A Novel Programmable Logic Array Structure with Low Energy Consumption

by

Shaohua Yuan

B.A.Sc., the University of British Columbia, 2006

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Applied Science

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

The University of British Columbia
(Vancouver)

April 2009

ABSTRACT

As modern integrated circuit design pushes further into the deep submicron era, the pseudo-random design structures become more and more difficult to fabricate and result in a yield reduction. To deal with process limitations due to photolithographic resolution, standard cell ASICs (SC-ASIC) may eventually need to be replaced by a more structured form of logic, such as programmable logic array (PLA). However, in order to compete with SC-ASIC, the PLA needs to be improved on delay, power and energy consumption.

Here, we will explore a novel PLA structure by combining one design having the best delay performance with a "product line merging process" to minimize power. We have simulated the different approaches on two sets of benchmark circuits using HSpice. As a result, the combination of the two methods produces the highest energy reduction among all prior PLA designs.

Next, algorithms are introduced for partitioning multi-output PLAs into smaller size sub-PLAs to further reduce delay and area. Finally, the performance of the improved PLA is compared with SC-ASIC. We found that the new PLA is faster or at least has the same speed as SC-ASIC implementation. However, the energy consumption is still more than twice as much as SC-ASIC design.

TABLE OF CONTENTS

ABSTRACT ii
TABLE OF CONTENTS iii
LIST OF TABLES iv
LIST OF FIGURES v
ACRONYMS vi
ACKNOWLEDGEMENTS vii
Chapter 1 INTRODUCTION 1
1.1 Research Motivation 1
1.2 Research Goals 5
1.3 Thesis Outline 6
Chapter 2 BACKGROUND 7
2.1 Conventional CMOS PLA Architecture 7
2.2 Power Minimization Using Super Product Lines 10
2.3 Power minimization by Input-Isolation 15
2.4 PLA Partitioning 20
2.5 Summary 21
Chapter 3 A NEW PLA WITH MINIMAL ENERGY 22
3.1 Molavi's PLA Operation 22
3.2 Tien's PLA Operation 26
3.3 Benchmark Circuits 31
3.4 HSpice Code Generation and Data Collection 32
3.5 Molavi PLA vs. Other Designs 35
3.6 Summary 38
Chapter 4 PLA PARTITIONING 39
4.1 PLA Partitioning 40
4.1.1 Molavi PLA Power and Delay Models 44
4.1.2 Partitioning Algorithm 49
4.1.3 Partitioning Results 52
4.2 Super Product Line in sub-PLA 54
4.2.1 Super Product Line Merging Algorithm 54
4.2.2 Super Product Line Results 56
4.3 Summary 58
Chapter 5 PLA VS. ASIC COMPARISONS 60
5.1 Standard Cell Flow and Data Collection 60
5.2 Area Comparison 62
5.3 Delay, Power and Energy Comparison 64
5.4 Summary 67
Chapter 6 CONCLUSION AND FUTURE DIRECTIONS 69
6.1 Contributions 72
6.2 Future Work 73
REFERENCES 75

LIST OF TABLES

Table 3.1: MCNC benchmark circuits information 31
Table 3.2: Tien, Wang and Molavi vs. conventional 36
Table 3.3: Molavi+Tien vs. conventional 37
Table 4.1: Molavi+Parition+Tien vs. Molavi and conventional design 59
Table 5.1: Conventional PLA, Molavi PLA, Molavi+Parition+Tien vs. SC-ASIC 68
Table 6.1: Different optimization approaches vs. conventional PLA design 70

LIST OF FIGURES

Figure 1.1: For this 2-input 2-output function, SC-ASIC implementation is 1x~3x faster, takes 5x less area and consumes 4x less power than PLA 3
Figure 2.1: Structure of clock-delayed PLA 8
Figure 2.2: Structure before and after producing super product lines 11
Figure 2.3: Power-saving graphs 14
Figure 2.4: Structure of Blair's PLA [16] 16
Figure 2.5: Structure of Kwang's PLA [17] 17
Figure 2.6: Structure of Wang's PLA [18] 18
Figure 2.7: Molavi's PLA structure [19] 19
Figure 2.8: Decomposition of PLAs 20
Figure 3.1: Charge-sharing problem in Wang's PLA 23
Figure 3.2: Molavi's PLA operation 24
Figure 3.3: The critical path timing diagram for Molavi PLA simulation 25
Figure 3.4: OR plane of a PLA 26
Figure 3.5: A simple example of merging product lines 28
Figure 3.6: Charge-sharing in super product line structure 29
Figure 3.7: Molavi's PLA super product line example 30
Figure 3.8: The scatter plot of the benchmark circuits 32
Figure 3.9: The design flow of PLA benchmark circuit simulation 33
Figure 3.10: Example of PLA benchmark circuit definition file (conv584) 33
Figure 3.11: Benchmark circuit "newtag" power consumption for different input vectors 35
Figure 4.1: Optimization process flow diagram 39
Figure 4.2: Single output partitioning vs. full circuit implementation (area) 41
Figure 4.3: Single output partitioning vs. full implementation (delay) 42
Figure 4.4: Single output partitioning vs. full implementation (power) 42
Figure 4.5: Single output partitioning vs. full implementation (PDP) 43
Figure 4.6: Molavi's PLA power module 45
Figure 4.7: Molavi's PLA critical path 45
Figure 4.8: Plot of HSpice simulation vs. calculated power and delay with respect to number of products 48
Figure 4.9: Example of weighted graph (*conv584*) 50
Figure 4.10: Example of greedy algorithm (*conv584*) 51
Figure 4.11(a): Multi-output partitioning vs. single output partitioning 52
Figure 4.12(b): Multi-output partitioning vs. single output partitioning 53
Figure 4.13: Example of PLA structure and corresponding PSG 56
Figure 4.14: Molavi+Partition vs. Molavi+Partition+Tien 57
Figure 5.1: The SC-ASIC design flow and CAD tools used 61
Figure 5.2: Example layout for Molavi's PLA (*misex1)* 62
Figure 5.3: PLA vs. SC-ASIC (area) 64
Figure 5.4: PLA vs. SC-ASIC (delay) 65
Figure 5.5: PLA vs. SC-ASIC (power) 66
Figure 5.6: PLA vs. SC-ASIC (PDP) 67

ACRONYMS

ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
CMOS	Complementary Metal-Oxide-Silicon
DSM	Deep Sub-Micron
FPGA	Field Programmable Gate Array
IC	Integrated Circuit
I/O	Input/Output
IPO	Inputs, Product terms, and Outputs numbers of a PLA
LUT	Look-Up Table
MCNC	Microelectronics Center of North Carolina
PLA	Programmable Logic Array
PSG	Power-Savings Graph
PVT	Process-Voltage-Temperature
RDR	Restrictive Design Rules
RTL	Register Transfer Level
ROM	Read-Only Memory
SC-ASIC	ASIC design synthesis using Standard Cell logic library
SoC	System on a Chip
SOP	Sum-Of-Product
VLSI	Very Large Scale Integration

ACKNOWLEDGEMENTS

Many thanks to my supervisor Dr. Resve Saleh. He has always been there ready to help and give me advice and direction. I also thank Dr. Steve Wilton and Dr. Guy Lemieux for serving on my thesis committee.

The SoC faculty have always been ready to answer questions and provide wonderful learning opportunities. I would also like to thank my colleagues at UBC SoC lab, Roberto Rosales, Roozbeh Mehrabadi, Sohaib Majzoub, and Michael Lee, for being kind and supportive.

Last but not the least, many thanks to my husband Xin Cui, who has been supporting me during all these days.

Funding for this research was provided by an NSERC Discovery Grant and an NSERC Postgraduate Scholarship.

Chapter 1 INTRODUCTION

1.1 Research Motivation

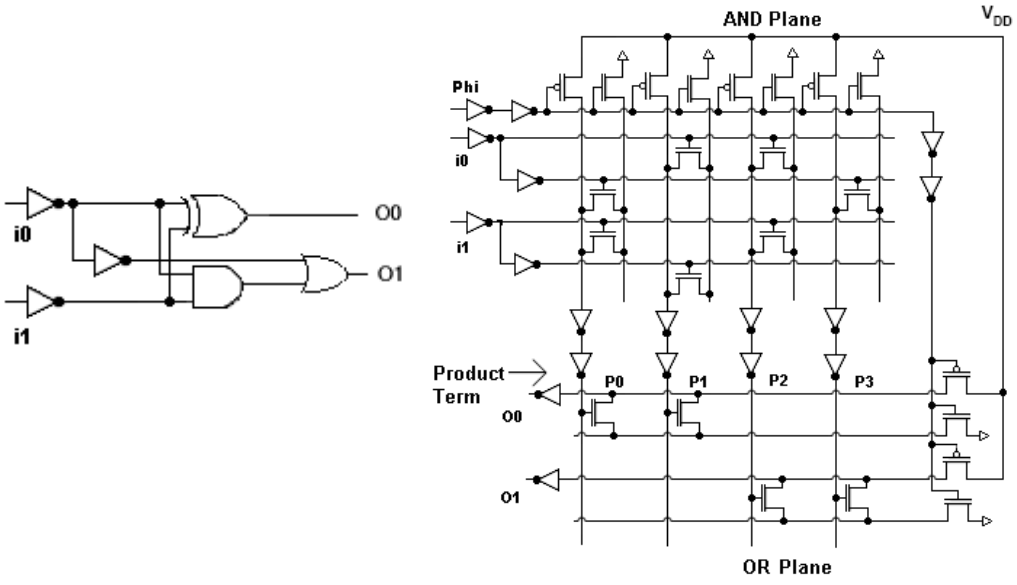
The aggressive scaling in modern integrated circuits (IC) design due to Moore's law [1] has resulted in a manufacturing process that is much more complicated than ever before. In deep submicron technology, the resolution requirements are so fine that the manufacturability of complex geometric patterns on a chip is increasingly difficult at each technology node. Specifically, the pseudo-random design structures, such as standard cell application-specified integrated circuits (SC-ASICs), cause a reduction in yield due to limitations in photolithographic resolution. Designers are now reconsidering the use of structured arrays to enhance the design yield. Also, foundries are establishing a set of restrictive design rules (RDRs) that limit the randomness of the layout [2]. Therefore, structured layouts with RDRs applied may be more appropriate for CMOS (complementary metal-oxide-semiconductor) than ASIC implementation in the future since the foundry can fine-tune the process to improve yield.

Structured arrays have been developed over the years in many forms, such as field-programmable gate array (FPGA) [3], structured ASIC [4], memory, and programmable logic array (PLA) [5]. FPGAs provide a general-purpose programmable logic array and are often the first type of logic design to be fabricated in a new technology node. However, it comes at a fairly high cost of area and much lower performance when compared to SC-ASIC implementation. Structured ASIC is a relatively new approach that contains an array of highly-structured and programmable logic and interconnect. The one

drawback is that all non-metal layers and some metal layers have already been fabricated and the user is only left with a few metallization layers to customize. A ROM block can be used for logic since it can store the outputs for all possible combinations of inputs. However, as the number of inputs increases, the memory size grows quickly and it becomes rather slow and consumes more power. A more area efficient way to implement this type of block, relative to the ROM, is a PLA. The reason is that logic functions typically have many input combinations that produce the same output [6].

PLAs have many other desirable features, such as simplicity, regularity and flexibility. These properties also allow accurate prediction of timing and power of PLA designs and make it a useful device in the realization of both combinational and sequential circuits. They have been around for many years and have been used in a variety of applications. Certain logic functions, such as the control units of processors, are often implemented using PLAs rather than random logic [7]. In [8] and [9], the authors used dynamic PLAs to design the control units of 1GHz microprocessors. In [10], the authors reported a 1GHz look-ahead adder using a dynamic PLA. In a novel hybrid FPGA architecture, logic tiles are configured as either PLAs or look-up tables (LUTs) [11]. The structured metal layout implemented on medium-sized PLAs is cross-talk immune, and can be 2x smaller and faster than a traditional standard-cell-based implementation of the same logic according to [12]. PLA in the field programmable form (i.e., FPLA) are also being used for silicon debugging purposes [13]. The regularity of the PLA structure also allows the use of duplicate PLA columns and rows for self-repair [14].

Although there are many applications and benefits, PLAs have some well-known limitations. In the years when PLAs were very popular, the major problem of PLAs relative to ASICs was the area. Since technology scaling now allows billions of transistors to be fabricated on a single chip, this is no longer as big a concern as before, although area may still affect yield. However, simulations of conventional PLA designs and respective SC-ASIC implementations show that the bigger problem is that PLAs generally consume more power. Furthermore, the delay of a PLA is much longer than the corresponding CMOS gate implementation. For a simple function of two inputs and two outputs for the PLA and ASIC designs shown in Figure 1.1, the ASIC implementation takes much less area, power and delay. Before PLAs can be widely-adopted and compete with SC-ASIC, PLAs need to reduce power consumption and delay. Several ideas have been proposed recently to improve the power efficiency, delay and energy consumption for NOR-NOR clocked dynamic PLAs [15]-[19], as described next.



(a) ASIC built from standard cells **(b) NOR-NOR clocked dynamic PLA**

Figure 1.1: For this 2-input 2-output function, SC-ASIC implementation is 1x~3x faster, takes 5x less area and consumes 4x less power than PLA

To reduce the power consumption, Tien et al. [15] proposed a new PLA architecture by combining certain product lines to form *super product lines*. In this case, the logic functions must be known in advance to optimize the result. Their approach depends on the maximum-weighted matching to find the optimal solution. By merging product lines, the switching probabilities for the super product lines are reduced compared to the original structure. Their results show that the power consumption could be reduced by 55.8% and the delay overhead is only 3.3% on average for the 25 benchmark circuits used for their simulations [15]. However, this super product line method suffers from a charge-sharing problem. A weak feedback transistor was used to solve this problem but it increases overall delay overhead by an additional 15.6%.

Many other researchers have modified the pull-up/pull-down path in PLA designs to improve both power and timing [16]-[18]. In contrast to Tien's PLA, these designs are more general as they do not require prior knowledge of the logic function to be implemented. Blair [16] suggested a pre-discharge scheme, which reduces dynamic power. This architecture forms a pseudo-NMOS structure in the evaluation phase, which is a ratioed design and therefore dissipates DC power. Kwang added an additional PMOS device in the DC path to eliminate the short-circuit current during evaluation, but Kwang's PLA suffers from glitches at the beginning of the evaluation phase that consumes AC power [17]. Wang employed a pseudo-footless architecture and suggested the use of a NAND-INV idle-mode inter-plane buffer. By doing this, Wang's design greatly reduced the switching activity of internal nodes [18]. However, Wang's design does not eliminate the glitches in the PLA. Molavi [19] improved Wang's design by

using a latch-based sensing circuit to further increase the speed of operation and to remove glitches.

All these designs significantly reduce the delay and power consumption compared with the conventional NOR-NOR dynamic CMOS PLA and at the same time introduce new problems which must be resolved. There are still other opportunities to improve PLA performance: we can partition a large PLA into sub-PLAs to reduce area and delay, and with possibility of decreasing power consumption. By combining these techniques, it may be possible to create a PLA that competes with ASIC designs in certain applications.

1.2 Research Goals

The goal of this thesis is to combine Molavi's speed improvements with Tien's power improvement and then apply output-partitioning to determine how close PLAs can come to ASICs in terms of speed, power and area at the 65nm technology node. In this research, we will analyze the design improvements mentioned in previous section, and discuss their advantages and disadvantages. We also investigate the effect of partitioning on the resulting PLA's performance. The sequence of steps in the work is as follows:

- Combine the Molavi and Tien's approaches to assess the improvement on a standard set of benchmarks
- Apply a new partitioning algorithm on the PLA for timing improvement
- Compare improved PLA design and layout against ASIC at 65nm CMOS technology node

We use two sets of benchmarks from Microelectronics Center of North Carolina (MCNC) [20] and Berkeley [21] to compare area, delay and power consumption among different PLA designs and between PLAs and SC-ASIC. We also extract parasitic capacitance and resistance in PLA critical path for delay optimization from layouts for more precise HSPICE [22] simulation. To achieve a reasonably fast partitioning algorithm, we need to find relatively accurate equations for area, delay and power consumption in terms of the number of inputs, outputs, and product terms. *The ultimate goal is to find a PLA architecture that can compete with the traditional ASIC design that uses standard cells.*

1.3 Thesis Outline

Chapter 2 provides background information on several conventional PLA designs and the advantage/disadvantage of each. We combine PLA designs by Tien and Molavi and present a comparison of the different approaches for a set of benchmarks in Chapter 3. In Chapter 4, we describe partitioning algorithms to improve PLA delay, power consumption and area. Combining the possible techniques, we compare the final PLA performances with SC-ASIC design in Chapter 5. Conclusions and future research directions are presented in Chapter 6.

Chapter 2 BACKGROUND

In this chapter, we provide background material on conventional clock-delayed CMOS PLAs and other previously published PLA designs. For each one, we describe the respective advantages and disadvantages. We also briefly introduce single-output logic and PLA partitioning.

PLAs can be implemented in either static or dynamic styles [7]. The static pseudo-NMOS design style is simple but has a disadvantage of DC-path power dissipation. Also, the ratioed design style tends to be relatively slow. A dynamic sum-of-product (SOP) domino logic design is better with regard to power savings, but the serial NMOS devices of the AND plane cause a large pull-down delay and potential charge-sharing problems. The static CMOS PLAs using a NAND-NAND or NOR-NOR structure tend to occupy a larger area but are faster and do not exhibit charge sharing. In fact, most recent advances in PLA delay and power reduction have been applied to dynamic CMOS clocked PLAs [7], [15]-[19]. Therefore, we will focus our efforts on the clocked CMOS PLAs in this thesis.

2.1 Conventional CMOS PLA Architecture

As a starting point, a conventional CMOS single-clock PLA is shown in Figure 2.1. This is a clocked NOR-NOR dynamic PLA. Even though both the first and second stages are exactly the same NOR functions, we still use the convention of calling the first stage the AND-plane and the second stage the OR-plane, which are separated by the inter-plane

buffers. The AND-plane on top produces the product terms from the inputs and passes the results via inter-plane buffers to the OR-plane which then sums the required product terms together to produce the outputs.

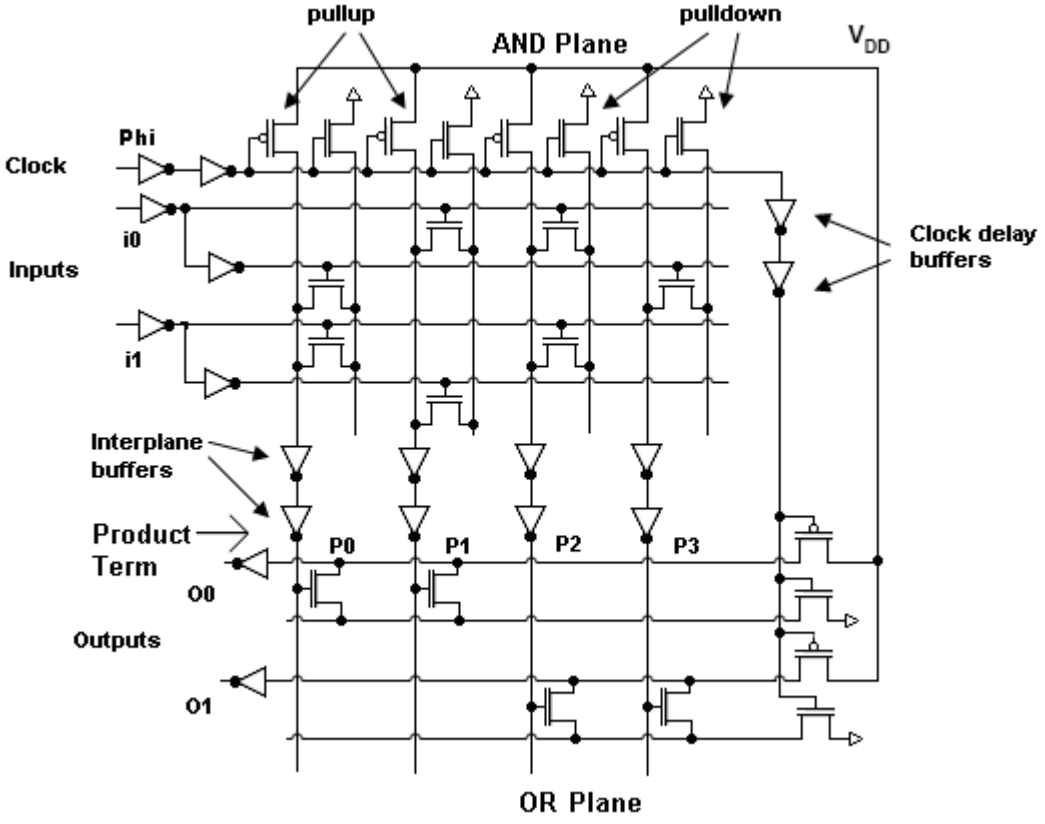


Figure 2.1: Structure of clock-delayed PLA

It is useful to understand the basic operation of this structure before going into structure modifications. The overall size of a PLA depends on the number of inputs, product terms, and outputs (IPO). The example in Figure 2.1 has 2 inputs, 4 product terms and 2 outputs, i.e., IPO=2x4x2. It produces the following two outputs:

$$O_0 = P_0 + P_1 = \overline{i_0} i_1 + i_0 \overline{i_1}$$

$$O_1 = P_2 + P_3 = i_0 i_1 + \overline{i_0}$$

The horizontal lines in the AND-plane are connected to the inputs or their complements. Only the input or its complement is part of a product term. In this PLA structure, the vertical lines connected to the pull-up transistors and inter-plane buffers act as the product lines. When an input line is part of a product term, it is connected to the gate of a pass transistor, which would discharge the product line through the pull-down line. When the clock signal "Phi" is low, the product lines are charged up to V_{DD}. This is called the pre-charge phase. The evaluation phase happens when "Phi" goes high. In this phase, the pre-charged product lines stay high if they are not connected to the pull-down lines via a pass transistor. Otherwise, they will be discharged low.

The major problem of a dynamic PLA is the potential for a race condition: if the output of the OR-plane begins evaluation before the AND-plane is stable, a problem occurs. The clock signal for the OR-plane is usually delayed to make sure it does not start evaluation before the product lines from AND-plane are all stable. This clock delay depends on the PLA size, such as number of product terms and number of inputs, and needs to be carefully designed. If the clock for the OR-plane comes too early, it may propagate the wrong signal value from the inter-plane buffers. Any output line that has been mistakenly discharged will not be able to regain its charge until the next pre-charging phase. On the other hand, if the clock is applied too late, this will slow down the circuit. However, some degree of safety margin is still needed to accommodate PVT (process-voltage-temperature) variations.

This PLA structure has nodes with a large number of input transistors connected to it. To pre-charge and discharge these large capacitances at the AND, OR, and intermediate

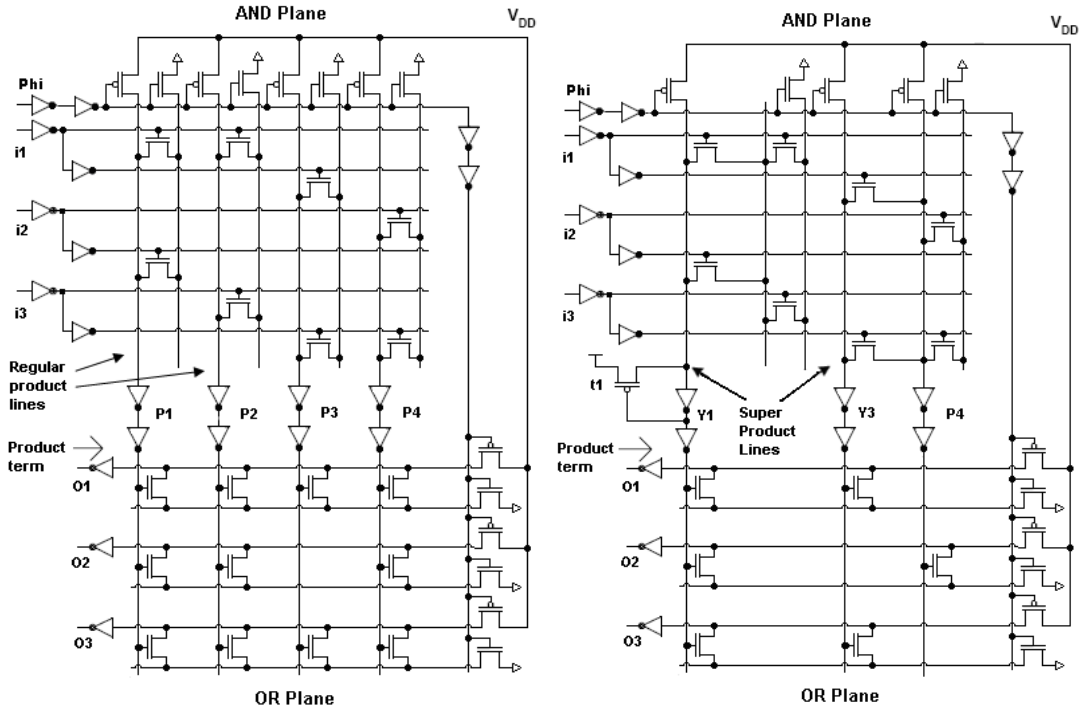
planes requires a great amount of power and time. Furthermore, the dynamic PLA has pre-charging and evaluation phases for both AND and OR planes. For a product line with *n* inputs, assuming each input has an equal probability of high and low, the product line only stays high when all the inputs are low and the probability of this case is $1/2^n$. Therefore, the probability of discharging for this product line is $1-(1/2^n)$ which is close to 1 for a large *n*.

2.2 Power Minimization Using Super Product Lines

The dynamic power of a circuit can be calculated as $P = \alpha C V_{DD}^2 f$, where α is the switching activity, *C* is the capacitance, V_{DD} is the supply voltage and *f* is the clock frequency [23]. If we consider the supply voltage and frequency to be constant, the dynamic power only depends on the capacitance and switching activity. One possible method to minimize PLA power consumption is to reduce switching activity. In [24] and [25], the authors proposed new algorithms to decrease the switching activity on product lines. In [24], a minimum power solution for dynamic PLAs, which consists only of prime implicants of the function, was proposed. In [25], the authors exploited the *don't care* set to reduce the cube switching activity. However, the improvement on the dynamic PLA of these methods is not significant due to the precharging and discharging process in each clock cycle of the dynamic logic.

Although these methods attempt to find product terms with lower switching activity, the NOR structure of dynamic PLAs inherently implies high-switching activity in product

lines and output lines. Some of these switching activities do not contribute to the final value of the output. For instance, in Figure 2.2(a) with output $O_2 = P_1+P_2+P_4$, if product line P_1 is high, the switching activity on product line P_2 and P_4 will have no effect on the output. Clearly, with the NOR structure and an assumption of a uniform distribution of high and low at the inputs, the probability of unnecessary switching is very high.



(a) before **(b) after**

Figure 2.2: Structure before and after producing super product lines

Tien et al. [15] proposed in a new structure for a dynamic PLA which merges a set of two product lines to form *super product lines* (see Figure 2.2(b)). The super product line adds NAND functionality to decrease the unnecessary switching activity and hence reduces power consumption. To illustrate their idea, let us consider the previous example where $O_2=P_1+P_2+P_4$, with $P_1 = \overline{(\overline{i_1} + i_2)} = i_1\overline{i_2}$ and $P_2 = \overline{(\overline{i_1} + \overline{i_3})} = i_1 i_3$. The switching

probability of product line P_1 is $1-\text{Prob}((\text{i}_1=1)\,\text{AND}\,(\text{i}_2=0))=1-1/2^2=0.75$ (similarly for P_2). Including a super product line $Y_1=\overline{(\overline{i_1}+i_2\overline{i_3})}=i_1(\overline{i_2}+i_3)=P_1+P_2$ and rewriting $O_2=Y_1+P_4$, the switching activity of Y_1 is $1-\text{Prob}((i_1=1)\,\text{AND}\,((\overline{i_2}=0)\text{OR}\,(i_3=1)))=1-1/2\times(1-1/4)=0.625$. This not only reduces the dynamic power, but also reduces the static power by decreasing the number of product lines. Combining related product lines into super product lines can lead to significant power savings.

The central issue of the super product line approach is to choose among product lines that provide the maximum power reduction. In order to verify the power improvement of the combination of product lines, Tien introduced a set of equations to estimate the product line and super product line power consumption. In Tien's model, power sources are divided into five parts: input lines, product lines in the AND plane, product lines in the OR plane, output lines, and the clock line. The total power cost for a product term P_j with its input and output component is given by [15]:

$$Power(P_j)=\frac{V_{dd}^2 f}{2}\left[\sum_{i=1}^{2n}C_{Ii}\text{Prob}^1(I_i)+\sum_{k=1}^{l}C_{Ok}\text{Prob}^0(O_k)\right]+PPower(P_j)$$

$$PPower(P_j)=\frac{V_{dd}^2 f}{2}(C_{ANDP}\text{Prob}^0(P_j)+C_{ORP}\text{Prob}^1(P_j))$$

where $2n$ and l are the numbers of input and output lines, respectively, C_{Ii} and C_{Ok} are the capacitive load due to these inputs and outputs, respectively, C_{ANDP} and C_{ORP} are the

capacitances of product line P in the AND and OR planes, respectively, and $Prob^0(P)$ and $Prob^1(P)$ are the 0-probability and 1-probability of product line P, respectively. The summation terms combine all the inputs and outputs, while the *PPower* term is the power consumption at a product line. Let super product term Y_i be obtained from two product terms P_i and P_j. Then, the function of Y_i becomes the OR function of P_i and P_j, i.e., $Y_i = P_i + P_j$. After combining two product lines, P_i and P_j, into a super product line, Y_i, the total power of the new line becomes:

$$SPower(Y_i) = \frac{V_{dd}^2 f}{2}\left[\begin{array}{l}\sum_{i=1}^{2n} C_{Ii}\,\mathrm{Prob}^1(I_i) + C_{ANDPi}\,\mathrm{Prob}((P_i = 0)\,\mathrm{AND}\,(P_j = 0)) \\ + C_{ORPi}\,\mathrm{Prob}((P_i = 1)\,\mathrm{OR}\,(P_j = 1)) + \sum_{k=1}^{l} C_{Ok}\,\mathrm{Prob}^0(O_k) + C_{gate}\end{array}\right]$$

$$+ \frac{(V_{dd} - V_t)^2 f}{2} C_{ANDPj} \times \mathrm{Prob}((P_i = 0)\,\mathrm{AND}\,(P_j = 1))$$

where C_{Ii} and C_{Ok} are the capacitive loads due to the new super product and I/O lines.

After estimating the power consumption of each product line, a power-savings graph (PSG) of Figure 2.3 is constructed to capture the power-saving potential if two product lines are merged. Each node of the graph, p_i, represents an existing product line, while each edge, *ps(i,j)*, represents the possibility of two product lines being merged. The weight of each edge is the amount of power that could be saved after merging. A maximum power saving is achieved when the maximum-weighted edges are chosen for the product line merging. However, not every merging action will result in the removal of

product terms in the OR plane. Therefore, only those product lines that are used by the same outputs are replaced by a super product line.

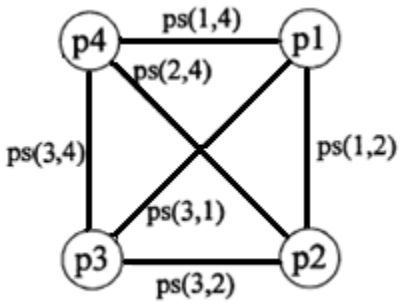


Figure 2.3: Power-saving graphs

Although Tien's PLA reduces power consumption, it does increase delay since some pass transistors are now connected in series. At the same time, the capacitive loads at input and output lines are reduced thereby decreasing the delay at I/O ports. Perhaps more importantly, this structure suffers from a charge-sharing problem. To solve the problem, Tien's PLA requires additional feedback transistors (t1 in Figure 2.2(b)) which degrade the performance of the circuits. Without a feedback transistor, the voltage drop at the super product line could be as low as half of the supply voltage. Furthermore, the improvement of Tien's PLA mainly depends on the outcome of the super product line merging process. The procedure of calculating power consumption, constructing PSG for each product line, and performing a maximum-weight matching requires more design effort, and occasionally the results may not be much better if all the product lines are loosely-related. However, the degree of improvement on average is worth the effort.

2.3 Other Power Minimization Techniques

Blair's PLA [16] uses the pre-discharged pseudo-NMOS design, as shown in Figure 2.4. Pass transistors are directly connected to ground, and pull-down NMOS transistors M_{pdc} are directly connected to product line node 'P' for the pre-discharge scheme. Different from the conventional PLA, this design uses a high clock for the pre-discharge phase. In this period, M_{pdc} discharges the node 'P' to ground. During the evaluation phase, the clock is low, so the discharge transistor is off and the pull-up PMOS transistor is on. If no pull-down is connected, the line goes high. If a pull-down is connected, then it forms a pseudo-NMOS structure. This is the source of speed bottleneck of this design. The voltage level of node 'P' depends on the inputs and the size ratio of the pull-up transistor to the pull-down pass transistors. Assuming a uniformly-distributed input probability, i.e., with 50% probability of being either high or low at any time, and the number of inputs is *n*, the chance of conducting a static current is $1-\left(1/2^{n}\right)$. The evaluation node is discharged low, and the probability that it stays low in the evaluation phase is nearly 1 for large n, which means the switching activities are thus nearly 0 and this saves power. However, the probability that the circuit becomes a pseudo-NMOS circuit is rather high, and this short-circuit power dissipation may offset the benefit gained. This is a significant shortcoming of this structure.

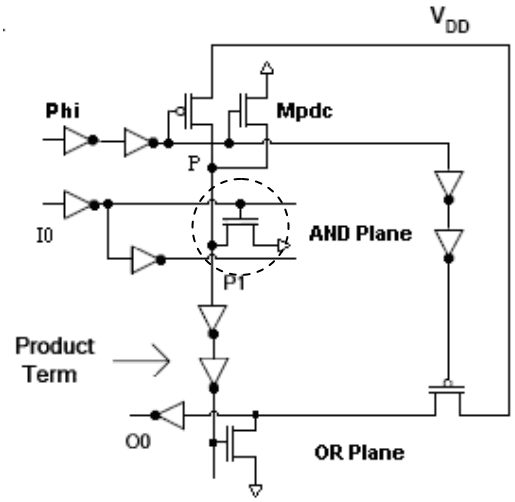


Figure 2.4: Structure of Blair's PLA [16]

To alleviate this problem, Kwang suggested the use of a conditional evaluation scheme [17]. As shown in Figure 2.5, another PMOS transistor (M_{EXT}) is inserted in series with the pull-up transistor. The pre-discharge operation is similar to that of Blair's PLA through M_{pdc}. However, during the evaluation phase, M_{EXT} will be "on" momentously while Phi propagates back to turn off M_{EXT} while pull-up transistor M_{pu} is on and M_{pdc} is off. This provides a pulse of current to charge node 'P' if no inputs are pulling it low. The problem of this design is that there is a delay through a series of gates to turn M_{EXT} off so there is still some short-circuits power dissipated. This delay cannot be avoided since it is used to ensure a correct evaluation in the AND plane.

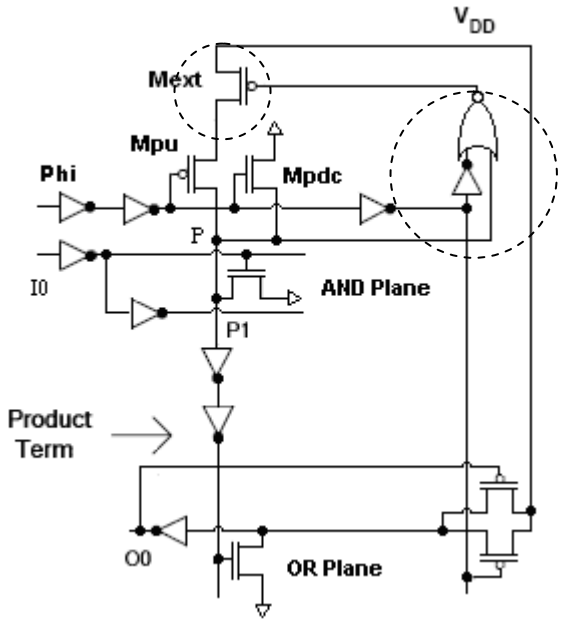


Figure 2.5: Structure of Kwang's PLA [17]

Wang employed a NAND-INV type of buffer to reduce this effect by removing the foot NMOS transistor (M_{pdc}) and inserting NMOS transistor (M_{ps}) into the product line path between the pull-up and pass transistors to form a pseudo-footless structure [18]. This is shown in Figure 2.6. When clock is low, Wang's PLA is in the pre-charge phase. Node 'P' is charged to high by the PMOS pull-up transistor M_{pu}. At the same time, node 'B' is high, and node 'C' is low. This ensures that there is no race condition in the OR plane. Also, transistor M_{ps} is off; the capacitance that needs to be charged up is quite small at node 'P' compared to the parallel-input nodes. When it goes into the evaluation phase, the clock is high, transistor M_{ps} is on, and the NAND gate turns into an inverter of P. Depending on the inputs, the voltage at node 'P' either stays the same or is pulled low. As shown in Figure 2.6, the most important design change here is the NAND gate intermediate buffer. The clock signal controls the new transistor M_{ps} and is also ANDed with product line to produce signal *B*. When the clock *Phi* is low, the NAND gate output

stays high, and it only changes to low when all the primary inputs that affect this product line are low. This significantly reduces the switching activity of the intermediate nodes since the possibility of these inputs being low at the same time is small. At any time, only one of the pull-up PMOS transistor M_{pu} or M_{ps} is on, which effectively mitigates the static current problem.

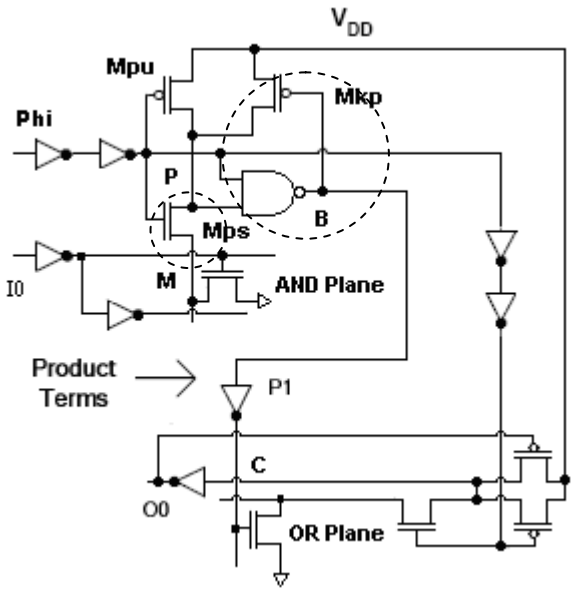


Figure 2.6: Structure of Wang's PLA [18]

However, since transistor M_{ps} is on, charge sharing between node 'P' and node 'M' occurs. Even though this may speed up the pull-down process, it also increases the time to charge node 'P' back to high whenever node 'P' is evaluated to be high. To solve this problem, a keeper transistor M_{kp} is included which is controlled by node 'B', to speed up the pull-up time. A proper sizing of the transistor M_{kp} and the NAND gate is needed to make this structure fast enough compared to other designs.

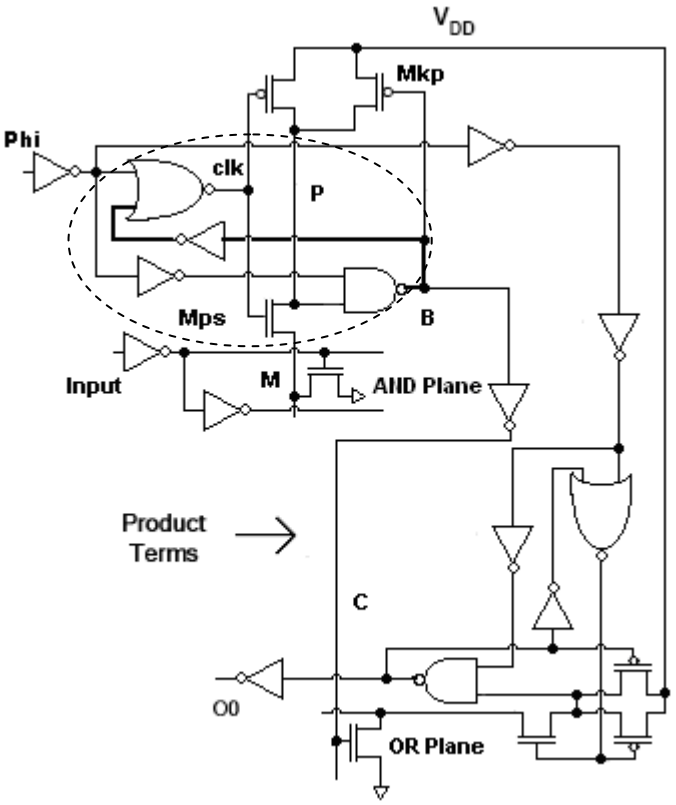
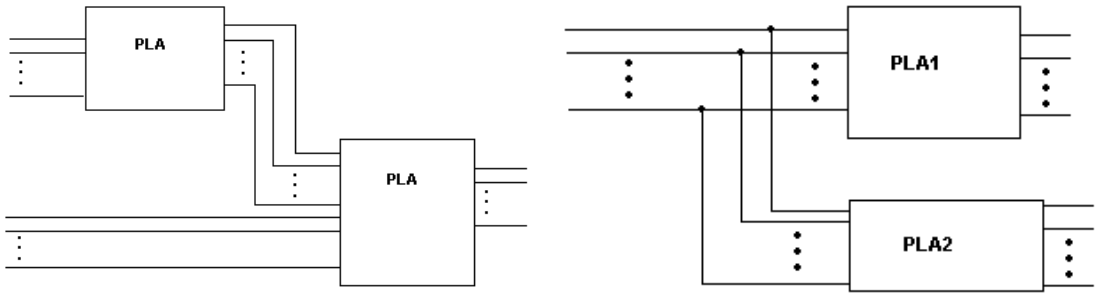


Figure 2.7: Molavi's PLA structure [19]

Wang's PLA has an improved power, improved delay and uses less energy compared with a conventional PLA [18]. However, the relatively slow rise time at node 'P' provides room for more delay improvement. Molavi proposes an alternative input-isolation scheme [19]. As illustrated in Figure 2.7, the charge-sharing between nodes 'P' and 'M' is prevented by adding a NOR gate that includes a feedback signal from the intermediate plane which propagates through the NOR gate to produce the *clk* signal. The output *clk* will turn off M_{ps} when all inputs that affect this product line are '0' to prevent charge-sharing. This pull-up circuit is somewhat more complicated, but it provides the shortest delay overall.

2.4 PLA Partitioning

The area and delay characteristics of a PLA are strongly correlated with the number of product terms in a PLA. With a smaller set of input/output and product terms, the PLA will take less area, operate faster, and consume less power. There are many previous studies in 80's and 90's on PLA partitioning for area and delay reduction [26]-[31]. Early work focused on reducing area [26]-[30]. Liu presented a min-cut partitioning of PLAs to reduce both area and cycle time [31]. As shown in Figure 2.8, decomposition of PLAs can be classified into two types: serial decomposition and parallel decomposition. In this thesis, we will concentrate on parallel decomposition, in which the output functions are partitioned into groups, and each group is realized by a smaller PLA.



(a) Serial decomposition **(b) Parallel decomposition**

Figure 2.8: Decomposition of PLAs

For a multiple-output PLA, the best delay for a single level of PLA logic can be obtained by partitioning the PLA into single-output sub-PLAs. However, the result of a single-output partitioning is very area-intensive. For outputs that share a great deal of the product terms, there will be too much area redundancy. For outputs with only a few product terms, it is very inefficient. PLA partitioning of this form also generally increases the total power of the circuits. A brute-force approach to find an optimal partitioning is to

try all possible output combinations. This has the properties of a NP-complete problem even if we assume only a bi-partition, that is, only two partitions are used. We will show in a later chapter that single-output partitioning gives the best delay with the worst power and area results, and the full PLA gives the best power with the worst delay results. Any other partitioning will produce results between these upper and lower bounds. The goal for a good PLA partitioning is to find an optimal point where area, delay and power consumption are balanced relative to some cost function.

2.5 Summary

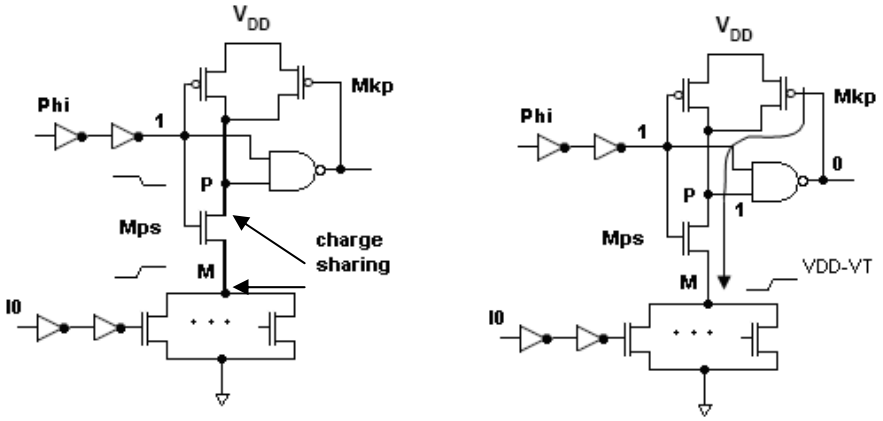
There are many techniques to improve the power and performance of a PLA. However, there is still a gap between the PLA and SC-ASIC and there is a room for more improvement. In the next several chapters, we will combine the design concepts of Molavi and Tien, and show its advantages over other PLA structures through the use of standard benchmark circuits. We will also discuss PLA partitioning in more detail and provide algorithms with cost functions to improve delay, area, and power consumption. After applying a PLA partitioning algorithm on the PLA designs that combine Tien and Molavi, we compare the best PLAs to SC-ASIC.

Chapter 3 A NEW PLA WITH MINIMAL ENERGY

In this chapter, we present Molavi's PLA structure and its operation in detail. By comparing the simulation results on a set of standard benchmark circuits, we verify the advantages and disadvantages of Molavi's PLA over other PLAs. We then combine Tien's super product line idea with Molavi's PLA structure to further improve the design. Results are presented to compare each method separately and then combined together to determine how effective this solution is at reducing delay, power and energy.

3.1 Molavi's PLA Operation

The motivation for Molavi's improvement is best illustrated by starting with Wang's PLA in Figure 3.1. As mentioned in the previous chapter, Wang's PLA has a slow rise-time when node 'P' evaluates to high. In the evaluation phase, *phi* is high, which turns on transistor M_{ps}. If all the inputs are zero, node 'P' should stay high. However, the voltage level at node 'M' is generally much lower than V_{DD}. Therefore, the charge at node 'P' will be shared with node 'M' through M_{ps} thereby decreasing the voltage level at node 'P', as shown in Figure 3.1(a). The output of the NAND gate turns on the keeper transistor M_{kp} which will pull up node 'P' to V_{DD} and node 'M' to V_{DD}-V_T, as illustrated in Figure 3.1(b). There are many input pass transistors connected to node 'M' which implies a large drain capacitance. Therefore, the delay to charge up node 'M' will be long. Moreover, node 'M' will be discharged when at least one of the inputs is high and this increases the power consumption.

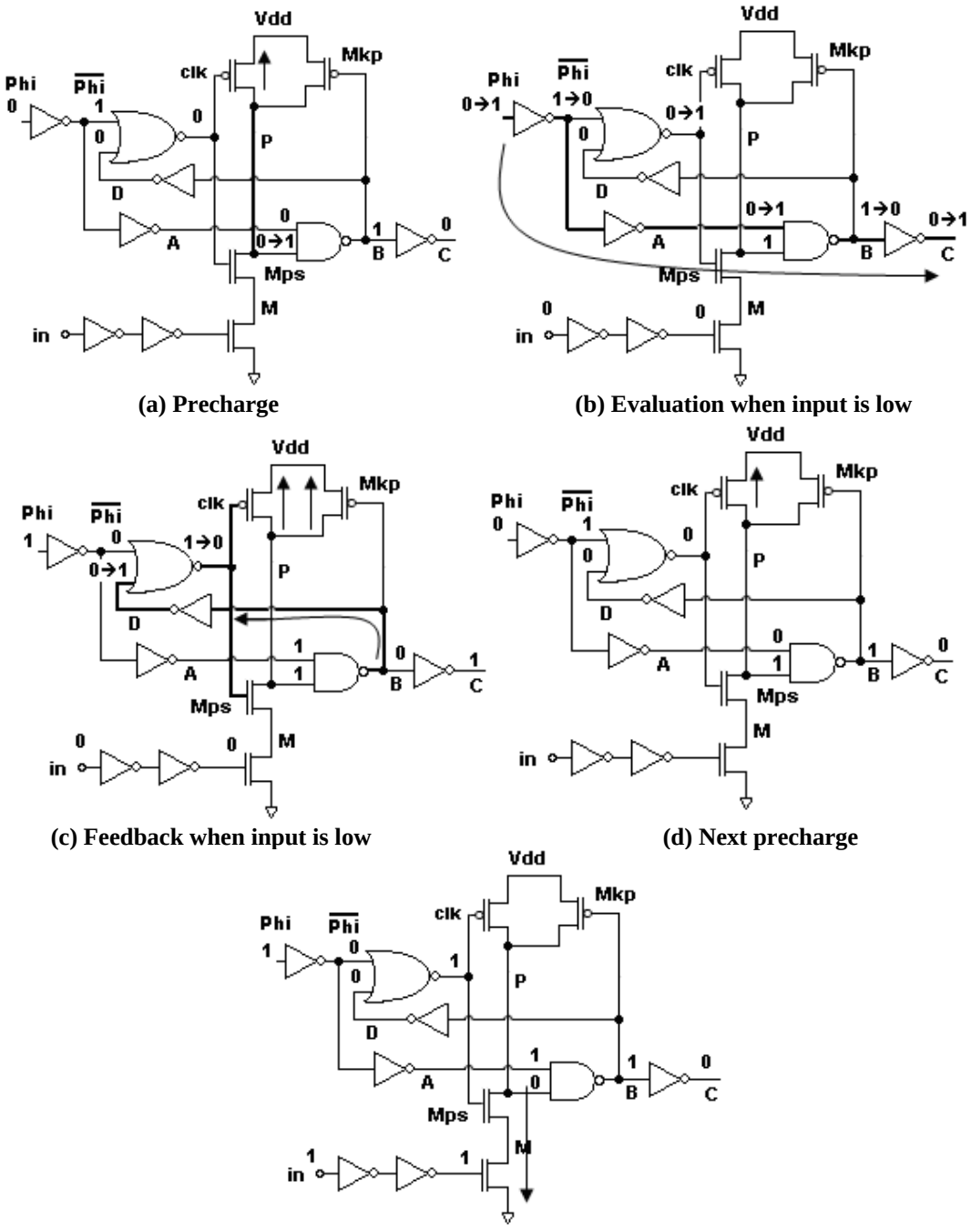


(a) **(b)**

Figure 3.1: Charge-sharing problem in Wang's PLA

Molavi solves this problem by including a latch-based sense-amplifier-like circuitry with a back-to-back inverter configuration. The operation of the critical path of the AND plane during one cycle of low input followed by one cycle of high input is shown in Figure 3.2. The clock signal that controls the PMOS pull-up and M_{ps} is not *phi* but a NORed signal that includes feedback from an inter-plane signal. During the precharge phase, the *clk* signal generated by the NOR gate turns on the pull-up to precharge the node 'P' to high (Figure 3.2(a)). During the evaluation phase, *phi* goes high which inverts signal *clk* and turns off the pull-up and turns on transistor M_{ps}. In the case that all inputs are low, node 'P' will stay high, and the sequence of nodes 'A', 'B' and 'C' are inverted in turn (Figure 3.2(b)). Different from Wang's design where charge-sharing between node 'P' and 'M' occurs in the entire evaluation phase, the low value of feedback loop from node 'B' will switch the NOR gate output signal, i.e. *clk*, to low and turn off transistor M_{ps}, which stops the charge flowing from node 'P' to 'M' (Figure 3.2(c)). As shown in the timing diagram of Figure 3.3, the node 'P' only drops by a small voltage and node 'M' does not charge up to V_{DD}-V_T as in Wang's design. This decreases the overall delay. However, the power improvement from this has been offset to some degree by the additional gates. In the case

that at least one input is high, the operation is much simpler (Figure 3.2(d) and (e)). Node 'P' is discharged through M_{ps} and any high input transistors to ground. Since the output of NAND gate is still high, the feedback path does not switch and *clk* signal stays high. The waveform diagrams for this case are also shown in Figure 3.3.



(a) Precharge

(b) Evaluation when input is low

(c) Feedback when input is low

(d) Next precharge

(e) Next evaluation when input is high

Figure 3.2: Molavi's PLA operation

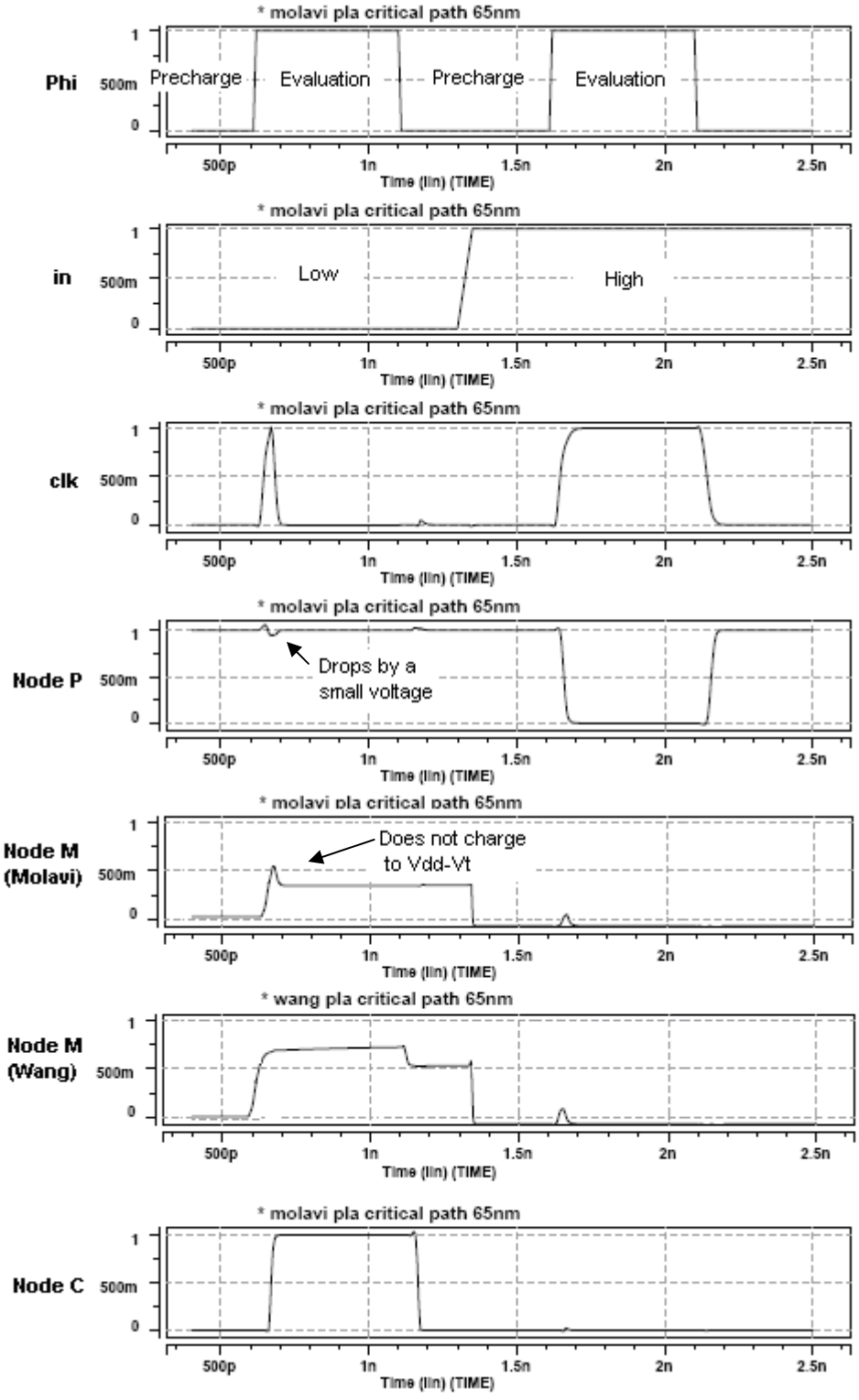


Figure 3.3: The critical path timing diagram for Molavi PLA simulation

3.2 Tien's PLA Operation

The dynamic PLA has a very high switching probability and this is a major reason for high dynamic power. In order to reduce the power consumption, it is logical to decrease the switching probability. During the precharge phase, the evaluation node is precharged to high, and the pull-down path has parallel-connected input transistors, which means that if any one of them is high, it will cause the evaluation node to be discharged. In Figure 3.4, we show an OR-plane structure. To have an output switching, we only need one of the product lines to evaluate to a high. For the function in this example, $O=A+B+C$, if product line A is high, it will discharge the output line which means product line B and C switching low will not affect the result. The probability of each product line staying high is small since that requires all the inputs affecting that product line to be low. If we assume 50% switching probability for all inputs, the probability of any product line being discharged could approach 1 for a large number of inputs. In other words, this high switching probability in the product lines will lead to high power.

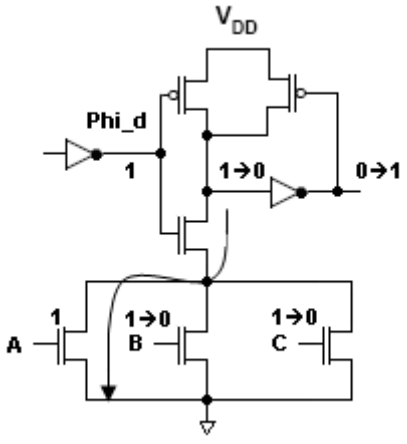


Figure 3.4: OR plane of a PLA

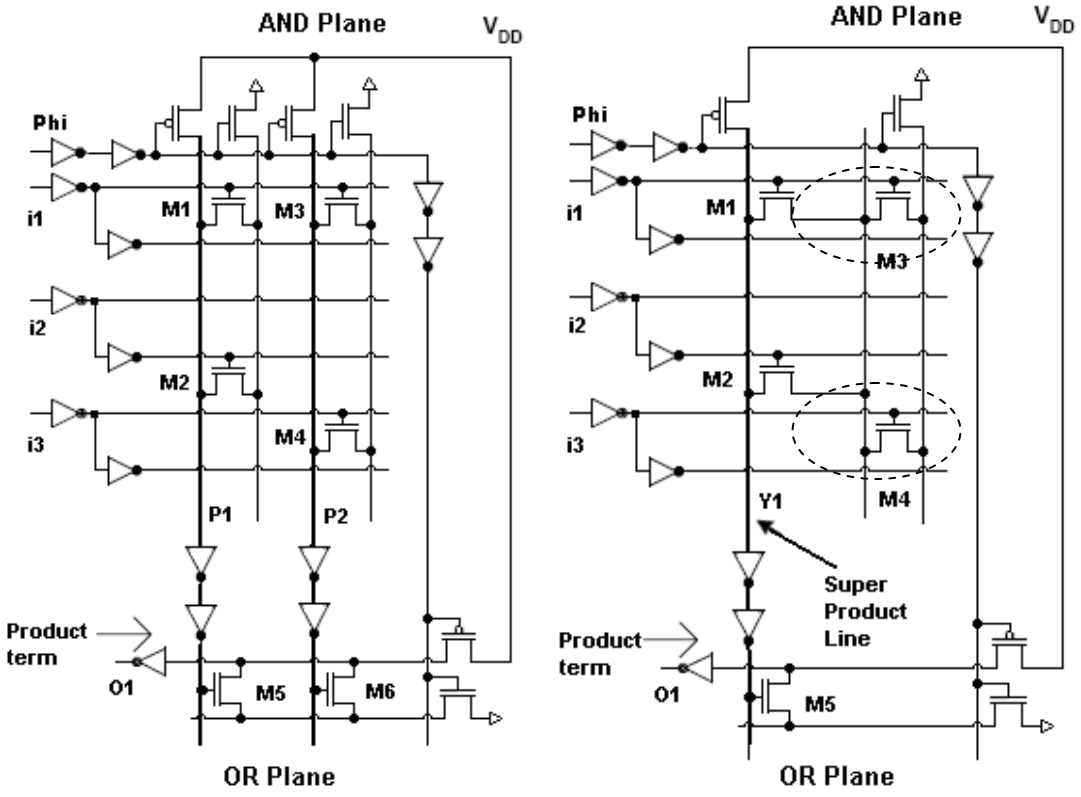
The main idea of Tien's approach to minimize power is to merge product lines into super product lines to reduce switching. Using Figure 3.5 as an example, product line P_1 and P_2 are used to evaluate output O_1, with

$$P_1 = \overline{(\overline{i_1} + i_2)} = i_1 \overline{i_2} \text{ and } P_2 = \overline{(\overline{i_1} + \overline{i_3})} = i_1 i_3$$

If P_1 is high, the switching of P_2 is not affecting the outputs, and when P_2 is high, the switching of P_1 is unnecessary. Assume i_1, i_2 and i_3 have a 50% probability of 0 or 1. The switching probability for both product lines is $1-(1/2)^2 = 0.75$. If we set $Y_1 = P_1 + P_2$ as a new function, remove the pull-down path of P_1 and pull-up path of P_2 and connect the input pass transistors (M_1 and M_2) to product line P_2, the new product line function Y_1, is

$$Y_1 = \overline{(\overline{i_1} + i_2 \overline{i_3})} = i_1 \overline{(i_2 \overline{i_3})} = i_1 (\overline{i_2} + i_3) = i_1 \overline{i_2} + i_1 i_3 = P_1 + P_2$$

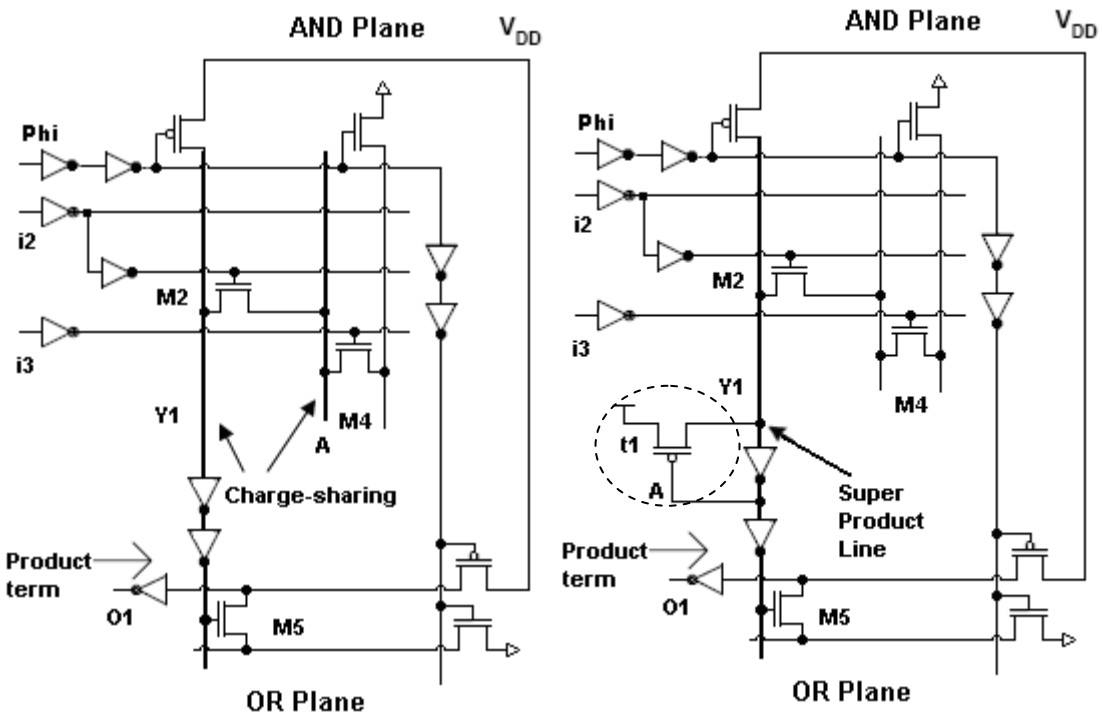
The new configuration is shown in Figure 3.5(b). Transistors M_1 and M_3 are controlled by the same signal, so the probability of that path being turned off is $1/2$. The probability of both M_2 and M_4 being turned on is $(1/2)\times(1/2) = 1/4$. Therefore, the switching probability of new super product line Y_1 is $1-(1/2)\times(1-1/4) = 0.625$ which is less than either P_1 or P_2. By merging product lines, we also remove some pull-up/down paths, inter-plane buffers, and OR-plane pass transistors which further reduces the power consumption.



(a) before **(b) after**

Figure 3.5: A simple example of merging product lines

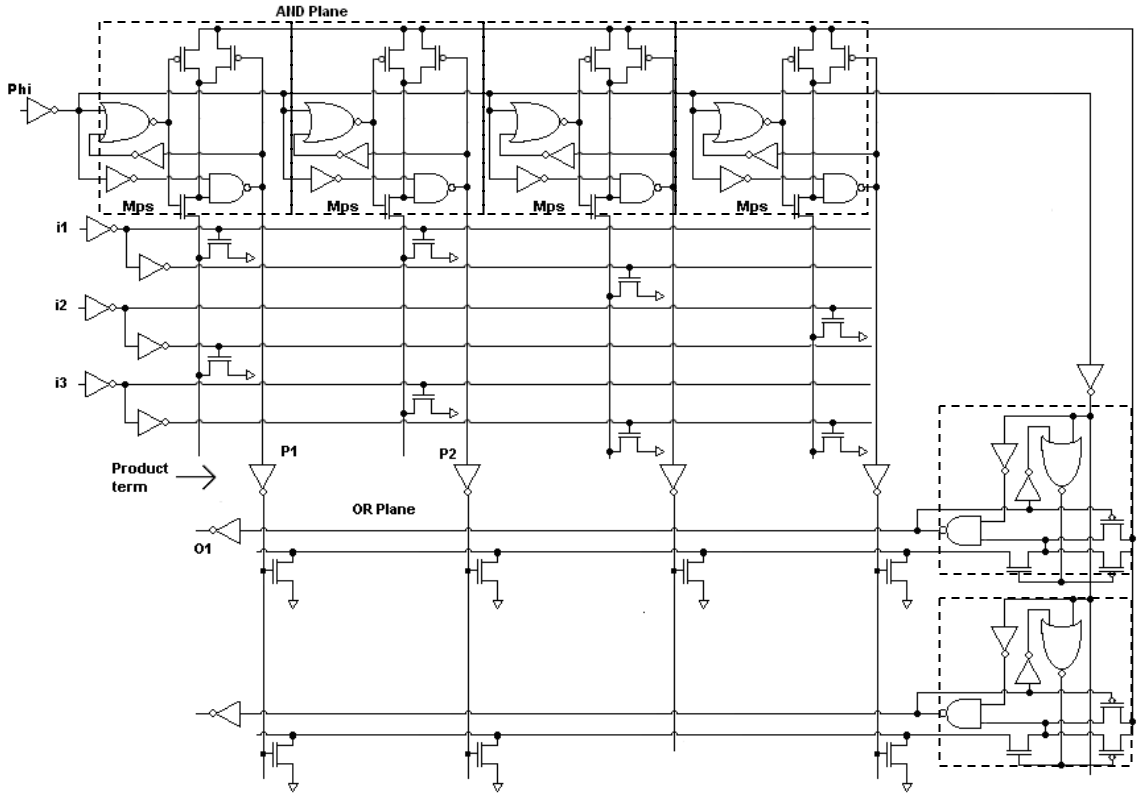
As shown in Figure 3.6, a series connection of pass transistors causes the charge-sharing problem in this super product line approach. During the evaluation phase, in the case that M_4 is turned off and M_2 is turned on, line *A* does not have a pull-up circuitry, and hence the voltage at that line is generally lower than V_{DD}. The charge on the product line will go through M_2 to node *A* and reduce the voltage at product line Y_1. A feedback transistor t_1 is added to pull-up the product line when it should evaluate to high. Since the voltage drop across transistor M_2 could be as large as half of V_{DD}, this feedback transistor is required. However, it degrades the performance of the circuit, and Tien claims in their paper that the delay overhead of adding the feedback transistor is 15.6% [15].



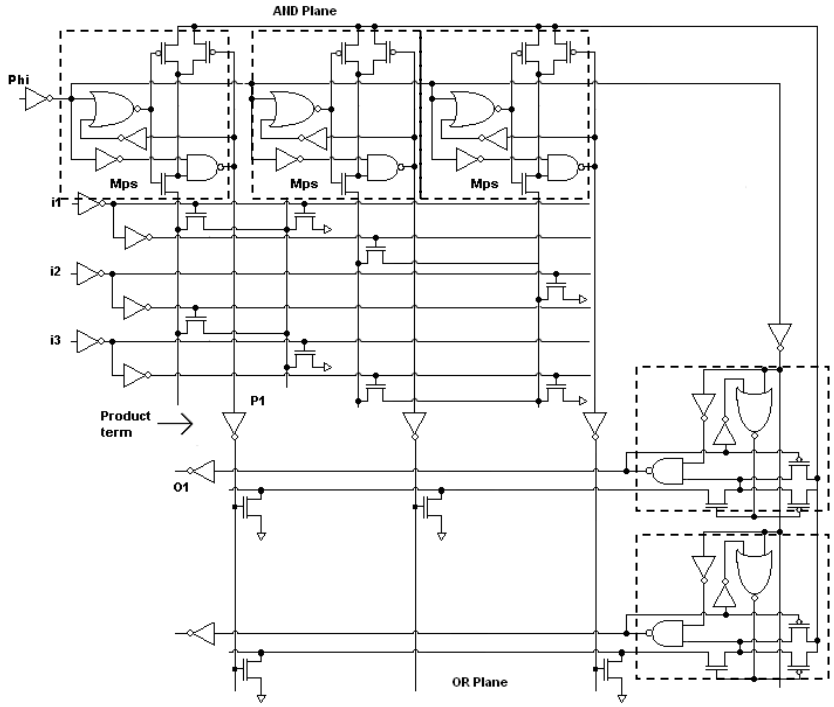
(a) without feedback transistor **(b) with feedback transistor**

Figure 3.6: Charge-sharing in super product line structure

The super product line idea can be used on other PLA designs rather than conventional design. Since the method does not affect the pull-up circuitry, we can combine this approach with Molavi's PLA structure to take advantage of its speed. The same example has been shown in Figure 3.7 with Molavi's design. Note that Molavi's design already has a feedback transistor. Therefore, an additional feedback transistor is not needed. Since the pull-up circuitry of Molavi's design is quite complicated, by removing one product line, we remove one NOR gate, one NAND gate, three inverters and pull-up transistors and also corresponding OR plane pass transistors. The power consumption is greatly reduced, and this is a key reason for the overall improvement.



(a) Molavi PLA before merging product lines



(b) Molavi's PLA after merging product lines

Figure 3.7: Molavi's PLA super product line example

Furthermore, in order to efficiently implement a PLA, logic functions are usually simplified using the Espresso heuristic logic minimizer [26]. This tool minimizes logic so that the product terms will be shared as much as possible by the several output function. This will allow the super product line approach to achieve good results on reducing power consumption, as shown in a later section.

3.3 Benchmark Circuits

To show the delay and power improvements of Tien, Wang and Molavi methods compared to the conventional PLA, a set of benchmark circuits from MCNC are used. This is chosen because Tien's super product line paper has used circuits from this set to generate their results [15]. We need to verify that, using their approach, we can reproduce their results. All the MCNC benchmark circuits are first minimized by the Espresso heuristic logic minimizer. The characteristics of those 24 circuits are shown in Table 3.1. One of the benchmark 'ryy6' used in Tien's work is not included here because it has only one output and not used for the partitioning in the later chapters.

	# of inputs	# of outputs	# of product term		# of inputs	# of outputs	# of product term
alu2	10	8	68	in4	32	20	212
alu3	10	8	66	in7	26	10	54
b9	16	5	119	max512	9	6	145
b10	15	11	100	max1024	10	6	274
bc0	26	11	179	newcond	11	2	31
bcd	26	38	117	newcpla2	7	10	19
chkn	29	7	140	signet	39	8	119
gary	15	11	107	t2	17	16	53
ibm	48	17	173	vg2	25	8	110
in0	15	11	107	x1dn	27	6	110
in1	16	17	106	X6dn	39	5	82
in2	19	10	136	x9dn	27	7	120

Table 3.1: MCNC benchmark circuits information

All PLA definition files are logic-optimized, logically-equivalent representations. Benchmark circuits include combinational circuits implementing various arithmetic and industrial control functions. Figure 3.8 is a scatter plot of their number of I/O's and product terms. Over 70% of the benchmark circuits have less than or equal to 30 I/O's, and more than 90% of them have less than 200 product terms. PLA circuits tend to be small since the delay and power of PLA are closely correlated with number of I/O ports and number of product terms. The 21 circuits chosen here include various I/O and product term numbers.

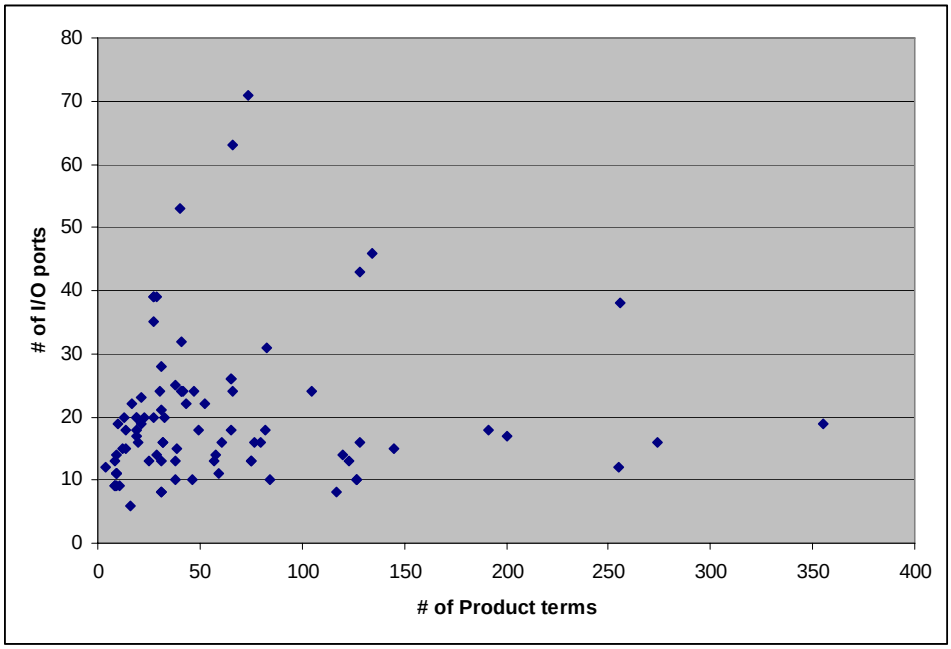


Figure 3.8: The scatter plot of the benchmark circuits

3.4 HSpice Code Generation and Data Collection

The design flow used in generating the simulation data for the benchmark circuits is outlined in Figure 3.9. A C++ program has been developed to automatically generate dynamic CMOS PLA circuits that are suitable for HSpice simulation. The analysis was carried out using 65nm technology with a 1V power supply for the conventional, Tien,

Wang and Molavi PLA designs. This program reads in a PLA definition file, (for example, a simple circuit *conv584* shown in Figure 3.10), and generates corresponding HSpice code for each design structure. Although some of the files were large, the flow was rather straight-forward to set up. This is made possible particularly due to the regularity of the PLA structure.

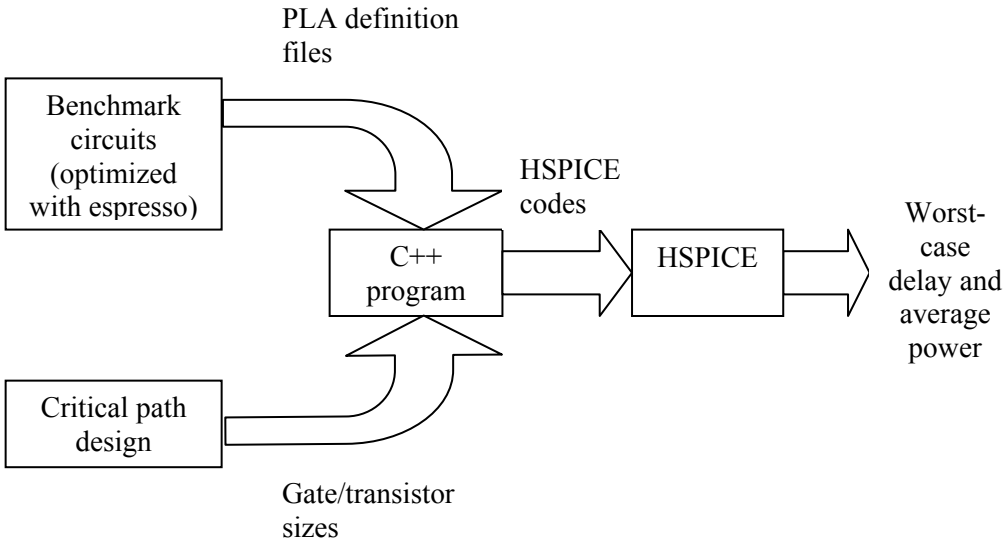


Figure 3.9: The design flow of PLA benchmark circuit simulation

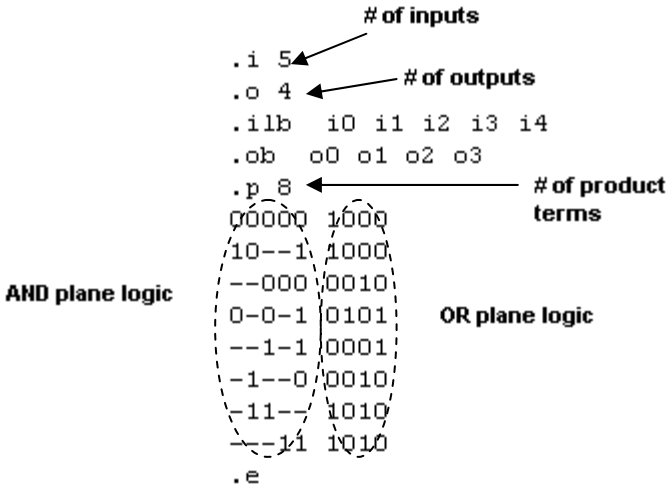


Figure 3.10: Example of PLA benchmark circuit definition file (conv584)

For each of the conventional, Tien, Wang and Molavi PLAs, a single level deep critical path is first designed to obtain pull-up/down, pass transistor, buffer and gate sizes. Then, the program reads in the IPO information of each circuit, and uses the transistor sizes from critical path to provide the corresponding number of pull-up/down, input, output, and inter-plane circuitry netlists in the HSpice file. The size of the clock buffers may change with the size of the PLA for correct operation. The places to insert the AND-plane and the OR-plane transistors depend on the '1's, '0's and '-'s in the definition file. The size of these HSpice files grows quickly and becomes quite large as the number of I/O ports and number of product terms increases.

To carry out the worst-case delay simulation, we need to consider both precharging and evaluation delay. For conventional PLAs, where the product line is connected to a large number of pass transistors, the drain capacitance affects both delays. Therefore, we need to find the number of input pass transistors connected to each product line and number of inter-plane pass transistors connected to each output line. For each output, we use the product line dependency to determine which path has the largest sum of output line capacitance and product line capacitance. For Wang's and Molavi's PLA, there is a NMOS transistor M_{ps} inserted between the product line and pass transistors. Their delays do not highly-depend on the number of input pass transistors. For comparison purposes, we choose the same input for all PLAs.

For the average power simulation, a counting sequence is used as the input. The outputs of the counting sequence are randomly assigned to the PLA inputs by the C++ program to

make the inputs pseudo-random. The problem is that the number of different input vector combination grows exponentially. To get the average power of all possible input vectors for each PLA circuit is time-consuming and may not be possible for circuits with a large number of inputs. However, the PLA has a regular structure and the average power consumed for different input vectors can be determined using only a few vectors. In Figure 3.11, we show the power consumed for different input vectors and the average power of a benchmark circuit with 8 inputs. We notice that after about 20 to 30 input vectors, the average power settles to a relatively constant value. Also, this is an acceptable approach since we are comparing the power of different PLAs with the same input vectors.

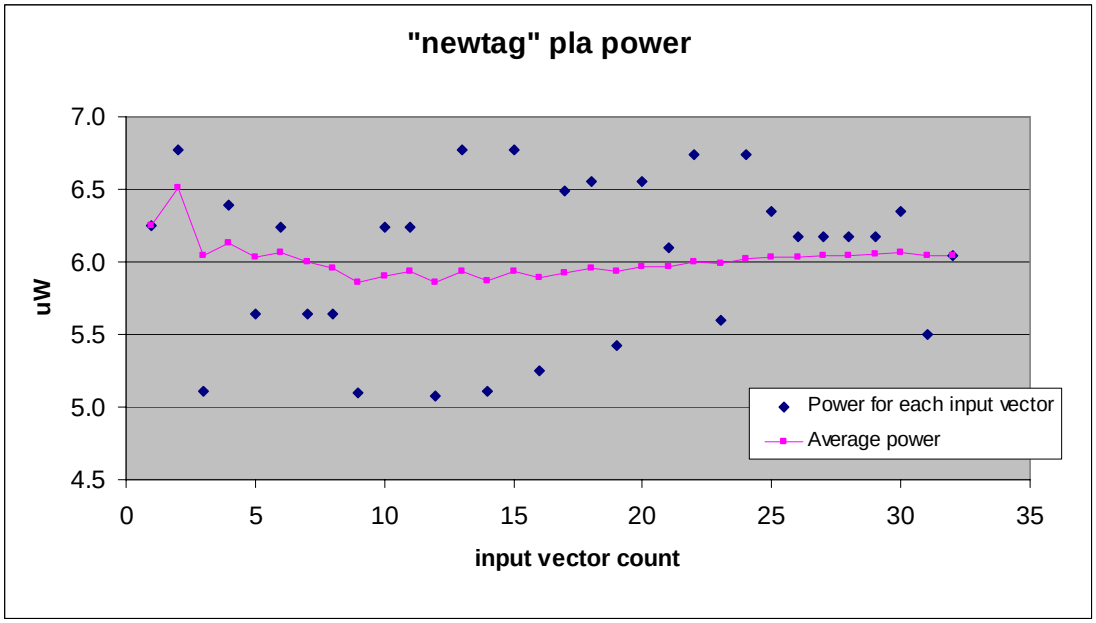


Figure 3.11: Benchmark circuit "newtag" power consumption for different input vectors

3.5 Molavi PLA vs. Other Designs

The power and delay performance of each benchmark circuit were determined using the methods described in the previous section. The results are normalized to the conventional

PLA design and provided in Table 3.2. We used 30 input vectors per benchmark circuits to get the average power consumption. Tien, Wang and Molavi's techniques reduce the power consumption for all the benchmark circuits. Wang has the best power improvement of 47%, followed by Molavi's 40% and Tien at 35%. Molavi's PLA has a lower power improvement than Wang's due to the additional feedback circuitry. Tien's super product line technique increases the delay of the circuit by 5%. On the other hand, Molavi improves the delay by 49% which is better than Wang's 40%. Finally, Molavi has the best PDP (power-delay product) improvement of 69% on average.

MCNC	Power improvement %			Delay improvement %			PDP improvement %		
design	Tien	Wang	Molavi	Tien	Wang	Molavi	Tien	Wang	Molavi
alu2	39%	19%	17%	-10%	33%	44%	32%	45%	54%
alu3	40%	15%	13%	-5%	44%	52%	37%	52%	58%
b9	31%	39%	38%	-2%	42%	58%	28%	72%	70%
b10	34%	50%	43%	-4%	40%	50%	31%	70%	71%
bc0	32%	59%	52%	3%	31%	43%	34%	71%	73%
bcd	27%	72%	70%	1%	43%	52%	28%	84%	86%
chkn	42%	60%	53%	-1%	31%	43%	42%	72%	73%
gary	31%	51%	44%	-2%	39%	50%	29%	70%	72%
ibm	33%	53%	48%	-1%	45%	47%	32%	74%	73%
in0	31%	51%	44%	-3%	39%	49%	29%	70%	72%
in1	32%	69%	62%	-2%	42%	50%	30%	82%	81%
in2	35%	54%	46%	0%	32%	43%	36%	68%	70%
in4	37%	64%	58%	6%	29%	41%	40%	75%	76%
in7	36%	42%	26%	-4%	52%	58%	34%	72%	69%
max512	40%	31%	18%	-3%	21%	35%	39%	46%	47%
max1024	36%	63%	52%	-4%	45%	47%	36%	70%	73%
newcond	42%	29%	5%	-16%	54%	58%	32%	67%	60%
newcpla2	29%	20%	0%	-23%	58%	60%	12%	66%	60%
signet	35%	39%	35%	-9%	35%	47%	29%	61%	65%
t2	26%	35%	33%	-12%	47%	54%	17%	66%	69%
vg2	39%	47%	38%	-5%	34%	45%	35%	65%	66%
x1dn	40%	52%	45%	-6%	35%	45%	37%	68%	70%
x6dn	27%	59%	61%	-8%	58%	62%	40%	63%	76%
x9dn	40%	54%	48%	-6%	33%	44%	37%	69%	70%
Average	**35%**	**47%**	**40%**	**-5%**	**40%**	**49%**	**32%**	**67%**	**69%**

Table 3.2: Tien, Wang and Molavi vs. conventional

Tien's super product line method only depends on logic simplification by merging product lines which could be used on any dynamic CMOS PLA designs. We combine this super product line idea with the Molavi design which has the best delay to further reduce power consumption. Since Molavi's PLA already has a feedback loop which solves the charge sharing effect, it does not need to include any extra feedback transistor. Therefore, using super product line method will not have as much delay overhead on Molavi design as on conventional design. As shown in Table 3.3, the power improvement of the Molavi design after Tien's super product line merging increased from 40% to 54%. However, the delay improvement is decreased from 498% to 43%. After combining Molavi and Tien's techniques, the resulting PLA has the best overall PDP improvement of 74%.

MCNC	Power improvement %	Delay improvement %	PDP improvement %
design	Molavi+Tien	Molavi+Tien	Molavi+Tien
alu2	40%	35%	61%
alu3	39%	46%	67%
b9	51%	55%	78%
b10	55%	45%	75%
bc0	62%	41%	77%
bcd	74%	49%	87%
chkn	68%	39%	81%
gary	55%	45%	75%
ibm	59%	43%	77%
in0	55%	44%	75%
in1	70%	46%	84%
in2	59%	40%	75%
in4	69%	41%	82%
in7	44%	53%	74%
max512	42%	29%	59%
max1024	60%	42%	77%
newcond	35%	48%	66%
newcpla2	16%	48%	56%
signet	50%	38%	69%
t2	42%	44%	68%
vg2	55%	38%	72%
x1dn	61%	38%	76%
x6dn	70%	52%	86%
x9dn	63%	36%	77%
Average	**54%**	**43%**	**74%**

Table 3.3: Molavi+Tien vs. conventional

3.6 Summary

In this chapter, we described the operation details for Molavi and Tien's PLA. Since Molavi's improvement is to change the pull-up circuitry to improve delay, and Tien's approach is to merge product lines to improve power, these two techniques are mutually orthogonal and can be combined to achieve the best power and performance improvement. We simulated the conventional PLA, Tien's improvement of the conventional PLA, Wang's PLA, Molavi's PLA and Tien's idea applied to Molavi's PLA. We found that the combination idea of Molavi and Tien results in the best power and PDP improvement over the conventional PLA.

Chapter 4 PLA PARTITIONING

In the previous chapter, we described a PLA design that combines Molavi's delay improvement and Tien's power improvement to reduce the overall energy by a large percentage. However, the method of merging product lines generally increases capacitances on the product lines and introduces a delay overhead on Molavi's PLA. In this chapter, we will discuss PLA partitioning to further improve the delay, and thereby further improve energy.

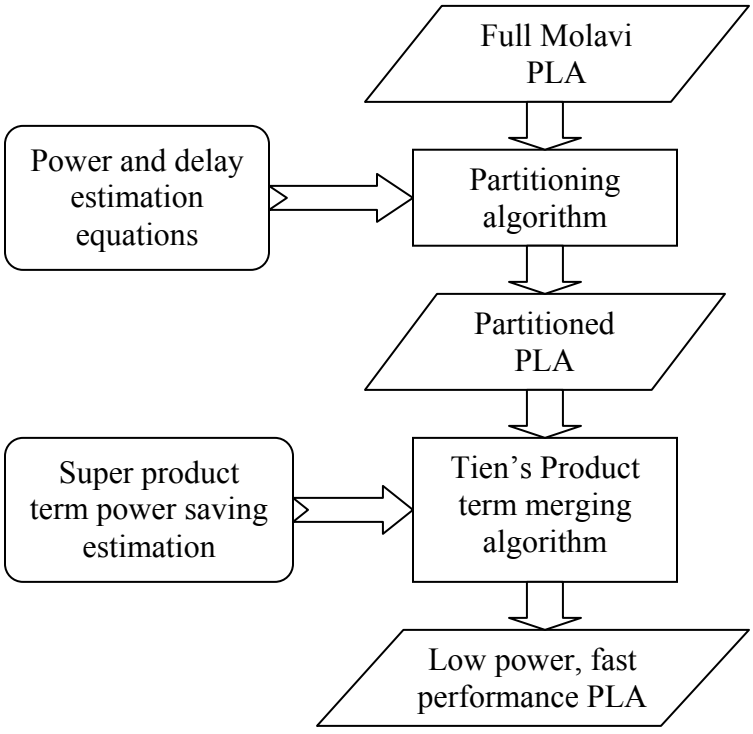


Figure 4.1: Optimization process flow diagram

The research question here concerns the order in which to apply the ideas. The Molavi, Tien and partitioning could be applied in any order. The optimization sequence that we have chosen is shown in Figure 4.1. We have Molavi design as the starting point. The

next step is to partition Molavi's PLA design. The algorithm considers product term partitioning and the effects on power and delay when dividing a large PLA structure into several smaller PLAs. The next step is to use Tien's idea to merge product lines in each partitioned sub-PLA to optimize power consumption. We will show the details of these steps in the following sections. The reasons for choosing the partitioning before merging super product lines is due to the nature of the partitioning algorithm, which will be made clear in section 4.2.

4.1 PLA Partitioning

The number of product terms in a PLA controls the number of pull-up/down circuits, inter-plane buffers, and parasitic capacitance and resistance on input and output lines. Therefore, the PLA delay is closely correlated with the number of product terms. Reducing the number of product terms in a PLA is an effective way of improving its speed. One way to partition a PLA is to build a separate PLA for each output. However, since product terms and inputs may be shared between outputs, this could cause significant redundancy in the circuitry. As shown in Figure 4.2, the total area of some circuits after single-output partitioning could increase as much as nearly 5 times the original circuit implementation. However, for more than half of the benchmark circuits, the total area after partitioning actually decreases. This is due to the fact that, in each partition, the independent primary inputs and product terms form smaller PLAs than the full implementation. The area of each partition is often much smaller compared with original circuit, and so the overall area is usually lower.

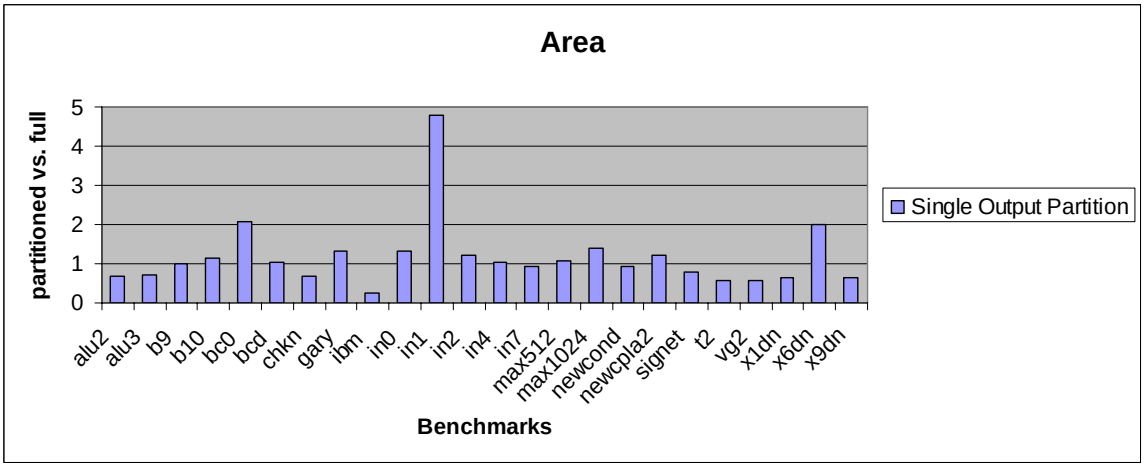


Figure 4.2: Single output partitioning vs. full circuit implementation (area)

It is important to note that a single-output block only includes the product terms and inputs that affects this particular output. This implies that the partitioning that results in the best delay is to create a set of single-output PLAs. Figure 4.3 shows the ratio of the worst-case delay after single-output partitioning and original PLA. In each circuit, the slowest outputs typically depend on the largest number of the product terms. There are several circuits that have less than 20% difference between the slowest single-output partition and original full PLA implementation, such as benchmark *newcond*. On the other hand, for many circuits where the outputs are dependent on small subset of product terms, the improvements are at least twice as fast. For benchmark *ibm*, the single-output partitioned PLA can run 3.5 times faster than its full implementation.

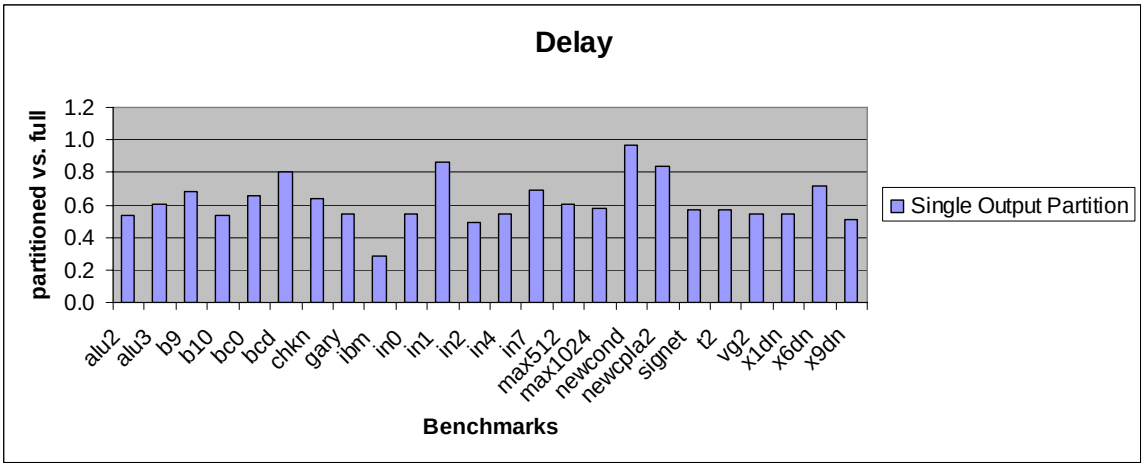


Figure 4.3: Single output partitioning vs. full implementation (delay)

The biggest problem of single-output partitioning is the power overhead. For those circuits with outputs that have many shared product terms and primary inputs, the circuit replication is very area intensive and consumes much more power. Figure 4.4 illustrates the power ratio before and after the single-output partitioning. The benchmark *in1* that takes more than 5 times the area after partitioning also consumes more than 9 times power. About half the benchmark circuits after single-output partitioning consume more than twice as much power than full PLA implementation.

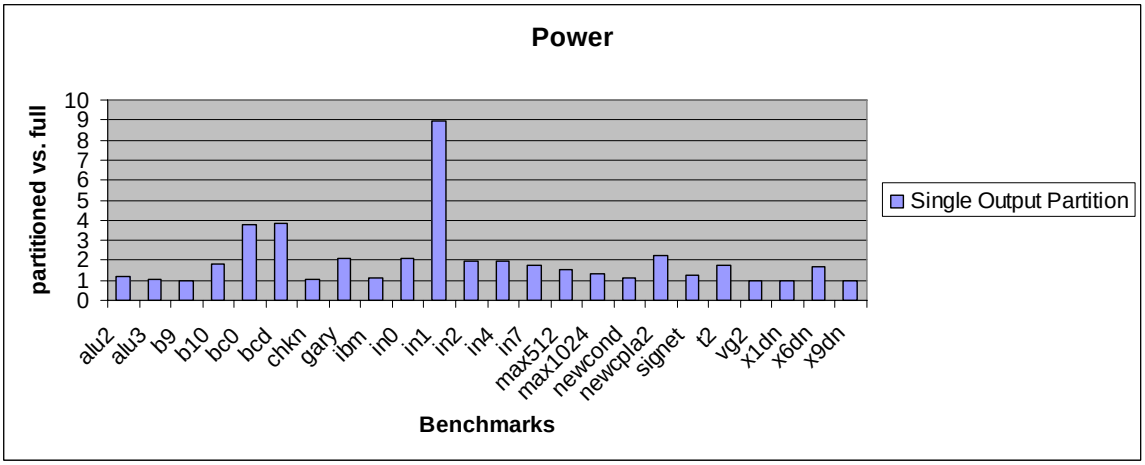


Figure 4.4: Single output partitioning vs. full implementation (power)

Even though the power overhead of partitioning is large for some circuits, the power-delay product after partitioning for more than half of the benchmark circuits is less than full implementation, as shown in Figure 4.5. This means partitioning is useful to minimize energy consumption. The outputs should be combined when they share inputs and product terms, in order to ensure that the area and power overhead is in the acceptable range.

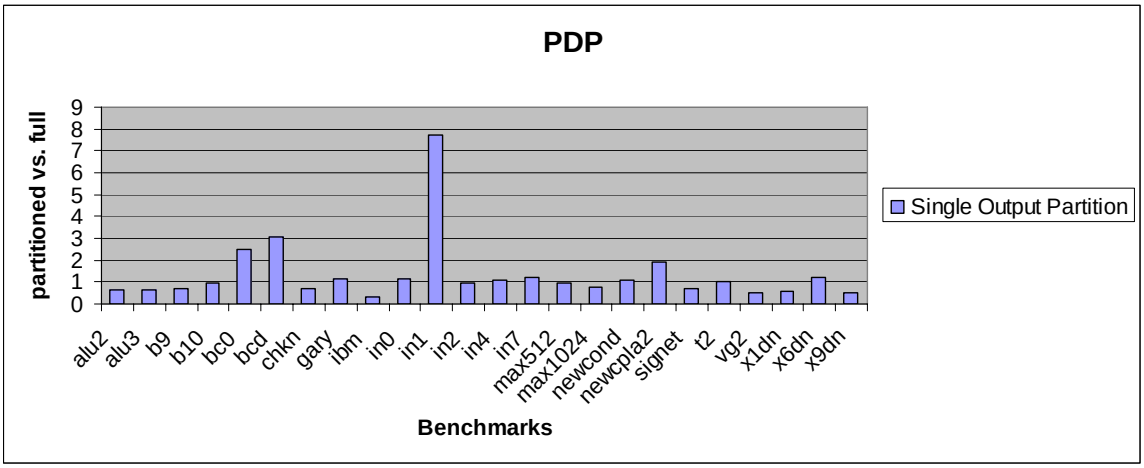


Figure 4.5: Single output partitioning vs. full implementation (PDP)

The simplest but least feasible way to find an optimal partitioning is to try all possible outputs combinations. However, this is a NP-complete problem. Even if we assume only a bi-partition, the number of different combinations grows exponentially with number of outputs. Single-output partitioning gives the best delay but the power consumption is the worst, and the full PLA gives a better power but the delay is the worst. Any other partitioning will have a delay that lies in the middle of these upper and lower bounds, and one point provides the minimum PDP. Before describing the partitioning algorithm, first we will discuss the power and delay models used for the cost function.

4.1.1 Molavi PLA Power and Delay Models

Since the algorithms for partitioning require knowledge of the area, power and delay of each candidate configuration, a set of fast but accurate models are needed to avoid lengthy simulation runs. These models must be validated with actual results before they can be used. Due to the structural regularity of a PLA, it is possible to formulate relatively accurate area, power and delay characteristics of a PLA before layout. In this chapter, we will focus on such power and delay models, while the area model will be described in the next chapter.

To compute the power consumption, we break the circuit into smaller parts to simplify the equations. Consider the PLA diagram in Figure 4.6; the total power will include clock power, input power, AND plane power and OR plane power:

$$Overall\_Power = Clock\_Power + \sum_{I} Input\_Power(I) + AND\_power + OR\_power$$

Basically, the switching characteristics of all the internal nodes depend on either the clock signal *phi* or the inputs. The switching activity of clock signal *phi* is 50% so the power consumption on the clock lines is simple and given by:

$$Clock\_Power = \frac{1}{2} V_{DD}^2 f \left(C_{phi} + C_{clk\_a} + C_{ff\_a} + C_{phi\_d} + C_{clk\_o} + C_{ff\_o} \right)$$

where V_{DD} is the power voltage, f is the clock frequency, and capacitances are as shown in Figure 4.7. Note that even though input lines I and I' are physically different lines, their functions are complementary. Assume the probability of input I being high is $Prob^1(I)$, we can express the input power consumption of a particular input I as:

$$Input\_Power(I) = \frac{1}{2} C_{in} V_{DD}^2 f \, \mathrm{Prob}^1(I)$$

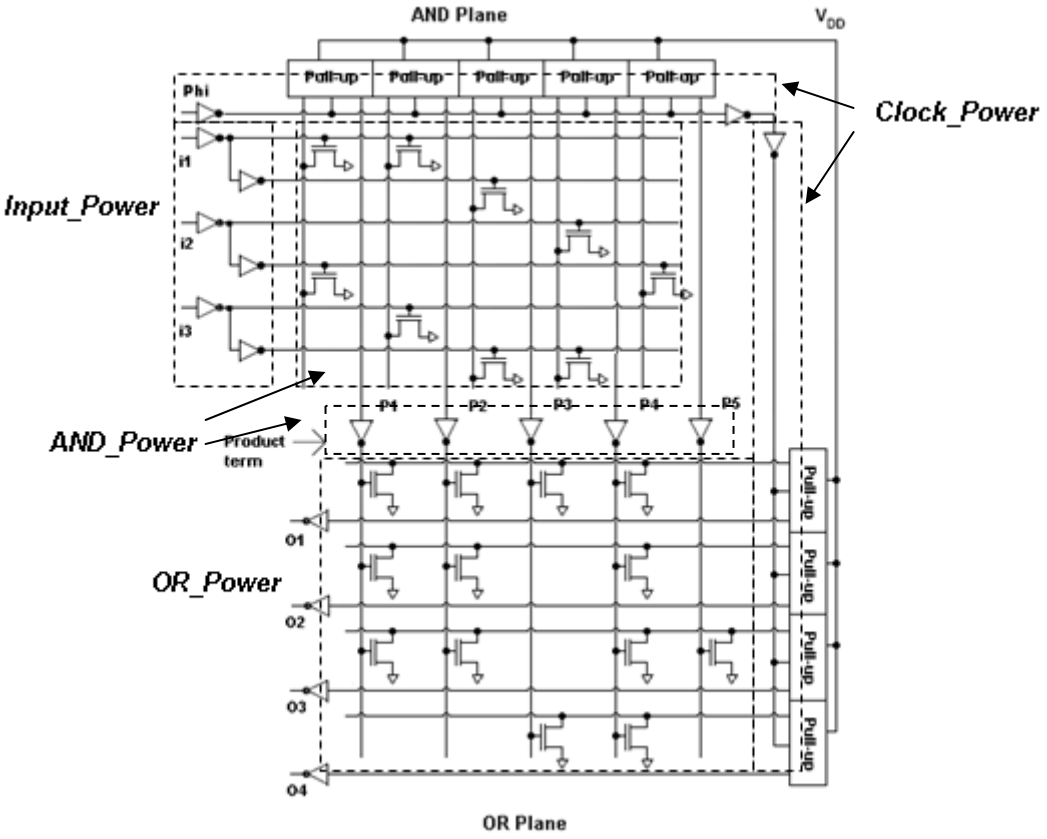


Figure 4.6: Molavi's PLA power module

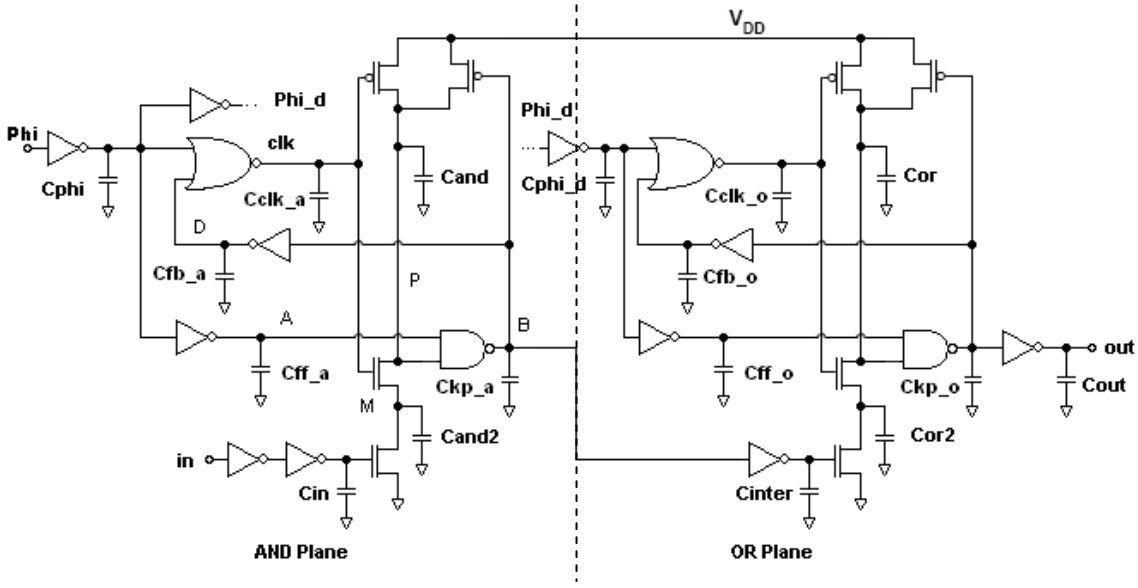


Figure 4.7: Molavi's PLA critical path

Assume that the probability of input I being low is $Prob^0(I)$. The product line is precharged high in every clock cycle. It stays high only when all the inputs I/I' connected to the product line are low. Let set S be the collection of all the inputs that affect the product line, and $I \in S$. The probability of one AND plane product line P staying high is:

$$\text{Prob}^1(P) = \prod_{I \in S} \text{Prob}^0(I)$$

which means the probability of a product line switching low is:

$$\text{Prob}^0(P) = 1 - \text{Prob}^1(P)$$

The power consumption of each product line together with the power consumption of feedback circuitry is:

$$product\_line\_power(P) = \frac{1}{2} V_{DD}^2 f \left(C_{fb\_a} + C_{and} + C_{and2} + C_{kp\_a} + C_{inter} \right) \text{Prob}^0(P)$$

The power consumed by the AND plane will be estimated by summing up all the power in the product lines:

$$AND\_power = \sum_{j=1}^{n} product\_line\_power(P_j)$$

where n is the number of product lines.

Similarly for the OR plane, node Q is precharged high and only discharged low when the product lines connected are all evaluate to high. Let the set U be the collection of all the product terms that affect the output, and $P \in U$. The probability of node Q being low is

$$\text{Prob}^0(Q) = 1 - \prod_{P \in U} \text{Prob}^0(P)$$

Power consumed by each output line can be estimated by:

$$output\_line\_power(Q) = \frac{1}{2} V_{DD}^2 f\left(C_{fb\_o} + C_{kp\_o} + C_{or} + C_{or2} + C_{out}\right)\text{Prob}^0(Q)$$

The power consumed by the OR plane will be the sum:

$$OR\_power = \sum_{k=1}^{m} out\_line\_power(Q_k)$$

where m is the number of outputs.

Now consider the critical path of Figure 4.7. The delay of a path containing logic gates and capacitive loads is computed with the equation: $D = \sum R_i C_i$ where R_i is the effective resistance (*Reff*) of the ith gate, and C_i is the capacitive load driven by the gate [26]. The dynamic CMOS PLA has precharge and evaluation phases. The precharge delay is measured from clock signal *phi* going low to the vertical product lines or the horizontal output lines charging up. The evaluation delay is measured from clock signal *phi* going high to the pull-down path discharge of the product or output lines. The delay depends on the size of the clock buffer, the NOR gate, pull-up transistor, pseudo-footless pass transistor, the NAND gate as well as the loads. The precharge delay t_p and evaluation delay t_e are computed as:

$$t_p = R_{phi}C_{phi} + R_{NOR}C_{clk\_a} + R_{up}C_{and} + R_{ps}C_{and} + R_{down}C_{and2} + R_{NAND}C_{kp} + R_{\text{inter}}C_{\text{inter}}$$

$$t_e = R_{phi}C_{phi} + R_{NOR}C_{clk} + R_{up}C_{or} + R_{ps}C_{or} + R_{down}C_{or2} + R_{phi\_d}C_{phi\_d} + R_{NAND}C_{kp} + R_{out}C_{out}$$

$$delay = t_p + t_{\text{margin}} + t_e$$

To ensure correct evaluation, the OR plane needs to wait for the AND plane results to be stable. A 10% t_{margin} is added for safety to take into account for timing changes due to process and environment variations. Figure 4.8 shows the ratio of the simulation and

estimation results from the equations just described. The capacitive loads are calculated according to the IPO numbers for the PLA. The gate or junction capacitances were added based on the number of connections necessary for each product line or output line when reading the ".pla" format netlist of each circuit. With the 10% safety margin, the delay calculation results are slightly higher than the simulation, which sets the upper bound of the real delay. However, the calculated power is not as precise as the delay calculation. It can be off by ±20%. This is because average power of PLA depends on the logic function and input vector and is not as closely correlated to the product terms as delay.

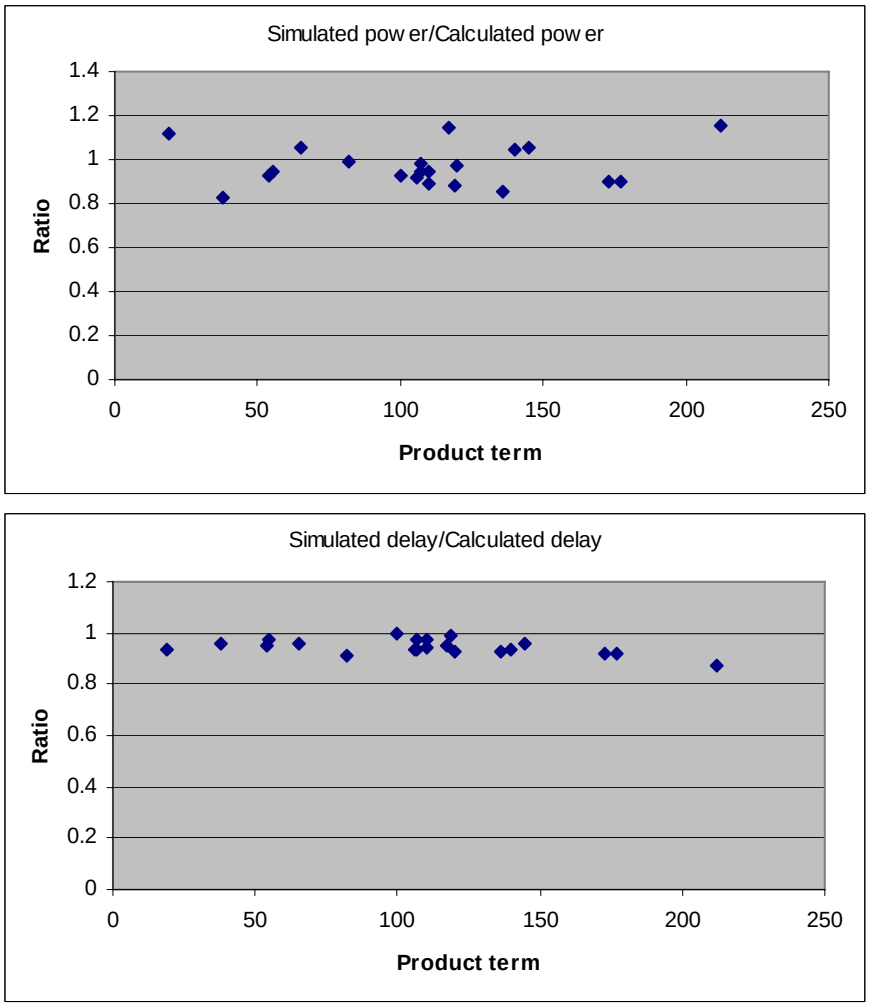


Figure 4.8: Plot of HSpice simulation vs. calculated power and delay with respect to number of products

4.1.2 Partitioning Algorithm

The goal during partitioning is to find the minimum of the PDP function by changing the combination of outputs. In our approach, we set no restrictions on the final sub-PLA sizes. Each PLA resulting from partitioning could have any number of inputs, outputs, and product terms, that are bounded only by the number of primary inputs, outputs, and total product terms. Also, we do not allow output splitting which means each output function only appears in one and only one of the sub-PLA. The partitioning problem is generally defined as: *given any particular PLA implementation of a circuit, find the partitioning of the outputs that results in the lowest cost function*.

The cost function could be a weighted linear combination of power and delay of the PLA, such as $Cost(f) = \alpha P + \beta D$, or their product combination if energy is the concern, such as $Cost(f) = \alpha PD$. One can include area or other metrics into the cost function that concerns designers. We can also normalize each term to original full implementation as $Cost(f) = \alpha\, P_{part} / P_{full} + \beta\, D_{part} / D_{full}$. In this cost function, we concentrate on the improvement over the full implementation. For our purpose of improving the energy, we will consider PDP as our metric here. In this chapter, we use $Cost(f) = PDP_{part} / PDP_{full}$ as the cost function.

For each benchmark circuit, we constructed a weighted graph where nodes represent the PLA outputs. Figure 4.9 shows an example of the graph for benchmark *conv584*. Each edge between nodes n_i and n_j has a weight w_{ij} to represent the number of common product terms shared between the partitions. The benchmark circuits used here were

optimized by Espresso, which means the full implementation has the maximum product term sharing among the outputs.

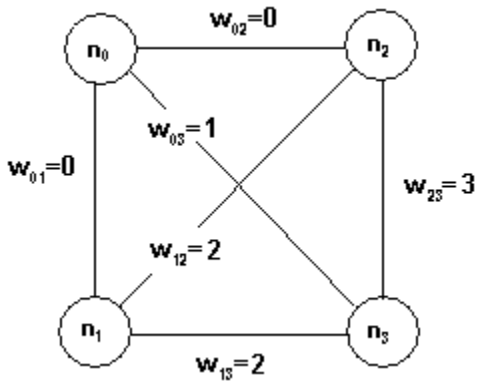


Figure 4.9: Example of weighted graph (*conv584*)

A simple algorithm for greedy partitioning is shown as follows:

Algorithm **Greedy Partitioning**
begin
Estimate PDP of full implementation of the circuit;
Estimate the total PDP of a single-output partitioning;
$\alpha_r = PDP_{single}/PDP_{full;}$
Build weighted graph for circuit, sort edges by weight, and un-label all edges;
while graph has unlabelled edges
begin
Choose unlabelled edge w_{ij} with largest weight and combine outputs as a new partition;
Re-estimate total PDP of new partitioning;
if new ratio $\alpha_p = PDP_{single}/PDP_{full}$ is less than α_r **then**
Set $\alpha_r = \alpha_{p;}$
Eliminate edge and update weights on other edges;
Un-label all edges;
else
Return to previous partitioning;
Label edge;
end if
end while
end Greedy Partitioning

An example of the algorithm is shown in Figure 4.10. We start with a single-output partitioning and carefully combine outputs together by considering the number of shared product terms. A label is used for each edge after it has been considered but the outputs

are not combined in this cycle (to prevent an infinite loop). For the *conv584* example, the edge $w_{23}=3$ is the largest weight, which means these two outputs share 3 product terms. If combining these two outputs results in a lower PDP, we will eliminate the edge w_{23}, update the weights on other edges and continue.

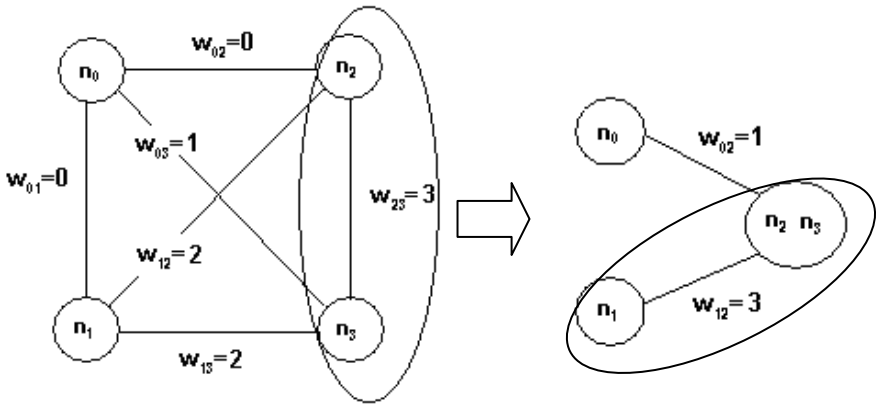


Figure 4.10: Example of greedy algorithm (*conv584*)

While this algorithm is greedy in nature, the runtime is $O(N^2)$ and will increase for circuits with a large number of outputs. For large circuits of having more than 30 outputs, an algorithm based on simulated annealing is used. The simulated annealing algorithm used here is described as follows:

Algorithm **Simulated Annealing**
begin
Start with a random partitioning;
Set $\alpha_r = PDP_{part}/PDP_{full;}$
Set initial Temperature = 1;
while Temperature > 0.001
do 10*number of outputs **iterations**
Randomly choose 2 output notes and merge
$\Delta\alpha = PDP_{new}/PDP_{full} - \alpha_r;$
$r=random(0,1);$
if $r<e^{-\Delta\alpha/T}$ **then**
Keep the merge;
Set $\alpha_r = \alpha_p;$
end if
end iteration
Temperature=Temperature *0.7;
end while
end Simulated Annealing

4.1.3 Partitioning Results

The results of the new partitioning from the algorithm over a full circuit implementation are shown in Figure 4.11 and Figure 4.12. Every benchmark circuit must be partitioned into at least two sub-PLAs in our approach. They are compared side-by-side with single-output partitioning results. Note that for half of the benchmark circuits, single-output partitioning is the optimal one that has the lowest PDP; therefore, the power, delay and area results before and after running the partitioning algorithm are the same.

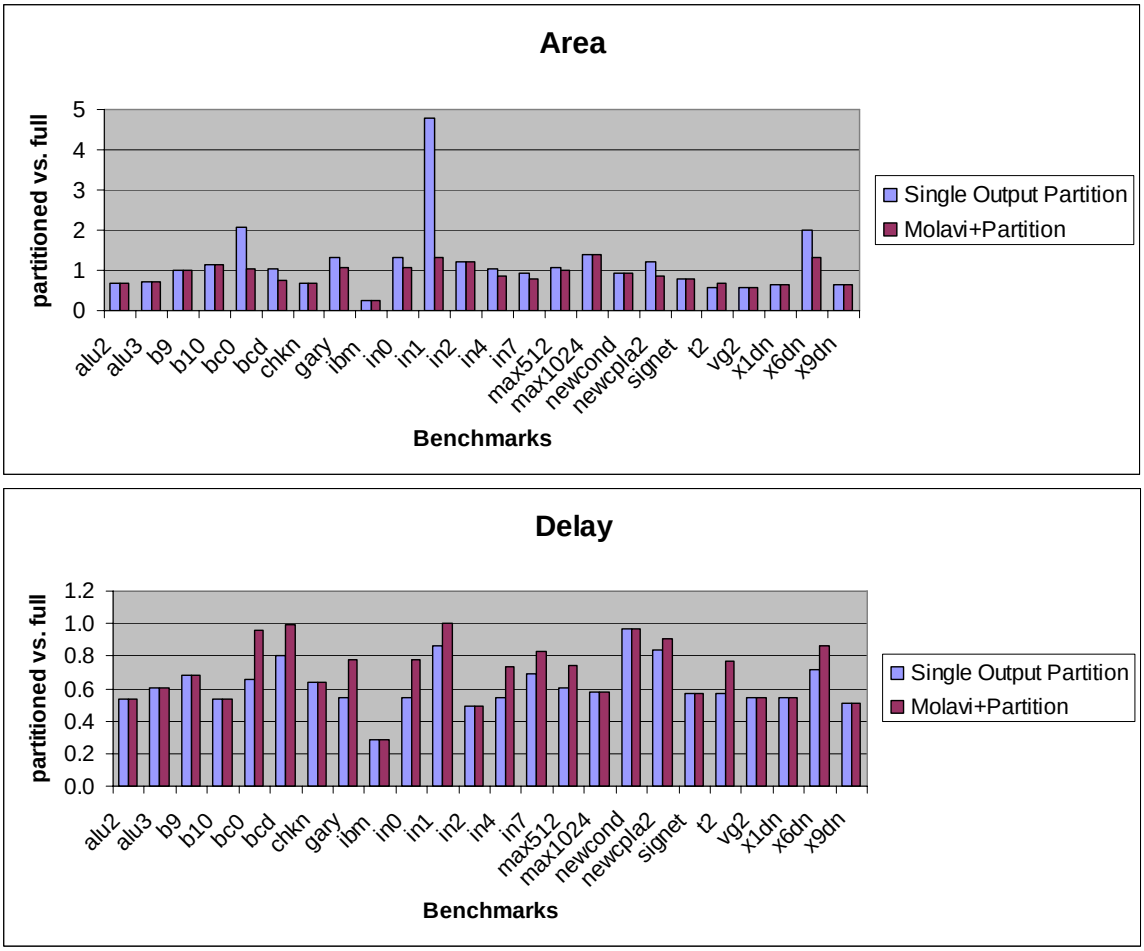


Figure 4.11: Multi-output partitioning vs. single output partitioning

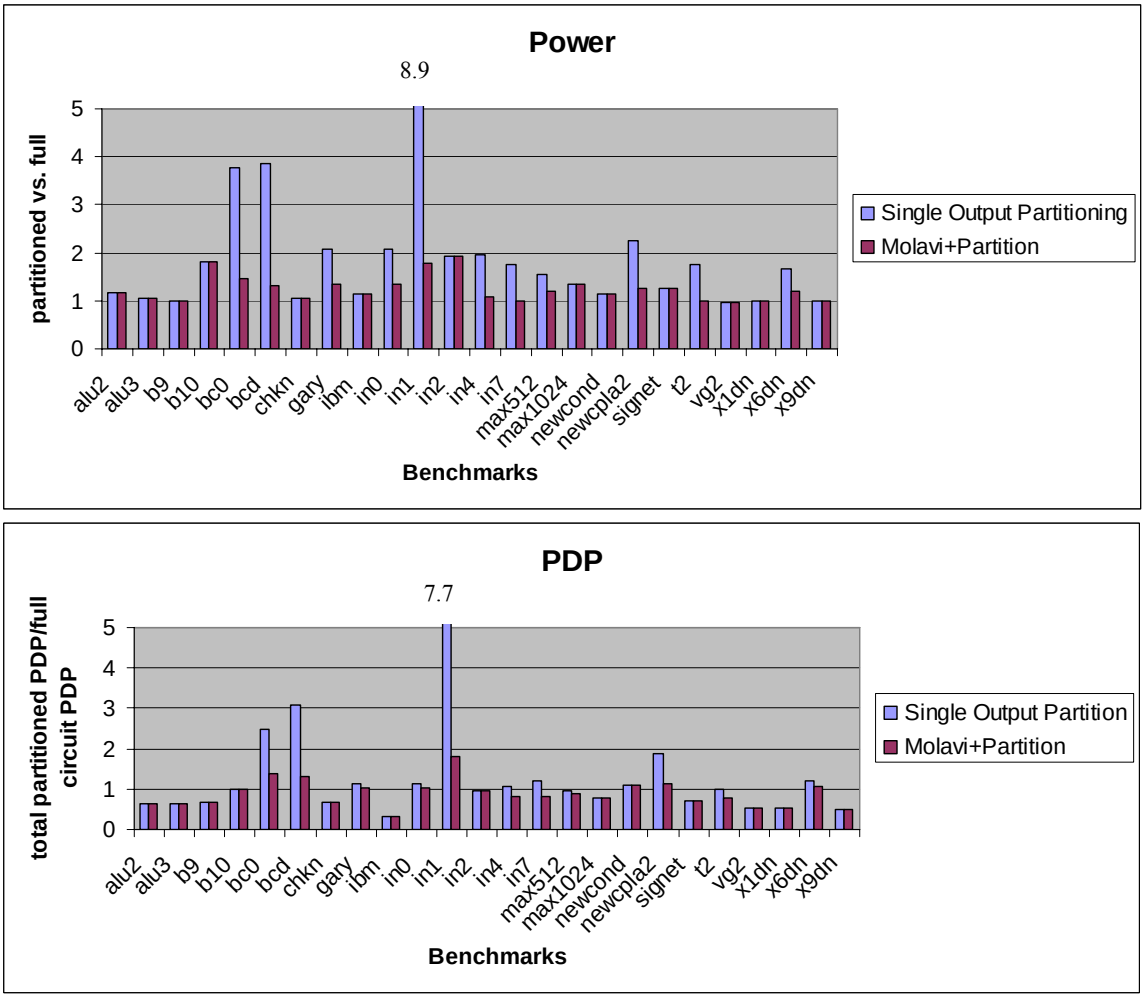


Figure 4.12: Multi-output partitioning vs. single output partitioning

Generally, the area taken for multi-output partitioning is smaller than single-output partitioning. This is because we combine the outputs by considering common product terms which reduces the redundancy. The area for benchmark *in1* decreases from about 5 times after single-output partitioning to less than 1.5 times after multi-output partitioning. Also, the power consumption for this circuit reduces from 9 times to less than 2 times. Multi-output partitioning will slightly increase the timing from single-output partitioning since the partitioned circuits are larger. For example, the delay for benchmark *in1* increases to the same delay as the full implementation of the circuit. The power of the

circuits is reduced by the multi-output partitioning. All of them have decreased to less than twice the power of the full implementation. The PDP for the circuits are also reduced. However, there are still one third of the benchmark circuits having larger energy consumption after multi-output partitioning than full implementation. We can still do better and, in the next step, we will use Tien's super product lines to further reduce the power.

4.2 Super Product Line in sub-PLA

On average, the total power consumption after the multi-output partitioning process is 25% larger than full circuit implementation. It is time to apply Tien's super product line method to further reduce the power. After partitioning, in each sub-PLA there will be a higher percentage of common product terms among the outputs. This is feature that Tien's approach can use to take advantage.

4.2.1 Super Product Line Merging Algorithm

Tien proved a Lemma in their work [15] that the optimum pair selection of product lines can be obtained from the maximum-weighted matching in the PSG (power-saving graph). Since every edge in the graph is the power saved by merging the two product lines, finding the matching that include the maximum weight of each pair of notes will result in maximum reduction of the total power consumption. The algorithm has two major steps: to build the PSG and to find the maximum-weighted matching of the PSG. The steps in building the PSG graph are briefly described below.

```
Algorithm Build PSG
begin
        foreach product line P_i do
        begin
                Add P_i node to PSG graph;
                Calculate power consumption of product line P_i;
                foreach other product line P_j do
                        if P_i and P_j used for same output then
                                calculate power saving after merging;
                                add edge ps(i,j) to PSG graph;
                        end if
                end foreach
        end foreach
end Build PSG
```

The power equations used in this algorithm for a super product line Y_i merging from product lines P_i and P_j were shown in Chapter 2. With these equations, a PSG can be constructed. The nodes of the graph are the product lines, and the edges between the nodes have a weight *ps(i,j)* representing the power saving when product lines P_i and P_j are combined together. Only the pairs of product lines that are used at the same time for the outputs can be merged; therefore, the graph is far from a complete graph. However, for any two product lines P_i and P_j that have the same output set, we can choose to stack P_i to P_j or P_j to P_i. These two choices generally result in different power savings. When we build the graph, we only choose the combination that can produce a larger power savings. In Figure 4.13 below, we show an example of the simplified Molavi's PLA and its PSG. The details of the Molavi pull-up circuitry are illustrated as a box to simplify the PLA diagram.

The maximum power savings can be found by using a maximum-weighted matching algorithm that produces an optimum solution in polynomial time [33]. The time complexity is bound by the maximum weight algorithm which is $O(N^3)$, where N is the

number of product lines. The solution for the example in Figure 4.13 is to merge P_1 and P_2 as super product line Y_1, merge P_3 and P_4 as super product line Y_2 and leave P_5 as regular product line.

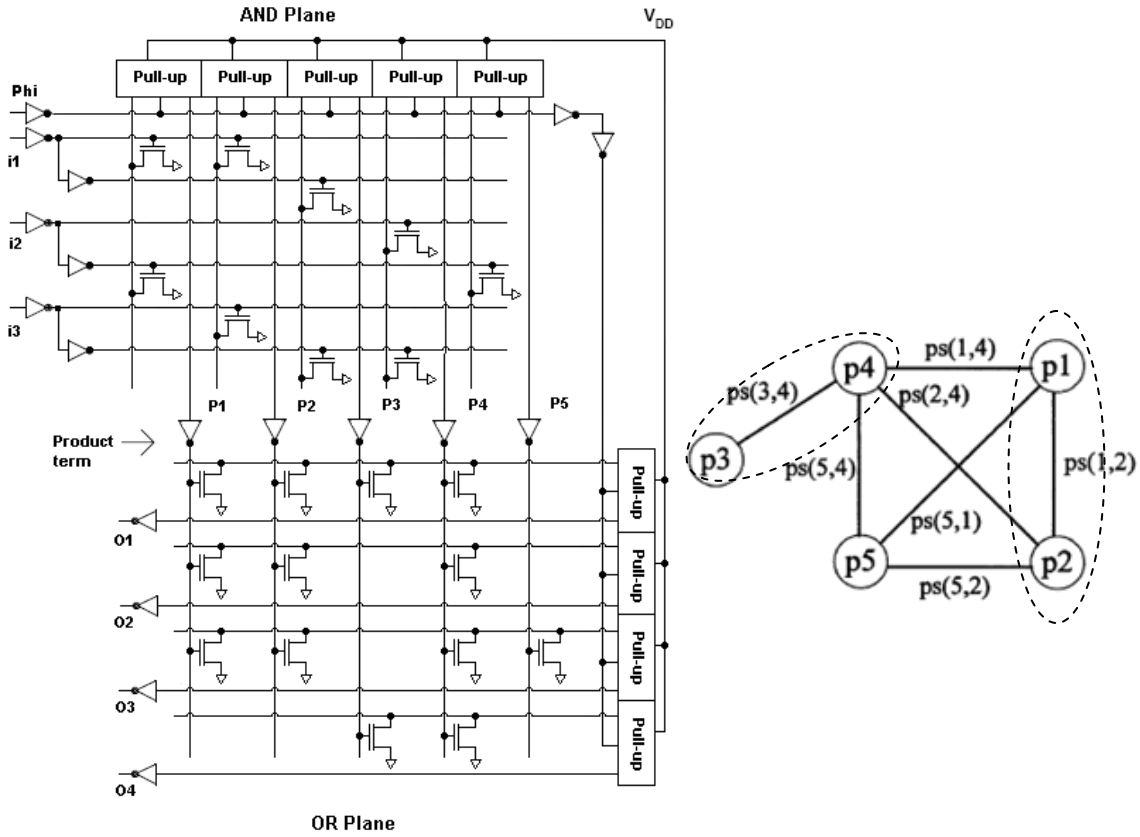


Figure 4.13: Example of PLA structure and corresponding PSG

4.2.2 Super Product Line Results

The power, delay and PDP of the benchmark circuits after product line merging are shown in Figure 4.14, and are compared side-by-side with the results before super product line merging. Not all the partitioned sub-PLAs allow for the super product line approach, but the total power is reduced for all the circuits. The delay increases slightly, and the PDP for most of the benchmark circuits are reduced to less than full implementation. There are circuits that, no matter how you do the partition, they always

have an energy consumption that increases. Since their outputs are sharing most of the product terms and are highly-dependent on each other, the full implementation is the best for minimum energy.

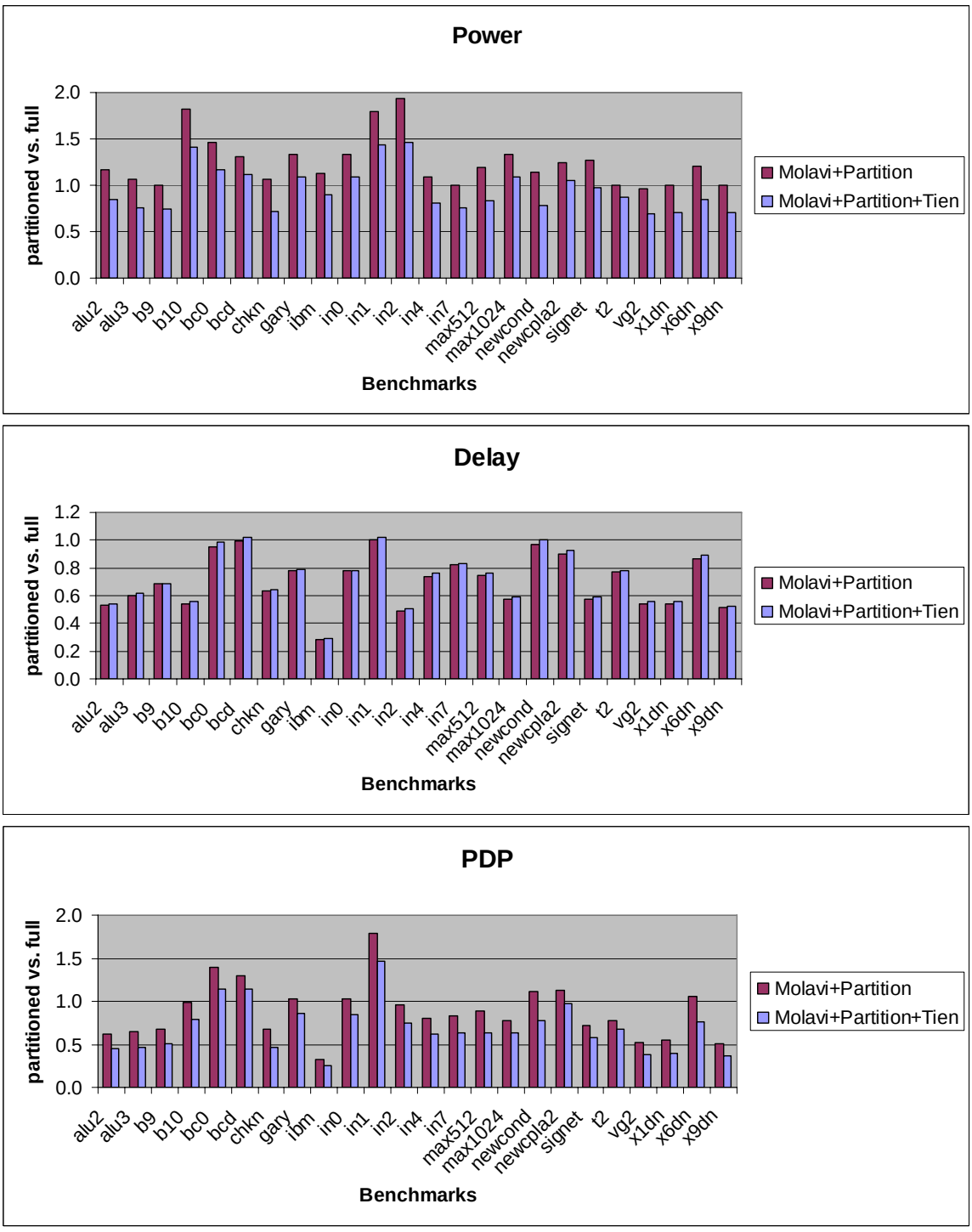


Figure 4.14: Molavi+Partition vs. Molavi+Partition+Tien

4.3 Summary

In this chapter, we considered PLA partitioning to reduce the delay and energy consumption. We start with single-output partitioning and have found that this has large area and power overhead. Outputs are combined by considering common product terms shared between the outputs. A simple algorithm is used to produce the multi-output partitioning. This reduces the total power consumption but increases delay. After partitioning the PLA into sub-PLAs, we apply Tien's super product approach to further reduce the power. The final results are show in the Table 4.1. There are three benchmark circuits that after partition and product line merging the PDP are getting worse. For these circuits, we should revert to the original Molavi implementation. Compared with full Molavi implementation, this method improves power by 5%, delay by 28% and PDP by 31% on average. As shown earlier in Table 3.3, a Molavi design with Tien's approach improves power consumption of conventional design by 54%. Because partitioning results in more power consumption, even after Tien's approach, the power improvement of partitioning over conventional design reduces to 44%. However, this method improves the delay ratio over conventional design from 43% to 63%. The improvement on PDP is also significant, increasing from 74% to 79%. This is highest energy improvement of any PLA reported to date.

	Power improvement	Delay improvement	PDP improvement	Power improvement	Delay improvement	PDP improvement
MCNC	Molavi+Partition+Tien vs. Molavi			Molavi+Partition+Tien vs. conv.		
alu2	16%	46%	55%	30%	70%	79%
alu3	25%	38%	54%	34%	70%	81%
b9	26%	31%	49%	54%	71%	85%
b10	-42%	45%	21%	19%	72%	77%
bc0	-17%	2%	-15%	44%	44%	69%
bcd	-12%	-2%	-15%	66%	51%	84%
chkn	28%	36%	54%	66%	63%	87%
gary	-8%	21%	14%	39%	60%	76%
ibm	11%	71%	74%	53%	85%	93%
in0	-8%	22%	15%	39%	60%	76%
in1	-44%	-2%	-47%	45%	49%	72%
in2	-47%	49%	26%	21%	71%	78%
in4	19%	24%	39%	66%	55%	85%
in7	25%	17%	37%	44%	65%	81%
max512	16%	24%	36%	31%	51%	66%
max1024	-8%	41%	36%	48%	69%	83%
newcond	22%	0%	22%	26%	58%	69%
newcpla2	-5%	7%	2%	-5%	63%	61%
signet	2%	41%	43%	37%	69%	80%
t2	13%	22%	33%	42%	64%	79%
vg2	31%	45%	62%	57%	70%	87%
x1dn	30%	44%	61%	61%	69%	88%
x6dn	15%	11%	24%	67%	66%	82%
x9dn	30%	47%	63%	63%	71%	89%
Average	**5%**	**28%**	**31%**	**44%**	**63%**	**79%**

Table 4.1: Molavi+Parition+Tien vs. Molavi and conventional design

Chapter 5 PLA VS. ASIC COMPARISONS

In the previous two chapters, we described approaches that can reduce power consumption and delay of PLAs. In this chapter, we will present the advantages and disadvantages of PLA structures with respect to standard cell ASICs (SC-ASIC) and evaluate the performance of the improved PLA structure by comparing with SC-ASIC designs for a set of benchmarks.

5.1 Standard Cell Flow and Data Collection

We use the Berkeley PLA Test set (which is included with Espresso as part of the SIS package [20]) in this chapter, because it is the basis of Mehrabadi's work [6] on conventional PLA vs. ASIC comparisons. These benchmark circuits also demonstrate that the results hold across different benchmarks. This benchmark is used for both PLA and SC-ASIC to ensure that both simulations start from a common RTL description. The design flow used to collect area, power and delay information for SC-ASIC implementation are shown in Figure 5.1. The sequence of steps is the same as Mehrabadi [6], except that we have one more step to generate power results. First, Synopsys dc_shell™ is used to compile each benchmark circuit to RTL and gate-level Verilog netlists. The RTL description defines each output as a sum of product terms and, in turn, the product terms are defined as a product of primary inputs. Timing constraints are specified to ensure the tool does not use minimum size gates and stays within the timing limits of the library. After RTL simulation, the circuits are compiled and optimized to generate gate-level Verilog netlist. A testbench is also generated from a wrapper

generator used by Mehrabadi. The gate-level simulation results are compared with RTL results and the outputs are checked to ensure they match the ones from the RTL simulation for the same inputs. Then, the gate-level netlist is passed to Cadence SOC Encounter™ for placement and routing. From this step, we obtain the layout area, extract line RC loads, and collect node activity. This information and the gate-level netlist are passed to the Synopsys Prime-Time™ and Prime-Power™ tools to obtain worst-case delay and average power of the circuits.

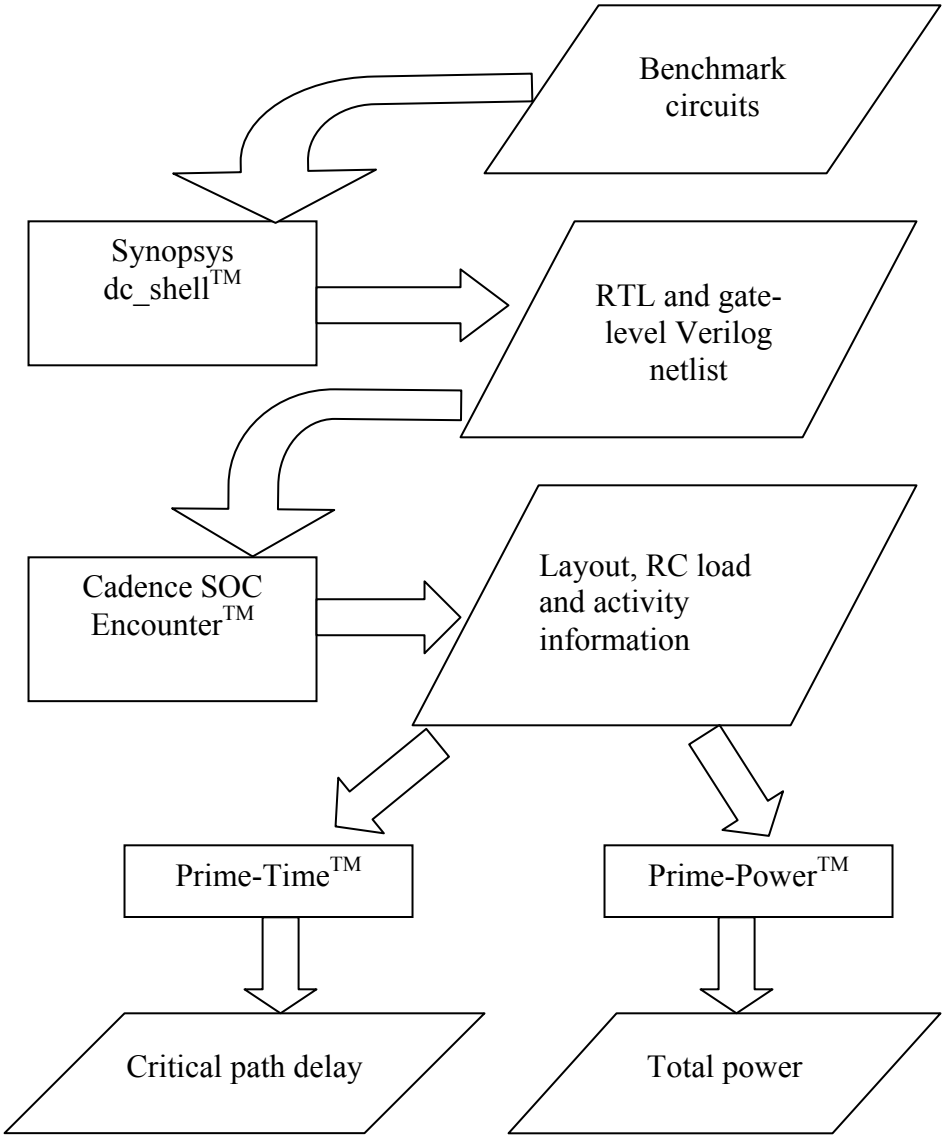


Figure 5.1: The SC-ASIC design flow and CAD tools used

5.2 Area Comparison

PLA area is easy to calculate since it is a regular structure. Without folding and other area reduction processes, the area of the PLA is directly proportion to number of inputs, outputs and product terms. A layout of a simple PLA benchmark circuit *misex1* with 8 inputs, 12 product terms and 7 outputs is shown in Figure 5.2. This is an implementation of Molavi's PLA without partitioning and super product line merging. The pull-up circuitry requires a lot of area compared with conventional PLA design, but this also reduces the precharging delay, as shown in the previous chapters.

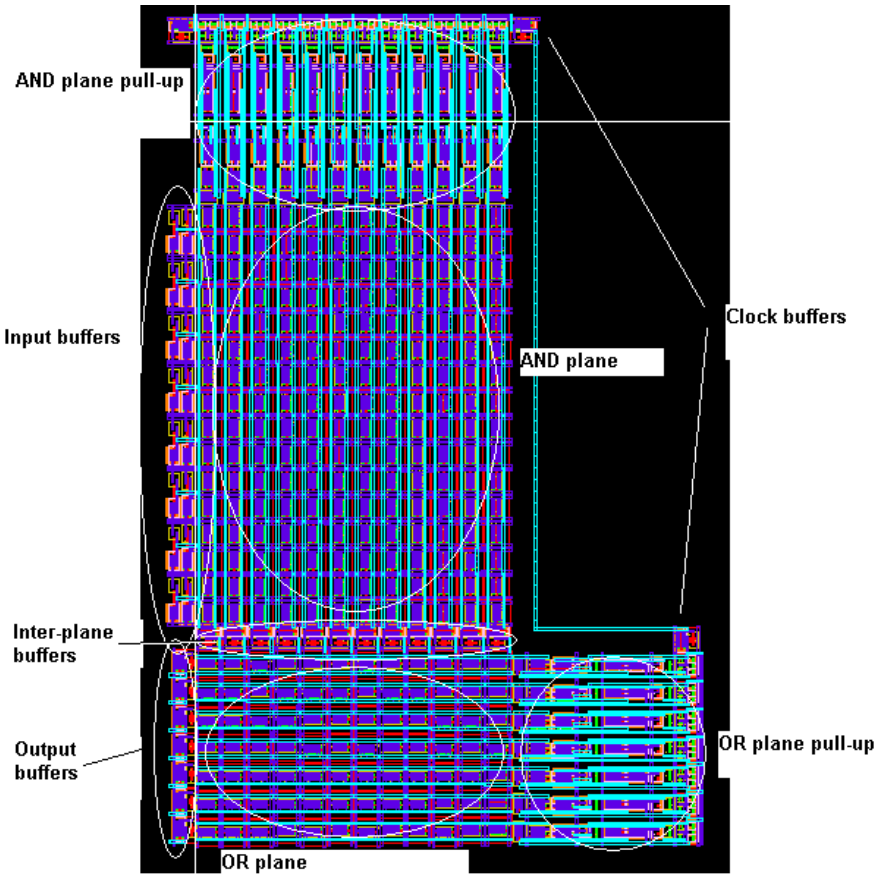


Figure 5.2: Example layout for Molavi's PLA (*misex1)*

The sizes of pass transistors and pull-up circuitry are relatively constant. We developed a simple empirical function that estimates the area of this PLA structure:

$$Area = 300 \times (product + output) + 85 \times product \times (input + output)$$

This function provides the area of a Molavi PLA when the number of inputs, outputs, and product terms are known. The area result is in terms of λ. For example, a PLA with 8 inputs, 12 product terms and 7 outputs, the area is estimated as 21000 λ^2; for 0.18μm technology, λ=0.1 μm, and the area is estimated to be 210 μm^2.

However, the size of SC-ASIC circuit is hard to predict from the netlist or the IPO number. We depend on the layout results from Encounter™. Figure 5.3 shows the ratio of PLA area over SC-ASIC area. The Molavi PLA used in this chapter uses minimum-sized devices wherever possible for a smaller area with the trade-off of a slower delay. The designs of the SC-ASIC benchmarks are limited by the timing constraint and minimum-sized gates are not allowed. Therefore, the area ratios are not as large as mentioned in the introductory chapter. The ratio between the full implementation PLA area and SC-ASIC layouts is between 1 to roughly 10, with about 70% lying below 4. After multi-output partitioning and merging product lines, the total area is reduced for most of the benchmark circuits. On average, the ratio decreases from 3.5 times to 2.5. After the reduction, the PLA implementation of a circuit still consumes more silicon area than SC-ASIC implementation, even in the case of using minimum-sized transistors in PLA design.

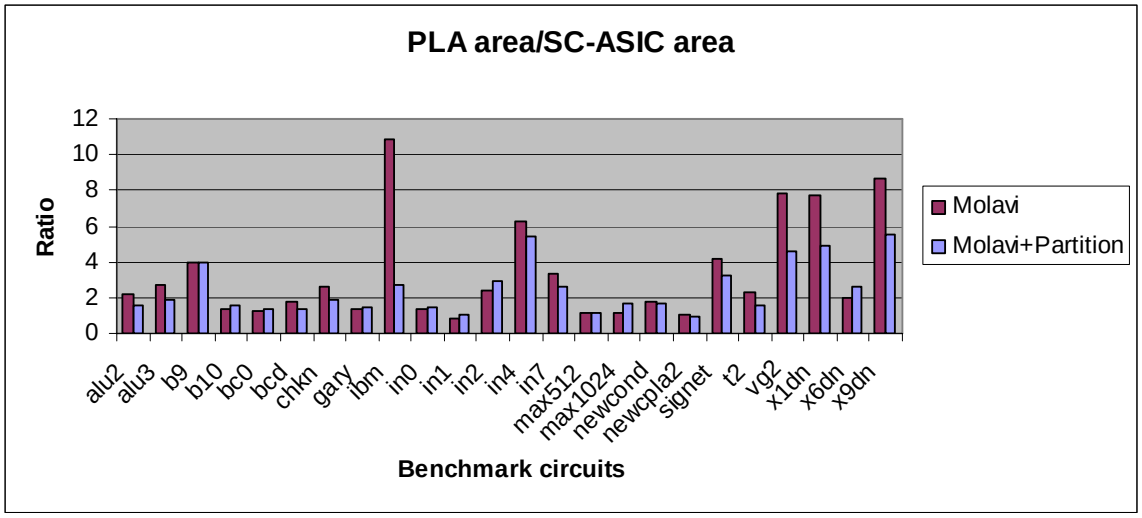


Figure 5.3: PLA vs. SC-ASIC (area)

5.3 Delay, Power and Energy Comparison

While the delay for the PLA is highly-correlated to number of product terms, there is no easy estimation for SC-ASIC circuit in terms of product line numbers. To optimize the critical path delay of the SC-ASIC circuit, the designer needs to reduce the number of slow gates in the path. Figure 5.4 illustrates the ratio of the worst-case delay for PLAs to the worst-case delay for SC-ASIC. The conventional PLA delays are generally larger than SC-ASIC delays. On average, the conventional PLA takes 80% more time than SC-ASIC. Molavi's PLA significantly improves PLA delay such that for more than half of the benchmark circuits, the PLA implementations are *faster* than SC-ASIC. On average, Molavi's PLA is as fast as SC-ASIC. The multi-output partitioning further reduces the PLA delay and, on average, the PLA delays after partitioning and merging product lines are about 66% of the SC-ASIC delays.

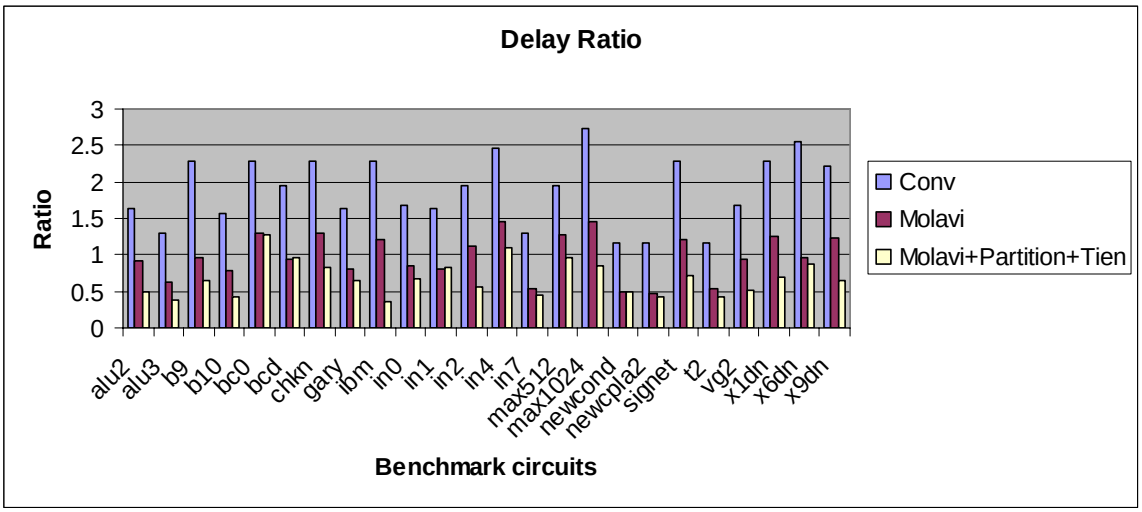


Figure 5.4: PLA vs. SC-ASIC (delay)

The power results are shown in Figure 5.5. Conventional PLA consumes much more power than SC-ASIC, and this is a main reason that SC-ASIC is more widely used than PLA for low power applications. On average, conventional PLA dissipates 6 times SC-ASIC power. Molavi's PLA improves power consumption to 3 times SC-ASIC power on average. After partitioning and super product line merging, PLA consumes this ratio is slightly increased but still around 3 times the SC-ASIC power. The ratio of partitioned and merged PLA power to SC-ASIC power varies from 25% to 12 times and is independent of the number of product lines. There are still two benchmark circuits that have ratios larger than 10 times. Also, for those two, partitioning and product line merging increased the power consumption. Excluding these two, most of the benchmark circuits have ratio less than 4 times, and the averages for both full PLA and partitioned PLA are 2 and 1.75 times, respectively.

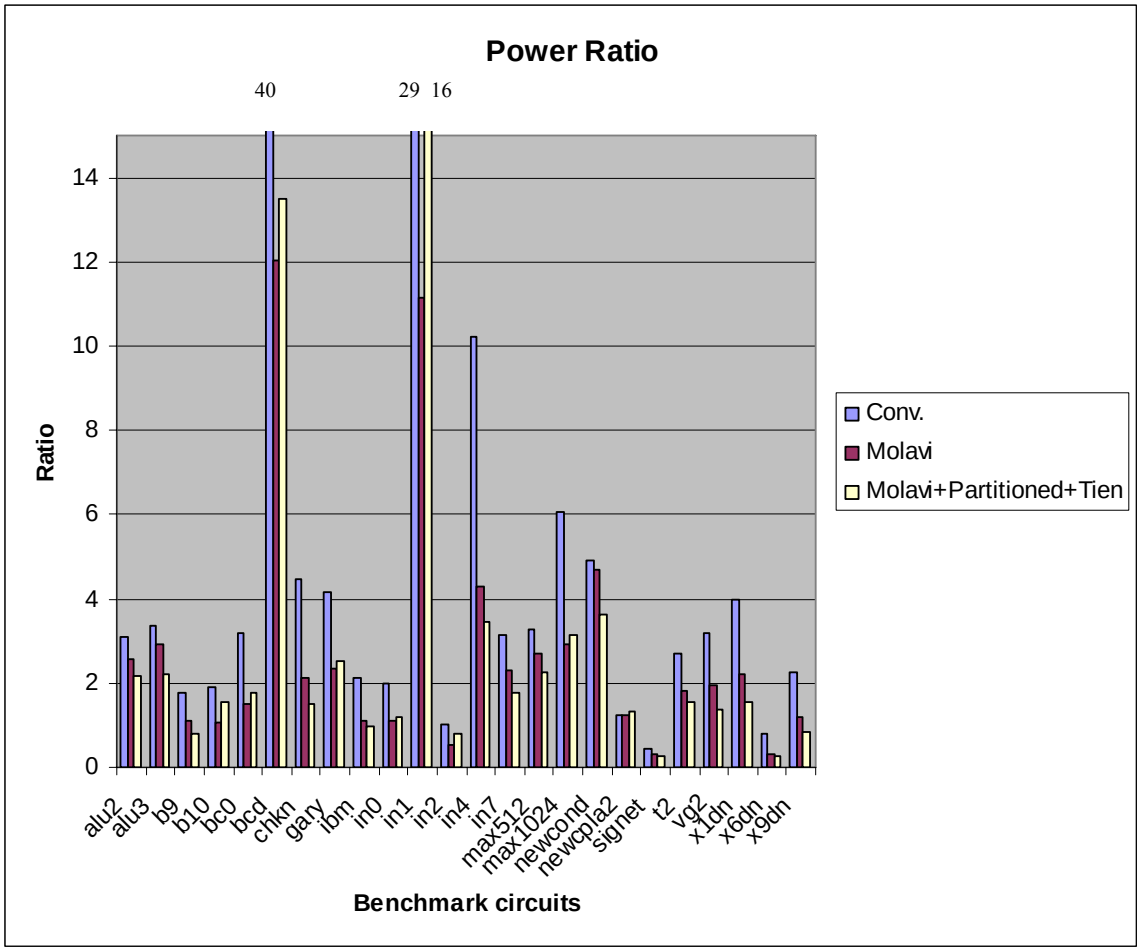


Figure 5.5: PLA vs. SC-ASIC (power)

As shown in Figure 5.6, the energy consumption for the PLA is generally larger than SC-ASIC. The conventional PLA on average consumes more than 10 times energy than SC-ASIC. Molavi design significantly reduces the energy consumption to 2.7 times. The partitioning and Tien's approach further reduce the delay and power such that the energy ratio reduces to 2.3 times.

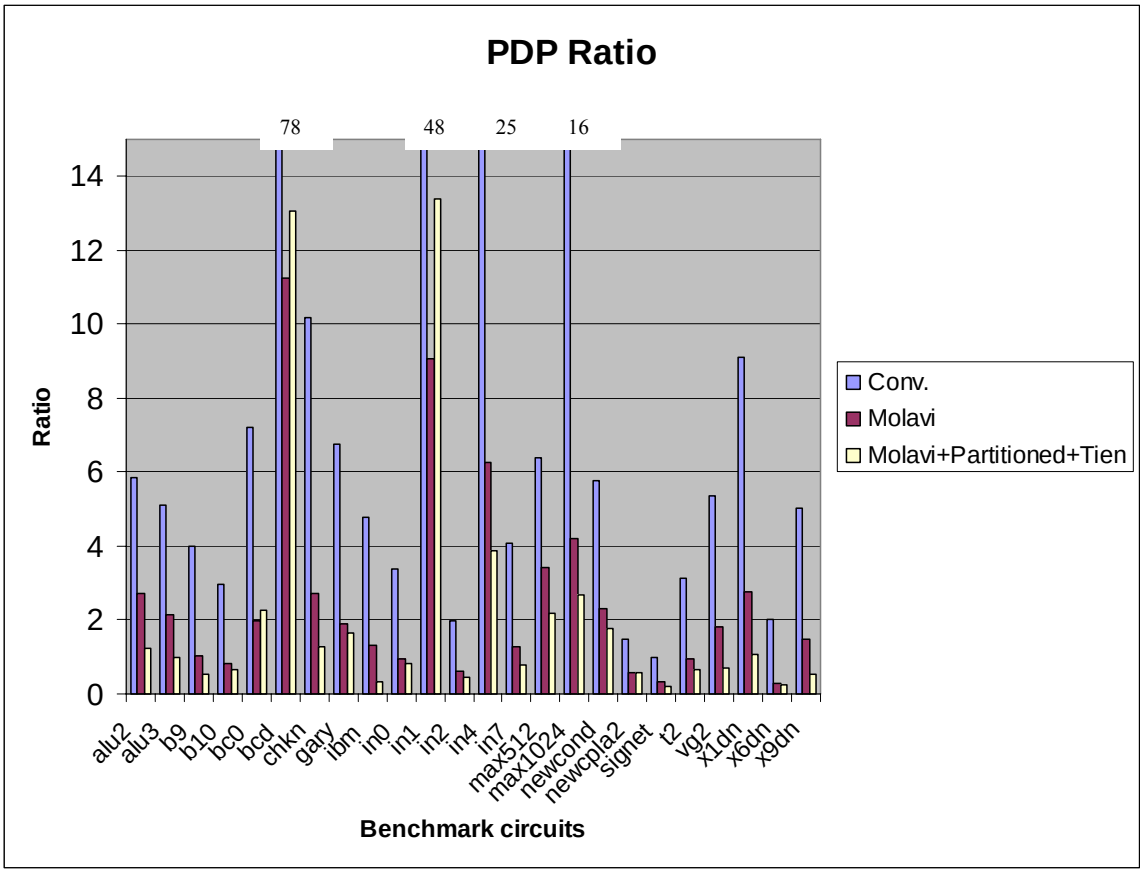


Figure 5.6: PLA vs. SC-ASIC (PDP)

5.4 Summary

In this chapter, we considered SC-ASIC performance of the same benchmark circuits as we used for the previous chapters. The power, delay and energy results for conventional PLA, and Molavi PLA with multi-output partitioning and super product over SC-ASIC design are shown in Table 5.1. As a general conclusion, the PLA in this work is 1X in delay, 2X in energy and 3X in power compared to SC-ASIC.

	Delay over ASIC			Power over ASIC			PDP over ASIC		
	conv	Molavi	M+P+T	conv	Molavi	M+P+T	conv	Molavi	M+P+T
alu2	163%	91%	49%	360%	299%	252%	585%	272%	123%
alu3	130%	62%	38%	394%	343%	260%	512%	214%	100%
b9	195%	96%	66%	228%	96%	66%	399%	104%	53%
b10	156%	78%	43%	190%	108%	154%	296%	84%	66%
bc0	228%	130%	128%	317%	152%	178%	722%	198%	227%
bcd	195%	94%	96%	4010%	1203%	1364%	7820%	1126%	1306%
chkn	228%	130%	83%	448%	210%	152%	1019%	273%	127%
gary	163%	81%	64%	416%	233%	254%	675%	189%	163%
ibm	228%	121%	35%	211%	110%	99%	480%	132%	35%
in0	169%	86%	68%	200%	112%	122%	338%	97%	82%
in1	163%	81%	83%	2935%	1115%	1614%	4769%	906%	1338%
in2	195%	111%	56%	102%	55%	81%	199%	61%	45%
in4	247%	146%	111%	1024%	430%	348%	2530%	627%	386%
in7	130%	55%	45%	314%	232%	176%	408%	127%	80%
max512	195%	127%	96%	328%	269%	226%	640%	341%	218%
max1024	163%	145%	85%	273%	145%	85%	1658%	422%	269%
newcond	117%	49%	49%	493%	468%	365%	577%	230%	179%
newcpla2	117%	47%	44%	125%	125%	132%	147%	59%	57%
signet	228%	121%	71%	44%	29%	28%	101%	35%	20%
t2	117%	54%	42%	268%	179%	155%	313%	97%	65%
vg2	169%	93%	51%	317%	197%	136%	536%	183%	70%
x1dn	228%	125%	70%	400%	220%	156%	910%	275%	109%
x6dn	117%	97%	87%	256%	97%	87%	203%	30%	23%
x9dn	221%	124%	65%	228%	118%	84%	504%	147%	55%
Average	**177%**	**98%**	**68%**	**578%**	**273%**	**274%**	**1098%**	**259%**	**217%**

Table 5.1: Conventional PLA, Molavi PLA, Molavi+Parition+Tien vs. SC-ASIC

Chapter 6 CONCLUSION AND FUTURE DIRECTIONS

As fabrication technology pushes further into the DSM era, structured architectures become more and more favorable to designers and for the manufacturing process. Traditional ASIC and other random design patterns are difficult to fabricate due to process limitations such as photolithographic resolution. FPGAs and structured ASICs offer programmability but incur rather high overheads on their speed, area and power. The PLA is being revisited as a research area with re-newed interest due to its regularity, simplicity and flexibility. To make PLAs more competitive in timing, power and area with SC-ASIC, the work in this thesis addressed methods to further optimize the PLA.

First, we described the operation details for Molavi's PLA and Tien's super product line method. Then, we developed a C++ program to automatically generate HSpice code for simulations over a set of benchmark circuits. We also provided the approach we used to obtain worst-case delay and average power for each PLA. It was observed that Tien's super product line method only depends on the AND-plane transistors and could be used on any dynamic CMOS PLA with different pull-up/down circuitry. A drawback of Tien's approach was the charge-sharing problem and, to solve it, a feedback transistor was needed. Using Tien's approach on Molavi's design has better power improvement than using Tien's approach on conventional design. Furthermore, Molavi's PLA has a feedback loop and does not need extra feedback transistor. The average delay, power and PDP results for different PLA designs over conventional design were generated. A

summary of the results are provided in the first 4 rows of Table 6.1. The findings for the unpartitioned implementation are:

1. Tien's approach applied on Molavi PLA has the best power and PDP improvement.
2. Molavi PLA has the best delay improvement

MCNC	Power improvement %	Delay improvement %	PDP improvement %
Tien+conv.	35%	-5%	32%
Wang	47%	40%	67%
Molavi	40%	49%	69%
Tien+Molavi	54%	43%	74%
Molavi+ Partition+ Tien	44%	63%	79%

Table 6.1: Different optimization approaches vs. conventional PLA design

Next, we considered partitioning of Molavi's PLA. The PLA delay has a close correlation to the number of product terms. Single-output partitioning of the PLA significantly reduces the delay. Also, on average, the partitioned PLA took less area than the full implementation. Even though the power overhead was large for some benchmark circuits, the PDP was improved for most of the benchmark circuits. To further reduce the PDP, we traded off delay improvement for power improvement by considering multi-output partitioning. In order to have fast estimation of power and delay of each sub-PLA, power and delay models were produced and validated with simulation for Molavi's PLA. We also presented a greedy algorithm for small- and medium-sized PLAs and a simulated annealing algorithm for large PLA with more than 30 outputs to perform the multi-output partitioning. Different cost functions were also discussed. For our purpose of minimizing

delay, $Cost(f) = PDP_{part} / PDP_{full}$ is used for both algorithms. Multi-output partitioning significantly reduced area and power consumption compared with single-output partitioning. The delay was increased for some benchmark circuits, but the PDP was decreased on average.

After multi-output partitioning, the Tien's super product line approach was applied to the sub-PLAs. The power improvement of Tien's approach was offset by the overhead of partitioning. The improvement of partitioned Molavi PLA after merging product line was only 4% over full Molavi design. In fact, Molavi with Tien's approach but without partitioning had a better result. However, the delay was significantly reduced by the partitioning, and after partitioning and merging product lines, the average PDP was reduced. The last row of Table 6.1 provides the results of partitioning with Molavi and Tien applied.

Finally, we presented a comparison between the improved PLA and SC-ASIC. A design flow was used to collect area, power and delay information for SC-ASIC implementation. The layout for a representative Molavi PLA was carried out and a simple function was developed based on this layout to estimate area from IPO number. The total area of the partitioned PLA is smaller on average than full implementation. Therefore, the ratio between PLA and SC-ASIC area has been reduced from 3.5 times to 2.5. The delay ratio was also significantly reduced by the partitioning from 1.8 times to 0.7. Even though the power ratio is reduced by half, the power consumption of partitioned PLA after product line merging was still 3 times higher than the SC-ASIC power. Without partitioning and

merging, Molavi PLA delay was about the same as SC-ASIC, but the power consumption is about 3 times higher.

6.1 Contributions

In this work, the tasks carried out were as follows:

- Investigated area, power, delay and energy analysis for several PLA structures and SC-ASIC implementations of some benchmark circuits
- Created a flow to generate HSpice code from PLA definition file for conventional, Tien, Wang and Molavi PLA designs
- Produced layout for Molavi PLA design to extract capacitive and resistive loads
- Presented power, delay and area equations of Molavi PLA design for fast estimation of circuit characteristics from IPO number

The summary of this research contributions are as follows:

1. Developed a new method combining Molavi's delay improvement approach and Tien's super product line approach to produce a PLA with the best PDP of 74% improvement over the conventional PLA.
2. Investigated algorithms to partition PLAs into smaller and faster sub-PLAs and combined partitioning with Tien's super product line approach. Partitioning a circuit and then applying Tien's super product line method improved the power consumption of Molavi's PLA by 5%, delay by 28% and PDP by 31%.

3. Assessed the overall impact of the improved PLA with respect to SC-ASIC. The Molavi design after partitioning and merging product lines, compared with conventional design, reduces PLA over SC-ASIC power ratio from 6X to 3X, delay ratio from 2X to 1X, and PDP ratio from 10 X to 2X.

6.2 Future Work

CAD tools for PLA layout, and power, and delay analysis are easy to develop because of their regularity. For new technologies, PLA designs should be easier to manufacture than pseudo-random design structures, such as SC-ASIC. However, after all the power and delay optimization approaches discussed in this thesis, the PLA still consumes more power and takes more area compared with SC-ASIC. Therefore, studies are still needed on suitable applications where PLAs are more attractive than SC-ASIC.

For example, Built-in-Self-Test (BIST) and Built-in-Self-Repair (BISR) can be used with PLAs to improve the yield of logic circuits. PLAs have a regular structure and it is quite easy to duplicate product lines and output lines for repairing purposes. BIST and BISR for Molavi's PLA (without the Tien approach included) have been developed by Alsaiari in [34]. However, ASIC has fault coverage limitations in terms of self-test, and it is almost impossible to apply self-repair. The PLA circuits are 100% testable while the ASIC fault coverage is limited by the pseudo-random vectors used in these tests. Yet, the area overhead of duplication is 2.2x the area of a basic PLA. Alsaiari claimed in [35] that, even though the PLA solution is expensive in terms of area, it is more suitable for self-

test and self-repair. There may also be opportunities to use PLA structures in post-silicon debugging applications, as described in [13]. Finally, it is anticipated that as technology scales to 32nm and below, the PLA may become a more attractive option.

REFERENCES

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits", *Electronics*, Vol. 38, No. 8, April 19, 1965

[2] M. Lavin, Fook-Luen Hens, and Greg Northrop, "Backend CAD flows for Restrictive Design Rules", *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference*, pp. 739-746, Nov. 2004

[3] J. Rose, R. Francis, D. Lewis, and P. Chow, "Architecture of Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency", *IEEE Journal of Solid-State Circuits*, Vol. 25, pp. 1217-1225, Oct. 1990

[4] B. Zahiri, "Structured ASICs: Opportunities and Challenges", *Proceedings 21st International Conference on Computer Design*, pp. 404-409, Oct. 2003

[5] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980

[6] R. Mehrabadi, "Structured Logic Arrays as an Alternative to Standard Cell ASIC", Master Thesis

[7] C. C. Wang et al, "A Low-power and High-speed Dynamic PLA Circuit Configuration for Single-clock CMOS", *IEEE Transactions on Circuits and Systems*, Vol. 46, No. 7, pp.857-861, July 1999

[8] S. Posluszny, "Design Methodology for a 1.0 GHz Microprocessor", *Proceedings IEEE International Conference on Computer Design*, pp. 17-23, 1998

[9] S. Posluszny, "Timing Closure by Design, a High Frequency Microprocessor Design Methodology," in *Proc, IEEE/ACM Des. Automat. Conf.*, pp. 712-717, 2000

[10] C. C. Wang, C. J. Huang, and K. C. Tsai, "A 1.0GHz 0.6um 8-bit carry lookahead adder using PLA-styled all-N-transistor logic," *IEEE Trans. Circuits Syst., Analog Digit. Signal Process.,* Vol. 47, No. 2, pp. 133-135, Feb. 2000

[11] L.G. Chen, K.W. Wang, J. M. Lai, and J. R. Tong, "A Novel Hybrid FPGA Architecture", *Solid-State and Integrated Circuit Technology, ICSICT apos; 06. 8th International Conference*, pp. 1947 – 1949, 2006

[12] S.P. Khatri, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Cross-talk immune VLSI design using a network of PLAs embedded in a regular layout fabric", *Computer Aided Design, 2000, ICCAD-2000, IEEE/ACM International Conference,* pp. 412-418, Nov. 2000

[13] S.R. Sarangi, A. Tiwari, and J. Torrellas, "Phoenix: Detecting and Recovering from Permanent Processor Design Bugs with Programmable Hardware", *The 39th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 26-37, Dec. 2006

[14] R. Treuer, H. Fujiwara, and V.K. Agarwal, "Implementing a Built-In Self-Test PLA Design", *IEEE Design and Test of Computers*, vol. 2, issue 2, pp. 37-48, April 1985

[15] T. Tien, C. Tsai, S. Chang, and C. Yeh, "Power Minimization for Dynamic PLAs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* Vol. 14, No. 6, pp.616-624 , June 2006

[16] G. I. M. Blair, "PLA design for single-clock CMOS", *IEEE Journal of Solid-State Circuits*, vol. 27, no. 8, pp.1211-1213 , Aug. 1992

[17] O. J. Kwang-Il, L. S. Kim, "A high performance low power dynamic PLA with conditional evaluation scheme", *Proceedings of the 2004 International Symposium on Circuits and Systems*, vol 2, pp. 881-4, May 2004

[18] J.S. Wang, C.R. Change, C. Yeh, "Analysis and Design of High-speed and Low-Power CMOS PLA", *IEEE J. Solid-State Circuits*, vol. 36, pp.1250-1262, Aug. 2001

[19] R. Molavi, S. Mirabbasi, and R. Saleh, "A High-Speed Low-Energy Dynamic PLA Using an Input-Isolation Scheme", *IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006 Proceedings*, pp.2885-2888, May 2006

[20] Microelectronics Center of North Carolina (MCNC) PLA benchmark, 1991

[21] Berkeley PLA Test Set, included under espresso-examples directory with the SIS package, Jan., 1988

[22] HSPICE: http://www.synopsys.com/products/mixedsignal/hspice/hspice.html

[23] W. Wolf, *Modern VLSI Design*, Perntice Hall, pp. 131, PTR 1998

[24] S. Iman, C.Y. Tsui, and M. Pedram, "PLA Minimization for Low Power VLSI Designs", CENG Tech. Rep., 1995.

[25] J. M. Tseng and J. Y. Jou, "Two-level logic minimization for low power," *ACM Trans. Des. Automat. Electon. Syst.*, vol. 4, no. 1, pp.52-69, 1999.

[26] R. L. Rudell, "Multiple-Valued Logic Minimization for PLA Synthesis", EECS Department University of California, Berkeley, Technical Report No. UCB/ERL M86/65, http://www.eecs.berkeley.edu/Pubs/TechRpts/1986/734.html, Jun. 1986

[27] D.A. Hodges, H.G. Jackson, R.A. Saleh, *Analysis and Design of Digital Integrated Circuits: In Deep Submicron Technology,* McGraw Hill, New York, 3rd Ed, 2004

[28] S. Roy, and H. Narayanan, "A New Approach to the Problem of PLA Partitioning Using the Theory of the Principal Lattice of Partitions of a Submodular Function", *Proceedings of the Fourth Annual ASIC Conference and Exhibit*, 1991

[29] P. G. Paulin, "Horizontal Partitioning of PLA-based Finite State Machines", *26th Conference on Design Automation,* pp. 333-338, June 1989

[30] M. J. Ciesielski, and S. Yang, "PLADE: A Two-Stage PLA Decomposition", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 11, pp. 943-954, Aug. 1992

[31] S. Liu, M. Pedram and A. Despain, "PLATO_P: PLA Timing Optimization by Partitioning", *1995 IEEE International Symposium on Circuits and Systems,* Vol. 3, pp. 1744-1747, May 1995

[32] J. Cong, H. Huang, and X. Yuan, "Technology Mapping and Architecture Evalution for *k/m*-Macrocell-Based FPGAs", *ACM Transactions on Design Automation of Electronic Systems*, Vol. 10, No. 1, pp. 3-23, Jan.2005

[33] J. Edmonds and E. L. Johnson, "Matching: A well-solved class of integer linear programs," in C*ombinatorial Structure and Their Applications*. New York: Gordon and Breach, pp. 89-92, 1970

[34] U. Alsaiari, and R. Saleh, "Power, Delay and Yield Analysis of BIST/BISR PLAs Using Column Redundancy", *Proceedins of the 8th International Symposium on Quality Electronic Design*, 2007

[35] J. Jou, "An Effective BIST Design for PLA", *The 14th Asian Test Symposium*, pp. 286, 1995