

# Low power mapping for AND/XOR circuits and its application in searching the best mixed-polarity\*

Wang Pengjun(汪鹏君)<sup>†</sup> and Li Hui(李辉)

Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China

**Abstract:** A low power mapping algorithm for technology independent AND/XOR circuits is proposed. In this algorithm, the average power of the static mixed-polarity Reed–Muller (MPRM) circuits is minimized by generating a two-input gates circuit to optimize the switching active of nodes, and the power and area of MPRM circuits are estimated by using gates from a given library. On the basis of obtaining an optimal power MPRM circuit, the best mixed-polarity is found by combining an exhaustive searching method with polarity conversion algorithms. Our experiments over 18 benchmark circuits show that compared to the power optimization for fixed-polarity Reed–Muller circuits and AND/OR circuits, power saving is up to 44.22% and 60.09%, and area saving is up to 14.13% and 32.72%, respectively.

**Key words:** AND/XOR circuits; low power mapping; MPRM; polarity optimization

**DOI:** 10.1088/1674-4926/32/2/025007

**EEACC:** 1130B

## 1. Introduction

The increasing complexity and high-performance requirements of modern integrated circuits (IC) have naturally led to very high power consumption, which results in the problems of package, thermal and stability in IC. Low power is one of the important objectives of VLSI designs. In particular, it has become the most important restraint exceeding area and performance in portable systems<sup>[1]</sup>. Once low power designs for VLSIs still focus on AND/OR circuits. Research indicates that AND/XOR circuits are more easily testable compared to traditional AND/OR circuits<sup>[2]</sup>, and they also provide a more compact structure in several applications<sup>[3, 4]</sup>, e.g., linear circuits, arithmetic circuits and telecommunication systems. Therefore, there is an urgent need for low power designs for AND/XOR circuits.

Technology mapping involves the optimal implementation of a logic function using gates from a given library, while low power mapping is a problem of mapping a technology independent circuit to a technology specific one with power as optimization metric<sup>[5]</sup>. At the logic design level, switching activity is the principal parameter for power reduction in low power mapping. Low power mapping algorithms try to minimize the power of NAND networks<sup>[6]</sup>, AND networks<sup>[7]</sup>, and XOR networks<sup>[8, 9]</sup>. In low power mapping for AND/XOR circuits, a multi-input AND/XOR gate needs be decomposed into a tree of two-input gates with minimizing switching activity<sup>[5]</sup>. For an  $n$ -input AND/XOR circuit,  $2^n$  fixed-polarity Reed–Muller (FPRM) realizations and  $3^n$  mixed-polarity Reed–Muller (MPRM) realizations are possible<sup>[10]</sup>. Usually, low power mapping for AND/XOR circuits is centered on FPRM realizations<sup>[3, 8]</sup>, but FPRM realizations

are a subset of MPRM realizations. Hence, using MPRM realizations is more likely to minimize power compared to using FPRM realizations in low power mapping for AND/XOR circuits.

In this paper, estimation models of the power and area for AND/XOR circuits are created based on an overall consideration of switching activity and load capacitance at nodes, and a fast low power mapping algorithm for AND/XOR circuits is proposed to optimize the circuit activity by analyzing the dynamic power model of static CMOS circuits and the transition characteristic of signal probabilities. Then, an algorithm of searching the best mixed-polarity is presented, where MPRM realizations under arbitrary polarities are obtained by using polarity conversion algorithms, and the best polarities are found by an exhaustive searching method. Finally, our searching strategy is used to solve several Microelectronics Center of North Carolina (MCNC) and International Symposium on Circuits Systems (ISCAS) benchmark problems, and computational results and comparisons are presented and discussed.

## 2. Estimation models of the power and area for two-input AND/XOR gate circuits

In low power mapping for AND/XOR circuits, we need to decompose a multi-input gate into a network of two-input gates. Figure 1 shows the model for two-input AND/XOR gates, where  $A_{\text{AND}}$  and  $A_{\text{XOR}}$  represent the area cost of AND gates and XOR gates.  $C_{\text{in,A}}$  and  $C_{\text{in,B}}$  are the input capacitance and  $C_{\text{out}}$  is the output capacitance. Thus, for any AND/XOR circuit including  $m$  two-input AND gates and  $n$  two-input XOR gates, its area estimation model is given by

$$E_{\text{area}} = mA_{\text{AND}} + nA_{\text{XOR}}. \quad (1)$$

\* Project supported by the National Natural Science Foundation of China (Nos. 61076032, 60776022), the Postdoctoral Science Foundation of China (No. 20090461355), the Postdoctoral Research Projects of Zhejiang Province, China, and the Natural Science Foundation of Zhejiang Province, China (No. Y1101078).

<sup>†</sup> Corresponding author. Email: wangpengjun@nbu.edu.cn

Received 17 August 2010, revised manuscript received 14 September 2010

© 2011 Chinese Institute of Electronics

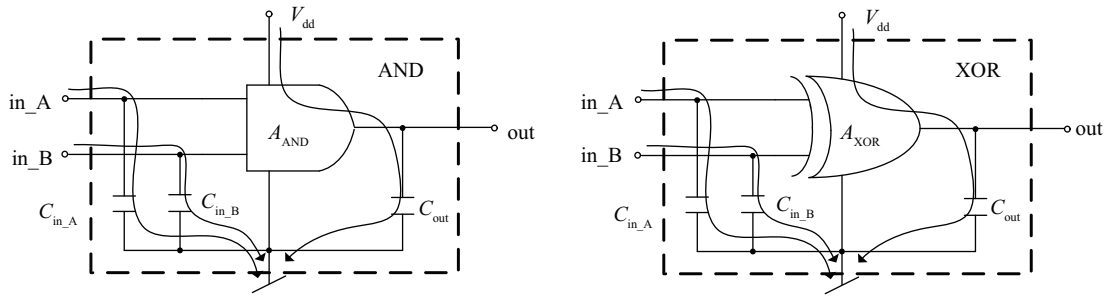


Fig. 1. Gate-level power model for CMOS circuits.

Power consumption consists of dynamic and static components in static CMOS digital circuits. Static losses are related to leakage and become significant in today's nanometer technologies<sup>[12]</sup>. Dynamic components directly depend on the switching frequency, which accounts for about 80% of the total power consumption<sup>[1]</sup>. So, only dynamic power dissipation is considered in this paper. If the wastage of output voltage is ignored, for any AND/XOR circuit holding a node set  $V$ , its power estimation model is

$$E_{\text{pow}} = 0.5V_{\text{dd}}^2 f_{\text{clk}} \sum_{v \in V} \left[ \sum C_{\text{in}}(v) + \sum C_{\text{out}}(v) \right] E_{\text{sw}}(v), \quad (2)$$

where  $V_{\text{dd}}$  is the supply voltage,  $f_{\text{clk}}$  is the clock frequency, and  $\sum C_{\text{in}}(v)$  and  $\sum C_{\text{out}}(v)$  are the input and output capacitance at a node  $v$ .  $E_{\text{sw}}(v)$  is the switching activity at node  $v$ , which represents the switch times of logic value at a node  $v$  in one clock cycle. In logic synthesis for AND/XOR circuits,  $V_{\text{dd}}$  and  $f_{\text{clk}}$  cannot be modified as the fixed technology, and  $\sum C_{\text{in}}(v)$ ,  $\sum C_{\text{out}}(v)$  or  $E_{\text{sw}}(v)$  can be optimized in physics design or technology mapping.

### 3. Low power mapping for AND/XOR circuits

If the signals are random variables in AND/XOR circuits, we define the signal probability of a node  $v$  as the probability of  $v$  being 1, denoted by  $\text{pr}(v)$ , or the probability of  $v$  being 0 is  $1 - \text{pr}(v)$ . The zero delay and spatial temporal independent model is assumed, where gate delays are assumed to be zero and thus signal transitions are ignored due to a glitch, primary inputs are assumed to be uncorrelated and the present input signal value is independent of those in the past<sup>[9]</sup>. In static logic, the switching activity  $E_{\text{sw}}(v)$  of a note  $v$  can be written as<sup>[9]</sup>

$$E_{\text{sw}}(v) = 2\text{pr}(v)(1 - \text{pr}(v)). \quad (3)$$

If the input signal probabilities and a decomposition tree are given, the probability of the internal signal can be computed as follows. For a two-input AND gate  $z = xy$ , let<sup>[7]</sup>

$$\text{pr}(z) = \text{pr}(x)\text{pr}(y). \quad (4)$$

For a two-input XOR gate  $z = x \oplus y$ , let<sup>[8]</sup>

$$\text{pr}(z) = \text{pr}(x) + \text{pr}(y) - 2\text{pr}(x)\text{pr}(y). \quad (5)$$

In low power mapping for static AND/XOR circuits, given an  $n$ -input and  $m$ -output AND/XOR with the signal probabilities of primary inputs  $\text{pr}(x_0), \text{pr}(x_1), \dots, \text{pr}(x_{n-1})$ , we construct a tree  $T = (V, E)$  of two-input AND/XOR gates with

the primary inputs as its leaves and the outputs as its roots such that

$$E_{\text{sw}}(T) = \sum_{v \in V} 2\text{pr}(v)(1 - \text{pr}(v)) \quad (6)$$

is minimized, where  $\text{pr}(v)$  is the signal probability of a note  $v$ .

The analysis of Eq. (3) shows that (1)  $E_{\text{sw}}(z)$  monotonically increases with the increase in  $\text{pr}(z)$  if  $0 < \text{pr}(z) < 0.5$ , (2)  $E_{\text{sw}}(z)$  monotonically decreases with the decrease in  $\text{pr}(z)$  if  $0.5 \leq \text{pr}(z) < 1$ , and (3)  $E_{\text{sw}}(z)$  is no greater than 0.5. Let  $\min\{a, b\}$  equal the lesser of  $a$  and  $b$ , and the switching activity minimization of two-input gate circuits is realized by using Min-Huffman algorithm with  $\min\{\text{pr}(z), 1 - \text{pr}(z)\}$ <sup>[7]</sup>.

It is known that  $\text{pr}(z)$  monotonically increases with the increase in  $\text{pr}(x)$  ( $\text{pr}(y)$ ) and  $\text{pr}(z)$  is no greater than  $\text{pr}(x)$  and  $\text{pr}(y)$  by analyzing Eq. (4). A low power mapping algorithm for the multi-input AND gate is described as follows.

#### Algorithm 1

(1) Sort the input variables of multi-input AND gates in descending order of signal probability; the variables whose signal probability is not greater than 0.5 include into the set  $C_1$ , or the variables whose signal probability is greater than 0.5 include into the set  $C_2$ .

(2) Go to step 3 if the number of elements is no less than 2 in  $C_1$ ; otherwise, insert the elements in  $C_1$  into the side of minimum signal probability in  $C_2$  and go to step 4.

(3) Combine the two variables of minimum signal probability in  $C_1$ , delete these two variables in  $C_1$ , insert the output variable into the side of minimum signal probability in  $C_1$ , and return to step 2.

(4) Go to step 6 if the number of elements is no greater than 2 in  $C_2$ ; otherwise, go to step 5.

(5) Go to step 6 if  $\min\{\text{the product of the signal probabilities of the two variables of minimum signal probability, the result of 1 minus the product of the signal probabilities of the two variables of minimum signal probability}\}$  is less than  $\min\{\text{the product of signal probabilities of the two variables of maximum signal probability, the result of 1 minus the product of signal probabilities of the two variables of maximum signal probability}\}$ ; otherwise, combine the two variables of maximum signal probability, delete these two variables in  $C_2$ , insert the output variable into  $C_2$  in order of signal probability, and return to step 4.

(6) Stop if the number of elements is no greater than 2 in  $C_2$ ; otherwise, go to step 7.

(7) Combine the two variables of minimum signal probability in  $C_2$ , delete these two variables in  $C_2$ , insert the output

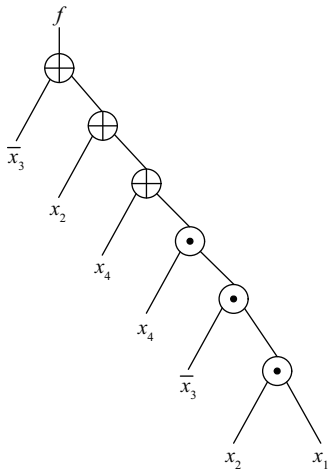


Fig. 2. Decomposition tree of the logic function  $f$ .

variable into the side of minimum signal probability in  $C_2$ , and return to step 6.

Analyzing Eq. (5) obtains the following features: (1)  $\text{pr}(z)$  monotonic increases with the increase of  $\text{pr}(y)$  ( $\text{pr}(x)$ ) and  $\text{pr}(z) > \text{pr}(x)$  ( $\text{pr}(z) > \text{pr}(y)$ ) if  $0 < \text{pr}(x) < 0.5$  ( $0 < \text{pr}(y) < 0.5$ ); (2)  $\text{pr}(z)$  monotonic decreases with the decrease of  $\text{pr}(y)$  ( $\text{pr}(x)$ ) and  $\text{pr}(z) > \text{pr}(x)$  ( $\text{pr}(z) > \text{pr}(y)$ ) if  $0.5 \leq \text{pr}(x) < 1$  ( $0.5 \leq \text{pr}(y) < 1$ ); (3)  $\text{pr}(z) \equiv 0.5$  if  $\text{pr}(x) = 0.5$  ( $\text{pr}(y) = 0.5$ ); (4)  $\text{pr}(z) < 0.5$  if  $0 < \text{pr}(x) < 0.5$  and  $0 < \text{pr}(y) < 0.5$  ( $0.5 < \text{pr}(x) < 1$  and  $0.5 < \text{pr}(y) < 1$ ); and (5)  $\text{pr}(z) > 0.5$  if  $0 < \text{pr}(x) < 0.5$  and  $0.5 < \text{pr}(y) < 1$  ( $0.5 < \text{pr}(x) < 1$  and  $0 < \text{pr}(y) < 0.5$ ). A low power mapping algorithm for a multi-input XOR gate is described as follows.

**Algorithm 2**

(1) Sort the input variables of multi-input XOR gates in descending order of  $\min\{\text{pr}(x_i), 1-\text{pr}(x_i)\}$ . Include these input variables into the set  $C_1$ .

(2) Stop if the number of elements is no greater than 2 in  $C_1$ ; otherwise, go to step 3.

(3) Combine the two variables of minimum  $\min\{\text{pr}(x_i), 1-\text{pr}(x_i)\}$  in  $C_1$ , delete these two variables in  $C_1$ , insert the output variable into  $C_1$  in order of  $\min\{\text{pr}(x_i), 1-\text{pr}(x_i)\}$ , and return to step 2.

In summary, there is the simplified description of the low power mapping algorithm for AND/XOR circuits. For example, we suppose that the input signal probabilities for a logic function  $f = x_2 \oplus \bar{x}_3 \oplus x_4 \oplus x_1x_2\bar{x}_3x_4$  are  $\text{pr}(x_1) = 0.3$ ,  $\text{pr}(x_2) = 0.4$ ,  $\text{pr}(x_3) = 0.4$ , and  $\text{pr}(x_4) = 0.7$ . The decomposition tree constructed by Algorithm 3 is shown in Fig. 2, where the summation of switching activities is 3.6731.

**Algorithm 3**

(1) Multi-input AND gates in AND/XOR circuits are decomposed by using Algorithm 1. The signal probabilities of tree roots and the switch activities of notes are evaluated by Eqs. (4) and (3).

(2) The root nodes of AND trees will be the primary inputs of XOR gates. Multi-input XOR gates in AND/XOR circuits are decomposed by using Algorithm 2. The switch activities of notes are evaluated by Eq. (3).

(3) The load capacitances of nodes are calculated accord-

ing to using gates from a given library and the network obtained by low power mapping. The power and area of the two-input gate circuit are estimated by Eqs. (2) and (1) based on  $V_{dd}$ ,  $f_{clk}$  and the switch activities of notes obtained by steps 1 and 2.

**4. Searching the best mixed-polarity**

An  $n$ -input AND/XOR circuit has  $3^n$  MPRM realizations identified by  $3^n$  different polarities, and the searching space is  $3^n$ . If an AND/XOR circuit is implemented with different polarities, the power and area will be consumed differently. Hence, searching the best mixed-polarity is to find the best mixed-polarity so that the cost of circuit is minimal by using someone searching method according to the cost function structured in anticipation.

**4.1. MPRM expressions and polarity conversions**

Any  $n$ -input and  $m$ -output MPRM realization can be represented in MPRM expression as<sup>[4]</sup>

$$f_k^P(x_{n-1}, x_{n-2}, \dots, x_0) = \bigoplus \sum_{i=0}^{2^n-1} b_{i,k} \pi_i, \quad (7)$$

where  $\bigoplus \sum$  is XOR operator,  $b_{i,k} \in \{0, 1\}$  is a MRPM coefficient, the polarity  $P$  can be expressed in a ternary form as  $p_{n-1}p_{n-2} \dots p_0$ , the product term  $\pi_i$  can be expressed as  $\dot{x}_{n-1}\dot{x}_{n-2} \dots \dot{x}_0$  and

$$\dot{x}_c = \begin{cases} 1, & i_c = 0, p_c = 0, \\ x_c, & i_c = 1, p_c = 0, \end{cases} \quad (8a)$$

$$\dot{x}_c = \begin{cases} 1, & i_c = 0, p_c = 1, \\ \bar{x}_c, & i_c = 1, p_c = 1, \end{cases} \quad (8b)$$

$$\dot{x}_c = \begin{cases} \bar{x}_c, & i_c = 0, p_c = 2, \\ x_c, & i_c = 1, p_c = 2. \end{cases} \quad (8c)$$

A MPRM expression is derived from a given Boolean expression or MPRM expression by applying the polarity conversions, such as a coefficient-map<sup>[13]</sup> or tabular techniques<sup>[14]</sup>. Tabular methods can be used for any number of variables and are suitable for both hand computation and computer solution. Combining the tabular with the expression way of multi-output logic functions, let the binary  $\langle i_{n-1}i_{n-2} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle$  denote a minterm or product term in tabular, where  $i_{n-1}i_{n-2} \dots i_0$  is the subscript of the minterm or product term, and  $o_k$  should contain either a 0 or a 1 to indicate the state of  $i_{n-1}i_{n-2} \dots i_0$  in each  $f_k$ . Therefore, MPRM expressions are obtained by using tabular techniques.

Based on the expression method of both minterm expressions and MPRM expressions in tabular, the generation of the new rows and the deletion of the equivalent rows are executed, and a mix-polarity conversion algorithm for the conversion from a given minterm expression to a polarity- $P$  MPRM expression can be obtained as follows.

**Algorithm 4**

(1) Read in an  $n$ -variable and  $m$ -output minterm expression, and list the minterms in binary form as  $\{\langle i_{n-1}i_{n-2} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle \mid 1 \leq l \leq t\}$ , let  $l = 1, c = n - 1$ .

(2) Go to step 5 if  $p_c = 2$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}1i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $p_c = 0$  and  $i_c = 0$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}0i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $p_c = 1$  and  $i_c = 1$ ; let  $l = l + 1$ , if  $l \leq t$ , then repeat step 2.

(3) Compare all newly generated rows with the original rows. If a pair of rows have the same  $i_{n-1}i_{n-2} \dots i_0$ , then the bitwise XOR operator is used for  $o_{m-1}o_{m-2} \dots o_0$  in these two rows. Delete these two rows if the result is 0; otherwise,  $o_{m-1}o_{m-2} \dots o_0$  of this original row is covered by the result and this newly generated row is deleted;

(4) Add the remaining newly generated rows into the tail of the original rows queue, reverse  $i_c$  of  $\langle i_{n-1}i_{n-2} \dots i_c \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle$  in table if  $p_c = 1$ .

(5) Let  $l = 1, c = c - 1$ , if  $c \geq 0$ , then return to step 2; otherwise, the table represents the polarity-P MPRM expression.

Tabular rows are dealt with by using AND/XOR operators under different polarity one by one. Mixed-polarity conversion algorithm for conversion forms a given polarity- $P_S$  MPRM expression to a polarity- $P_D$  MPRM expression, is described as follows:

**Algorithm 5**

(1) Read in an  $n$ -variables and  $m$ -output polarity- $P_S$  MPRM expression, and list product terms in binary form as  $\{\langle i_{n-1}i_{n-2} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l | 1 \leq l \leq t\}$ , let  $Q = P_S \oplus P_D = (q_{n-1}q_{n-2} \dots q_0)_3$ , where  $q_c = (p_c)_S \oplus (p_c)_D, l = 1, c = n - 1$ .

(2) Go to step 5 if  $q_c = 0$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}0i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $q_c = 1$  and  $i_c = 1$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}1i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $q_c = 2$  and  $i_c = 0$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}0i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $q_c = 3$  and  $(q_c)_S = 2$  and  $i_c = 1$ ; otherwise, generate a new row  $\langle i_{n-1}i_{n-2} \dots i_{c+1}1i_{c-1} \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle_l'$  if  $q_c = 3$  and  $(p_c)_S = 1$  and  $i_c = 0$ , let  $l = l + 1$ , if  $l \leq t$ , then repeat step 2.

(3) Compare all newly generated rows with the original rows. If a pair of rows have the same  $i_{n-1}i_{n-2} \dots i_0$ , then the bitwise XOR operator is used for  $o_{m-1}o_{m-2} \dots o_0$  in these two rows. Delete this two rows if the result is 0; otherwise,  $o_{m-1}o_{m-2} \dots o_0$  of this original row is covered by the result and this newly generated row is deleted.

(4) Add the remaining newly generated rows into the tail of the original rows queue. Reverse  $i_c$  of  $\langle i_{n-1}i_{n-2} \dots i_c \dots i_0, o_{m-1}o_{m-2} \dots o_0 \rangle$  in table if  $q_c = 3$ .

(5) Let  $l = 1, c = c - 1$ , if  $c \geq 0$ , then return to step 2; otherwise, the table represents the polarity- $P_D$  MPRM expression.

**4.2. An algorithm of searching the best mixed-polarity**

The cost function is an important measure of the circuit performance. Usually, the lower the cost value, the higher the performance. To get the integrated optimization of the area and power, the cost function  $Cost(P)$  of an AND/XOR circuit under a polarity  $P$  is constructed as

$$Cost(P) = E_{area}(P)w/E_{area\_max} + E_{pow}(P)(1 - w)/E_{pow\_max}, \tag{9}$$

Table 1. Parameters of gates in mcnc.genlib.

Gate	Area (unit)	$C_{in\_A}$ (nF)	$C_{in\_B}$ (nF)	$C_{out}$ (nF)
AND	3	2	2	1
XOR	5	4	4	2

where  $E_{area}(P)$  and  $E_{pow}(P)$  can be evaluated by Eqs. (1) and (2),  $E_{area\_max}$  and  $E_{pow\_max}$  are the maximum area and power obtained by an exhaustive searching process, and  $w$  is the weighting given to components area and power,  $0 \leq w \leq 1$ , as input by the user. A lower value of  $w$  stresses on power minimization, or a higher value of  $w$  tends to area optimization.

The best mixed-polarity of AND/XOR circuits can be obtained by using an exhaustive searching method. In the searching process, the first MPRM expression is converted from a given Boolean expression by using Algorithm 4, and other MPRM expressions are converted from a MPRM expression under neighbor polarity by using Algorithm 5. A Gray code of single-bit-difference ternary strings is used for traversing all polarities in the shortest computation time. Based on the analysis of Algorithm 5, we know that the polarity  $0 \leftrightarrow 1$  conversion is the most effective, the polarity  $0 \leftrightarrow 2$  conversion is the second, and the polarity  $1 \leftrightarrow 2$  conversion is the least. So, the evolution of Gray code is  $1 \rightarrow 0 \rightarrow 2 \rightarrow 2 \rightarrow 0 \rightarrow 1$ . For example, the route for a 2-variable Gray code is  $11 \rightarrow 10 \rightarrow 12 \rightarrow 02 \rightarrow 00 \rightarrow 01 \rightarrow 21 \rightarrow 20 \rightarrow 22$ . An algorithm of searching the mixed-polarity can be characterized as follows:

**Algorithm 6**

(1) The MPRM expression under polarity  $(11 \dots 1)_3$  is converted from a given Boolean expression by using Algorithm 4. The MPRM expression is mapped by Algorithm 5, the power and area of the AND/XOR circuit are estimated by Eqs. (2) and (1), the cost of circuit is evaluated by Eq. (9), and the best solution is initialized;

(2) The current polarity is achieved based on Gray code. The MPRM expression under current polarity is converted from the MPRM expression under neighbor polarity by using Algorithm 5. The power and area of the AND/XOR circuit are estimated by Eqs. (2) and (1), and the cost of the circuit is evaluated by Eq. (9);

(3) Update the best solutions based on the minimum cost value. Stop and output the best polarity if each polarity has been traversed; otherwise, return step 2.

**5. Experimental results**

The proposed algorithms have been coded in C and simulated using a Pentium-IV processor on a Linux environment. In our program, we applied our algorithms on the standard PLA format of the ISCAS and MCNC benchmark. If there was no PLA format, then we applied SIS to convert from other formats to the PLA format. We used the minimizer SIS Espresso for getting the minimal two-level AND/OR circuits from selected benchmarks<sup>[6]</sup>.

In the simulation, the clock frequency and supply voltage were assumed to be 20 MHz and 5 V, and the SIS mcnc.genlib library was used for mapping. The main parameters of mcnc.genlib are shown in Table 1, where ‘Area’ is the area cost of the single gate, and ‘ $C_{in\_A}$ ’, ‘ $C_{in\_B}$ ’ and ‘ $C_{out}$ ’

Table 2. Comparison of results between the optimization method for MPRM realizations, FPRM realizations and AND/OR circuits.

Benchmark	MPRM										FPRM <sup>[9]</sup>		AND/OR <sup>[7]</sup>		
	$w = 0.00$		$w = 0.25$		$w = 0.50$		$w = 0.75$		$w = 1.00$		Area	Power	Area	Power	
Name	In/Out	Area	Power	Area	Power	Area	Power	Area	Power	Area	Power	Area	Power	Area	Power
tcheck	3/3	11	21.7	11	21.7	11	21.7	11	21.7	11	21.7	11	21.72	21	36.68
ver	4/1	44	78.5	44	78.5	44	78.5	44	78.5	44	78.5	56	104.14	54	87.44
xor5	5/1	20	50.0	20	50.0	20	50.0	20	50.0	20	50.0	20	50.00	237	312.23
rd53	5/3	160	298.7	160	298.7	160	298.7	160	298.7	160	298.7	160	298.74	459	637.07
newapla2	6/7	105	146.0	111	141.5	105	146.0	105	146.0	122	141.0	138	144.64	105	146.04
pope	6/48	3302	4086.4	3302	4086.4	3302	4086.4	3302	4086.4	3302	4086.4	3416	4512.15	3942	5565.15
rd73	7/3	678	1122.6	678	1122.6	678	1122.6	678	1122.6	678	1122.6	678	1122.55	2619	3206.26
max128	7/24	3129	3461.3	3284	3276.6	3284	3276.6	3129	3461.3	3751	3185.8	3447	4726.62	2460	3481.34
rd84	8/4	1250	1995.6	1250	1995.6	1250	1995.6	1250	1995.6	1250	1995.6	1250	1995.60	6111	7085.89
sqrt8	8/4	397	576.6	397	576.6	397	576.6	397	576.6	397	576.6	397	576.62	477	697.86
f51m	8/8	786	1175.6	786	1175.6	786	1175.6	786	1175.6	786	1175.6	786	1175.63	951	1346.47
9sym	9/1	663	3435.5	663	3435.5	663	3435.5	663	3435.5	2252	3435.5	2252	3435.49	2661	3249.86
apex4	9/19	21503	20129.0	22595	13987.3	22449	14100.6	22449	14100.6	22595	13987.3	22149	22708.08	27606	31236.18
prom1	9/40	37985	18683.6	38221	18476.6	37985	18683.6	37985	18683.6	47466	17862.5	47126	45992.45	60966	67775.03
newtpla2	10/4	203	219.7	203	219.7	203	219.7	203	219.7	203	219.7	205	225.88	210	277.67
sao2	10/4	2200	2049.2	2209	2021.8	2209	2021.8	2209	2021.8	2209	2021.8	2791	2617.17	1575	1885.15
br1	12/8	1027	801.6	1030	792.0	1030	792.0	1030	792.0	1030	792.0	2572	2095.47	1266	1331.63
newapla	12/10	331	354.7	331	354.7	331	354.7	331	354.7	331	354.7	331	354.70	318	434.99

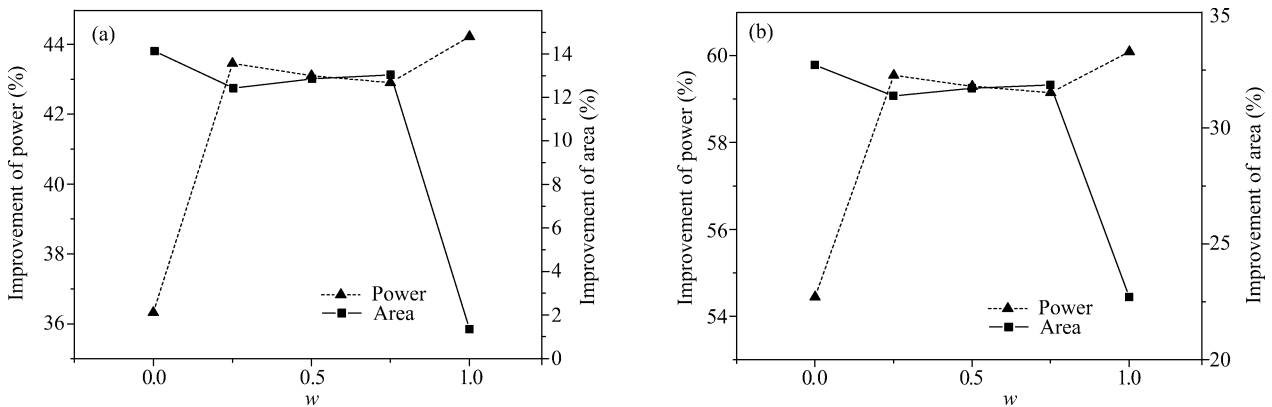


Fig. 3. Improvement in power and area for MPRM minimization over (a) FPRM realizations and (b) AND/OR minimization with different  $w$ .

are the input and output capacitance.

In our experiments, the signal probability of each primary input was set to 0.5, and 18 medium and small benchmark instances were selected randomly for testing. To evaluate the effectiveness of our algorithm, for 5 different  $w$ , the results obtained by our optimization method compared with those obtained by the optimization method for FPRM realizations in Ref. [9] and AND/OR circuits optimization in Ref. [7]. The results are shown in Table 2. The name of benchmark ('name'), the number of input and output ('in' and 'out'), the area (unit) and power ( $\mu\text{W}$ ) of circuits ('area' and 'power'), the results of using our method, the optimization method for FPRM realizations and AND/OR circuits ('MPRM', 'FPRM' and 'AND/OR') are presented.

Table 2 shows that results obtained by our algorithm are much better than the optimization method for FPRM realizations, where we have a power saving up to 44.22% and an area saving up to 14.13% on average, and results obtained by our algorithm are much better than the optimization method for AND/OR circuits in the cases of xor5, rd53, rd73, rd84, prom1 and br1, where we have a power saving up to 60.09% and an

area saving up to 32.72% on average. Figures 3(a) and 3(b) show that a graphical representation of the power and area improvements in MPRM minimization over FPRM realizations and AND/OR minimization, where it obtains the following features: (1) the area of circuits can be minimized when  $w = 0$ , but the ability of power optimization is poor; (2) both the power and the area can be optimized if  $1 < w < 0$ , and the performance of integrated optimization is best when  $w = 0.5$ ; and (3) the power of circuits is optimal when  $w = 1$ , yet the ability of area optimization is worse. The CPU time of our algorithm is proportional to the scale of circuits. The overall simulation time for large circuits like newapla is over 33 min, or for smaller circuits like tcheck almost 0.

## 6. Conclusion

In this paper, low power mapping for AND/XOR circuits with switching activity minimization is addressed and its application in searching the best polarity is discussed. We presented a fast low power mapping algorithm for AND/XOR circuits that attempts to minimize the switching activity of nodes. In

this algorithm, we structured estimation models of the power and area for AND/XOR circuits to measure the cost of circuits. Finally, we proposed searching the best mixed-polarity algorithm to optimize the cost of MPRM circuits based on the polarity conversion algorithms and the exhaustive searching method. We experiment with our algorithms against the well-known benchmarks. The experimental results show that searching the best mixed-polarity algorithm can effectively reduce both the power and the area compared with the traditional optimization algorithms, and the best integrated optimization of power and area is found to be  $w \approx 0.5$ . For an  $n$ -input AND/XOR circuit, the searching space is  $3^n$ . Searching the best polarity algorithm is infeasible in CPU time for large-scale logic circuits. In the future, we will develop a searching algorithm combined by Monte Carlo algorithms, which can find the near optimal solutions in the polynomial time.

## References

- [1] Roy K, Prasad S C. Low-power CMOS VLSI circuit design. Canada: Simultaneously, 2000
- [2] Sasao T. Easily testable realizations for generalized Reed–Muller expressions. *Transactions on Computers*, 1997, 46(6): 709
- [3] Wang P, Lu J, Chen K, et al. Low power polarity conversion based on the whole annealing genetic algorithm. *Journal of Semiconductors*, 2008, 29(2): 298
- [4] Al Jassani B A, Urquhart N, Almaini A E A. Manipulation and optimisation techniques for Boolean logic. *Computers and Digital Techniques*, 2010, 4(3): 227
- [5] Büyüksahin K M, Najm F N. Early power estimation for VLSI circuits. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(7): 1076
- [6] Tsui C Y, Pendram M, Despain A M. Technology decomposition and mapping targeting low power dissipation. *Design Automation Conference*, 1993: 68
- [7] Zhou H, Wong D F. Exact gate decomposition algorithm for low-power technology mapping. *International Conference on Computer-Aided Design, Digest of Technical Papers*, 1997: 575
- [8] Narayanan U, Liu C L. Low power logic synthesis for XOR based circuits. *International Conference on Computer-Aided Design, Digest of Technical Papers*, 1997: 570
- [9] Zhou H, Wong D F. Optimal low power XOR gate decomposition. *Design Automation Conference*, 2000: 104
- [10] Al Jassani B A, Urquhart N, Almaini A E A. Minimization of incompletely specified mixed polarity Reed Muller functions using genetic algorithm. *International Conference on Signals, Circuits and Systems*, 2009: 1
- [11] Brzozowski I, Kos A. Calculation methods of new circuit activity measure for low power modeling. *International Conference on Signals and Electronic Systems*, 2008: 133
- [12] Brzozowski I, Kos A. A new approach to power estimation and reduction in CMOS digital circuits. *Integration, the VLSI Journal*, 2008, 41(2): 219
- [13] Cheng J, Chen X, Faraj K M, et al. Expansion of logical function in the OR-coincidence system and the transform between it and maxterm expansion. *Computers and Digital Techniques*, 2003, 150(6): 397
- [14] Zhang X Y, Wang L L, Zhou X G. Efficient RM conversion algorithm for large multiple output functions. *ICSICT*, 2008: 2300