Ternary and Quaternary Decision Diagrams

Haidar Ali Massoud — Dept. Of Comp. & Inf. Eng. Beirut Arab University, Beirut, Lebanon

El Moussaoui Ali Ahmad — Department of Information System, Sahel General Hospital, Beirut, Lebanon

Haraty Ramzi. — Division of Comp. Science & Mathematics Lebanese American University, Beirut, Lebanon

Shirahama Hiroyuki — Department of Technologies, Faculty of Education Ehime University, Matsuyama, Japan

Sakomoto Masahiro — Dept. of Comp. science, Eng. Fact. Saitama Institute of Technology, Saitama, Japan

Abstract – This paper develops a new method of defining, analyzing, and implementing the ternary functions by ternary decision diagrams (TDDs), also in this paper a novel Quaternary Decision Diagrams (QDDs) for quaternary logic functions has been proposed. Decision Diagrams (DD's) are a data structure convenient for representation logic functions. Ternary and Quaternary Decision Diagrams (TDD's, QDD's) are graph representation of their functions and they are a natural extension of the binary decision diagrams (BDD's) to the ternary- and quaternary-valued logic cases. Thus, the diagrams enable us to evaluate Multiple-Valued logic functions.

Keywords – Switching function, Graph theory, Decision diagram, Multi-valued logic, Galois theory

1. Introduction

Various methods exist to represent Boolean functions. A truth table is the most straightforward method. Another methods are the sum-of-products (SOP), and Decision Diagrams (DDs). Decision diagrams are a data structure which is widely used for representation purpose that leads to efficient manipulation and implementation. Binary decision diagram (BDD) are commonly used in binary logic synthesis, since they can represent complex functions with many variables [1][2]. Recently, ternary decision diagrams (TDDs) had caught some attention among decision diagram researchers, it provides an alternative representation technique and the potential has yet to be explored in binary and ternary logic functions [3][4]. Ternary logic functions are represented by a rooted, directed, and three-branched three-terminal graph [5][6][7].

While the QDDs are still a newly explored in this paper for representation of logic functions . QDDs are similar to BDDs and TDDs except that each non-terminal node has four terminal nodes.

2. Decision Diagram

DDs are an alternative representation of logic function. TDDs are generalization of BDD derived by allowing third edge. But TDDs are more complex than BDDs and so is QDDs. In general, the complexity of BDD is $O(2^n/n)$. Where the complexity of TDD is $O(3^n/n)$ and the complexity of QDD is $O(4^n/n)$. The Complexity of TDD is greater than BDDs complexity because each type of TDDs has some redundancy. But, theoretical bound for size of TDDs is considerably greater than practical size of TDDs [7][8].

2.1 Ternary Decision Diagram (TDD)

The TDD has been developed to be an alternative representation of ternary logic functions. Let $f = x^0 f_0 \vee x^1 f_1 \vee x^2 f_2$ be the Shanon expansion of an arbitrary three-valued function. Where, $f : T^n \rightarrow T$, $T = \{0, 1, 2\}$. Then the sub-graphs of the TDDs represents f_0, f_1 and f_2 as shown in fig (1).

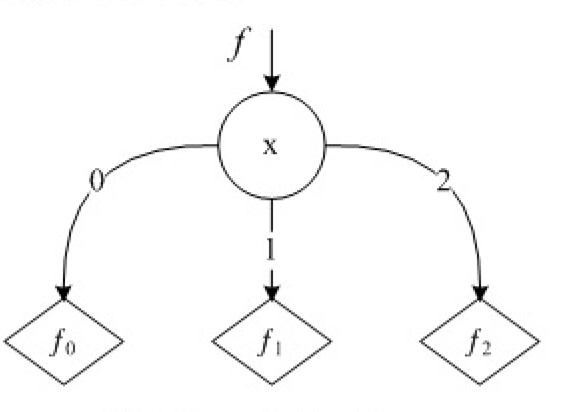

Fig (1). Ternary Decision Diagram

2.2 Quaternary Decision Diagram (QDD)

The QDD is an extension of the BDD and TDD to four-valued case. It is a special case of a Multiple-Valued Decision Diagram (MVDD). The QDD have many manipulation techniques. QDD is a representation of quaternary functions so it must have 4 terminal nodes (0-, 1-, 2- and, 3-terminal nodes), and four outgoing edges (0-, 1-, 2- and, 3-edge) as shown in fig (2). Where $f: T^n \rightarrow T$, $T = \{0, 1, 2, 3\}$.

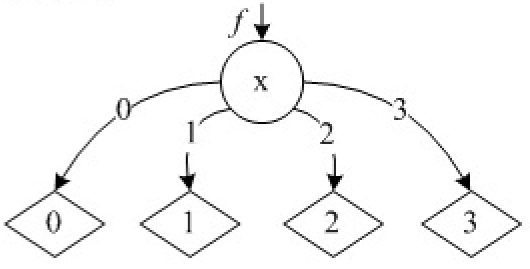

Fig (2). Quaternary Decision Diagram

For solving the Unknown problem we could use the QDD as shown in fig (3) for $f(x) = \{0, 1, 2, 3\}$.

$$f(x) = \begin{cases} 0 & \text{: If x is false.} \\ 1 & \text{: if x is true.} \\ 2 & \text{: if x is not true and not false (third value).} \\ u & \text{: if x is perhapse true and perhapse false(unknown)} \end{cases}$$

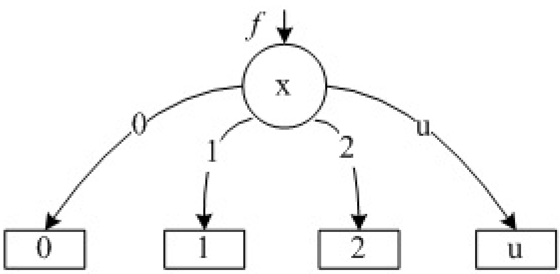

Fig (3). QDD with unknown

2.3 Minimization Rules

This paper develops Ordered Quaternary Decision Diagram (OQDD) and two new minimization rules. Called Reduction rules for Ordered Quaternary Decision Diagram (ROQDD). These rules save time and space to have duplicate QDDs by only coping a pointer to the root nodes. And an equivalence checking can be performed immediately by just looking at the root node.

2.3.1 Ordered Quaternary Decision Diagram (OQDD)

An Ordered QDD (OQDD) is a QDD where input variables appear in a fixed order in all paths of the graph and no variable appears more than once in a path.

2.3.2 Reduced Ordered Quaternary Decision Diagram (ROQDD)

a) Jump and node elimination

Suppose a node with several outgoing edges pointing to the same node, then the jump and node elimination rule as shown in fig (4) .

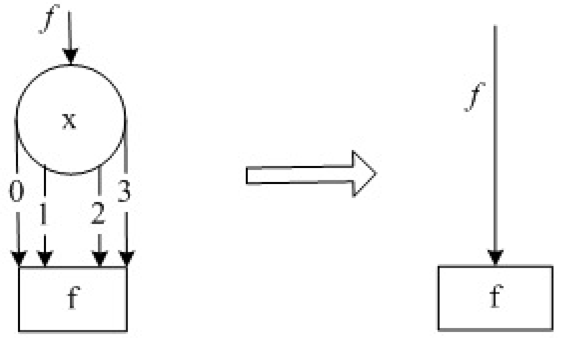

Fig (4). Node elimination and jump

b) Shared environment

On the other hand, assume nodes and their outgoing edges pointing to same functions in the same order, then merge them in one shared node as shown in fig (5).

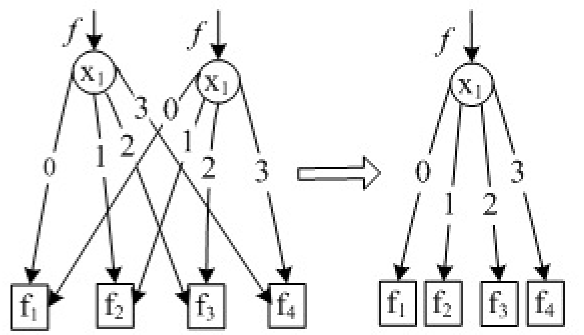

Fig (5). Node Sharing

Consider Maximum function of table (1) where the complete QDD tree is given as shown in fig (6). By implementing the Minimization rules on the tree of fig (6), then the Reduced Ordered Decision Diagram Tree can be achieved as shown in fig (7), where g = {0, 1} and, k = {0, 1, 2}.

Table (1) Maximum function.

MAX	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

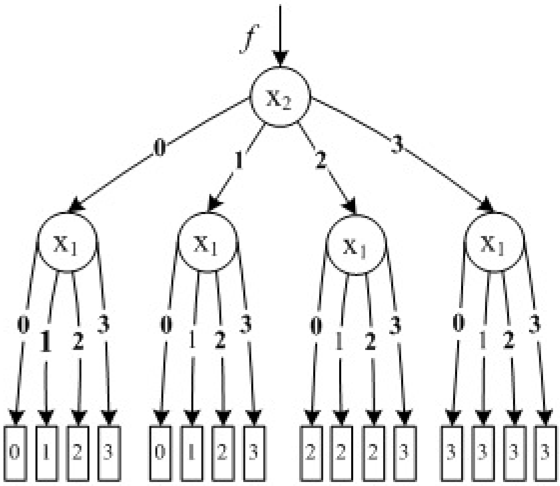

Fig (6). Complete tree

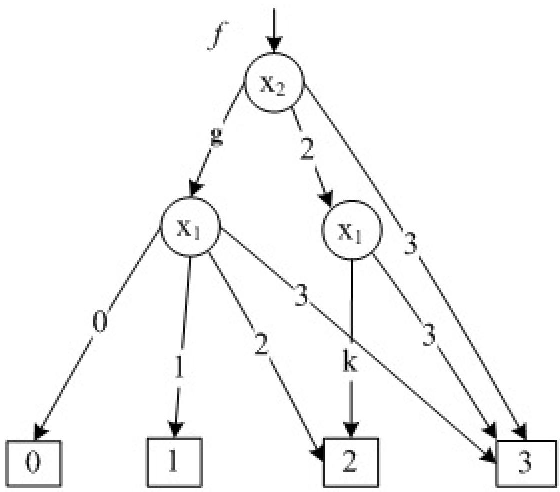

Fig (7). ROTTD.

3. Applications on Ternary and Quaternary Decision Diagram

3.1 Minimum an maximum function

Consider Minimum operation of table (2), then the TDD tree has been designed and minimized as shown in fig (8), where h = {1, 2}.

Table (2). Minimu Operation

Min	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

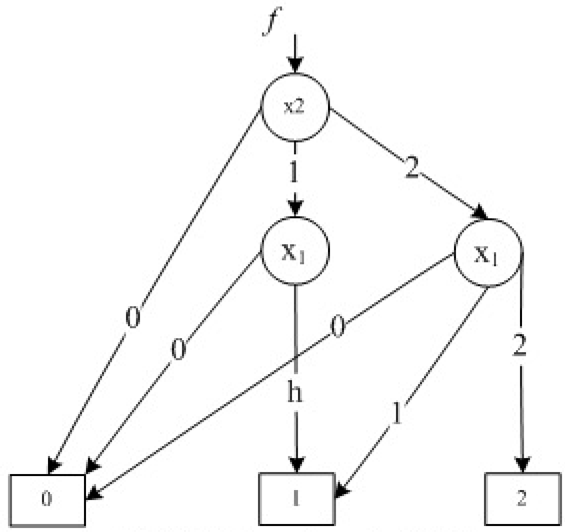

Fig (8). Minimum Operation in TDD

Also the Maximum operation of table (3) is implemented as shown in fig (9), where h = {1, 2}.

Table (3). Maximum Operation

Max	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

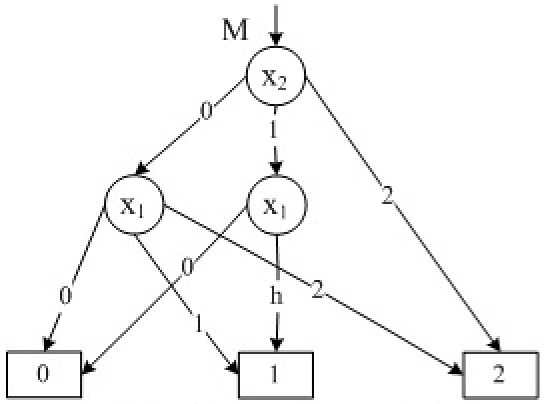

Fig (9). Minimum Operation in TDD.

3.2 Ternary-Half-Adder

Table (4) shows the sum and carry of the ternary-half-adder, and fig (10) shows its TDD implementation, where g = {0, 1} and h = {1, 2}.

Table (4). sum and carry

x1	x2	Sum(x1,x2)	Carry
0	0	0	0
0	1	1	0
0	2	2	0
1	0	1	0
1	1	2	0
1	2	0	1
2	0	2	0
2	1	0	1
2	2	1	1

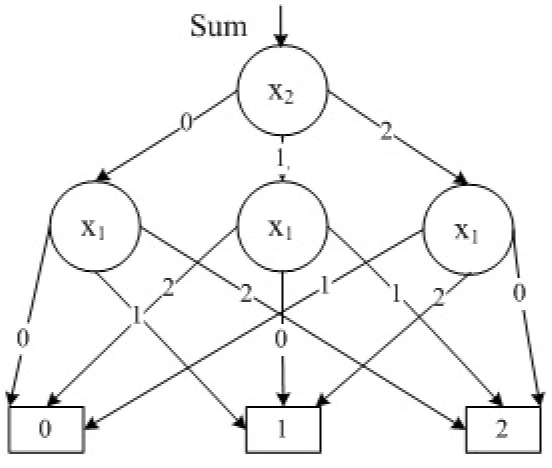

Fig (10)(a). Sum.

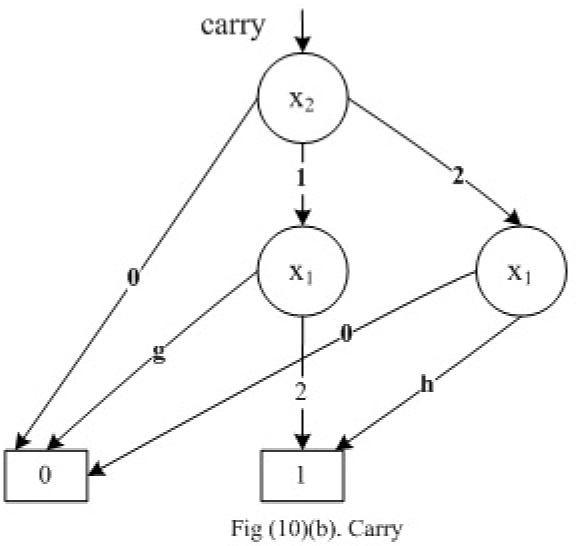

Fig (10)(b). Carry

3.3 Product Operation

The TDD's implementation of Product operation in table (5) is given in fig (11).

E (3) = {0,1,2}, where g = {0, 1}.

Table (5). Product Operation in TDD.

x1	x2	P
0	0	0
0	1	0
0	2	0
1	0	0
1	1	0
1	2	0
2	0	0
2	1	0
2	2	1

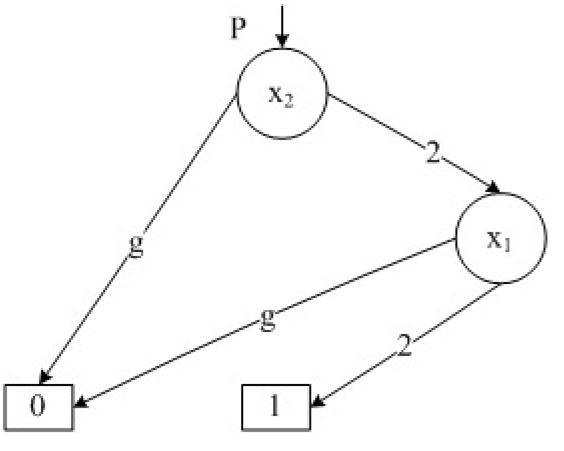

Fig (11). Product Operation in TDD.

4. Ternary function with Unknown (QDD)

The ternary decision diagram has three different values see section (2.1). Consider a ternary function with unknown then the ternary vector input is improved to a quaternary-valued output. F: {0, 1, 2}→{0, 1, 2, u}.

4.1 MIN Operation in QDD

The quaternary decision diagram (QDD) of fig (12) is the implementation of the Quaternary minimum function of table (6), where **u** means unknown, and h = {1, 2}.

Table (6). Minimum Operation

MIN	0	1	2	u
0	0	0	0	0
1	0	1	1	u
2	0	1	2	u
u	0	u	u	u

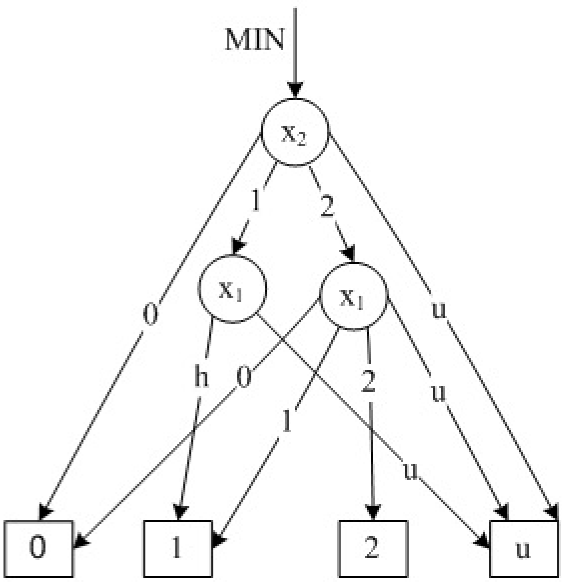

Fig (12). Minimum operation

4.2 Maximum Operation in QDD

The Quaternary Maximum operation of table (7) has been designed with the quaternary decision diagram (QDD) as shown in fig (13). Where **u** means unknown, g = {0, 1}, and j = {0, 1, u}.

Table (7). Maximum operation

MAX	0	1	2	u
0	0	1	2	u
1	1	1	2	u
2	2	2	2	2
u	u	u	2	u

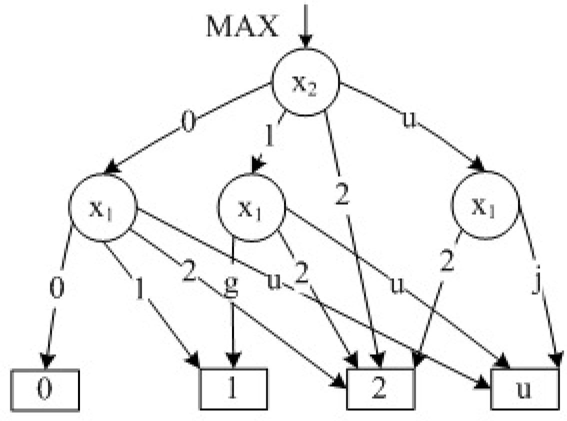

Fig (13). Maximum operation

4.3 Galois finite field.

4.3.1 GF (4) Addition [8]

Tables (8) shows the addition tables for GF (4), and fig (14) shows its QDD.

Table (8) GF(4) Addition

A	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

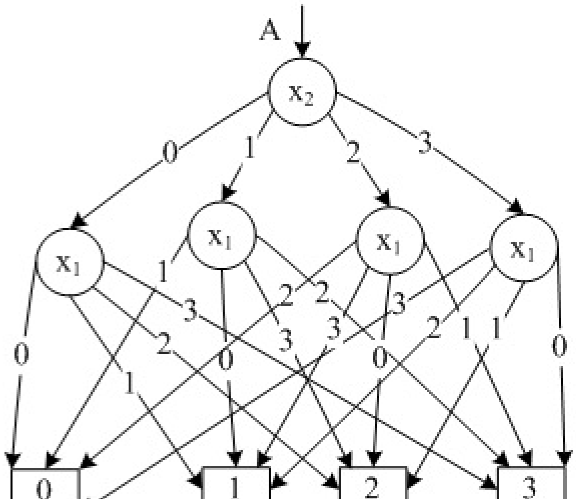

Fig (14). GF(4) Addition in QDD.

4.4. GF (4) Multiplication

Table (9) shows the Multiplication for GF(4), where the QDD's is shown in fig (15).

Table (9). GF(4) Multiplication

M	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

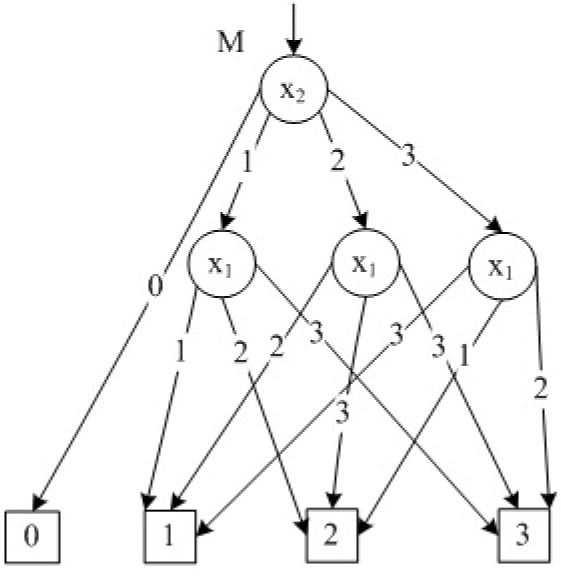

Fig (15). GF(4) Multiplication in QDD.

5. CONCLUSION

In this paper, we have developed a new method of defining, analyzing, and implementing the ternary functions by ternary decision diagrams (QDDs), also a novel Quaternary Decision Diagrams (QDDs) for quaternary logic functions has been proposed. The paper has developed the Ordered Quaternary Decision Diagram (OQDD) and the Reduction rules for Ordered Quaternary Decision Diagram (ROQDD).

The unknown problem has been solved by improving the ternary function with don't care where the function for a ternary vector input is improved to a quaternary-valued output. Also we developed a new method of defining, analyzing, and implementing the quaternary functions by using quaternary decision diagrams.

6. REFERENCES

[1] Y. Iguchi, T. Sasao, M. Matsuura, "On Properties Of Kleene TDDs", Proc.ASP-DAC'97, 473-476, 1997.

[2] T.Sasao and M. Fujita (ed.), Representations of discrete functions, Kluwer Academic Publishers, 1996.

[3] Sim Poh Ching, Mohamad Kadim Suaidi "Ternary Decision Diagram", IEEE Transaction on Computers, 76-79 , 2002 IEEE.

[4] BINARY DECISION DIAGRAMS AND APPLICATIONS FOR VLSI CAD. Shini-ichi Minato, NTT LSI Laboratories, Kanagawa, Japan. 1996 by Kluwer Academic Publishers.

[5] G. Jennings, J, Isaksson P. Lindgren "Ordered Ternary Decision Diagrams and the Multivalued Compiled Simulation of Unmapped Logic", IEEE Transaction on Computers, 1994 IEEE.

[6] T. Sasao, 1997 Japan ,"Ternary Decision Diagrams", IEEE Transaction on Computers, 1997 IEEE.

[7]G. Jennings, 1995, Sweden " Accurate Ternary-Valued Compiled Logic Simulation of Complex Logic Networks by OTDD Composition", IEEE Transaction on Computers, 1995 IEEE.

[8] D. Jankovic ,R. S. Stankovic, "Complexity of Ternary Decision Diagrams for representation of Discrete functions". IEEE Transaction on Computers, 17-20, 2001 IEEE.