2 Variable Reed Muller Binary Decision Diagrams

OH, P. and ALMAINI, A. E. A.
School of Engineering
Napier University
Edinburgh
EH10 5DT
Scotland, UK

Abstract - This paper proposes a variation of Decision Diagram, the 2VRMBDD. It outlines the background for RMBDD expanded with respect to one variable and the new 2VRMBDD when the expansion is with respect to two variables. An example is realised using 2VRMBDD and implemented using Reed-Muller Universal Logic Modules (RM-ULM). The resulting solution is variable order dependent. Lastly the total number of possible solutions is outlined in this paper.

Key-Words: - Reed Muller, Binary Decision Diagrams, Fixed Polarity Reed Muller, Switching Functions, Circuits Minimisation.

1 Introduction

Decision Diagrams (DD) were first introduced by Akers [1] and were then further improved by Bryant [2]. DDs are graphical way of representing switching functions and provide an alternative optimisation technique to the use of Karnaugh maps, truth tables and algebraic representations [3].

There are many advantages in expressing switching functions using Reed-Muller expansions. Exclusive-OR (EX-OR) realisations are easily testable [4]. Secondly, it can often lead to more efficient solutions than the standard Boolean function realisation, in terms of number of gates or number of gate connections [5,6].

Reed-Muller Binary Decision Diagrams (RMBDD) can be used to realise multi-level representations of Generalised Reed-Muller (GRM) expressions [7,8]. Being a counterpart of Binary Decision Diagrams (BDDs), in RMBDD realisation, a function, f, is initially expressed in a tree format, called the Reed-Muller Binary Decision Tree (RMBDT). Merger and Deletion operations are then employed in order to achieve a Reduced Ordered Reed-Muller Binary Decision Diagram (RORMBDD).

2 Principle of RMBDD

A given n-variable Reed-Muller expansion can be expressed as follows:

$$\begin{aligned} f(\dot{x}_{n-1}, \dot{x}_{n-2}, ..., \dot{x}_0) &= \oplus\sum_{i=0}^{2^n-1} b_i P_i \\ &= b_0 \oplus b_1 \dot{x}_0 \oplus \cdots \\ &\oplus b_{2^n-1} \dot{x}_{n-1} \dot{x}_{n-2} \cdots \dot{x}_0 \end{aligned} \quad \textbf{(1)}$$

$\oplus$ denotes modulo-2 addition
P_i denotes a product term
$b_i \in \{0,1\}$
$i = 0,1,...,2^n - 1$
$\dot{x}_i = x_i$ or $\overline{x_i}$

There are 2^n Fixed Polarity RM expansions where each variable can be true or complemented but not both.

Definition: RMBDT is a graphical representation of a RM expression and contains non-terminal nodes, terminal nodes and edges. The terminal nodes, also known as leaves, contain 0's and 1's.

Non-terminal nodes are represented by circles labelled with the splitting variables.

According to Shannon [9] expansion, a RM expression f can be expanded with respect to any variable x_i as shown in Equation (2):

$$\begin{aligned} & f(x_{n-1}, x_{n-2}, ..., x_0) \\ & = \overline{x_i} f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 0, x_{i-1}, ..., x_0) \\ & \quad \oplus x_i f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 1, x_{i-1}, ..., x_0) \end{aligned} \quad \textbf{(2)}$$

If $\dot{x}_i$ is present in its true form throughout the GRM expansion, it is called the Positive Davio or Positive Polarity, where $\dot{x}_i = x_i$ [10]. Substituting $\overline{x_i} = x_i \oplus 1$ in Equation 2 gives:

$$\begin{aligned} f &= f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 0, x_{i-1}, ..., x_0) \oplus \\ & \quad x_i \begin{bmatrix} f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 0, x_{i-1}, ..., x_0) \oplus \\ f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 1, x_{i-1}, ..., x_0) \end{bmatrix} \\ &= f_i^0 \oplus x_i \left[f_i^0 \oplus f_i^1 \right] \end{aligned} \quad \textbf{(3)}$$

If the splitting variable $\dot{x}_i$ is present in its complemented form, where $\dot{x}_i = \overline{x_i}$, throughout the GRM expansion; it is called a Negative Davio or Negative Polarity [10]. By substituting $x_i = \overline{x_i} \oplus 1$ in Equation 2, the following equation is obtained:

$$\begin{aligned} f &= f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 1, x_{i-1}, ..., x_0) \oplus \\ & \quad \overline{x_i} \begin{bmatrix} f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 0, x_{i-1}, ..., x_0) \oplus \\ f(x_{n-1}, x_{n-2}, ..., x_{i+1}, 1, x_{i-1}, ..., x_0) \end{bmatrix} \\ &= f_i^1 \oplus \overline{x_i} \left[f_i^0 \oplus f_i^1 \right] \end{aligned} \quad \textbf{(4)}$$

The above is illustrated in Figure 1, where a RM expansion is expanded using Shannon equation with respect to any splitting variable $\dot{x}_i$, as indicated in the non-terminal node. This is repeated until the sub-functions are reduced to logical ‘1’ and ‘0’. X in Figure 1 is equal to 0 if x_i acts as the splitting variable but is equal to 1 if $\overline{x_i}$ is used.

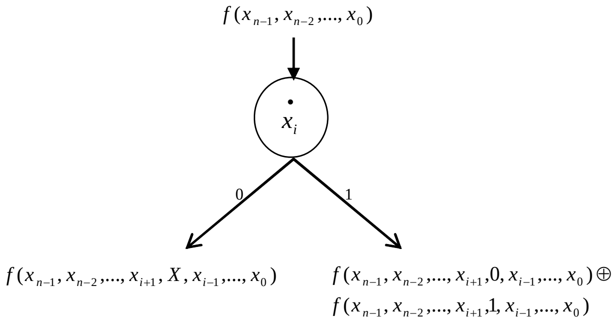


Figure 1 Reed-Muller Function Realised into Two Branches

Once a RMBDT is drawn, it is operated upon using the Merger and Deletion operators in order to obtain a Reduced RMBDD. The RMBDD is said to be Reduced if and only if both the operations are not applicable to that RMBDD. The Reduced RMBDD is Ordered if the same variable(s) is (are) used at each level, forming the RORMBDD.

In Fixed Polarity RM expression realisation, it means that a variable has to exist either in Positive or Negative Polarity. Therefore, within the same level, the variable(s) used is (are) of the same polarity.

The final RORMBDD obtained can then be directly mapped onto hardware implementation diagram with the use of RM-ULMs.

2.1 Variable Order

As in Standard Boolean BDDs, the size (measured by the number of non-terminal nodes) of a ROBDD is heavily affected by its variable orderings.

In the RM domain, any n-variable GRM expansion has $n!$ possible RORMBDDs, each of which is generated with a different variable ordering.

This property of BDD in both Standard and RM domains has been extensively researched world-wide in the past decades. Techniques that have been developed so far have only managed to produce an ordering of ‘near-best’ solutions, with minimal OBDD size [7,8, 11-14]. Exhaustive search is still the only way to find the minimum solution.

3 2-Variable RMBDD

The advantage of optimisation using BDD is the direct implementation of non-terminal nodes using 2:1 Universal Logic Modules (ULM) having the splitting variable acting as a control input [12]. The equivalent digital multiplexer used for RMBDD implementation is RM-ULM [15]. Given the same function, it can be expanded using two variables at each level instead and these two variables can be used as the control inputs of a 4:1 RM-ULM [12,16]. The use of 4-input ULM modules requires less levels and modules for the implementation of a given function at the cost of larger modules.

3.1 Two-Variable Realisation

A given RM function can be expanded with respect to any two variables at a time. The sub-functions will be continuously expanded until they reach logical '0' or '1'. If the number of variables is odd, then a single variable is used first.

3.2 The Procedures

1. Merger
   If two non-terminal nodes have any of their outgoing branches pointing to similar sub-branch, $x_m x_n$, then the said sub-branch can be shared. Figure 2 shows an example of sharing non-terminal node of $x_m x_n$.

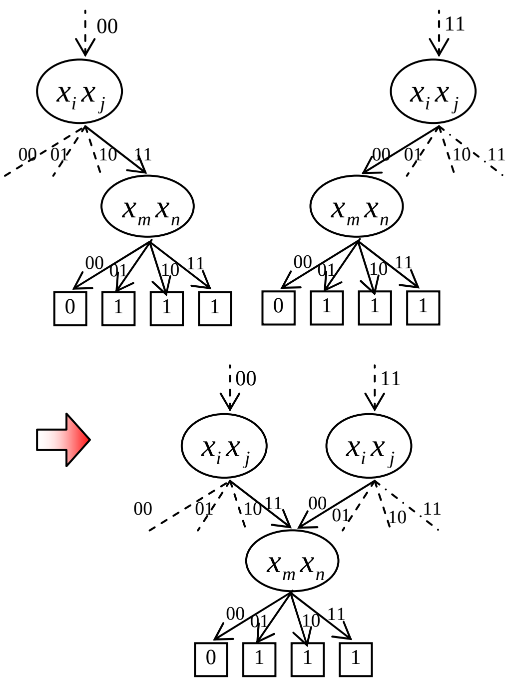


Figure 2 Elimination of a 2-Variable Node

2. Node Deletion
   If any of the $x_i x_j$'s high edges is pointing to a logical '0', it means none of the true form of the said variables exists at all. Therefore the node can be deleted and its incoming edge is directed to its '00' sub-branch, Figure 3(a) and 3(b).

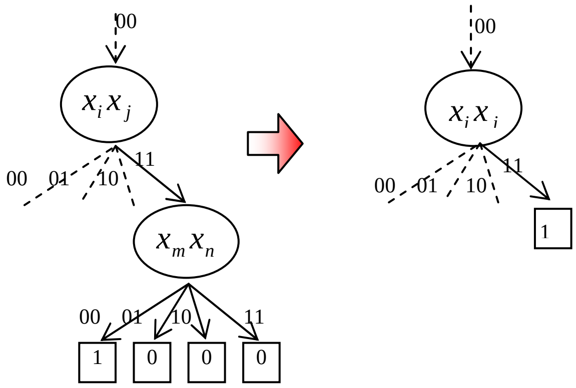


Figure 3 a) Two-Variable Node Deletion

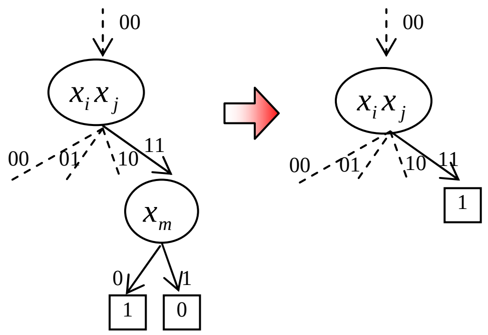


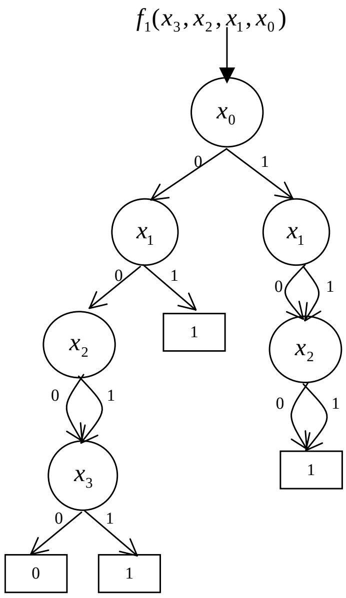
Figure 3 b) Single-Variable Node Deletion

Example:

Consider a four-variable example taken from [7].

$$f_1(x_3, x_2, x_1, x_0) = x_3 \oplus x_3 x_2 \oplus x_1 \oplus x_0 \oplus x_2 x_0 \oplus x_1 x_0 \oplus x_2 x_1 x_0 \quad \textbf{(5)}$$

The RORMBDD solution for f_1 obtained with single variable expansion is shown in Figure 4(a) and the K-map equivalent is shown in Figure 4(b).



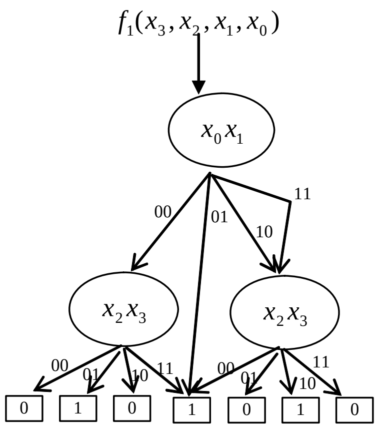
a

x_3x_2 \ x_1x_0	00	01	11	10
00		1	1	1
01		1	1	
11	1			
10	1			

b

Figure 4 a) RORMBDD for f_1. using single variable nodes. b) Karnaugh-map equivalent for f_1.

The 2VRORMBDD solution for f_1 is shown in Figure 5(a) and the K-map equivalent of the 2VRMBDD solution is shown in Figure 5(b).



a

x_3x_2 \ x_1x_0	00	01	11	10
00		1	1	1
01		1	1	
11	1			
10	1			

b

Figure 5 2VRORMBDD for f_1. a) using 2-variable nodes. b) Karnaugh-map equivalent for f_1.

Figures 4(a) and 5(a) result in the RM-ULM implementations given in Figures 6(a) and 6(b) respectively.

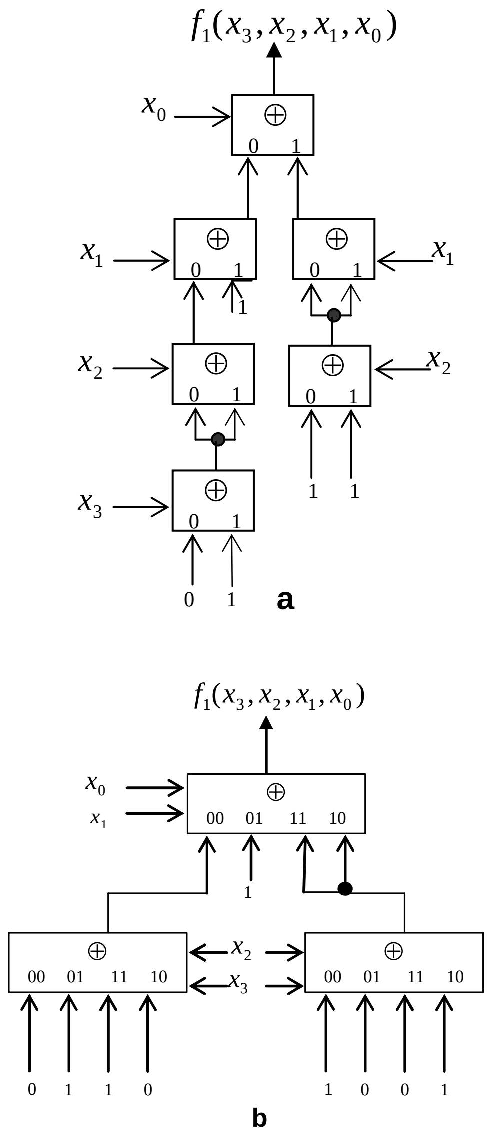


Figure 6 ULM diagrams for f_1 using a) 2:1 ULMs and b) 4:1 ULMs.

The example above shows that the ULM layers and the number of nodes required have been reduced. This leads to the reduction of the number of ULM modules required if a given function is drawn using a 2VRMBDD, compared to single variable RMBDD, with the penalty of using larger modules.

The logic diagrams for the RM-ULMs used in Figures 6 (a) and 6(b) are given in Figure 7.

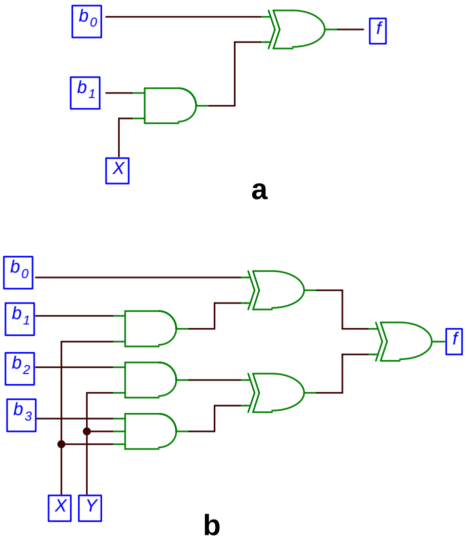


Figure 7 RM-ULM Logic Diagrams for a) 1-Variable RM-ULM and b) 2-Variable RM-ULM.

Different variable orders produce different sets of RORMBDD, as in single variable RMBDD. Figure 8 shows the 2VRORMBDD for function f_1 with x_3x_1, x_2x_0 variable order:

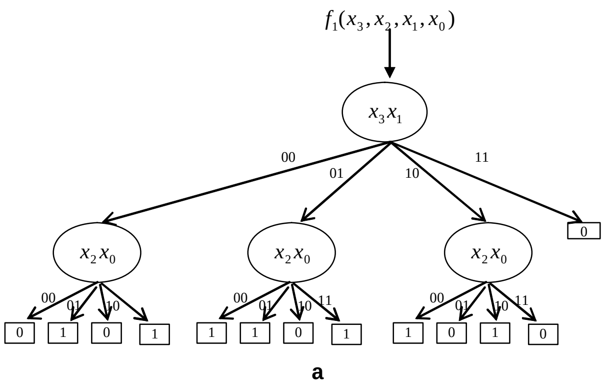


Figure 8 2VRMBDD with different variable ordering for f_1.

4 Single VRMBDD and 2VRMBDD

Each Boolean function f with n variables has 2^n GRM expansions. Each of these GRM expansions has $n!$ variable orders if single variable RMBDD is used. Whereas in 2VRMBDD, the total number of possible variable orders tc_n of each GRM expression is governed by whether n is odd or even number.

If n is odd, ($n \geq 3$), single variable node is used at the top level, then 2 variable node at the second level down and so on. The tc_n is therefore:

$$tc_{no} = n\left[\frac{(n-1)\,!!\,(n-2)\,!!}{2^{\frac{n-1}{2}}}\right] \quad \textbf{(6)}$$

For even values of n, where $n \geq 4$, the given function is realised using 2 variable nodes at each level, the tc_n is found to be:

$$tc_{ne} = \frac{n!!(n-1)!!}{2^{\frac{n}{2}}} \quad \textbf{(7)}$$

Table 1 shows the difference between tc_n and $n!$ for n of up to 20.

Table 1 Difference between the total variable orders for single and 2VRMBDD

Number of variables, n	Total variable orders with 2VRMBDD, tc_n	Total variable orders with single VRMBDD, $n!$
3	3	6
4	6	24
5	30	120
6	90	720
7	630	5040
8	2520	40320
9	22680	362880
10	113400	3628800
11	1.20E+06	39916800
12	7.48E+07	4.79E+08
13	9.70E+07	6.23E+09
14	6.80E+08	8.72E+10
15	1.20E+10	1.31E+12
16	8.20E+10	2.09E+13
17	1.39E+12	3.56E+14
18	1.25E+13	6.4E+15
19	2.38E+14	1.22E+17
20	2.38E+15	2.43E+18

5 Conclusion

The proposed 2VRMBDD is an alternative RMBDD form. It results in the reduction of ULM levels and hence shorter path for the signal to travel. It also reduces the number of possible variable orders for a given function. The variable order within a given level is found to have no effect and simply permutes the outgoing edges of that level. The final outcome (RORMBDD) is still dependent on the variable order used. It is also recommended that existing techniques [18] be used first to find the best polarity and best order which should reduce the search space and possibly the module count.

References:

[1] Akers, S.B., *Binary Decision Diagrams.* IEEE Transactions on Computers, 1978. C-**27**(6): p. 509-516.

[2] Bryant, R.E., *Symbolic Manipulation of Boolean Functions Using a Graphical Representation.* IEEE 22nd Design Automation Conference, Paper 42.1, 1985: p. 688-694.

[3] Bryant, R.E., *Graph-Based Algorithms for Boolean Function Manipulation.* IEEE Transactions on Computers, 1986. **C-35**(8): p. 677-691.

[4] Reddy, S.M., *Easily Testable Realisations for Logic Functions*. IEEE Transactions on Computers, 1972. **C-21**(11): p. 1183-1188.

[5] Green, D., *Modern Logic Design*. Electronic Systems Engineering Series, ed. E.L.D.U.o. Bristol). 1986, Addison-Wesley.

[6] Almaini, A.E.A., *Electronic Logic Systems*. Third Edition ed. 1994, UK: Prentice Hall.

[7] McKenzie, L.M., Xu, L. and Almaini, A. E. A. *Graphical Representation of Generalised Reed-Muller Expansions*. in *Workshop on Application of RM Expansion in Circuits Design*. 1993. Hamburg, Germany.

[8] Xu, L., and McKenzie, L., *Constructing Reed-Muller Universal Logic Module Networks from Reed-Muller Binary Decision Diagrams.* Microelectronics Journal, 1996. **27**: p. 23-36.

[9] Shannon, C.E., *A Symbolic Analysis of Relay and Switching Circuits.* AIEE Transactions, 1938. 57: p. 713-723.

[10] Sasao, T., *Switching Theory For Logic Synthesis*. 1999, USA: Kluwer Academic Publishers. 362.

[11] Drechsler, R., Becker, B. and Gockel, N. *A Genetic Algorithm for Variable Ordering of OBDDs*. in *ACM/IEEE International Workshop on Logic Synthesis*. 1995. Tahoe City.

[12] Almaini, A.E.A., and Woodward, M. E., *An Approach to the Control Variable Selection Problem for Universal Logic Modules.* Digital Processes, 1977. **3**: p. 189-206.

[13] Fujita, M., Matsunaga, Y. and Kakuda, T., *On Variable Ordering of Binary Decision Diagrams for the Application of Multi-Level Logic Synthesis.* IEEE, 1991: p. 50-54.

[14] Ishiura, N., Sawada, H. and Yajima, S., *Minimization of Binary Decision Diagrams Based on Exchanges of Variables.* 1991: p. 472-475.

[15] Xu, L., and Almaini, A. E. A., *Full-Custom Design of Reed-Muller Universal Logic Modules.* International Jounal of Electronics, 1993. **74**(4): p. 605-613.

[16] Almaini, A.E.A., Miller, J. F. and Xu, L., *Automated Synthesis of Digital Multiplexer Networks.* IEE Proceedings of Computers and Digital Techniques, 1992. **139**(4): p. 329-334.

[17] Minato, S., Ishiura, N. and Yajima, S. *Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation*. in *27th ACM/IEEE Design Automation Conference*, 1990: p. 52-57.

[18] Almaini, A.E.A., and Zhuang, N., *Using Genetic Algorithms for the Variable Ordering of Reed-Muller Binary Decision Diagrams.* Microelectronics Journal, 1995. **26**: p. 471-480.