group</th>
<th>Testing Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU (L01–L20)</td>
<td>8.4%</td>
</tr>
<tr>
<td>TCAS (T01–T20)</td>
<td>12.2%</td>
</tr>
<tr>
<td>Random (R01–R40)</td>
<td>2.6%</td>
</tr>
<tr>
<td>Weighed average</td>
<td>6.4%</td>
</tr>
</tbody>
</table>

In summary, we have the following findings in the empirical studies.

1. When a Boolean expression is in canonical IDNF (e.g. T08 and T09), no test cases can satisfy the six newly proposed strategies to supplement the MUMCUT strategy for double fault detection. Under such circumstances, the MUMCUT strategy can detect all double faults.
2. When every term of a Boolean expression has just one literal (e.g. L18), there are no test cases satisfying the SMFP, SPMFP and SMUTP strategies.

3. When every term of a Boolean expression has at most 1 missing literal (e.g. L01, T02 and T20), there are no test cases satisfying the PMNFP and SPMFP strategies.

4. In general, the size of test sets satisfying various strategies under study can be arranged in the order of MUMCUT << MUMCUT-and-Six << MAX-A < MAX-B < MAX-A-and-SMOTP. Hence, for double fault detection, the MUMCUT-and-Six strategy is more cost effective than the MAX-A-and-SMOTP strategy.

5. Last, but not least, Table 12.7 summarizes the mean size of test sets, as a percentage of the corresponding exhaustive test set, required by each strategy. The combined weighed average of the three groups still indicates that the MUMCUT-and-Six strategy uses less resources (approximately 30.5%) than the MAX-A-and-SMOTP strategy (approximately 48.4%) for double fault detection.

Finally, the aim of the empirical studies is to further complement our insights into these test case selection strategies as derived in previous chapters. For example, Points 1–3 above give us further insights into these test case selection strategies. As always, the empirical studies have certain limitations and threats to the validity of the results. For example, one limitation is the choice of the Boolean expressions. Forty Boolean expressions are extracted from real avionic software such as LRU and TCAS, hence reflecting some real-life aspects of software specifications. Another 40 Boolean expressions are randomly generated by other researchers used in their empirical study [60]. We did not generate any Boolean expressions for the empirical studies because this may induce further unnecessary bias towards the subjects of the empirical studies. On the other hand, related to the validity of the results, it may be inappropriate for the readers to generalize the results and observations (e.g. Points 4 and 5 above) to other situations as the actual results may vary from expressions to expressions.
Chapter 13

Conclusions

13.1 Summary

This study focused on the detection of double faults that may occur within Boolean expressions. A fault-based testing approach was used to model double fault classes, as the combinations of two single fault classes.

Since there are altogether 81 different ways to form a double fault based on the nine single fault classes related to terms and literals in a Boolean expression considered in this study, we further divided them into three categories:

1. Double faults related to terms only;
2. Double faults related to literals only; and
3. Double faults related to a term and a literal.

As the order of the occurrences of two faults in a Boolean expression may result in different faulty expressions, we further studied double faults from two groups, namely double faults with and without ordering. The former refers to the situation that the ordering of occurrences of the two single faults may result in non-equivalent faults, while the latter refers to the situation that the ordering of occurrences will not affect the final faulty expression.

For the first category (Chapter 4), there are 15 types of double fault without ordering resulting in 27 distinct non-equivalent faulty expressions. There are 25 types of double fault with ordering resulting in 53 faulty expressions.

For the second category (Chapter 6), there are 10 types of double fault without ordering resulting in 19 distinct non-equivalent faulty expressions. There are 16 types of double fault with ordering resulting in 33 possible faulty expressions.

For the third category (Chapter 8), there are 20 types of fault classes of double faults without ordering resulting in 36 different faulty expressions. There are 40 types of fault classes of double faults with ordering resulting in 78 faulty expressions.

After comparing expressions from these two groups of every category, we found that some of them are equivalent to each other. Once the equivalence was con-
sidered, there are a total of 92 (=31+19+42) different double-fault expressions, in which 31 double-fault expressions are from the first category, 19 and 42 double-fault expressions are from the second and third categories.

In this thesis, the detection conditions of all 92 double-fault expressions have been proven in Chapters 5, 7 and 9. Instead of simply presenting them in Boolean format [25, 26], the detection conditions were presented as conditions satisfied by test cases in $B^n$. Such categorization based on certain properties of test sets helps in identifying and developing test case selection strategy in this study.

Further study showed that test case selection strategies aimed to detect single faults, such as BASIC, MUMCUT, MAX-A and MAX-B strategies, cannot guarantee to detect all double faults. After a thorough analysis of these detection conditions, it was revealed that only some double-fault expressions can be detected by these test case selection strategies.

After comparing two approaches of developing new test case selection strategies for double fault, proposing new ones from scratch without referring existing strategies, and proposing new ones to supplement existing test case selection strategies, we decided to use the second approach. There were three reasons for choosing the second approach:

(1) It can guarantee to detect both single and double faults;

(2) Some existing strategies can guarantee to detect certain double faults within Boolean expressions. It would be simpler to develop new strategies that focus only on the undetected double faults; and

(3) It is more practical, due to time and resource limitation, testing practitioners may opt to use existing test case selection strategies to assure the program is single-fault free. If there is still time for further testing, they may opt to use supplementary test case selection strategies to detect double faults.

As a result, six new test case selection strategies were developed to supplement existing test case selection strategies to detect the remaining undetected double faults.

Finally, experiments were performed to analyse the cost of these newly proposed test case selection strategies in terms of numbers of test cases. For 80 Boolean expressions studied in the experiments, the results showed that all these six strategies together with the MUMCUT strategy require on average, 30.5% of the entire input domain which is much more cost-effective than the MAX-A-and-SMOTP, MAX-A and MAX-B strategies which requires 48.4%, 43.7% and 46.6% of the entire input domain.
13.2 Achievements

This thesis presents an investigation of double faults within Boolean expressions. Boolean expressions are important and fundamental in software engineering. They can be used to not only model the program specification [3, 2, 34, 45], but also represent the behaviour of the software system [11].

This work extended the previous works [9, 26, 33, 53, 56] that focused on single fault to double faults, a special instance of multiple faults. We used fault-based approach to model double fault classes, as the combinations of two single fault classes. Instead of assuming two single faults are independent to each other, our study also addresses the interaction between two faults.

Rather than studying fault coupling [22, 23, 41, 42], we investigated the detection conditions of double faults. Since existing test case selection strategies [56, 10] cannot guarantee to detect double faults, test case selection strategies were developed based on proved detection conditions. This thesis proposed a solution that guarantees the detection of studied double faults. As the test cases are selected based on the specification, the solution can guarantee to reveal the failures in any implementation, provided those failures are caused by the studied double faults.

The following lists the achievement of this thesis:

1. Analysis of double faults with and without ordering within Boolean expressions (Chapters 4, 6 and 8). Study from two such groups helps to gain a better understanding of how these faults interact with each other, and hence, results in better fault detection. After comparing the double-fault expressions from these two groups, there are a total 92 double-fault expressions.

2. Analysis of detection conditions of double faults expressions (Chapters 5, 7 and 9). The detection conditions of all possible double faults within Boolean expressions were achieved from a theoretical perspective. The detection conditions help to not only develop test case selection strategies [9], but also to build fault classes hierarchies [26, 33]. Moreover, the detection conditions of double fault classes were used to analyse the relationship between some single and double fault classes [29].

3. Theoretical analysis on some existing test case selection strategies in detecting double faults (Chapter 10). Double-fault expressions that can be detected by these strategies were identified.

4. Proposal of six new test case selection strategies to supplement the MUMCUT strategy for double fault detection (Chapter 11). Even though these test case selection strategies were developed from specification-based perspective, they
can also be used on white-box testing, such as condition coverage and decision coverage. This is because the logical decisions in a program can always be transformed to an equivalent expression in IDNF [59]. The experimental study showed that all these six strategy together with the MUMCUT strategy require on average, 30.5% of the entire input domain.

13.3 Future Work

Several issues need to be considered in the future. Although we have proven that the six newly proposed test case selection strategies can guarantee the detection of double faults within Boolean expressions, empirical studies described in this thesis showed that the average size of test cases required is 30.5% of the entire input domain. The test sets satisfying the MUMCUT-and-Six strategy are actually the union of the test sets satisfying each of these 7 strategies.

In [60], empirical results show that different approaches, such as the greedy-union and incremental-union, can be used to achieve a smaller sized test set for the MUMCUT test case selection strategy. Similarly, we can also apply these approaches for the MUMCUT-and-Six strategy, which guarantees the detection of all double faults, in order to generate smaller sized test sets. Further work may include empirical studies (1) on how to integrate these seven individual test case selection strategies for achieving small sized test sets, and (2) on how the ordering of these seven strategies can affect the size of the resulting test sets.

In [32, 29, 33], fault class hierarchy has been studied as to use minimal test cases to detect most of the fault classes. Similarly, a hierarchy may exist between different double fault classes. Hence, such work may also help to develop test case selection strategies aimed at smaller sized test sets.
Appendix A

Works Arising from the Study

This appendix lists all works derived from this research project in the interest of full disclosure.

A.1 Peer-Reviewed Publications

This section lists Peer-Reviewed Publications derived from this research project which can be found in Bibliography section.


This summarizes the major contents in Chapters 4 and 5.


This summarizes the major contents in Chapters 6 and 7.


This summarizes the major contents in Chapters 8 and 9.

A.2 Unpublished Technical Reports

This section lists unpublished technical reports derived from this research project.


Bibliography


