Minimization of OR-XNOR Expressions Using Four New Linking Rules

Khalid. Faraj
Computer Science
Wajdi Institute of Technology
Jerusalem, Mount of Olives. P.O. Box 19014
Palestine

Abstract:-This paper presents a minimization algorithm for fixed polarity dual Reed-Muller expressions (FPDRMs) for completely specified functions. For an \( n \)-variable function there are \( 2^n \) different distinct FPDRMs. Minimum FPDRMs is one with the fewest number of sums. The minimization algorithm for the two-level dual Reed-Muller expressions has been developed, based on a four set of rules. The four new linking rules have been introduced to minimize single output dual Reed-Muller (OR/XNOR) expressions. The new rules are demonstrated using Karnaugh maps for some random functions.

Keywords:-Product of Sum, Dual Reed_Muller expression, Maxterms, Boolean functions.

1. Introduction

The increasing complexity of chip designs and the continuous development of smaller size fabrication processes present new challenges to the existing tools. Future synthesis tools are required to handle millions of gates in a realistic time. Computer-Aided Design (CAD) tools became critical for design and verification of Very Large Scale Integrated (VLSI) digital circuits. Since most of the research has focused on developing algorithms for AND/OR or NAND/NOR circuits. An alternative description of a Boolean function is Reed-Muller expansion [1, 2]. It employs modulo-2 arithmetic and is also unique and canonical for a given Boolean function. The application of XOR/AND and XNOR/OR gates has some advantages over other implementations. In practice, it is well known that many useful circuits such as arithmetic units and parity checkers are heavily XOR oriented and it is more economical to implement their modulo-2 expressions [3-6]. Some authors [7, 8] even conjecture that it is generally more economical to base logic design on modulo-2 expressions rather than conventional OR expressions. Recent progress in circuit technology makes the use of OR/XNOR gates feasible, especially with the development of the new technologies and the arrival of various programmable gate array (FPGA) devices. A major other characteristic of the XNOR logic is the numerous possible canonical representations of switching functions it provides. There are several kinds of OR/XNOR circuits. The FPDRM is one of the canonical OR/ XNOR expressions. FPDRMs are a generalization of Positive Polarity Reed-Muller expressions (PPDRM). A PPDRM is unique for a completely specified function, is an OR/XNOR expressions with only un-complemented (positive) literals. Each variable in the FPDRM can appear either in un-complemented or complemented form but not both. For an \( n \)-variable completely specified Boolean function there are \( 2^n \) distinct FPDRMs. There are techniques for converting from POSs to PPDRM or FPDRM [9-13].

In the domain of combinational logic synthesis, logic minimization plays a vital role in determining the area and performance of the synthesized circuit. Logic minimization based on product of sum (POS) using OR-AND gates is a well studied area. However, minimization based on fixed polarity dual Reed-Muller (FPDRM) using OR-XNOR gates has received relatively lesser attention. The problem of finding a FPDRM of the given Boolean function with the minimum number of cubes has both theoretical and practical value. From the theoretical point of view OR-XNOR is the most general Dual Reed-Muller form. From the practical point of view, XNOR gates and OR-XNOR have numerous applications.
in logic synthesis. In particular, it has been shown that the OR-XNOR representation of Boolean functions is typically more compact than the POS representation. In this paper we present four new rules that can be used to reduce the FPDRM expressions in terms of sums terms and literals number respectively.

2. Preliminaries

In this section, essential definitions and notations are presented. These are important for the understanding of the paper.

A literal is a Boolean variable in positive or negative polarity.

A cube \( C \) is a sum term composed of literals using Boolean OR operation.

Two cubes coincide in variable \( x \) if \( x \) does not appear in the cubes or if \( x \) appears in the cubes in the same polarity.

Two cubes differ in variable \( x \) if they do not coincide in variable \( x \). The distance \( D \) between two cubes is the number of variables, in which the cubes differ.

A dual Reed-Muller is minimum if it contains the minimum number of cubes among all the OR- XNOR of the given Boolean function.

A variable in the cube can have three forms: (1) positive polarity; (2) negative polarity; (3) don’t-care.

Proposition: The XNOR of two cubes that have distance 1 can be represented by a single cube.

The XNOR function is '0' whenever either, but not both, of its two inputs is '0'. Thus, A XNOR B is 0 when only one of either A or B is 0. The truth table of the XNOR (\( \otimes \) ) function is as follows:

**Table 1:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ( \otimes ) B</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The fundamental properties of the XNOR operation are given as follows:

\[
x \otimes x = 1
\]

\[
x \otimes 0 = \overline{x}
\]

\[
x \otimes 1 = x
\]

**The Distributive Law:** This law shows that combinations of OR and XNOR can be written in an expanded form.

\[
(A + (B \otimes C) = (A + B) \otimes (A + C)
\]

**Definition 2.1** An \( n \)-variable Boolean function can be expressed as:

\[
f(x_{n-1}, x_{n-2}, ..., x_0) = \bigoplus_{i=0}^{2^n-1} (d_i + M_i)
\]

Where ‘\( \prod \)’ represents logical products (AND), the ‘+’ is an OR operation and \( i \) is a binary \( n \)-tuple \( i = [i_0, i_1, ..., i_n] \). \( M_i \) is a sum term

\[
M_i = \bigoplus_{k=n-1}^0 x_k = x_{n-1} + x_{n-2} + ... + x_0
\]

and

\[
x_k = \begin{cases} x_k & i_k = 0 \\ \overline{x_k} & i_k = 1 \end{cases}
\]

Alternatively, any Boolean function can be represented by a FPDRM expression as:

\[
f(x_{n-1}, x_{n-2}, ..., x_0) = \bigoplus_{i=0}^{2^n-1} (c_i + S_i)
\]

Where ‘\( \otimes \)’ is XNOR operator, \( [c_2^n, c_2^{n-2}, ..., c_0] \) is the truth vector of the function \( f \), \( c_i \in \{0,1\}, i = [i_0, i_1, ..., i_n] \). \( S_i \) represents a Sum term as

\[
S_i = \bigoplus_{k=n-1}^0 x_k = x_{n-1} + x_{n-2} + ... + x_0
\]

and

\[
x_k = \begin{cases} 0 & i_k = 0 \\ \overline{x_k} & i_k = 1 \end{cases}
\]

**Definition 2.2** A polarity vector \( (p_{n-1}, p_{n-2}, ..., p_0) \) for a FPDRM of an \( n \)-variable Boolean function is a binary vector with \( n \) elements, where \( p_i = 0 \) indicates the variable
In an un-complemented form \((x_i)\), while \(p_i = 1\) indicates the variable \(x_i\) in the complemented form.

**Property 1** For an \(n\)-variable Boolean function, there are \(2^n\) FPDRM expansions corresponding to \(2^n\) different polarity numbers. Each of such expansions is a canonical representation of a completely specified Boolean function.

Maxterms can be identified by expanding a Kronecker sum of \(n\) basis vectors of the form \([0 \ x_i]\) for ‘0’ polarity and \([0 \bar{x}_i]\) for ‘1’ polarity.

The FPDRM can be deduced by substituting the coefficient vector \(c\) as shown in equation (9).

\[
f(x_{n-1},x_{n-2},...,x_0) = \{[0 \ x_{n-1}] *[0 \ x_{n-2}] * ... *[0 \ x_0]\} \odot c \tag{9}
\]

Where ‘\(\odot\)’ represents matrix multiplication based on OR and XNOR [9-13].

For example for a zero polarity \((p = 0)\), with \(n = 2\) the basis vector is generated as follows:

\[
[0 \ x_1]*[0 \ x_0] = [0 + 0 \ 0 + x_0 \ x_1 + 0 \ x_1 + x_0]
\]

**3 Minimization Algorithm**

This section describes a new theory for minimizing a single output completely specified fixed polarity dual Reed-Muller expressions which is based on applying four new rules. The new rules are applied repeatedly to link the sum terms in order to decrease the number of sum terms or literals in the OR/XNOR representation. However, before introducing the four rules, we introduce and prove the following identity [14-15]:

\[
[(a + x_i) \otimes (b + x_i)] = [(a \otimes b) + x_i] \tag{10}
\]

This can be verified as the following: by complementing the left side of equation (10) as follows:

\[
[(a + x_i) \otimes (b + x_i)] = \bar{x_i} \oplus b \bar{x_i} = x_i (a \oplus b)
\]

where ‘\(\oplus\)’ is XOR operator.

Applying the Complement once again of the last expression, the following is obtained

\[
x_i (a \oplus b) = [(a \otimes b) + x_i]
\]

Therefore,

\[
[(a + x_i) \otimes (b + x_i)] = [(a \otimes b) + x_i]
\]

In order to minimize the FPDRM expressions, four new linking rules are thus introduced. The four linking rules are operations which join two cubes, all of which decrease the number of sum terms or literals in the expressions.

**The four linking rules are:**

2. **The combining rule:**

Two cubes can be combined if the distance \((D)\) between them is one as follows:

\[
(A + B) \otimes (A + B) = B \tag{11}
\]

This can be verified as shown: by taking the complement of left side of equation (11)

\[
(A + B) \otimes (A + B) = \overline{AB} \otimes \overline{AB} = \overline{(A \oplus A)} = \overline{B}(1)
\]

Applying the Complement once again for the last expression yields the following:

\[
(A + B) \otimes (\overline{A + B}) = B
\]

2. **The elimination rule:**

\[
(A + B) \otimes B = (A + B) \tag{12}
\]

Can be explained as follows:

Factoring out \(B\) from the equation yields to the following:

\[
(A + B) \otimes B = B + (A \otimes 0)
\]

Applying equation (3) yields the result below.

\[
(A + B) \otimes B = (A + B)
\]

3. **Increase of order:**

This rule is proved as follows:

\[
(A + B) \otimes \overline{A} = (A \oplus B) \otimes 0 \tag{13}
\]

Replacing \(\overline{A}\) by \((A \otimes 0)\) using equation (3) yields the following:

\[
(A + B) \otimes \overline{A} = (A + B) \otimes A \otimes 0
\]

Factoring \(A\) from the first two terms yields the following:

\[
A + (B \otimes 0) \otimes 0
\]

Replacing \((B \otimes 0)\) by \(B\) using equation (3), gives the final result.

\[
(A + B) \otimes \overline{A} = (A + B) \otimes 0
\]

4. **The joining rule:**

\[
(A + B) \otimes (A + B) = A \oplus \overline{B} \tag{14}
\]

Applying equation (3) yields the following:

\[
= [A + B] \otimes [(A \otimes 0) + (B \otimes 0)]
\]

\[
= (A + B) \otimes [(A + B) \otimes (A + 0) \otimes (B + 0) \otimes (0 + 0)]
\]
but, 
\[(A+B) \otimes (A + B) = 1\]
thus, 
\[(A+B) \otimes (\overline{A} + \overline{B}) = 1 \otimes A \otimes B \otimes 0\]
hence, 
\[(A+B) \otimes (\overline{A} + \overline{B}) = (A \otimes 1) \otimes (B \otimes 0)\]
therefore, 
\[(A+B) \otimes (\overline{A} + \overline{B}) = A \otimes \overline{B}\]

The four rules are illustrated using Karnaugh maps in Figures 1 to 4. Their strategy is to apply the rule which give the greatest benefit first, if any combining is possible then combining is applied, if not then elimination, if neither of these then Increase of order, and if none of these then joining. This algorithm is given in Figure 5.
According to the four new rules, we propose the following algorithm to minimize the OR-XNOR expressions:

Minimizing Function(G)
    Do {
        done_any = false;
        If (can do combining)
            Do combining(G); done-any = true;
        Else if (can-do exclusion)
            Do exclusion (G ); done-any = true;
        Else If (can do combining)
            Do combing (G); done-any = true;
        Else If (can do combining)
            Do combing(G); done-any = true;
    } while (done-any)
    Return G

The following example illustrates the rules using Karnaugh map, and compares between POSs and the Dual Reed-Muller expressions by showing that the Dual Reed-Muller for some functions will give more compact expression.

**Example 1**
Consider the following expression:
\[ f(A,B,C,D) = \Pi (2,3,5,6,8,9,12,15) \]
Fig. 6 shows two Karnaugh maps. Fig. 6.a contains loops that result in the minimized POS form,

\[ F = (A + B + \overline{C}) \cdot (A + \overline{C} + D) \cdot (A + B + C + \overline{D}) \cdot (\overline{A} + B + C + D) \cdot (A + C) \cdot (A + B + C) \]

Where the ‘\cdot’ is logical (AND).

The realization of POS form requires 6 OR gates and the overall expression is comprised of 20 literals. The Karnaugh map on Fig. 6.b contains loops that obey the new rules. All 0 values are looped an odd number of times and all 1 values are looped an even number of times.

This results in the following expression:
\[ F = (A + C) \otimes (B + D) \otimes (A + \overline{C}) \]

This only requires 3 OR gates and is comprised of 6 literals and two XNOR gates.

Dual Reed-Muller can be minimized using a Karnaugh map. This is similar to minimization product of sums expressions, but '1' terms can be covered an even number of times, while the '0' terms must be covered an odd number of times. As shown in example one.

**5. Conclusion**
The use of the Dual-Reed-Muller representation to represent and manipulate switching functions in logic synthesis systems is discussed. An algorithm for minimizing fixed dual polarity Reed-Muller representations for single-output completely specified switching (OR/XNOR) functions is presented, in which
four rules are used to determine the minimum expression in terms of sum terms and literal numbers respectively. This algorithm has been tested on some random functions. The four rules have been demonstrated using a Karnaugh maps.

References

Adaptive Filtering of Heart Rate Variation Signal Based on an Efficient Model

S. SEYEDTABAII1 & R. SEYEDTABAII2
1Shahed University
IRAN
stabaii@shahed.ac.ir
2Tehran University of Medical Science
IRAN
roheila2054@yahoo.com

Abstract: -In order to provide accurate heart rate variability indices of sympathetic and parasympathetic activity, the low frequency and high frequency components of an RR heart rate signal must be adequately separated. Adaptive filters aid the separation of the low frequency sympathetic and high frequency parasympathetic components from an ECG R-R interval signal, enabling the attainment of more accurate heart rate variability measures. This paper studies the performance of an efficient (short size) case based model in adaptive filtering of heart rate signal that renders analogous results, in the encountered case, to what higher order conventional FIR model adaptive filter may yield.

Key-Words:- Heart rate variability, adaptive filter, FIR model, first order equalizer.

1- Introduction

Heart rate variability (HRV) is a measure of alterations in heart rate derived by measuring the variation of RR intervals. HRV parameters have been shown to aid assessment of cardiovascular disease [1]. Heart rate is influenced by both sympathetic and parasympathetic (vagal) activity. The influence and balance of both branches of the autonomic nervous system (ANS) have been termed sympathovagal balance and is reflected in the RR interval changes. A low frequency (LF) component of HRV has been proposed as reflecting both sympathetic and parasympathetic effects on the heart and generally occurs in a band between 0.04 Hz and 0.15 Hz. The influence of vagal efferent modulation of the sinoatrial node can be seen in the high-frequency band (HF), loosely defined between 0.15 and 0.4 Hz and known as respiratory sinus arrhythmia (RSA) because it occurs at the respiratory frequency. The magnitude of this high frequency band has been demonstrated to be associated with the extent of cardiac parasympathetic activity in pharmacological autonomic blockade studies [2], respiratory sinus arrhythmia, cardiac vagal tone, and respiration: within and between-individual relations.

The ratio of power in the LF and HF components (LF/HF) has been used to provide an estimate of cardiac sympathovagal balance, although this measure remains indispute [3]. Nevertheless, several studies have indicated that when considered jointly, HF and LF HRV may provide useful information about both sympathetic and parasympathetic influences upon the cardiac cycle [4].

Spectral HRV is a measure of power in various frequency bands. To determine the RSA amplitude over a period of time, frequency domain, time domain and phase domain approaches have been analyzed [5]. Presented in [6] is an adaptive filter, which can separate the LF and HF components and therefore acquires separate spectral analysis measures. In this paper an adaptive filter with a new model structure, just with a few parameters, is introduced which performs analogous to a higher order FIR model adaptive filter in the encountered case.