

A Power Driven Two-Level Logic Optimizer

Jyh-Mou Tseng Jing-Yang Jou

Department of Electronics Engineering
National Chiao Tung University
Hsinchu, Taiwan 300, Republic of China
jyjou@bestmap.ee.nctu.edu.tw

Abstract - In this paper we present Boolean techniques for reducing the power consumption in two-level combinational circuits. The two-level logic optimizer performs the logic minimization for low power targeting static PLA, general logic gates and dynamic PLA implementations. We modify Espresso algorithm by adding our heuristics that bias the logic minimization toward lowering the power dissipation. In our heuristics, signal probabilities and transition densities are two important parameters. The experimental results are promising.

1. Introduction

Recently, some methods on optimizing combinational circuits for low power are proposed. A method exploiting don't-care set to reduce transition density was presented in [2]. The method proposed in [3] modified Espresso[4] package by changing some heuristics of sub-algorithms of Espresso. However they only consider the way on expansion of cubes. *Power Prime Implicant*[6] method is used to identify the set of all implicants that are sufficient and necessary for obtaining a minimum power solution. However, the method is only based on signal probability. All the theorems proposed in [6] are not valid when the transition density is considered.

In order to come out a practical solution, the heuristic-based approach is adopted in this paper. We modify the Espresso package similar to [3]. However, we consider the situation where some input signals switching simultaneously which is not considered in [3]. A good heuristic method to select the directions of expansion is developed. The don't care set is also exploited in our approach to reduce the transition density of cubes. In reduction of cubes, we come out a new heuristic in ordering of cubes. We formulate the irredundant covering problem of cubes as a minimum covering problem with weighted function. The experimental results on static PLA, dynamic PLA and general AND-OR implementations are promising.

This paper divides into five sections. In section 2, the method of power estimation including power model and calculation of transition density are presented. Our proposed algorithms will be shown in section 3. In section 4, the results of our power reduction techniques for PLA and AND-OR implementations will be discussed. The conclusion of the paper is in section 5.

2. Power Estimation

Signal probability and transition density are the two popular metrics used in statistical power estimation

techniques. The signal probability of a signal $x(t)$ denoted as P_x^1 is the expected value of a logical signal being equal to one at time t . The transition density D_x is the expected number of transition happened in a clock period.

Because reconvergent fanout usually exist at outputs, the spatial correlation among inputs must be considered for accurate estimation of signal probability. In order to account for the possibility of different inputs switching at the same time, in the paper, we will thus use the equation proposed by Huzefa Mehta[9] to calculate the transition density for better accuracy. The method has been implemented by using BDD[8] method.

2.1 The Static PLA Model

A typical PLA circuit has two planes, namely, AND plane and OR plane. Assume m is the number of product terms, n is the number of primary inputs and l is the number of primary outputs. Our objective is to minimize the power consumption written as

$$P_{total} = \sum_{i=1}^n \frac{1}{2} D_{I_i} C_{I_i} V^2 f + \sum_{j=1}^m \frac{1}{2} D_{p_j} C_{p_j} V^2 f + \sum_{k=1}^l \frac{1}{2} D_{O_k} C_{O_k} V^2 f$$

In P_{total} , the power supply voltage V is fixed, and the transition density D_{I_i} of every input I_i is given.

C_{I_i} , C_{p_j} and C_{O_k} are capacitances of input-line i , product-line j and output-line k , respectively. According to [5], the AND plane of PLAs consist of long polysilicon lines which consume a large proportion of the power dissipated in PLAs. The increase in the number of connections in the AND plane does not dramatically alter the capacitance of each line[10]. Therefore, the reduction of power dissipation in product-lines and output-lines becomes the main objective. In order to simplify our problem, we assume the capacitances of product-lines and output-lines are proportional to the length of long polysilicon. The cost function of power minimization is stated as follow :

$$Cost(f) = \sum_j^m (2 \cdot n + 1) \cdot D_{p_j} + \sum_k^l m \cdot D_{O_k} .$$

2.2 The Dynamic PLA Model

The structure of dynamic PLA is similar to the static PLA. The difference is the product lines and the output lines are precharged to high in dynamic PLA. The power dissipation occurs when a line is from high to low or from low to low. Therefore, the cost function is

$$Cost(f) = \sum_j^m (2 \cdot n + 1) \cdot P_{p_j}^0 + \sum_k^l m \cdot P_{O_k}^0.$$

2.3 The AND-OR Model

This model is somewhat in the technology independent sense. Based on this power model, we show in the later section that the two-level logic minimization for low power will eventually results in the better results after the complete synthesis tasks are performed which include multi-level logic minimization and technology mapping.

We assume the capacitance of a node is proportional to the number of fanout it drives. The cost function is

$$Cost(f) = \sum_i^n D_{I_i} \cdot L_{I_i} + \sum_j^m D_{p_j} L_{p_j} + \sum_k^l D_{O_k} L_{O_k}$$

where L_a is the fanout number of node a . We assume L_{O_k} is equal to 1.

3. Logic Minimization Techniques for Low Power

Espresso is a well-known two-level logic minimizer for low area. The objective of Espresso is to minimize the number of product terms of the cover. Espresso consists of seven routines. The essential_prime, expand, irredundant and reduce routines are the four major routines we modify.

3.1 Essential Prime

When the target of logic minimization is for area, the essential prime implicants must be part of any cover. If the target of the minimizer is for low power, the essential prime implicants may not be essential any more. In order to increase the opportunity of reduction, we don't employ the procedure Essential_prime in our algorithm under PLA implementation.

3.2 Expand

3.2.1 The Ordering of Cubes for EXPAND

The order in which the cubes are expanded is important because it affects which cubes will be covered and dropped. If a cube with higher transition density is covered by other cubes, we can drop it and the corresponding reduction of transition density is larger. Therefore we process the cubes according to the increasing weights where the weight of each cube is shown in Table 1.

Implementations	W_c
Static PLA	D_c
Dynamic PLA	P_c^0
AND - OR	$D_c + \sum_{\forall k \in I_c} D_k$

Table 1 The weight of each cube for the ordering of expand.

3.2.2 Select Directions of Expansion

Given a function F with its ON-set F^{ON} and OFF-set F^{OFF} , the matrix $M(F^{ON})$ associated with a cover F^{ON} is the matrix obtained by stacking the row vectors representing each of the cubes of F^{ON} . Similarly $M(F^{OFF})$ is a matrix

representation of F^{OFF} . The **blocking matrix B** of a cube c is a 0-1 matrix and its elements is defined as

$$B_{ij} = \begin{cases} 1 & \text{if } \{(c_j = 1) \text{ and } (M(F^{OFF})_{ij} = 0) \text{ or} \\ & (c_j = 0) \text{ and } (M(F^{OFF})_{ij} = 1) \} \\ 0 & \text{otherwise.} \end{cases}$$

The notation $c_j = 1$ ($c_j = 0$) means that the cube to be expanded contains the literal x_j (\bar{x}_j). The **lowering set L** is defined as a set of variables whose corresponding columns have at least a 1 in each row of B. The lowering set defines the expanded cubes c^+ by

$$c^+(L, c)_j = \begin{cases} c_j, & j \in L, \\ 2 & \text{otherwise.} \end{cases}$$

Raising set is the variables that are not in the lowering set.

The expanded cube c^+ is a implicant of F^{ON} if and only if L is a column covering of B. Espresso makes the number of entries in L as small as possible such that the opportunity of the corresponding directions of expansion is larger. However we select L such that the corresponding expanded cube has lower transition density. Because a primary input with high transition density has larger effect on its output, we like to select those variables with higher transition densities to be expanded. In other words, we select L such that the sum of transition densities of variables in L is minimum. Therefore we give each column of B a weight and solve the **weighted covering problem**. The weight of each column is the transition density of its corresponding variable under AND-OR and static PLA model. Under dynamic PLA model, the weight of each column of B is related to its signal probability. Assume the selected lowering set is $L = \{l_1, l_2, \dots, l_k\}$, then the

$P_{c^+}^0$ of the expanded cube is $1 - \prod_{j=1}^k P_{l_j}^1$. In order to use the method of weighted covering problem, we use the logarithm of $P_{l_i}^1$ of input l_i as its weight.

3.2.3 Expansion Does Not Include Don't Care Set

After the directions of expansion are selected, the variables in raising set can always be raised. However the expansion of any particular variable may not be always advantageous for reduction of transition density. The reduction of transition density includes two parts, one is the difference between the original cube and the expanded cube, the other is the transition densities of cubes covered by the expanded cube. $R(c^+, c)$ can be computed by as following :

$$R(c^+, c) = D_c - D_{c^+} + \sum_{\forall c_i \subset c^+} D_{c_i}.$$

In the procedure of expansion, we expand one direction at a time in descending order of transition densities of variables in raising set, and then check whether the expansion is advantageous or not. If $R \geq 0$, the expansion is advantageous. On the other hand, if $R < 0$, the expansion is disadvantageous and we don't expand in that direction.

Now we consider the dynamic PLA model. We assert that

the expansion is always advantageous under dynamic PLA model. The proof is omitted here due to the space limitation.

Theorem 1 : Given a n-input AND gate with signal probabilities and transition densities of its inputs $\vec{X} = (X_1, X_2, \dots, X_n)$. The expansion of a variable in raising set can always reduce the switching activity under dynamic PLA model.

Furthermore, The reduction of transition density under AND-OR model includes the difference between D_{c^+} and D_c , the transition densities of cubes covered by the expanded cube c^+ , and the reduction of the load of the raised input x_i and the corresponding reduction is D_{x_i} . Therefore, the reduction $R(c^+, c)$ is

$$R(c^+, c) = D_c - D_{c^+} + \sum_{\forall c_j \subset c^+} (D_{c_j} + \sum_{\forall k \in I_j} D_k) + D_{x_i}.$$

3.2.4 Expansion Includes Don't Care Set

The don't care set is usually used to reduce the transition density. The transition density of primary output f will change to f^+ when the don't care set is used for minimization. The reduction $R(f^+, f)$ is equal to the difference between D_{f^+} and D_f . Therefore, when the don't care set is used for minimization, the reduction $R(c^+, c)$ is modified by adding the term $D_f - D_{f^+}$.

3.3 Irredundant Cover

Irredundant cover selects subset of cover, that is still a cover. The objective of irredundant covers for area is to reduce the maximum number of cubes in the cover. The removal of a maximum number of redundant implicants requires considering the mutual covering relations among implicants. The problem has been formulated as a covering problem [4]. When the target is low power, the transition density of each cube should be paid attention to. Hence the weighted covering problem is used to solve this problem.

In AND-OR implementation, we must consider the effect of increasing capacitances of primary inputs due to the increase of literals. Therefore, the cost of a cube in AND-OR model includes transition density of its output and its primary inputs, i.e., $D_{c_i} \times L_{c_i} + \sum_{I_j \in I_{c_i}} D_{I_j}$. The set I_{c_i} contains those

primary inputs that are the inputs of cube c_i . On the other hand, the weight of a cube in static PLA model only includes its transition density D_{c_i} , and the weight of a cube in dynamic PLA model is $P_{c_i}^0$.

3.4 Reduce

The advantage of reduce is to increase the opportunities for expansion cubes into different selections to get a better solution. In order to increase the opportunity of reduction, we removed the procedure of essential prime as described before. Furthermore, we also change the ordering of reduce. We alternate between two ordering methods : (1) Order the cubes in descending order of the transition density. (2) Order the cubes in ascending order of distance from the largest cube;

break ties by ordering cubes of equal distance in descending order of size. The first method will process the cube with higher transition density first. It has higher chance to lower transition density.

4. Experimental Result And Discussion

The version of algorithms and heuristics discussed in section, called **Espower**, has been implemented in C and integrated into the Espresso package. In this section we will describe the processes of our experiments and the results.

4.1 The Static PLA Model

In this section we will show the experimental result under static PLA model. We conducted experiments on a subset of circuits from the MCNC benchmark. We use the random generator to generate one thousand signal probabilities set for each circuit. We first generate the signal probability between 0.1 and 0.9 for each input, then generate the transition density according to the rules shown below

$$D_j \leq 2 \times P_j^1, \quad D_j \leq 2 \times P_j^0.$$

The results are shown in Table 2. The values shown in Table 2 are the average of one thousand simulations. On average, the cost of power has 11.68 % reduction.

Circuit	Espresso			Espower		
	Power		Area	Power		Area
	AND	OR	Cube	AND	OR	Cube
5xp1	152.0	216.3	65	109.0	212.6	63.899
9sym	24.1	24.1	86	23.2	24.4	87.000
Z5xp1	122.3	204.4	65	114.1	202.7	64.443
b12	218.7	72.2	43	197.8	72.2	43.000
bw	43.7	148.7	22	38.3	148.7	22.005
clip	124.5	218.6	120	96.5	224.7	123.333
misex1	28.9	21.9	12	17.0	21.9	12.000
rd53	26.0	29.3	31	16.9	29.3	31.000
rd73	103.6	129.9	127	24.5	129.9	127.000
rd84	95.7	301.2	255	31.6	301.2	255.000
sao2	45.9	36.3	58	22.3	36.3	58.000
squar5	49.5	53.9	25	42.4	54.4	25.242
Total	1024.9	1456.8	909	733.6	1458.3	911.922

(a)

	Espower vs. Espresso			
	AND	OR	AND + OR	Cubes
Saving	+ 28.42%	- 0.1%	+ 11.68%	- 0.32%

(b)

Table 2 The experimental results under static PLA model.

4.2 The Dynamic PLA Model

Table 3 shows the experimental results for the dynamic model. We have 1.44 % improvement in power reduction compared to Espresso package. The corresponding reduction of area is 0.55 %. The reduction of cost in the AND plane is only 0.79% because the expansion of a variable is always advantageous for both area and power reductions as proved in Theorem 1. On the other hand, the reduction of OR plane is 4.01%. The reduction is obtained from the reduction of cubes and the use of Don't care set. The power consumption of

AND-plane is larger because the signal probability $P_{p_j}^0$ is almost equal to 1 when the number of inputs is large.

	Espanner vs. Espresso			
	AND	OR	AND +	Cubes
Saving	+0.79%	+4.01%	+1.44%	+0.55%

Table 3 The experimental results under dynamic PLA model.

4.3 The AND-OR Model

Because two-level logic minimization is only one of the important operations used in the multi-level logic minimization procedure. There are still a lot of other operations needed to produce a good multi-level logic implementation. In order to measure the contribution of our two-level logic optimization for low power on the power reduction of the final multi-level logic with little distortion, we only use the gcx (common cube extraction) command after Simplify or Espanner to make the circuit in multi-level forms. The procedure of experiment is shown in Table 4. The purpose of using this model is to demonstrate the impact of our method on the general multi-level logic synthesis environment. Table 5 is the experimental result. We have 8.27 % improvement in power consumption compared to the Simplify package. The corresponding reduction of area is 10.37 %. If we use general AND-OR to implement a circuit, the increase of literals will also increase the input capacitance of the circuit. The transition densities of primary inputs are usually larger than those of internal nodes and output nodes. The expansion of a variable is thus usually advantageous not only for area but also for power because it reduces the load of the expanded variable. The main source for lowering power is in selecting the directions for expansion. The procedure will make larger impact when the transition densities of primary inputs have larger variances among each other. The area of Simplify is the largest and its power consumption is also the largest. In fact, the power consumption of the inputs accounts for over 70% of the total power. Therefore, when we use techniques for lowering the transition density, the increase of capacitance is also an important consideration.

- ```

=====
1. read_pla
2. simplify or espanner
3. gcx
4. read_library Nand-Nor.genlib
5. map
6. power_estimate
=====

```

Table 4 The procedure of experiment.

|       | Simplify(Simp) |        | Espanner (Espow) |        | Simp. vs. Espo. |       |
|-------|----------------|--------|------------------|--------|-----------------|-------|
|       | Area           | Power  | Area             | Power  | Area            | Power |
| 5xp1  | 212            | 3237.0 | 215              | 3136.8 | 3.10            | 3.10% |
| 9sym  | 420            | 5665.5 | 423              | 5384.2 | -0.714%         | 4.97% |
| Z5xp1 | 212            | 3207.2 | 208              | 3140.0 | 1.887%          | 2.10% |
| b12   | 165            | 2696.0 | 161              | 2691.7 | 2.424%          | 0.16% |

|              |             |                |             |                |               |              |
|--------------|-------------|----------------|-------------|----------------|---------------|--------------|
| bw           | 270         | 3742.5         | 259         | 3828.9         | 4.074%        | -2.31%       |
| clip         | 417         | 5880.7         | 377         | 5144.2         | 9.592%        | 12.52%       |
| misex1       | 88          | 1272.5         | 84          | 1192.1         | 4.545%        | 6.32%        |
| rd53         | 90          | 1189.4         | 90          | 1215.2         | 0%            | -2.17%       |
| rd73         | 226         | 2969.8         | 219         | 2867.4         | 3.097%        | 3.45%        |
| rd84         | 741         | 6789.5         | 466         | 4807.0         | 37.112%       | 29.20%       |
| sao2         | 262         | 4107.0         | 269         | 3983.1         | -2.672%       | 3.02%        |
| squar5       | 108         | 1518.6         | 107         | 1390.0         | 0.926%        | 8.47%        |
| <b>Total</b> | <b>3211</b> | <b>42275.7</b> | <b>2878</b> | <b>38780.6</b> | <b>10.37%</b> | <b>8.27%</b> |

Table 5 The experimental results under AND-OR model.

## 5. Conclusion

In this paper we have presented Boolean techniques to reduce power consumption of Boolean function in the sum of the products form and have very good results in static PLA and AND-OR implementation. In dynamic PLA implementation, we also prove that the power reduction and area reduction have close relationship. These results will help us to incorporate the techniques for power reduction into multi-level logic synthesis packages.

## Reference

- [1] Anantha P. Chandrakasan, Samuel Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design," IEEE J. Solid-State Circuits, vol. 27, No. 4, pp. 473-483 Apr. 1992.
- [2] A. Shen, S. Devadas, A. Ghosh, and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of Combinational Logic Circuits," Proceedings of the Intl Conference on CAD, pp.402-407, Nov. 1992.
- [3] R. Iris Bahar, Fabio Somenzi, "Boolean Techniques for Low Power Driven Re-Synthesis," ICCAD, Nov. 1995.
- [4] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis," Boston, Massachusetts: Kluwer Academic Publishers, 1984.
- [5] R. Hossian and A. Albicki, "Low Power via Reduced Switching Activity and its Application to PLAs," Proceeding of IEEE International ASIC Conference, pp. 100-103. Sept. 1994.
- [6] Sasan Iman, Massoud Pedram, "Two-Level Logic Minimization for Low Power," ICCAD, Nov. 1995.
- [7] A. Ghosh, S. Devadas, J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," 29th ACM/IEEE Design Automation Conf., pp. 253-259, June, 1992.
- [8] F. Najm, "Transition Density, A Stochastic Measure of Activity in Digital Circuits," Proc. 28th ACM/IEEE Design Automation Conf., pp. 644-649, June, 1991.
- [9] Huzefa mehta, "Accurat Estimation of Combination Circuit Activity," Proc. 32th ACM/IEEE Design Automation Conf., pp. 618-622, June, 1995.
- [10] A. Tragi, "Energy Complexity and Delay Comparison of Dynamic and Static PLA Design Style," Advanced Research in VLSI, Paul Cosleben (Ed), pp. 371-390, Stanford, CA, 1987.