Two-level AND-XOR Network Synthesis with Area-Power Trade-off

Sambhu Nath Pradhan **Santanu Chattopadhyay**
Dept. of E & ECE, IIT Kharagpur-721302, India

Abstract

As AND-XOR network results in much better realization and requires fewer product terms than AND-OR realization, AND-XOR network has encouraged researchers to look for efficient minimization and synthesis tools for it's realization. Among several canonical representations of AND-XOR networks, popular and the most testable one is the Fixed Polarity Reed Muller (FPRM) form. In this paper we have used GA (genetic algorithm) to select the polarities of the variables of the AND-XOR network. The polarities are selected based on the optimization of area, dynamic power and leakage power of the resulting circuit. This is the first ever effort to incorporate leakage power consideration in the variable polarity selection process. Here, we have presented new leakage power model of AND and XOR gates at 90nm technology. The area (in terms of number of product terms) results obtained are superior to those reported in the literature. The dynamic power minimization also results in their optimum values for many circuits that we have tested. It also enumerates the trade-offs present in the solution space for different weights associated with area, dynamic power and leakage power of the resulting AND-XOR network.

Index Terms

Fixed Polarity Reed Muller form, AND-XOR network, genetic algorithm, switching power, leakage power, Binary Decision Diagram.

1. Introduction

Logic minimization, in the domain of combinational logic synthesis, plays an important role in determining area and power of the synthesized circuit. Logic optimization techniques can be broadly classified into two categories. The first one is targeted to the PLA structure which is a two-level minimizer. On the other hand, a multilevel minimizer aims at reducing the circuit area by extracting common subexpressions within the subfunctions. Whether it is a two level logic circuit or multilevel logic one, these are basically the manifestation of AND-OR logic synthesis. However, in XOR dominated application-specific circuits, the XOR-based synthesis produces better realizations. Researchers have long back established the suitability of XOR based representation in different domains of applications, such as, linear circuits, arithmetic circuits, telecommunication circuits etc. In such cases, XOR-based realizations often produce more compact circuit than the OR-based realizations. When realized as PLAs, XOR circuits offer high testability. It is also worthwhile to note the fact that several technologies like Field Programmable Gate Arrays (FPGAs) have made the delay and area of all types of gates equal. For example, in Xilinx look-up-table-type FPGA, the basic combinational block can realize any function of upto five variables with same area and delay.

Similarly, the three-Mux architecture of Actel is quite suitable for XOR realization. Cli6006 from Concurrent Logic includes a 2-input XOR gate in its basic granularity block. Two-level realizations often form the basis for the multilevel minimization tools. Thus, for good realization of combinational functions, it is very important to start with a good two-level decomposition of it. Accordingly, lot of research works have been carried out in XOR-based circuit synthesis and this has been surveyed in Section 2. There are several types of AND-XOR based circuits, namely positive polarity Reed-Muller (PPRM), fixed polarity Reed-Muller (FPRM), pseudo Reed-Muller, generalized Reed-Muller, XOR sum of products, Kronecker and pseudo Kronecker forms [6]. Each of these circuits has its own advantages. As far as XOR synthesis is concerned, this paper concentrates on the synthesis of FPRM circuits only. The works reported in the literature so far have the following shortcomings. Most of the works target area minimization and do not account for power. However, the advent of portable electronic devices has made low power design an increasingly important research area. Though the works presented in [23] talks about low power XOR-based circuit synthesis, it targets a multi-level realization. It may be noted that all the previous works ignored the leakage power consumed by the circuit under the premises that leakage power is quite insignificant as compared to the dynamic power. The International Technology Roadmap for Semiconductors (ITRS) projects an exponential increase in the leakage power with minimization of devices [24]. As the technology drops below 65nm feature sizes, subthreshold leakage is expected to exceed the total dynamic power. As leakage current becomes the major contributor to power consumption, the industry must

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

reconsider the power equation that limits system performances, chip sizes and cost.
With this background, the current work proposes a solution to the following problem.

Given a multiinput, multioutput Boolean function F and weight factors, perform a FPRM synthesis of minimizing the weighted sum of area (number of product terms), dynamic power (switching activity) and leakage power.

The motivation comes from the following example:

Example 1: Consider a multi-output function F consisting of subfunctions,

$$f_1(x_3, x_2, x_1) = \Sigma(1, 3, 5, 6, 7),$$
$$f_2(x_3, x_2, x_1) = \Sigma(1, 3, 5, 6)$$

AND-OR realization of the function F requires 4 product terms: $x_1x_2x'_3$, x'_1x_3, x'_2x_3 and x_3. For AND-XOR realization with positive polarities for all the three variables, the FPRM forms of f_1 and f_2 are,

$$f_1 = x_3 \oplus x_1x_2 \oplus x_1x_2x_3$$
$$f_2 = x_3 \oplus x_1x_2$$

Thus AND-XOR realization of the function F requires 3 product terms.
To solve the above problem we have made a Genetic Algorithm (GA) based formulation to identify the polarity assignment to the input variables.
Reed-Muller expansion [14] has long been proposed for realizing Boolean function in the positive polarity AND-XOR form. An n-variable Boolean function f can be expressed in the following Canonical Reed-Muller expansion of 2^n terms:
$f(x_1, x_2, ... x_n) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus ----- \oplus a_2n\text{-}_1\, x_1x_2....x_n$, where $a_i \in \{0, 1\}$. All x_i input variables appear in true form in the expansion. A number of modified versions of this basic canonical form have been studied. One in which the variables are allowed to take both positive and negative polarities, is known as Generalized Reed-Muller (GRM) form. Here, the polarity of a variable can vary between the product terms. The constrained version in which each of the variables is allowed to assume only one of the polarities that is maintained throughout the function is known as Fixed Polarity Reed Muller (FPRM) form [14]. This particular form results in much lesser number of product terms than the original Reed-Muller form and is the most testable among all general canonical forms.
The rest of the paper is organized as follows: section 2 gives the previous work on AND-XOR synthesis. Section 3 presents our AND-XOR synthesis approach. Power estimation is given in Setion 4. Using GA, AND-XOR synthesis solution is given in Section 5. Sectionn 6 presents experimental results.

2. Previous Works on AND-XOR Logic Synthesis

Two level AND-XOR minimizers have been illustrated in [1, 2, 3, 4]. It is seen that AND-XOR logic results in much better realization and requires fewer product terms than AND-OR realization as explained by Sasao and Besslich [5]. For example, on an average a five variable function requires 7.46 product terms in minimum SOP (sum-of-product), while 6.16 product terms in minimum ESOP (exclusive sum of product) expressions as shown by Sasao[6]. A number of heuristic algorithms for AND-OR-XOR minimization have been presented in [7, 8, 9]. Dubrova et al. in [8], Debnath and Sasao in [10] have given an upper bound on the complexity of algorithm and number of product terms. Sasao et al. in [11, 5, 12, 13] dealt with the problem of minimizing the two-level AND-XOR PLAs and proposed some heuristic approaches and used both true and complemented variables in mod-2 SOPs. An upper bound in the number of product terms in minimum ESOPs has also been derived. Reed-Muller expansion [14] has long been proposed for realizing Boolean function in the positive polarity AND-XOR form. A number of modifed versions of this basic canonical form have been studied. The representation in which a variable can have either positive or negative polarity (which is consistent throughout the function) is known as Fixed Polarity Reed-Muller Form (FPRM) as given by Davio and Deschamps [15]. This particular form results in much lesser number of product terms than the original Reed-Muller form and is the most testable among all general canonical forms. It also has applications in function classification, Ashenhurst and other decomposition methods, and multilevel design [15]. It has also been applied in image processing and other AND-XOR minimizers as addressed by Besslich and Riege [1]. A heuristic approach has been proposed by Sarabi and Perkowski [16] to find out the best polarity assignment for FPRM realization. Chattopadhyay et al.[2] presented a scheme for FPRM network synthesis based on genetic algorithm. The scheme results in considerable improvement over [16]. However, both these works address area minimization of single-output functions. Chattopadhyay et al. [3] addressed the area minimization problem for FPRM realization of multioutput functions. Graphical method of FPRM synthesis has been presented by Tran [17] and Green [18]. However these methods suffer from the limitation that they can handle only a small number of inputs effectively. A mapping based FPRM synthesis has been presented by Khan and Alam in [19] and [20], which are based on relationship between the minterms of the given system and FPRM polarities. In [21] a Binary Decision Diagram (BDD) based method has been presented that uses an optimized BDD representation as the starting point of the FPRM coefficient generation procedure. But the generation of the optimal BD tree may itself require a major effort. A detailed survey of the work done so far in this context has been given in [22].

3. AND-XOR Problem Formulation and Synthesis

Given an n-input, m-output Boolean function, the objective is to realize it as a Fixed Polarity Reed-Muller (FPRM) expansion. In the FPRM expansion, each variable maintains a fixed polarity- either positive or negative throughout the expansion. That is, a variable appears either in true or in complemented form. This consists of the following steps.

1. Obtain a disjoint cube representation of the function.
2. Determine the polarity assignment to each variable.
3. Decompose the network into a consistent polarity one.

First of all the sub-functions given in terms of AND-OR cubes are converted to a set of disjoint cubes. Now the ORs can be replaced with XORs, without affecting the functionality. Once we have obtained a set of disjoint cubes, next task is to determine the polarity assignment $< p_1, p_2, p_3,\ldots,p_n >, p_i \in \{0, 1\}$ to the variables $< x_1, x_2, x_{3, \text{-------}} x_n >$ in order to minimize the number of product terms and/or the associated power consumption in the FPRM realization. Each variable can attain either positive or negative polarity. A variable with polarity *1* occurs uncomplemented in all product terms, whereas variable with polarity *0* occurs only in complemented form. To obtain the FPRM form, the literals having polarities different from that assigned to the corresponding variables are replaced by $1 \oplus X_i$, where $X_i = x_i,\ if\ p_i = 1$ and x_i', otherwise. Before going to explain the GA based synthesis of AND-XOR logic, let us formulate the different power dissipation components in the circuit.

4. Power Estimation

The power consumed by a CMOS VLSI circuit can be categorized into two classes [25]: dynamic and static power. Dynamic power dissipation in CMOS circuit occurs when the circuit nodes are switching their states.

$$P_{dynamic} \approx P_{switching} = 0.5 C_L V^2_{DD} \alpha_L f$$

here, α_L is the switching activity at the output node of the circuit, V_{DD} is the supply voltage, f is the frequency of operation, C_L is the physical capacitance at the output node. Thus, all other factors remaining same (may be determined by technology, speed etc), the controllable parameter is the switched capacitance, $\alpha_L.C_L$. Here, we considered expected switching activity (ESA) as the estimation of switching power.

4.1 Switching probability estimation

ESA is defined as the expected number of changes that occur at the outputs of the gates in a combinational logic circuit. For a change in the primary input to the system we assume that primary inputs are uncorrelated, that is, *Prob(input = 1) = Prob(input = 0) = 1/2*, and are statistically independent of each other. The expected switching activity at the output of a logic gate, *F*, is denoted by the term *ESA(F)*. We first begin by considering the ESA for a single logic gate. Output of the logic gate changes its state if the current state of the output differs from the previous one. Thus the probability of the output of a gate changing state is: *Prob (current output = 0)*Prob (previous output = 1) + Prob (current output = 1)*Prob (previous output = 0)*. Since we assume that the probability does not change with time, ESA (of logic gate) = 2*Prob (output = 0) * Prob (output = 1). The ESA for an *i* input AND gate (with primary inputs) is $2(1/2^i)[1- (1/2^i)]$.
For $\mathbf{n_1}$ such AND gates in the first level, switching activity is

$$\boldsymbol{Psw\_first\_total} = \sum_{n1} 2*[(1/2)^i]*[1-(1/2)^i]$$

To compute ON-probability of second-level XOR gates, we consider the Boolean function implemented by them. If a gate with *j* inputs realizes a function with *k* ON-terms, the ON-probability is given by, $\boldsymbol{k/2^j}$. Thus, switching activity of the node is $\boldsymbol{2*[k/2^j]*[1-k/2^j]}$

4.2 Leakage power estimation

To get the leakage power of the gates, all gates are implemented in pseudo NMOS logic style. This is because basic PLA structure is pseudo NMOS type, the number of transistors required is low, so the area of the resulting PLA structure is small. The problem in this structure is that when one of the pull down path is ON, there is static power dissipation. In this paper we have also considered this short circuit power at the time of leakage power calculation. All inverters are implemented in static CMOS logic style. All the leakage power has been calculated in CADENCE SPECTREE at 90nm UMC technology. Here, all the transistors are the general purpose transistors whose minimum supported length is 80nm for the pull down transistor and 120 nm for the pull up transistor. For all the cases, the length and width of pull up transistor is 500nm and 120nm respectively. The widths of the pull down transistors are adjusted accordingly to get the correct output. Probability dependent leakage power of a gate is given by ,

$$P_{leakage} = V_{dd} \sum_k S_k * I_k \quad \text{----------} \ (1)$$

k is over all the possible input states of the gate. S_k is the probability of state k and I_k is the leakage current of state k. V_{dd} is the supply voltage. In 90nm technology, supply voltage is 1 volt.
Same type of gates in a circuit may have different number of inputs and when number of inputs is very large, direct application of Eq. 1 to estimate leakage power is quite infeasible. So we have estimated leakage in a different way as given in the following section.

4.2.1 AND gate leakage

Table 1 shows the leakage power of a 2-input AND gate for all the input combinations. Column 2 gives the leakage power of the NAND gate and column 3 gives the leakage power of the AND gate considering inverter at the output of the NAND gate.

Table 1: Leakage power of two input NAND and AND gate

Inputs A B	Leakage Power of NAND (watt)	Leakage Power of AND (watt)
0 0	1.554 pico	34.7815 nano
0 1	1.994 nano	36.774 nano
1 0	1.399 nano	36.179 nano
1 1	14.040 micro	14.0495 micro

Input A is assumed to be nearer to the the output. For ‘00’ combination, both the pull down transistors are off, so the leakage power is the minimum for this case. For the combination ‘01’ lower transistor of the pull down network is ON, so the leakage value of this combination is higher than that of ‘10’ combination. When both the pull down transistors are ON, leakage is high.

Leakage power of inverter when the input is high is 34.78 *nano watt* and it is 9.5 *nano watt* when the input is low. We have simulated upto 10 input AND gate and obtained the leakage power values. Width of the transistors in the series are adjusted such that the lower most transitor’s width is 200nm. To maintain correct output voltage level, for each level above this lowest level, transistor’s width is incremented by 100nm compared to the previous level. From all the leakage values obtained, we observed that the leakage values maintained a specific pattern. For example, when all the inputs are high, for ‘11’ it is 14.0495uw, for ‘111’ inputs it is 14.0095uw, for ‘1111’ the leakage value is 13.97uw. Difference in leakage between ‘11’ and ‘111’ inputs is 40nw and leakage difference between ‘11’ and ‘1111’ is about 79.5nw. So, difference in leakage between ‘11’ and an n-input AND gate is nearly a multiple of 40nw. This difference is captured by *(n-1)*40* nw. Let the probability of becoming all input high be P_{all_high}, then extrapolated leakage power for all high input is P_{all_high} **{14049.5 – (n-2)*40}*nw. Similarly, for other cases the extrapolated leakage power is $(1-P_{all\_high})*\{36.49 -(n-2)*0.1\}$ nw. Here, *n* is the number of input. So, extrapolated leakage power for AND gate having *n* number of inputs is given by,

$P_{all\_high}*(13969.5-40*n)+(1-P_{all\_high})*(36.69 -0.1*n\}$-----(2)

4.2.2 XOR gate leakage

During the implementation of the XOR logic gate, sharing of the Boolean function at each level is considered. This reduces the number of transistors. In this approach two input XOR gate requires seven transistors, three input XOR gate requires eleven transistors, four input XOR gate requires only fifteen transistors. So for *n* input XOR gate, the number of transistors required is *3 + (n-1)*4*. Fig. 1 and 2 show three and four input XOR gates respectively. Width of the lowest level transistor is 200nm. For each level above this, transistor’s width is incremented by 100nm compared to the previous level. This ensures correct output voltage level.

We simulated the XOR gate upto 10 inputs and noted the leakage power values. It has been observed that for a particular XOR gate, when even number of inputs are ON, leakage power is the same. For odd number of ON inputs, the leakage power is also same but the value is different from that for even number of ON inputs. This is expected as for any even combination of ON inputs, the pattern of transistors ON and OFF are similar. For odd combination of ON terms also this is true. Table 2 shows the leakage power of three input XOR gate. Table 3 gives the simulated leakage value upto eight input XOR gate. Here, we do not consider the inverter at the output. For example, for three input XOR as depicted in Fig 1, reported leakage values in Table 3 are upto the point X.

Number of input	Leakage power (watt) for even number of ON terms in a combination	Leakage power (watt) for odd number of ON terms in a combination
3	6.896 nano	13.82 micro
4	10.35 nano	13.78 micro
5	13.96 nano	13.76 micro
6	17.32 nano	13.74 micro
7	20.6 nano	13.73 micro
8	24.11 nano	13.72 micro

To estimate the leakage power of *n* input XOR gate, ON probability at the output of the XOR gate is calculated. Let, this probability be P_{ON}. From the simulated leakage table of the XOR gates, leakage power for odd and even combination of ON terms of *n* input XOR gate is extrapolated as there is a specific pattern among the leakage values. For two input XOR gate, the leakage value is calculated using Eq. 1.

P_{ON} is the probability of XOR gate being high. The output of XOR is high when odd number of inputs are ON in a combination. So, for the even number of ON inputs, the probability of getting high at the XOR output is $(1-P_{ON})$. From Table 3, it is observed that leakage values of the input combination having even number of ON terms are 6.896nw (for three input) ≈ 2*3.45, 10.35 nw (for three input) ≈ 3*3.45, which are multiple of 3.45 nw. So, leakage value of the input combination having even number of ON inputs is given by, $(1-P_{ON})*(n-1)*3.45$ *nw*, --------- (3)

where, *n* is the number of inputs. For three input XOR, the leakage for odd number of ON inputs is 13.82 uw, for four input case, it is 13.78uw. So, the difference is 40nw. For the combination having odd number of ON inputs, this difference in the leakage value is captured by the term, *tot_inc* which is calculated from the leakage table in the following way:

```
    inc=40nw; tot_inc=0.0;
FOR i = 1 TO (n-3 )  // n  is the number of input
    inc =inc/i ;
    tot_inc = tot_inc + inc;
END FOR
```

So, for n input XOR gate, leakage value of the combination having odd number of ON terms is given by,

$$P_{ON}*(13820 - tot\_inc)\ nw\text{----------- (4)}$$

Now, consider the inverter at the output. When input is high leakage power of inverter is 34.78nw and for low input it is 9.48 nw. Probability of the input of the inverter to be high is P_{ON} and the probability of the input of the inverter becombining low is $(1-P_{ON})$. So, for the inverter at the output, leakage value is

$$P_{ON} * 34.78 + (1- P_{ON})*9.48nw,\text{ --------- (5)}$$

Combining Eqn. 3, 4 and 5, for n input ($n \geq 3$) XOR gate leakage is given by,

$$(1-P_{ON})*(n - 1)*3.45 + P_{ON} * (13820 - tot\_inc) + P_{ON} * 34.78 + (1- P_{ON})*9.48\ nw\text{ -----------------(6)}$$

Table 2: Leakage power of three input XOR gate

Input combination A B C	Leakage power (watt)
0 0 0	6.896 nano
0 0 1	13.82 micro
0 1 0	13.82 micro
0 1 1	6.896 nano
1 0 0	13.82 micro
1 0 1	6.896 nano
1 1 0	6.896 nano
1 1 1	13.83 micro

Table 3: Leakage power of XOR gate

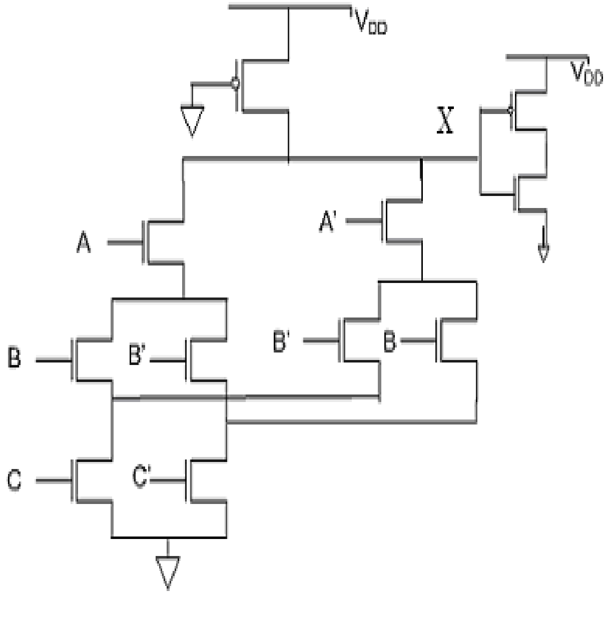


Figure 1: Three input XOR gate

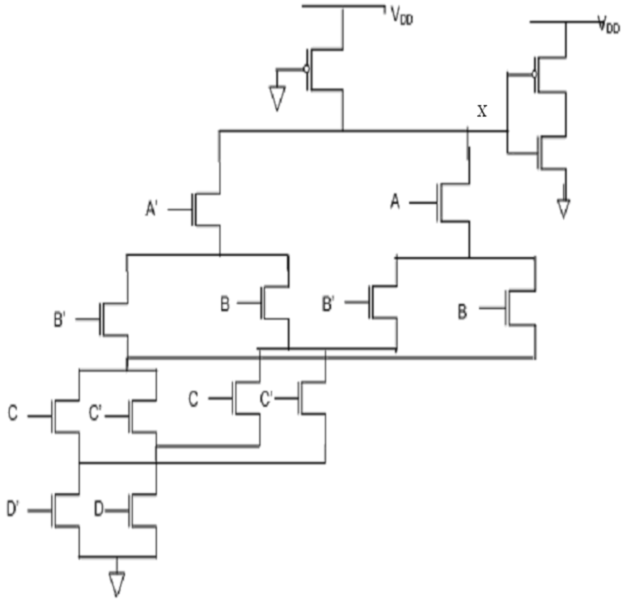


Figure2: Four input XOR gate

5. Genetic Algorithm Formulation

In this section, we present our GA based strategy for polarity identification.

5.1 Solution Representation

In this problem, a solution can be most elegantly represented as an n-bit bitstring $P = < p_1, p_2, p_3, \ldots\ldots, p_n >$, $p_i \in (0, 1)$. $p_i = 0$ means the corresponding input variable is assigned negative polarity, while $p_i = 1$ implies that the variable be of positive polarity. For example, for a 3-input function, the chromosome “010” indicates that the first and third variables are of negative polarity, while the second variable is of positive polarity.

5.2 Fitness Measure

The fitness measure must be devised for each problem to be solved by GA. Given a particular chromosome, the fitness measure returns a single numerical fitness or figure of merit, which is supposed to be proportional to the utility or ability of the individual which that chromosome represents. Number of product terms in the FPRM expansion of a function corresponding to the polarity assignment depicted by a chromosome is taken as the area measure for the chromosome. Procedure to calculate expected switching activity (ESA) and leakage power computations are the same as discussed in Section 4.1 and 4.2. respectively. The ESA values over all rows and columns in both AND- and XOR-plane are summed up to get the ESA value to be used as the dynamic power component in the fitness measure of each chromosome. Total leakage power value is the leakage power component in the fitness measure of each chromosome. To get area-power trade-off, we next need to combine the area, ESA values and leakage value computed above. For this, the values are first normalized. To carry out normalization, we compute the maximum area, maximum ESA and maximum leakage required by any chromosome in the first generation. Let this value be A_{max}, E_{max}, and L_{max} respectively. For a chromosome *C*, let the area be *AC, ESA* be *EC* and leakage power be *LC*. Fitness of *C* is then given by

$$fitness(C)= w_1*(AC/A_{max}) + w_2*(EC/E_{max}) + w_3*(LC/L_{max})$$

The weightages w_1, w_2 and w_3 can be set by the designer with $w_1+w_2+ w_3 = 1$

5.3 Genetic Operators

The following operators have been used to evolve GA.

5.3.1 Crossover: In GA formulation, the selection of chromosomes participating in crossover may not be uniformly random. In fact it is biased towards the chromosomes with better fitness value in this work. For this purpose, whole population is sorted according to the fitness value. A certain percentage of population (here, 20%) with better ftness value is considered to be the "best class". To select a chromosome participating in crossover, first a uniform random number between 0 and 1 is generated. If the number is greater than 0.5, a chromosome from the best class is selected randomly. Otherwise a chromosome is selected from the entire population. Let the population size be *n* and the cardinality of the best class be *m*. Then the probability of a chromosome getting selected for crossover is *0.5/m + 0.5/n,* for chromosomes belonging to the best class. Whereas, for a chromosome not belonging to the best class, the probability of it getting selected is *0.5/n*. Since *m* is much lesser than *n*, the probability of a chromosome belonging to the best class being selected is more than that of a chromosome not belonging to the best class. This approach of selecting more fit chromosomes to participate in crossover leads to the generation of better off-springs as compared to truly random one. After selecting two parent chromosomes to participate in crossover, a random point is selected, around which, parts of the chromosomes are exchanged to create the off-springs.

5.3.2 Mutation: Mutation operator is very important in bringing variety into the population. As the population size is finite, crossover operator alone cannot bring enough variation in the population. Thus the solution quality and rate of convergence suffers. The mutation operator brings sudden variation into the chromosomes introducing newer search options. The mutation operator simply modifies a chromosome by complementing one of its bits at some randomly chosen point.

5.3.3 Direct Copy: We copy best 20% chromosomes directly to the next generation. This ensures that best chromosomes are always maintained between the generations and do not inadvertently get degraded by crossover or mutation.

5.3.4 Termination: The GA terminates when there is no improvement in result over the previous 50 generations. The best chromosome at that generation is taken as the final solution.

6. Experimental Result

The proposed idea of AND-XOR network synthesis is coded in C language and applied to a number of benchmark circuits from NCSU benchmark suit LGSynth93, on a Linux based platform using Pentium-IV processor. Simulation is carried out for a large number of generations and GA is terminated when there is no further improvement in cost function for 50 consecutive generations. Population size is taken as 200 for larger circuits with inputs more that 15 and for circuits with less input, population is varied from 20 to 100. Crossover and mutation factors are 0.6 and 0.2 respectively.

First we present a comparison of our work with the AND-XOR minimizer approaches available in the literature from the point of view of area minimization. Also we compare the dynamic power component. Table 4 shows the area and power comparisons. As it can be seen from the table, our approach is quite compatible with the approaches in the literature. Column 2 of Table 4 gives the number of inputs and outputs of the circuits. Column *min-switching* gives the optimum switching value among all possible input polarity assignments of the circuit. Using our GA approach, switching results corresponding to switching based decomposition is given at the last column of Table 4. From the switching results it is observed that in all cases, our GA based approach reaches the optimum switching possible.

Next we present the result of area-power trade-offs achievable in our approach. For this purpose we have varied w_1(area weightage), w_2(dynamic power weightage) and w_3(leakage power weightage) in a range of 0 to 1. Table 5 presents some of the example cases for the values of w_1, w_2 and w_3. In particular we have presented the results for w_1, w_2, w_3 values of *(1,0,0), (.5,.5,0), (0,1,0), (0,.5,.5), (0,0,1), (.5,.25,.25), (.5,0,.5)*. The circuits in bold face show area-power trade-offs under various weight assignments. Last column shows the maximum CPU time required among all the decompositions. It may be noted that the area results are the number of product terms, while dynamic power results are the total switching activities and leakage power is in microwatts. To summarize the results, we have computed the number of circuits having minimum area, switching and leakage value for each weighted decomposition. For an example, out of 31 circuits, 30 circuits give minimum area value for the weights *(1,0,0)*. For other cases, number of circuits having minimum area is less. So, for only area based decomposition, area weightage should be 1 and maximum area saving can be achieved but switching and leakage power are high in this case. In fact, switching is 3.5% higher than it's minimum switching value and leakage is 4.8% higher than the minimum leakage value. We have also computed some ratios. For getting area ratios, the results for the case *(1,0,0)* (that is, area weightage 1 and rest two are 0s) is taken as unity. Other area values have been divided by this quantity. Average of all these ratios have been noted in the last row of the Table 6. Similarly, ratios have been calculated for switching activity and leakage, taking the values for the cases *(0,1,0)* and *(0,0,1)* as references respectively. From Table 5, it is observed that minimum switching and leakage are obtained for the decomposition *(0,1,0)* and *(0,0,1)* respectively. So, maximum number of circuits which result minimum switching is for full weightage to switching activity but in this case, area and leakage values are 8% and 5.4% higher compared to their minimum values. As leakage power dissipation is at it's minimum for the weightage *(0,0,1)*, one can choose this weightage beyond 65nm, where leakage becomes the dominating factor. But for this weightaged decomposition area and switching are 10.9% and 2.7% higher than their corresponding minimum values. In fact, in this case, area increase is maximum.

As observed from the Table 5 and 6, the best results are obtained for the weightage *(0.5, 0.25, 0.25)* for which area, switching and leakage are 2.8%, 2% and 2% higher compared to their respective minimum values. It may also be noted that for the weightage *(0.5, 0.5, 0)* area, switching and leakage results are also good. In this case, area, switching and leakage values are 2.7%, 2% and 2% higher compared to their minimum results. So, we can decompose the circuits for weightage *(0.5,0.25,0.25)* or *(0.5,0.5,0)* for which area, switching and leakage values are within acceptable limits.

However, it must be mentioned that these weightages can best be adjusted by the designer based upon the target technology, design constraints, and optimization criterions.

Conclusion

Variable polarity selection plays an important role in efficient realization of FPRM representation of a function. In this paper, we have tackled this problem using a genetic algorithm-based formulation. The scheme not only concerns about the minimization of the number of product terms but also takes into account the total power dissipation of the circuit. A trade-off has been performed for the weighted minimization of area, switching and leakage power. It has been shown that a range of solutions can be achieved with varying degree of area, switching and leakage power minimization using different weightages associated with the area requirement, switching and leakage power consumption of the resulting two-level AND-XOR circuit. To the best of our knowledge consideration of leakage power dissipation of the resulting circuit during variable polarity selection of AND-XOR circuit synthesis process has been done for the first time in this work.

References

[1] P.W. Besslich and M. Riege. An efficient program for logic synthesis of Mod-2 Sum Expressions. In *Proc. of Euro ASIC*, pages 136-141, 1991.

[2] S. Chattopadhyay. Synthesis of low power Fixed Polarity Reed-Muller Networks. In *Proc. of Int. Conference/ Exhibition on High Performance Computing in Asia Pacific Region*, volume 6th, December 2002.

[3] S. Chattopadhyay and P. P. Chaudhuri. Genetic algorithm approach for minimizing multi-output Fixed Polarity AND-XOR functions. In *Proc. of Asia PacificConference on Hardware Description Languages*, 1996.

[4] S. Chattopadhyay, S. Roy, and P. P. Chaudhuri. Synthesis of highly testable Fixed Polarity AND-XOR canonical networks- a Genetic Algorithm based approach. *IEEE Trans. on Computers*, 45(4):487-490, April 1996.

[5] T. Sasao and P. Besslich. On the complexity of Mod-2-Sum PLAs. *IEEE Trnas.on Computers*, 39(2), February 1990.

[6] T. Sasao. *AND-EXOR expressions and their optimization*. Kluwer Academic Publishers, Editor, Logic Synthesis and Optimization, Boston, 1993.

[7] T. Sasao. On the complexity of three-level logic circuits. In *Int. Workshop on Logic Synthesis*, pages 1-15, North Carolina, May 1989.

[8] E.V. Dubrova, D.M. Miller, and J.C. Muzio. AOXMIN: A three-level heuristic AND-OR-XOR minimizer for Boolean functions. In *Proc. 3rd International Workshop on the Applications of the Reed-Muller Expansion in Circuit Design*, pages 209-218, Oxford, U.K., September 1997.

[9] E.V. Dubrova, D.M. Miller, and J.C. Muzio. AOXMIN-MV: A heuristic algorithm for AND-OR-XOR Minimization. In *Proc. 4th International Workshop on the Applications of the Reed-Muller Expansion in Circuit Design*, 1999.

[10] D. Debnath and T. Sasao. An optimization of AND-OR-EXOR three-level networks. In *Proc. of IEEE-ACM Asia Pacific Design Automation Conference*, pages 545-550, January 1997.

[11] T. Sasao. EXMIN2: A simplification algorithm for Exclusive-OR-Sum-of-Products expressions for multiple valued input two-valued output functions. *IEEE Transaction on CAD*, 12(5):621-632, May 1993.

[12] N. Koda and T. Sasao. An upper bound on the number of products in minimum ESOPs. In *Workshop on Applications of the Reed-Muller Expansions in Circuit Design (Reed-Muller '95)*, Makuhari, Japan, August 1995.

[13] N. Koda and T. Sasao. A method to simplify multiple output AND-EXOR expressions. *Trans. IEICE*, J79-D-1(2):43-52, February 1996.

[14] I. Reed. A class of multiple-error-correcting codes and their decoding scheme. *IRETrans. on Inf. Theory*, PGIT-4:48-49, 1954. [15] M. Davio, Y. Deschamps, and A. Thayse. *Discrete and switching Functions*. George and McGraw-Hill, NY, 1978.

[16] A. Sarabi and M. Perkowski. Fast exact and quasi-minimal minimization of highly testable Fixed -Polarity AND/XOR

canonical networks. In *Proc. of IEEE-ACM Design Automation Conference*, volume 29th, pages 30-35, 1992.
[17] Tran A. Graphical method for the conversion of minterms to Reed-Muller coefficients and the minimization of exclusive-or switching circuits. In *IEE Proc. onComputer and Digital Technique*, volume 134, pages 93-99, 1987.
[18] Green D.H. Reed-Muller expansions of incompletely specified functions. *IEE Proc. on Computer and Digital Technique*, 134:228-236, 1987
[19] Khan M.M.H.A. and Alam M.S. Mapping of fixed polarity Reed-Muller coefficients from minterms and the minimization of fixed polarity Reed-Muller expressions. *Int.J. Electron*, 83:235-248, 1997.
[20] Khan M.M.H.A. and Alam M.S. Mapping of on-set fixed polarity Reed-Muller coefficients from on-set canonical sum of products coefficients and minimization of pseudo Reed-Muller expressions. *Int. J. Electron*, 86:225-268, 1999.
[21] Purwar S. An efficient method of computing generalized Reed-Muller expansion from the binary decision diagram. *IEEE Trnas. on Computers*, 40:1298-1301, 1991.
[22] S. Aborhey. Reed-Muller tree-based minimization of fixed polarity Reed-Muller expansions. *IEE Proc. on Computer and Digital Technique*, 148(2), March 2001.
[23] Unni Narayanan and C.L. Liu. Low power logic synthesis for XOR based circuits. In *Proc. of IEEE-ACM Int. Conference on CAD*, 1997.
[24] S. Thompson, P.Packan and M. Bohr. “MOS Scaling: Transistor Challenges for the 21st Century,” Intel Technology Journal, Q3,1998.
[25] K. Roy, S. C.Prasad. “Low-Power CMOS VLSI Circuit Design,”John Wiley & Sons, Inc, 2000.

Acknowledgement

This work was supported by SMDP-II project sponsored by Department of Information Technology, Government of India.

Santanu Chattopadhyay received his B.E. in Computer Science and Technology from Calcutta University, West Bengal, India in 1990 and M.Tech. in Computer and Information Technology from Indian Institute of Technology, Kharagpur, India in 1992. He completed his Ph.D. in Computer Science and Engineering from Indian Institute of Technology, Kharagpur, India in 1996. Currently, he is working as an Associate Professor in the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India. His research interests are VLSI circuit design and test, system-on-chip and network-on-chip design, low power system design.

Sambhu N. Pradhan received the B.E. and M.E. degrees in Electronics Engineering from the Kalyani University and the Bengal Engg. & Science University (Shibpur) in 2002 and 2004 respectively. Currently he is a research scholar in the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India. His area of research includes low-power VLSI design and testing.

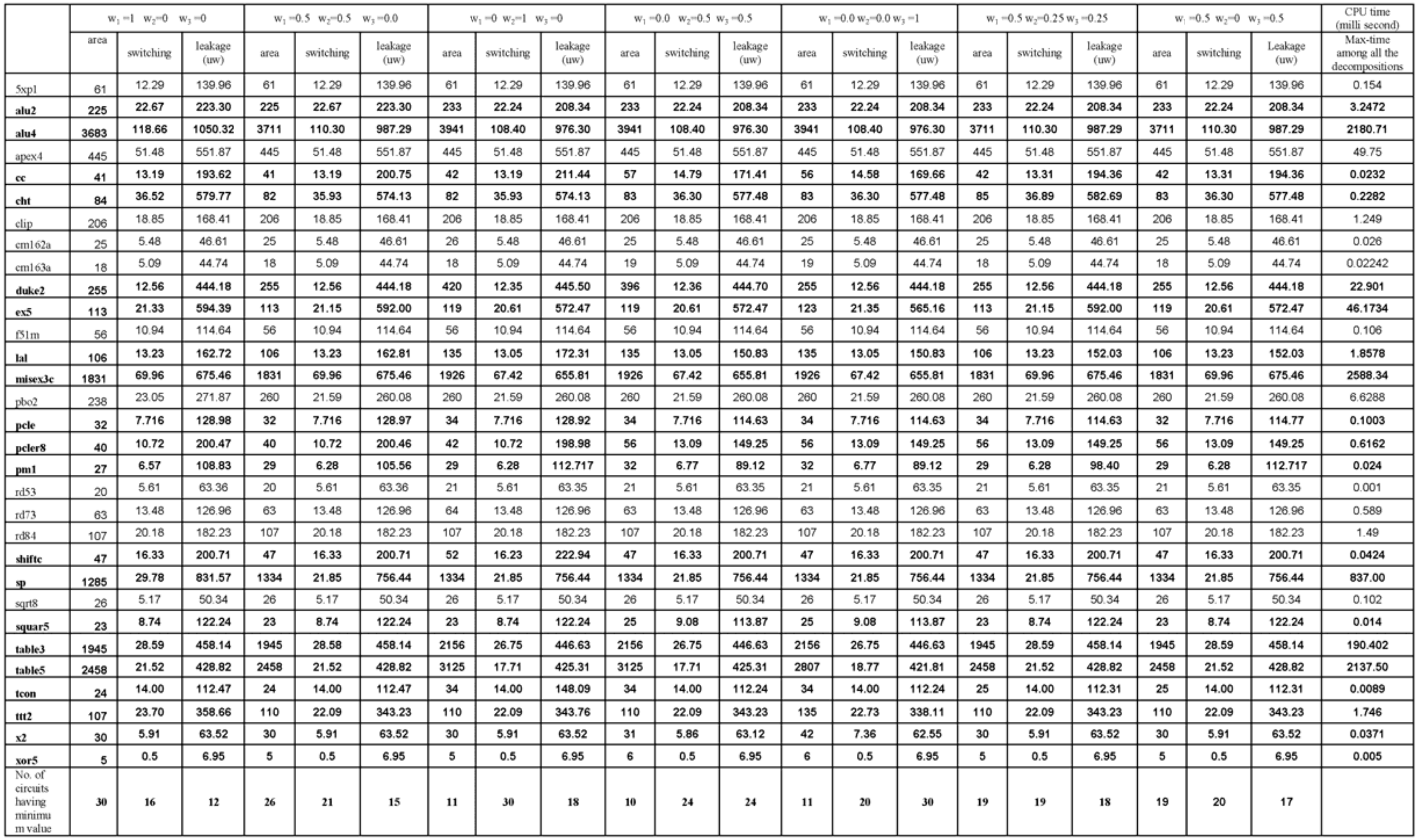
Table 4: area and switching activity comparison

Circuit	Inputs/outputs	N0. of product terms			Switching activity	
		Our approach (w_1 =1 w_2=0 w_3 =0)	Min-Area RMTree [22]	Heuristic [22]	Exhaustive min-switching	Our approach (w_1 =0 w_2=1 w_3 =0)
5xp1	7/10	61	61	61	12.2945	12.29
alu2	10/6	225	-	-	22.2382	22.24
alu4	14/8	3683	3683	4334	-	108.40
apex4	9/19	445	445	512	-	51.48
cc	21/20	41	-	-	-	13.19
cht	47/36	84	-	-	-	35.93
clip	9/5	206	206	206	18.8465	18.85
cm162a	14/5	25	-	-	5.4842	5.48
cm163a	16/5	18	-	-	5.0951	5.09
duke2	22/29	255	255	267	-	12.35
ex5	8/63	113	113	171	20.6075	20.61
f51m	8/8	56	56	73	10.9432	10.94
lal	26/19	106	-	-	-	13.05
misex3c	14/14	1831	1831	1890	-	67.42
pbo2	15/15	238	-	-	21.5907	21.59
pcle	19/9	32	-	-	-	7.716
pcler8	27/17	40	-	-	-	10.72
pm1	16/13	27	-	-	6.2778	6.28
rd53	5/3	20	20	20	5.6093	5.61
rd73	7/3	63	63	63	13.4765	13.48
rd84	8/4	107	107	164	20.1814	20.18
shiftc	8/21	47	-	-	16.2329	16.23
sp	16/46	1285	-	-	-	21.85
sqrt8	8/4	26	-	-	5.1756	5.17
squar5	5/8	23	22	22	8.7422	8.74
table3	14/14	1945	1945	2203	-	26.75
table5	17/15	2458	2458	2807	-	17.71
tcon	17/16	24	-	-	-	14.00
ttt2	24/21	107	-	-	-	22.09
x2	10/7	30	-	-	5.8684	5.91
xor5	5/1	5	5	5	0.5	0.5

Table 6: Average area, switching and leakage value

Weightage	Area	Switching	leakage
w_1 =1.0 w_2=0.0 w_3 =0.0	1	1.035	1.048
w_1 =0.5 w_2=0.5 w_3=0.0	1.007	1.014	1.039
w_1 =0.0 w_2=1.0 w_3 =0.0	1.080	1	1.054
w_1 =0.0 w_2=0.5 w_3=0.5	1.111	1.015	1.002
w_1 =0.0 w_2=0.0 w_3 =1.0	1.109	1.027	1
w_1 =0.5 w_2=0.25 w=0.25	1.028	1.021	1.016
w_1 =0.5 w_2=0.0 w_3 =0.5	1.027	1.019	1.020

Table 5: area, switching and leakage for different weights



	w_1 =1 w_2=0 w_3 =0			w_1 =0.5 w_2=0.5 w_3 =0.0			w_1 =0 w_2=1 w_3 =0			w_1 =0.0 w_2=0.5 w_3 =0.5			w_1 =0.0 w_2=0.0 w_3 =1			w_1 =0.5 w_2=0.25 w_3 =0.25			w_1 =0.5 w_2=0 w_3 =0.5			CPU time (milli second)
	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	Leakage (uw)	Max-time among all the decompositions
5xp1	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	0.154
alu2	225	22.67	223.30	225	22.67	223.30	233	22.24	208.34	233	22.24	208.34	233	22.24	208.34	233	22.24	208.34	233	22.24	208.34	3.2472
alu4	3683	118.66	1050.32	3711	110.30	987.29	3941	108.40	976.30	3941	108.40	976.30	3941	108.40	976.30	3711	110.30	987.29	3711	110.30	987.29	2180.71
apex4	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	49.75
cc	41	13.19	193.62	41	13.19	200.75	42	13.19	211.44	57	14.79	171.41	56	14.58	169.66	42	13.31	194.36	42	13.31	194.36	0.0232
cht	84	36.52	579.77	82	35.93	574.13	82	35.93	574.13	83	36.30	577.48	83	36.30	577.48	85	36.89	582.69	83	36.30	577.48	0.2282
clip	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	1.249
cm162a	25	5.48	46.61	25	5.48	46.61	26	5.48	46.61	25	5.48	46.61	25	5.48	46.61	25	5.48	46.61	25	5.48	46.61	0.026
cm163a	18	5.09	44.74	18	5.09	44.74	18	5.09	44.74	19	5.09	44.74	19	5.09	44.74	18	5.09	44.74	18	5.09	44.74	0.02242
duke2	255	12.56	444.18	255	12.56	444.18	420	12.35	445.50	396	12.36	444.70	255	12.56	444.18	255	12.56	444.18	255	12.56	444.18	22.901
ex5	113	21.33	594.39	113	21.15	592.00	119	20.61	572.47	119	20.61	572.47	123	21.35	565.16	113	21.15	592.00	119	20.61	572.47	46.1734
f51m	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	0.106
lal	106	13.23	162.72	106	13.23	162.81	135	13.05	172.31	135	13.05	150.83	135	13.05	150.83	106	13.23	152.03	106	13.23	152.03	1.8578
misex3c	1831	69.96	675.46	1831	69.96	675.46	1926	67.42	655.81	1926	67.42	655.81	1926	67.42	655.81	1831	69.96	675.46	1831	69.96	675.46	2588.34
pbo2	238	23.05	271.87	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	6.6288
pcle	32	7.716	128.98	32	7.716	128.97	34	7.716	128.92	34	7.716	114.63	34	7.716	114.63	34	7.716	114.63	32	7.716	114.77	0.1003
pcler8	40	10.72	200.47	40	10.72	200.46	42	10.72	198.98	56	13.09	149.25	56	13.09	149.25	56	13.09	149.25	56	13.09	149.25	0.6162
pm1	27	6.57	108.83	29	6.28	105.56	29	6.28	112.717	32	6.77	89.12	32	6.77	89.12	29	6.28	98.40	29	6.28	112.717	0.024
rd53	20	5.61	63.36	20	5.61	63.36	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	0.001
rd73	63	13.48	126.96	63	13.48	126.96	64	13.48	126.96	63	13.48	126.96	63	13.48	126.96	63	13.48	126.96	63	13.48	126.96	0.589
rd84	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	1.49
shiftc	47	16.33	200.71	47	16.33	200.71	52	16.23	222.94	47	16.33	200.71	47	16.33	200.71	47	16.33	200.71	47	16.33	200.71	0.0424
sp	1285	29.78	831.57	1334	21.85	756.44	1334	21.85	756.44	1334	21.85	756.44	1334	21.85	756.44	1334	21.85	756.44	1334	21.85	756.44	837.00
sqrt8	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	0.102
squar5	23	8.74	122.24	23	8.74	122.24	23	8.74	122.24	25	9.08	113.87	25	9.08	113.87	23	8.74	122.24	23	8.74	122.24	0.014
table3	1945	28.59	458.14	1945	28.58	458.14	2156	26.75	446.63	2156	26.75	446.63	2156	26.75	446.63	1945	28.59	458.14	1945	28.59	458.14	190.402
table5	2458	21.52	428.82	2458	21.52	428.82	3125	17.71	425.31	3125	17.71	425.31	2807	18.77	421.81	2458	21.52	428.82	2458	21.52	428.82	2137.50
tcon	24	14.00	112.47	24	14.00	112.47	34	14.00	148.09	34	14.00	112.24	34	14.00	112.24	25	14.00	112.31	25	14.00	112.31	0.0089
ttt2	107	23.70	358.66	110	22.09	343.23	110	22.09	343.76	110	22.09	343.23	135	22.73	338.11	110	22.09	343.23	110	22.09	343.23	1.746
x2	30	5.91	63.52	30	5.91	63.52	30	5.91	63.52	31	5.86	63.12	42	7.36	62.55	30	5.91	63.52	30	5.91	63.52	0.0371
xor5	5	0.5	6.95	5	0.5	6.95	5	0.5	6.95	6	0.5	6.95	6	0.5	6.95	5	0.5	6.95	5	0.5	6.95	0.005
No. of circuits having minimum value	30	16	12	26	21	15	11	30	18	10	24	24	11	20	30	19	19	18	19	20	17	

Table 5: area, switching and leakage for different weights

	w_1 =1 w_2=0 w_3 =0			w_1 =0.5 w_2=0.5 w_3 =0.0			w_1 =0 w_2=1 w_3 =0			w_1 =0.0 w_2=0.5 w_3 =0.5			w_1 =0.0 w_2=0.0 w_3 =1			w_1 =0.5 w_2=0.25 w_3 =0.25			w_1 =0.5 w_2=0 w_3 =0.5			CPU time (milli second)
	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	leakage (uw)	area	switching	Leakage (uw)	Max-time among all the decompositions
5xp1	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	61	12.29	139.96	0.154
alu2	**225**	**22.67**	**223.30**	**225**	**22.67**	**223.30**	**233**	**22.24**	**208.34**	**233**	**22.24**	**208.34**	**233**	**22.24**	**208.34**	**233**	**22.24**	**208.34**	**233**	**22.24**	**208.34**	**3.2472**
alu4	**3683**	**118.66**	**1050.32**	**3711**	**110.30**	**987.29**	**3941**	**108.40**	**976.30**	**3941**	**108.40**	**976.30**	**3941**	**108.40**	**976.30**	**3711**	**110.30**	**987.29**	**3711**	**110.30**	**987.29**	**2180.71**
apex4	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	445	51.48	551.87	49.75
cc	**41**	**13.19**	**193.62**	**41**	**13.19**	**200.75**	**42**	**13.19**	**211.44**	**57**	**14.79**	**171.41**	**56**	**14.58**	**169.66**	**42**	**13.31**	**194.36**	**42**	**13.31**	**194.36**	**0.0232**
cht	**84**	**36.52**	**579.77**	**82**	**35.93**	**574.13**	**82**	**35.93**	**574.13**	**83**	**36.30**	**577.48**	**83**	**36.30**	**577.48**	**85**	**36.89**	**582.69**	**83**	**36.30**	**577.48**	**0.2282**
clip	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	206	18.85	168.41	1.249
cm162a	25	5.48	46.61	25	5.48	46.61	26	5.48	46.61	25	5.48	46.61	25	5.48	46.61	25	5.48	46.61	25	5.48	46.61	0.026
cm163a	18	5.09	44.74	18	5.09	44.74	18	5.09	44.74	19	5.09	44.74	19	5.09	44.74	18	5.09	44.74	18	5.09	44.74	0.02242
duke2	**255**	**12.56**	**444.18**	**255**	**12.56**	**444.18**	**420**	**12.35**	**445.50**	**396**	**12.36**	**444.70**	**255**	**12.56**	**444.18**	**255**	**12.56**	**444.18**	**255**	**12.56**	**444.18**	**22.901**
ex5	**113**	**21.33**	**594.39**	**113**	**21.15**	**592.00**	**119**	**20.61**	**572.47**	**119**	**20.61**	**572.47**	**123**	**21.35**	**565.16**	**113**	**21.15**	**592.00**	**119**	**20.61**	**572.47**	**46.1734**
f51m	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	56	10.94	114.64	0.106
lal	**106**	**13.23**	**162.72**	**106**	**13.23**	**162.81**	**135**	**13.05**	**172.31**	**135**	**13.05**	**150.83**	**135**	**13.05**	**150.83**	**106**	**13.23**	**152.03**	**106**	**13.23**	**152.03**	**1.8578**
misex3c	**1831**	**69.96**	**675.46**	**1831**	**69.96**	**675.46**	**1926**	**67.42**	**655.81**	**1926**	**67.42**	**655.81**	**1926**	**67.42**	**655.81**	**1831**	**69.96**	**675.46**	**1831**	**69.96**	**675.46**	**2588.34**
pbo2	238	23.05	271.87	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	260	21.59	260.08	6.6288
pcle	**32**	**7.716**	**128.98**	**32**	**7.716**	**128.97**	**34**	**7.716**	**128.92**	**34**	**7.716**	**114.63**	**34**	**7.716**	**114.63**	**34**	**7.716**	**114.63**	**32**	**7.716**	**114.77**	**0.1003**
pcler8	**40**	**10.72**	**200.47**	**40**	**10.72**	**200.46**	**42**	**10.72**	**198.98**	**56**	**13.09**	**149.25**	**56**	**13.09**	**149.25**	**56**	**13.09**	**149.25**	**56**	**13.09**	**149.25**	**0.6162**
pm1	**27**	**6.57**	**108.83**	**29**	**6.28**	**105.56**	**29**	**6.28**	**112.717**	**32**	**6.77**	**89.12**	**32**	**6.77**	**89.12**	**29**	**6.28**	**98.40**	**29**	**6.28**	**112.717**	**0.024**
rd53	20	5.61	63.36	20	5.61	63.36	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	21	5.61	63.35	0.001
rd73	63	13.48	126.96	63	13.48	126.96	64	13.48	126.96	63	13.48	126.96	63	13.48	126.96	63	13.48	126.96	63	13.48	126.96	0.589
rd84	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	107	20.18	182.23	1.49
shiftc	**47**	**16.33**	**200.71**	**47**	**16.33**	**200.71**	**52**	**16.23**	**222.94**	**47**	**16.33**	**200.71**	**47**	**16.33**	**200.71**	**47**	**16.33**	**200.71**	**47**	**16.33**	**200.71**	**0.0424**
sp	**1285**	**29.78**	**831.57**	**1334**	**21.85**	**756.44**	**1334**	**21.85**	**756.44**	**1334**	**21.85**	**756.44**	**1334**	**21.85**	**756.44**	**1334**	**21.85**	**756.44**	**1334**	**21.85**	**756.44**	**837.00**
sqrt8	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	26	5.17	50.34	0.102
squar5	**23**	**8.74**	**122.24**	**23**	**8.74**	**122.24**	**23**	**8.74**	**122.24**	**25**	**9.08**	**113.87**	**25**	**9.08**	**113.87**	**23**	**8.74**	**122.24**	**23**	**8.74**	**122.24**	**0.014**
table3	**1945**	**28.59**	**458.14**	**1945**	**28.58**	**458.14**	**2156**	**26.75**	**446.63**	**2156**	**26.75**	**446.63**	**2156**	**26.75**	**446.63**	**1945**	**28.59**	**458.14**	**1945**	**28.59**	**458.14**	**190.402**
table5	**2458**	**21.52**	**428.82**	**2458**	**21.52**	**428.82**	**3125**	**17.71**	**425.31**	**3125**	**17.71**	**425.31**	**2807**	**18.77**	**421.81**	**2458**	**21.52**	**428.82**	**2458**	**21.52**	**428.82**	**2137.50**
tcon	**24**	**14.00**	**112.47**	**24**	**14.00**	**112.47**	**34**	**14.00**	**148.09**	**34**	**14.00**	**112.24**	**34**	**14.00**	**112.24**	**25**	**14.00**	**112.31**	**25**	**14.00**	**112.31**	**0.0089**
ttt2	**107**	**23.70**	**358.66**	**110**	**22.09**	**343.23**	**110**	**22.09**	**343.76**	**110**	**22.09**	**343.23**	**135**	**22.73**	**338.11**	**110**	**22.09**	**343.23**	**110**	**22.09**	**343.23**	**1.746**
x2	**30**	**5.91**	**63.52**	**30**	**5.91**	**63.52**	**30**	**5.91**	**63.52**	**31**	**5.86**	**63.12**	**42**	**7.36**	**62.55**	**30**	**5.91**	**63.52**	**30**	**5.91**	**63.52**	**0.0371**
xor5	**5**	**0.5**	**6.95**	**5**	**0.5**	**6.95**	**5**	**0.5**	**6.95**	**6**	**0.5**	**6.95**	**6**	**0.5**	**6.95**	**5**	**0.5**	**6.95**	**5**	**0.5**	**6.95**	**0.005**
No. of circuits having minimum value	**30**	**16**	**12**	**26**	**21**	**15**	**11**	**30**	**18**	**10**	**24**	**24**	**11**	**20**	**30**	**19**	**19**	**18**	**19**	**20**	**17**	