行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※

※ ※

※ 硬體描述語言低功率架構設計(1/3) ※

※ ※

※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑個別型計畫　□整合型計畫

計畫編號：NSC 90－2213－E－002－108－

執行期間：2001 年 8 月 1 日至 2002 年 7 月 31 日

計畫主持人：賴飛羆 教授

共同主持人：

計畫參與人員：阮聖彰（博士班研究生）

陳柏宏（博士班研究生）

蔡坤霖（博士班研究生）

顏宇宏（碩士班研究生）

蘇浩志（碩士班研究生）

何佳霖（碩士班研究生）

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份

□赴大陸地區出差或研習心得報告一份

☑出席國際學術會議心得報告及發表之論文各一份

□國際合作研究計畫國外研究報告書一份

執行單位：國立台灣大學電機工程系

中華民國 2002 年 9 月 27 日

行政院國家科學委員會專題研究計畫期中報告

硬體描述語言低功率架構設計(1/3)

Low power design in hardware description language

計畫編號：NSC 90-2213-E-002-108-
執行期限：91 年 8 月 1 日至 91 年 7 月 31 日

主持人：賴飛熙　國立台灣大學電機工程系
計畫參與人員：阮聖彰　國立台灣大學電機工程系

一、中文摘要

近年來，能量消耗的問題已經是最重要的設計考量之一，這直接關係到手持式設備操作時間的長短。大多數在管線電路上的低功率研究都著眼於邏輯區塊的最佳化。在所有減少功率消耗的技術上，使用二分割的方法可以比較有效地降低功率消耗，其方法是將一個給予的電路分割成兩塊子電路，並且在一個時間區段內只有其中一個子電路執行運作，因而降低了不需要的訊號傳遞。然而，一個系統的運作時間是與能量消耗有關，並非與功率消耗有關。因此，在本計畫便針對降低 0 與 1 的轉換機率提出一個二分割的演算法來降低組合邏輯電路與管線暫存器的能量消耗情況。

關鍵詞：低功率、二分割演算法、管線電路、

Abstract

Energy consumption has recently emerged as one of the most critical design constraints. It directly relates to the operating time of a portable device. Most researches on pipelined circuits address the optimization of logic blocks to achieve low power. Among the power reduction techniques, the bipartition approach is comparatively effective as it partitions a given circuit into two subcircuits such that only a selected subcircuit is activated at a time, hence reducing unnecessary signal transitions. However operation time of a system is correlated to energy dissipation rather than power dissipation. Thus, this project proposes a bipartition algorithm which aims to reduce switching probabilities such that the energy dissipation of combinational block as well as pipelined registers are reduced.

Keywords: low power, bipartition algorithm, pipeline circuit

二、緣由與目的

隨著可攜式裝置的普及化，如手機、手提式電腦、個人數位助理等等，這些裝置對我們日常生活的影響也越來越大。然而、在不插電的狀況下，如何延長這些裝置的使用時間是非常重要的考量點。更重要的是，在下一代的中央處理器中，由於時脈不斷的提升，高速運算消耗大量功率引發中央處理器過熱的現象，也嚴重影響了電腦系統的穩定性及效能。

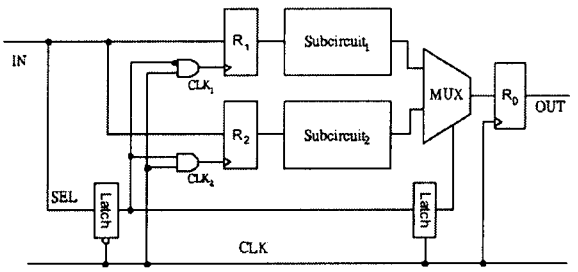
在電路設計方面，管線電路已被廣泛地使用在設計數位系統上，諸如行動電話或個人數位助理等。對於這些產品而言，低能源消耗與產品可靠度、效能或電池使用壽命同等重要。

為了降低管線電路的功率消耗，已有許多研究與方法被提出，其中包括先計算(precomputation)、拴鎖管線暫存器(gated pipelined registers)與電路分割(circuit partition)等。先計算乃是將部分的暫存器失能，而這些暫存器並不會影響到輸出的結果。

本計畫的目的在於提出一個有效控制能量消耗的架構與方法。因此在本年度計畫中，我們針對二分割架構提出一個簡單且有效的演算法，利用 Shannon 展開式對電路在電晶體層次上進行分割。並將之運用於重要的管線電路之上。

三、研究方法與演算法

在本計畫中，我們針對二分割電路做一個詳細的研究。首先，我們將一個管線電路分割成兩塊子電路，如圖一，研究的目的在於使此電路 R_1 與 R_2 的轉換機率減少，並降低兩塊子電路 $Subcircuit_1$ 和 $Subcircuit_2$ 的熵值(Entropy)，熵值與電路之功率消耗及面積有關，因此降低熵值便能有效控制功率消耗情形。事實上，當我們嘗試去減少 R_1 與 R_2 的轉換機率時，熵值會同時下降，因而，我們所需注重的便是如何有效地使每一塊子電路的輸入訊號轉態次數降到最低。由於轉換機率是與輸入資料息息相關的，因此我們便需針對每塊電路資料輸入的順序進行探討。



圖一. 低功率二分割架構

首先，我們考量一個具有 n 個輸入的電路，若我們想要將這塊電路利用 Shannon 展開式進行二分割，那麼將會有 n 種不同的組合。對我們而言想要找出哪一種組合是最好的並不困難，只需利用暴力法(brute force)將每種情況測試一次便可，而這樣的時間複雜度通常是可以被接受的。

說明白一些，我們對於電路中的每一個輸入腳位進行下列兩個動作：

- 利用輸入腳位將原始電路一分為二
- 利用測試輸入次序計算每一塊分割電路的轉換機率總和

一個具有較少轉換機率的結果將會由這兩個簡單的步驟中得到。分割完成的電路便可於模擬中得知其省電結果。

四、成果與討論

本計畫中，二分割演算法利用 C++語言進行實現，並利用 PowerMill 等測試工具針對測試電路進行模擬評估。

模擬的結果如表一所示，與原電路相比，經過二分割演算法的電路在功率與延遲的乘積上，大約可降低百分之二十五點六，若只考量電路的暫存器部分，則可節省約 28.8%左右。然而有些電路並不適合利用二分割演算法進行電路分割，在實驗結果中，我們以負數來表示並沒有任何能量節省。

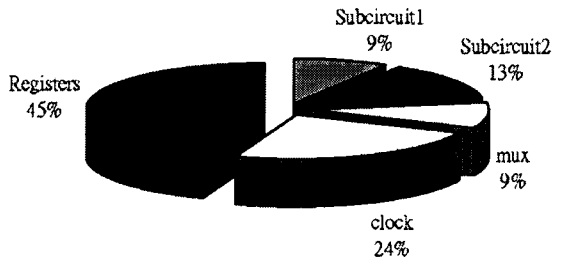
我們將二分割電路平均功率消耗的分佈情形利用圓餅圖顯示在圖二。就如圖上所顯示，管線暫存器依然是管線架構中消耗最多功率的一個部分，約佔了 45%。而時脈(Clock)在電路中代表了兩個拴鎖器(latch)與 AND 閘，這一部份由於具有高頻率的轉換，因此也佔了相當一部份的功率消耗。而主要的子電路約只消耗總功率的 22%，這證明了我們所提出來的二分割演算法能夠有效地降低組合邏輯電路的熵值來達到低功率的目的。

在此年度的計畫中，我們先提出了一個二分割的演算法，並利用選擇輸入腳位來分割原始電路，使二分割架構中的轉換次數降到最低，以求得減少功率消耗的情形。在未來，我們將不止針對功率消耗的情形進行考量，也將針對電路的延遲進行改進。

在計畫進度方面，我們已經將一般電路中最耗電的部分進行討論與改進，這將有助於我們對硬體描述語言低功率架構設計進行更進一步的研究。未來，只需將此一方法與結果運用在硬體描述語言，便可針對硬體描述語言所設計出之電路進行二分割，便能達到省電的效果，這可以幫助使用硬體描述語言的晶片設計者更快速且有效的達到低功率的目的。

表一.原始電路與二分割電路模擬結果比較表

Circuits	Original		Bipartition		
	power	delay	power	delay	ER%
sao2	2867	14.7	2609	12.3	24.2%
5xp1	2885	16.1	2630	15.1	8.8%
bw	2844	26.2	2735	14.8	45.9%
clip	3236	14.3	2943	12.3	21.7%
con1	1362	2.6	1247	3.5	-24.6%
misex1	1936	5.7	1827	6.0	0.3%
rd53	1297	7.9	1485	7.9	-14.0%
rd73	2222	9.7	2197	8.0	18.6%
xor5	1119	5.5	1306	5.6	-17.6%
9sym	4147	9.3	3052	8.6	31.9%
Average	2270	11.2	2099	9.4	25.6%



圖二. 二分割架構平均功率消耗分佈情形

五、成果發表

1. S.-J. Ruan, E. Naroska, C.-L. Ho, and F. Lai, "Power Analysis of Bipartition and Dual-Encoding Architecture for Pipelined Circuits," *in proceeding of IEEE International Conference on Computer Design (ICCD), Sep. 2002.*
2. S.-J. Ruan, E. Naroska, Y.-R. Chang, F. Lai, and U. Schwiegelshohn, "ENPCO: An Entropy-Based Partition-Codec Algorithm to Reduce Power for bipartition-codec architecture in Pipeline Circuits," IEEE Transactions on Very Large Scale Integration Systems.
3. S.-J. Ruan, E. Naroska, Y.-R. Chang, C.-L. Ho, and F. Lai, " Energy analysis of bipartition architecture for pipeline circuits," *in proceeding of IEEE Asia Pacific Conference on Circuits and Systems (ACCAS), Oct. 2002.*

六、參考文獻

[1] S.-J. Ruan, E. Naroska, C.-L. Ho, and F. Lai, "Power Analysis of Bipartition and Dual-Encoding Architecture for Pipelined Circuits," *in proceeding of IEEE International Conference on Computer Design (ICCD), Sep. 2002.*

[2] W. Ye, and M. J. Irwin, "Power Analysis of Gated Pipeline Registers," *in Proceeding of Twelfth Annual IEEE International ASIC/SOC Conference, pp. 281-285, 1999.*

[3] S.-J. Ruan, E. Naroska, Y.-R. Chang, C.-L. Ho, and F. Lai, " Energy analysis of bipartition architecture for pipeline circuits," *in proceeding of IEEE Asia Pacific Conference on Circuits and Systems (ACCAS), Oct. 2002.*

[4] S.-J. Ruan, J.-C. Lin, P.-H. Chen, F. Lai, K.-L. Tsai, and C.-W. Yu, "An Effective Output-Oriented Algorithm for Low Power Multipartiton Architecture," *in IEEE Int'l. Conf. on Electronics, Circuits and Systems, pp. 609-612, 2000.*

[5] S.-J. Ruan, E. Naroska, Y.-R. Chang, F. Lai, and U. Schwiegelshohn, "ENPCO: An Entropy-Based Partition-Codec Algorithm to Reduce Power for bipartition-codec architecture in Pipeline Circuits," IEEE Transactions on Very Large Scale Integration Systems

[6] G. Yeap, *Practical Low Power Digital VLSI Design.* Kluwer Academic, 1998.

[7] M. Munch, B. Wurth, R. Mehra, J. Sproch, and N. When, "Automating RT-Level Operand Isolation to Minimize Power Consumption in Datapaths," *in Proceeding of Design, Automation and Test in Europe Conference and Exhibition, pp. 624-631, 2000.*

[8] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design.* New York:

John Wiley, 2000.

[9] H. Kapadia, L. Benini, and G. D. Micheli, "Reducing Switching Activity on Datapath Buses with Control-Signal Gating," *IEEE J. Solid-State Circuits, vol. 34, pp. 405-414, March 1999.*

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

年　月　日

報告人姓名	賴飛麟	服務機構及職稱	國立台灣大學 教授
會議時間地點	January 2002 Melbourne, Australia	本會核定補助文號	NSC 90-2213-E-002-108
會議名稱	(中文) 第7屆亞太計算機系統結構會議 (英文) Seventh Asia-Pacific Computer Systems Architecture Conference		
發表論文題目	(中文) 衛守：低功率快取記憶体之有效過濾機制 (英文) Sentry Tag: An Efficient Filter Scheme for Low Power Caches		

報告內容應包括下列各項：

一、參加會議經過

二、與會心得

三、考察參觀活動（無是項活動者省略）

四、建議

五、攜回資料名稱及內容

六、其他

一、參加會議經過

這個會議結合其它領域的會議在同一地點召開，幾乎涵蓋所有資訊相關之研究，例如：人机介面，理論及演算法多媒体、資料庫等。會議假Monash大學校園內舉辦，自然也吸引很多學生前來參加，增添不少熱鬧氣氛。每篇論文有30分鐘報告的時間，講者能夠清楚介紹其研究內容，也有机會回答聽眾之問題，作完分之溝通。於表論文者來自台灣、新加坡、日本、澳洲、德國、紐西蘭、英國、比利時。美國可能受到911事件之影響，並無參與，可見其對世人，尤其是美國人之衝擊。

二、與會心得

群組會議之安排，讓參與之研發人員也有机會聽聽其它領域之報告，無形中增加學習之机會又不用付額外的費用，這種感覺真好。聽了日本來的報告，覺得他們系統建置的環境非常完整，有了一些新的想法後，

不但能夠用模擬的方式來做驗証，也可以用IC實作來證實其理論，更加具有說服力。所謂一個國家工業實力之深厚，我想指的就是這樣子吧！

三、考察參觀活動

無

四、建議：

要把台灣建設為亞太中樞，除了交通便捷等有形的海空、無形的網路外，最重要的是內容，也就是說經濟的實力、工業之競爭力、環環相扣，以學校所扮演的角色而言，提昇研發的質與量、培養優秀的人才、又富創新力的學生，是我們責無旁貸，努力爭取國際會議承辦最大的好處，是眾多的學生、或業界之工程師，可以最少的代價，

未參與、學習、建議政府無論如何要在這方面大力投資，不能因為預算短缺，就失次這個項目。

五、提回資料名稱及內容。

會議論文集一冊。

ENERGY ANALYSIS OF BIPARTITION ARCHITECTURE FOR PIPELINED CIRCUITS

Shanq-Jang Ruan, Edwin Naroska, Yen-Ren Chang, Chia-Lin Ho and Feipei Lai*

Dept. of EE and Dept. of CSIE
National Taiwan Univeristy
Taipei 106, Taiwan
Phone:+886-2-23914116, Fax:886-2-23637204,
flai@cc.ee.ntu.edu.tw

Computer Engineering Institute
University of Dortmund
Dortmund 44221, Germany
Phone:+49-231-7552406, Fax:+49-231-7553251,
edwin@ds.e-technik.uni-dortmund.de

ABSTRACT

Energy consumption has recently emerged as one of the most critical design constraints. It directly relates to the operating time of a portable device. Most researches on pipelined circuits address the optimization of logic blocks to achieve low power. Among the power reduction techniques, the bipartition approach is comparatively effective as it partitions a given circuit into two subcircuits such that only a selected subcircuit is activated at a time, hence reducing unnecessary signal transitions. However operation time of a system is correlated to energy dissipation rather than power dissipation. In this paper, we propose a bipartition algorithm which aims to reduce switching probabilities such that the energy dissipation of combinational block as well as pipelined registers are reduced. Transistor-level simulation results show that our proposed algorithm reduce not only the power dissipation but also delay for most of the benchmark circuits.

1. INTRODUCTION

Pipelined circuits have been widely utilized in designing unpluged digital systems, e.g.cellular phones, laptops or PDAs. For these devices low energy consumption is crucial to meet the design requirements in reliability, performance and battery life. As a matter of course, the lead circuit designers have to cope with excessive power dissipation, which in no doubt urges for synthesis tools specifically designed for energy efficient circuits.

Towards minimizing the power dissipation of pipelined circuits, a lot of works have been done including the methods in precomputation [1], gated pipelined registers [2, 3], guarded evaluation [4, 5] and circuit partition [6, 7, 8]. Precomputation disables the parts of the pipelined registers which do not affect the output value [1]. The authors of [2, 3] successfully reduce the switching activities of the pipelined registers by using control signals to gate the clock. In [4, 5], the authors isolate the operands which result in redundant operations to save unnecessary power dissipation. Chen *et al.* separate the original circuit into two parts: a small part (with respect to area) that is active most of the time and a bigger part which is infrequently activated. Because of the separation, power is saved as the switching activity is confined to the smaller part most of the time. In [7], Choi and Hwang partition the combinational logic block of a pipelined circuit into multiple subcircuits through a recursive application of Shannon expansion with respect to the selected input variables. Furthermore, Ruan shows that partitioning circuits into more than two subcircuits often fails in saving power due to the overhead introduced by the additional circuitry [8].

In this paper, we propose a bipartition algorithm to reduce the switching probability of two subcircuits by using Shannon expansion to select a partition variable that results in minimal switching probability of the circuit. In addition, bipartitioning also reduces length of the critical path. This implies that we can extend the operation time of an unpluged system due to the reduced power-delay product. To validate the results, we employ an accurate transistor-level power estimator[1] to obtain energy dissipation for bipartition algorithm.

Comparison to other partition techniques

As stated in the introduction, there are several other partition techniques designed for lowering power consumption, but not for energy dissipation[2]. In [6], cir-

The author was sponsored in part by Synopsys Inc.
This work was sponsored in part by the National Science Council of Taiwan under Grant NSC 90-2213-E002-108

[1] EPIC PowerMill was developed by EPIC Design Technology, INC.

[2] Energy dissipation is a measure of power-delay product.

cuits are bipartitioned based upon profile information. The approach detects the frequent output pattern and synthesizes them into a small subcircuit. However, for selecting the active subcircuit additional control circuitry is required, which increases time delay. [7] and [8] partition circuits using Shannon expansion. As the circuit architectures in [7, 8] do not contain the latches and AND gates shown in Fig. 3 the circuits may suffer from glitches that can cause incorrect behaviors. This unfortunately can not be detected while running gate-level simulation. The authors of [7] and [8] estimated power/energy dissipation in gate-level is less accurate than that in transistor-level let alone the power dissipation of each component of the partition architecture is still unclear.

In contrast to the other approaches, we provide the following viewpoints:

- we bipartite circuit in terms of reducing switching probability instead of area;
- we adopt accurate transistor-level simulation. This ensures the correctness of circuit behavior and energy dissipation; and
- the power dissipation of each component in the bipartition architecture is disclosed and analyzed.

The paper is organized as follows. In Section 2, we provide some basic concepts and briefly introduce our bipartition architecture, while Section 3 explains the bipartition algorithm. Section 4 describes the experimental results that confirm the effectiveness of the proposed method. The conclusion of the paper is summarized in Section 5.

2. PRELIMINARIES

2.1. Power Distribution

In this paper, we adopt a single-phase edge-triggered pipelined circuit shown in Fig. 1 that is used in a large number of ASIC applications. Pipelined circuits have no state feedback lines because they do not need information about the past cycles for running the present cycle. In order to compare energy dissipation results obtained by bipartition method, we begin our the power analysis with the original benchmark circuits. We implement the circuits presented in Fig. 2 using the primitive standard cells provided by CCL[3] (0.8μm technology). Power dissipation were estimated using EPIC PowerMill. Fig. 2 shows that the pipelined register

[3]CCL stands for Computer and Communication Research Labs and is one of the members of Industrial Technology Research Institute in Taiwan.

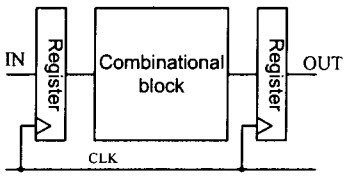


Figure 1: A combinational pipelined circuit

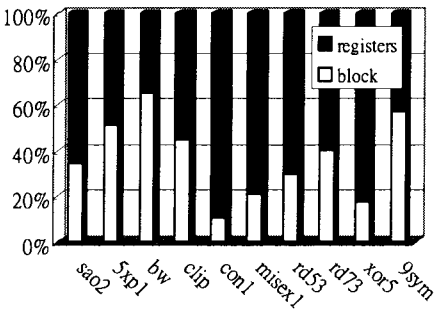


Figure 2: Power dissipation for several MCNC benchmarks

takes a large fraction of total power dissipation in most circuits. On average they account for 57.5% of the total power budget. Obviously, the registers are almost the largest contributor to the power budget of a pipelined circuit. We can say that pipelined registers should be taken into account when optimizing for low power.

2.2. Power Estimation Using Entropy

Consider a combinational logic circuit with m-bit input X and n-bit output Y. The power consumption can be obtained as follow [9]:

$$Power \propto \frac{2^{n+1}}{3n(m+n)} H(Y)[H(X) + H(Y)], \quad (1)$$

The entropy measures $H(X)$ and $H(Y)$ are typically obtained by monitoring the signals X and Y during high-level simulation of the circuit. In general, the outputs of a logic gate make fewer transitions than its inputs. It can be theoretically shown that $H(Y) \leq H(X)$. The entropy is defined as follows:

$$H(x) = \sum_{i=1}^{2^k} p_i \cdot \log_2 \frac{1}{p_i}, \quad (2)$$

where x is a discrete variable which can take k different values, while p_i is the probability that x takes the i-th

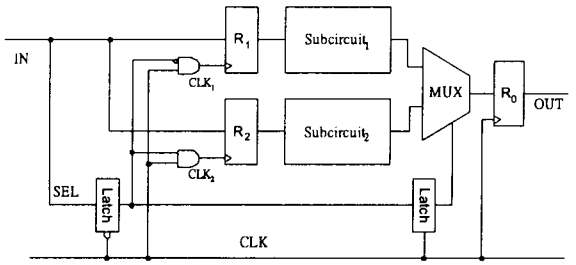


Figure 3: Bipartition architecture for low power.

value x_i. It should be noted that entropy is intuitively related to switching activity [10].

2.3. Bipartition Architecture

In this paper, our bipartition architecture is based on Shannon expansion. In Fig. 3, an input variable SEL has been selected and the original combinational block is transformed into two co-factor subcircuits by applying Shannon expansion. Power reduction is achieved by disabling one of the two subcircuits during operation. Note that the choice of SEL is critical to this approach.

Example: Considering the three-input, two-output function $\mathbf{f} = [f_1 f_2]$, where $f_1 = ab + ab'$ and $f_2 = ab + bc + ac$. The function is represented by the following truth table:

abc	$f_1 f_2$
000	00
001	10
010	00
011	11
100	00
101	01
110	11
111	11

Let us assume that the inputs cycle thorough $000 \rightarrow 111 \rightarrow 000\ldots$ most of the time. As a result, all input register bits will flip nearly at each clock cycle.

If we select a as SEL signal, the table will be partitioned as follows:

$a=0$		$a=1$	
bc	$f_1 f_2$	bc	$f_1 f_2$
00	00	00	00
01	10	01	01
10	00	10	11
11	11	11	11

Where the output function are $f_1 = c$ and $f_2 = bc$ when $a = 1$, and $f_1 = b$ and $f_2 = b + c$ when $a = 0$. Because register R_1 is activated for $SEL = 0$ and R_2 for $SEL = 1$, input pattern 000 will be stored in R_1 while pattern 111 is routed to R_2. In case of a input sequence that alternates between 000 and 111 both register will stop toggeling.

3. PROPOSED METHOD

3.1. Problem Formulation

As previously discussed in Section 1, we want to biparition a pipelined circuit such that the switching probabilities of R_1, R_2 are low and so are the entropies of $Subcircuit_1$ and $Subcircuit_2$. In fact, if we reduce the switching probilities of R_1 and R_2, we also obtain low entropy of $Subcircuit_1$ and $Subcircuit_2$. Therefore, the bipartition problem can be simplified down to a task of finding an input variable such that the total input switching probability of both partitions is minimal. As the switching probability is data dependent we need a input pattern sequence which is typically applied to the circuit. E.g., this test sequence can be obtained during simulation.

3.2. Energy-Efficient Bipartition Algorithm

Consider a circuit with n input pins. If we want to bipartition it by using Shannon expansion, there will be n different combinations. In order to find the best solution we test each of these combinations. As there are only n combinations the runtime for this brute force approach is usually acceptable.

In detail, we perform the following operations for each input pin:

- Bipartition the original circuit by the input pin.
- Calculate the total switching probabilities of the two partitions based on the test input sequence.

A solution with minimal switching probability will be selected as the final result.

4. EXPERIMENTAL RESULTS

The bipartition algorithm has been implemented in C++ on a Pentium III desktop PC. We used SIS[4] to synthesize our partition results and estimate power dissipation by PowerMill. The benchmark circuits with input test patterns taken from LGSynth91 were used to demonstrate our algorithm. In the experiment, 5v supply voltage and a clock frequency of 20MHz were

[4]SIS is a sequential circuit synthesis systems which was developed by U.C. Berkeley.

Table 1: Power dissipations of the registers of original circuit and bipartition architectures

Circuits	Orig_Reg	Bipar_Reg	PR%
sao2	1884	1557	17.4%
5xp1	1411	1113	28.2%
bw	991	669	32.5%
clip	1781	1344	24.5%
con1	1214	756	37.7%
misex1	1526	898	41.2%
rd53	910	673	26.0%
rd73	1330	991	25.5%
xor5	924	625	32.3%
9sym	1779	1485	16.5%
Average			28.8%

Table 2: Energy consumption of original circuit and bipartition architectures

Circuits	Original		Bipartition		
	power	delay	power	delay	ER%
sao2	2867	14.7	2609	12.3	24.2%
5xp1	2885	16.1	2630	15.1	8.8%
bw	2844	26.2	2735	14.8	45.9%
clip	3236	14.3	2943	12.3	21.7%
con1	1362	2.6	1247	3.5	-24.6%
misex1	1936	5.7	1827	6.0	0.3%
rd53	1297	7.9	1485	7.9	-14.0%
rd73	2222	9.7	2197	8.0	18.6%
xor5	1119	5.5	1306	5.6	-17.6%
9sym	4147	9.3	3052	8.6	31.9%
Average	2270	11.2	2099	9.4	25.6%

assumed. The rugged script of SIS was used to optimize the benchmarks. The registers of the output part are unchanged in our architectures so that we do not consider the effect of these registers. The delay, area and power units are nS, $128\mu m^2$ and μW in our experiments, respectively.

Table 1 shows the average pipelined registers power reduction is 28.8%. These results are consistent with earlier discussion which shows the proposed algorithm reduces switching activity. Table 2 shows the power and delay for the original circuits as well as the bipartitioned circuits. The **ER%** column represents the reduction of power-delay product. As shown in Table 2, performing bipartition reduces not only power but aslo delay time of the circuit. On average, energy (power-delay product) reduction of 25.6% could be obtained. Note that some circuits are not suitable for bipartition architecture and have no energy reduction (negative reduction rate).

In Fig. 4, we demonstrate the average power distribution of bipartition architecture. As shown in Fig. 4, the pipelined registers are still the largest power contributors in a pipeline stage even though their contribution decreased from 57.5% to 45%. "Clock" represents two latches and AND gates, which also consumes a large amount of power in a bipartition architecture due to the frequent switching. As shown in the figure the subcircuits only consume 22% of the total power, which proves that our bipartition algorithm can effectively reduce entropy to obtain low power in combinational blocks.

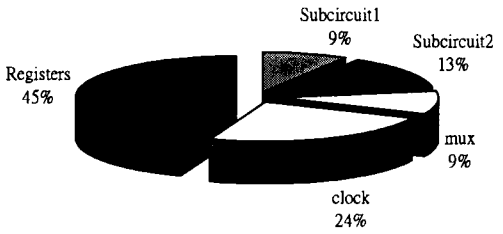


Figure 4: Average power dissipation of bipartition architecture

5. CONCLUSION

We have proposed a simple bipartition algorithm that selects an input variable to bipartite the original circuit which results in minimal switching probability in bipartition architecture in order to reduce power consumption. Further, we reduced not only the power of pipelined registers but also the delay of the circuit.

This paper also analyzes power distribution, delay and energy before and after applying our partitioning approach.

REFERENCES

[1] M. Alidina, J. Monterio, S. Devadas, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-Based Sequential Logic Optimization for Low Power," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 426–436, Dec. 1994.

[2] W. Ye and M. J. Irwin, "Power Analysis of Gated Pipeline Registers," in *Proceeding of Twelfth Annual IEEE International ASIC/SOC Conference*, pp. 281–285, 1999.

[3] H. Kapadia, L. Benini, and G. D. Micheli, "Reducing Switching Activity on Datapath Buses with Control-Signal Gating," *IEEE J. Solid-State Circuits*, vol. 34, pp. 405–414, March 1999.

[4] V. Tiwari, S. Malik, and P. Ashar, "Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1051–1060, Oct. 1998.

[5] M. Munch, B. Wurth, R.Mehra, J. Sproch, and N. Wehn, "Automating RT-Level Operand Isolatoin to Minimize Power Consumption in Datapaths," in *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pp. 624–631, 2000.

[6] S.-J. Chen, R.-J. Shang, X.-J. Huang, S.-J. Ruan, and F. Lai, "Bipartition and Synthesis in Low Power Pipelined Circuits," *IEICE Trans. Fundamentals of Electronics, Communication and Computer Scientce*, vol. E81-A, pp. 664–671, Apr. 1998.

[7] I.-S. Choi and S.-Y. Hwang, "Circuit Partition Algorithm for Low-Power Design Under Area Constraint Using Simulated Annealing," *IEE Proc. Circuit Devices Syst.*, vol. 146, pp. 8–15, Feb. 1999.

[8] S.-J. Ruan, J.-C. Lin, P.-H. Chen, F. Lai, K.-L. Tsai, and C.-W. Yu, "An Effective Output-Oriented Algorithm for Low Power Multipartition Architecture," in *IEEE Int'l. Conf. on Electronics, Circuits and Systems*, pp. 609–612, 2000.

[9] G. Yeap, *Practical Low Power Digital VLSI Design*. Kluwer Academic, 1998.

[10] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design*. New York: John Wiley, 2000.

ENPCO: An Entropy-Based Partition-Codec Algorithm to Reduce Power for bipartition-codec architecture in Pipelined Circuits

Shanq-Jang Ruan Edwin Naroska Yen-Ren Chang
Feipei Lai Uwe Schwiegelshohn

***Abstract*—This paper proposes a new algorithm to synthesize low power bipartition-codec architecture for pipelined circuits. The bipartition-codec architecture has been introduced as an effective power reduction technique for circuit design. The entropy-based partition-codec (ENPCO) algorithm extends this approach as it optimizes for both: power and area. It uses entropy as a criterion to balance between power and area. The ENPCO algorithm is composed of two phases: first, it clusters the output vectors with high occurrence into a group, moving all remaining output vectors into another group. The first group will be encoded in order to save power. Secondly, based on circuit entropy, output patterns are moved between both groups in order to balance power consumption and area overhead. A number of Microelectronic Center of North Carolina (MCNC) benchmarks were used to verify the effectiveness of our algorithm. Results demonstrate that ENPCO algorithm can achieve low power with less area overhead than the single-phase algorithm introduced in [1].**

I. INTRODUCTION

In modern VLSI chip designs, power dissipation has become one of the major concerns (besides area and speed) due to the widespread demand for high performance computing in portable systems. Because of its high throughput, pipelining is popularly used in such systems. Unfortunately, the registers that are inserted into the circuit to separate pipeline stages are a major source of power consumption [2]. Therefore, increasing system clock frequency without sacrificing low power properties requires advanced pipelined synthesis algorithms.

In today's VLSI circuits, the dominant fraction of average power dissipation is attributed to dynamic power dissipation caused by the switching activity of gate outputs [3]. As a result, many power optimization techniques that try to reduce switching activities to the minimum at various design levels have been proposed in recent years ([4], [5], [3] are good surveys to previous research efforts).

For logic level designs, two major low power techniques, *precomputation* and *gated-clock* are often applied. Alidina, *et al.* first proposed a *precomputation-based* scheme, which selectively disables the inputs of a sequential logic circuit to achieve low power consumption [6]. Another precomputation scheme is circuit partitioning. The idea of partition for low power is that in behavioral descriptions of hardware, a small set of computation often accounts for most of the computational complexity as well as power dissipation. The authors of [7] extract CCCs (Common-Case Computation) during the design process and simplify the CCCs in a stand alone manner to achieve power saving. In 1999, Ruan, *et al.* proposed a bipartition-codec architecture which treats each output value of a combinational circuit as one state of a FSM, so that the most transitive states will be extracted to build a small sub-circuit [1]. Additionally, Choi and Hwang partition a combinational circuit into multiple sub-circuits through the recursive application of Shannon expansion with respect to the selected input variables [8]. While these approaches could successfully reduce power consumption they also incur some area overhead. Moreover the relationship between the number of partitions and power/area is still unclear. Ruan, *et al.* showed that partitioning circuits with more than two-ways often does not save power due to the overhead of duplicated input latches and output multiplexors [9]. Benini *et al.* proposed a *gated-clock* approach to build low power FSMs [10]. The approach eliminates unnecessary glitches in the idle states of FSMs. In [11] and [12] a FSM is decomposed into a number of coupled sub-machines so that most of the time state transitions of high probability will be confined to smaller sub-machines.

Additionally, power estimation plays a significant role in low power design as it helps rating different design alternatives with respect to power consumption. Power estimation is especially important at higher level of abstraction in order to provide synthesis algorithms with information that can be used to optimize for low power. Various approaches for power estimation have been developed. Taking entropy-based estimation as example, it is an interesting solution to the problem of how to predict the area and power dissipated by a digital system for which an input/output description is available [13], [14], [15], [16].

In this paper, we propose an algorithm called *entropy-based partition-codec algorithm* (ENPCO) which optimizes pipelined circuits for low power while incurring only small area overhead in most cases. Power saving is achieved by generating a bipartition-codec architecture from the original circuit. The ENPCO algorithm includes two phases: given a circuit described by PLA format, we first bipartition the PLA into two sub-PLAs. The first sub-PLA only includes patterns that frequently occur on the outputs (note that we assume that each *input* pattern have equal probability) while the other PLA is built from the remaining patterns. In the second phase, encoding is applied on the highly active sub-PLA and entropy is used to estimate the area of the entire architecture in order to fine tune both sub-PLAs (i.e., patterns are moved from one sub-PLA to the other in order to decrease area overhead). ENPCO algorithm decreases are overhead from 45.3% to 31.8% compared to the single-phase algorithm proposed in [17] while achieving almost the same power saving effects.

We illustrate our synthesis flow in Fig. 1. $Bipartition$ is the first phase of ENPCO algorithm while $FineTune$ and the remaining processes belong to second phase. In order to verify the results, our synthesis flow from the PLA specification to the transistor-level implementation and an accurate switch-level power estimation tool, EPIC powermill [1], is used to estimate power consumption. The experimental results prove that ENPCO algorithm can save power consumption for most circuits of the MCNC benchmarks.

The major difference between bipartition-codec and precomputation architectures is that precomputation only disables some of the input pins to reduce switching activity of the combinational logic. However, the remaining input signals may also incur redundant switching activity in the entire combinational logic. Furthermore, precomputation does not account for the power dissipation of pipeline registers. Conversely, bipartition-codec architecture separates the combinational logic which ensures they will not influence each other. Moreover we take power dissipation of pipeline registers into account by applying the codec structure. This helps reducing power dissipation significantly.

The paper is organized as follows. Section 2 presents the basic models of power dissipation and area complexity of CMOS circuits. Section 3 describes the bipartition-codec architecture which is proposed in [1]. In section 4, we formulate the problem and propose the ENPCO algorithm. Section 5 demonstrates the experimental results to verify

Shanq-Jang Ruan is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (email: stj@orchid.ee.ntu.edu.tw)

Edwin Naroska and Uwe Schwiegelshohn are with the Computer Institute Engineering, University of Dortmund, Dortmund 44221, Germany (email: {edwin|uwe}@ds.e-technik.uni-dortmund.de)

Yen-Ren Chang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (email: ychang@bulls.csie.ntu.edu.tw)

Feipei Lai is with the Department of Electrical Engineering and Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (email: flai@cc.ee.ntu.edu.tw)

[1] EPIC Powermill was developed by DPIC Design Technology, INC.

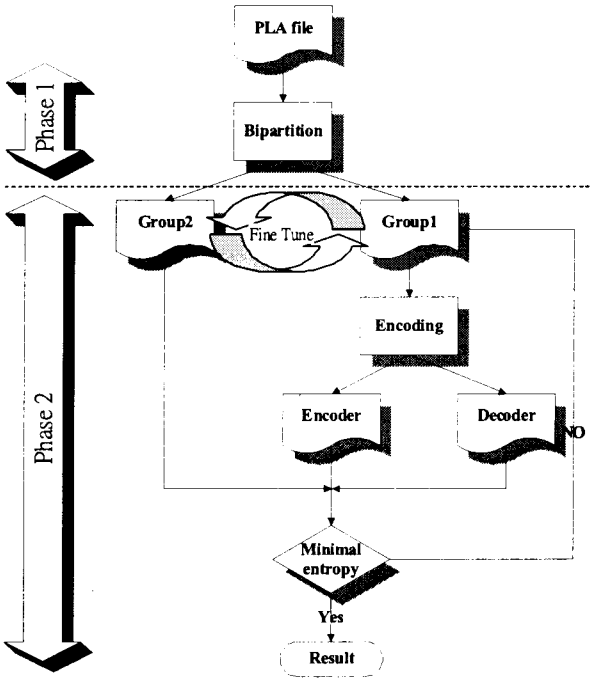


Fig. 1. Synthesis Flow

and prove the feasibility of our algorithm. We conclude in section 6.

II. Preliminaries

In this section, we first describe the power dissipation model of CMOS circuits [4]. This leads to a relationship between power consumption and switching activity. Moreover, we describe an area estimation model based upon the input/output entropies [16].

A. Power Dissipation Model

In a digital CMOS circuit, the major source of power dissipation is switching activity so that the power consumption in digital CMOS circuits can be simply written as follows [4]:

$$P = 0.5 \times C_L \times V_{dd}^2 \times E(sw) \times f_{clk}, \tag{1}$$

where C_L is the loading capacitance, V_{dd} denotes the supply voltage, and f_{clk} is the clock frequency. These three parameters are primarily determined by the fabrication technology and circuit layout considerations such as transistor sizing. $E(sw)$ is the average transition number per clock cycle (referred to as the transition density), which can be determined by evaluating the logic function and the statistical properties of the input vectors. Obviously, equation (1) relates the dominant power dissipation to the switching activity of a CMOS circuit.

B. Area Estimation Model

For efficiency consideration, we adopt the area complexity model proposed by Cheng *et al.* [16]:

$$L(n, d, X) = (1 - d) \cdot k \cdot 2^n \cdot H(X), \tag{2}$$

where n is the number of input pins, d is the fraction of *don't care* terms in the truth table of the circuit, k is constant, and $H(X)$ is the entropy of the outputs. Let P be a set of probabilities $P = \{p(s_1), p(s_2), \cdots, p(s_{2^m})\}$ where $p(s_i)$ is a *probability function* $p : S \longrightarrow \{0..1\}$ which returns the frequency of an output pattern s_i. I.e.,

$$p(s_i) = \frac{x_i}{2^n},$$

where x_i is the number of appearances of pattern s_i at the outputs of the circuit if all possible 2^n input patterns are assigned to the circuit exactly once (note that some x_i may be 0). Then, the entropy $H(X)$ is defined as follows:

$$H(X) = \sum_{i=1}^{2^m} p_i \cdot \log_2 \frac{1}{p_i}, \tag{3}$$

where m is the number of output pins of the circuit. Note that entropy is intuitively related to switching activity. For example, a n-input signal that rarely toggles suggests that the word-level values are relatively stagnant and many values will probably not appear. This skew occurrence probability gives a low entropy measure. Conversely, if signals are highly active, all word-line values are very likely to appear with the same probability. This maximizes entropy of the signals [18].

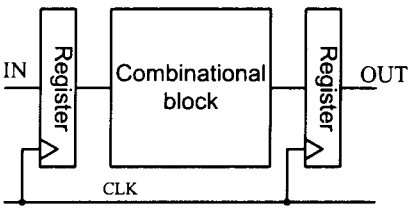


Fig. 2. A combinational pipelined circuit

III. Bipartition-Codec Architecture

A pipeline stage can be represented by a combinational logic block separated by distinct registers as shown in Fig. 2.

Fig. 3 is a bipartition architecture without codec, where GCB selects a group to be enabled. Only one group can be active in a cycle. After bipartitioning a circuit, we apply the encoding technique introduced in [19] to the highly active sub-circuit ($Group_1$). To reduce power consumption of the registers of the highly active sub-circuit ($Group_1$), we replace it with a codec structure that consists of an encoder and a decoder (see Fig. 4). Note that the encoder not only encodes the outputs with minimal Hamming distance but also generates a SEL signal to choose the data path to be activated. In detail, the bipartition-codec architecture operates as follows. The input IN feeds into the Encoder and the registers of $Group_2$. Signal SEL and the encoding outputs become valid before the rising edge of the global clock CLK. If SEL=1, the gated clock CLK_1 will be activated and CLK_2 be stopped when the next rising edge of CLK arrives. At this moment the encoding output passes through register R_1 and propagates into the decoder. Note that the input IN will not propagate through R_2. Hence the Decoder is selected to compute the outputs while $Group_2$ is idle.

Note that the presence of the low-enabled latch is needed for a correct behavior, because SEL may have glitches that must not propagate to the AND gate when CLK is high. The high-enabled latch memorizes the SEL function during a period of the global clock so that the multiplexer can select a correct output. The two latches work as a master-slave flip-flop. The interested reader may refer to [1] for further analysis of the architecture.

Note that the performance of the bipartition-codec architecture depends on the selection of output vectors that are assigned to $Group_1$ (encoder/decoder). Unfortunately, for n different output patterns there

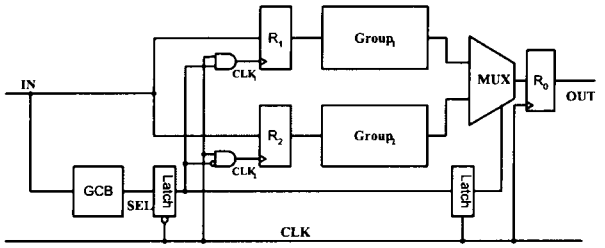


Fig. 3. A bipartition architecture

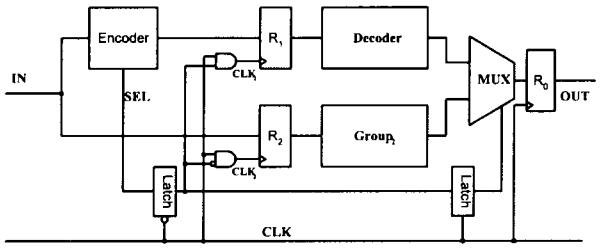


Fig. 4. A bipartition-codec architecture

are $\binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{n-1} = 2^n - 2$ possible bipartitioning configurations. As a result, testing all configurations becomes computational expensive for large n.

IV. ENPCO Algorithm

A. Problem Formulation

The total area estimation model is:

$$\begin{aligned} Total\_Area &= area_E + area_D + area_{G2} + overhead \\ &= L_E(n_E, d_E, X_E) + L_D(n_D, d_D, X_D) + \\ &\quad L_{G2}(n_{G2}, d_{G2}, X_{G2}) + overhead \\ &= L_E(n, d_E, X_E) + L_D(n_D, d_D, X_D) + \\ &\quad L_{G2}(n, d_{G2}, X_{G2}) + overhead, \end{aligned} \tag{4}$$

where $area_E$, $area_D$ and $area_{G2}$ represent the area of Encoder, Decoder and $Group_2$ after being synthesized by SIS, respectively. The last term *overhead* consists of latches, logic gates, registers and multiplexers shown in Fig. 4. Note that we replaced n_E and n_{G2} with n because the numbers of input pins for *Encoder* and $Group_2$ are the same as the number of input pins of the original circuit.

For our optimization approach we use a slightly simplified version of Equation 4 to estimate area consumption of the bipartition codec architecture by neglecting the *overhead* term which is a constant throughout the optimization approach:

$$\begin{aligned} Total\_Area' &= L_E(n, d_E, X_E) + L_D(n_D, d_D, X_D) + \\ &\quad L_{G2}(n, d_{G2}, X_{G2}) \\ &= k \cdot 2^n((1 - d_E)H(X_E) + (1 - d_{G2})H(X_{G2})) + \\ &\quad k \cdot 2^{n_D}(1 - d_D)H(X_D). \end{aligned} \tag{5}$$

B. Two Phases ENPCO Algorithm

The input of the algorithm is the set of different output patterns $S = \{s_1, s_2, s_3, \cdots s_q\}$ of the circuit (note that here only pattern that actually appear at least once on the outputs are in S). Its output is a set of output pattern $Q \subset S$ that are assigned to the encoder/decoder part of the bipartition architecture.

PROCEDURE SelectCandidates(S, p)
  Input: (randomly ordered) set of output states S,
    probability function p
  Output: candidate set C
  Initialization: $C = \emptyset, prob = 0$;
BEGIN
  Sort $S = (s_1, s_2, \cdots s_q)$ by decreasing $p(s_i)$;
  WHILE $prob \leq 0.5$ **OR** p(first element in S) $> \frac{1}{q}$ **LOOP**
    $s = \{$first element in $S\}$;
    $C = C \cup \{s\}$;
    $prob = prob + p(s)$;
    $S = S \setminus \{s\}$;
  END LOOP;
  RETURN C;
END PROCEDURE;

Fig. 5. Algorithm to select candidates (phase 1)

The algorithm is divided into two phases. In the first phase we select a set of candidates $C \subset S$ which might be moved to the encoder/decoder part of the architecture. These candidates are selected by their frequency of appearance $p(s_i)$ on the outputs of the circuit. The motivation behind this approach is to assign a few highly active output pattern to the encoder/decoder section of the bipartition circuit in order to keep the small (with respect to area) part of the architecture active most of the time while the bigger part is mostly idle.

The second phase selects from the set of candidates C a subset of pattern $Q \subseteq C$ that are finally assigned to the encoder/decoder part.

Phase 1: Figure 5 shows the algorithm used for Phase 1. In order to select a "promising" set of candidates for the second phase of our algorithm, C is determined by two main rules. First, all patterns that have a probability above the average value $\frac{1}{q}$ are added to candidate set C. If the sum of probabilities of all patterns (*prob*) in C is less than $1/2$ then further pattern are added (in decreasing probability order) until $1/2$ is reached.

Phase 2: After determining the candidate set C, the next step is to select a subset of elements from C that will be assigned to the encoder/decoder section of our architecture. However, our selection does not only focus on reducing power but also on reducing the area required to implement the entire architecture. Hence, our goal is to find a non-empty set $Q \subseteq C$ that if assigned to the encoder/decoder part of our architecture results in a minimal total area complexity. Note that an exhaustive search is impractical as $O(2^{|Q|})$ permutations must be analyzed. Hence, we use a greedy approach to find an acceptable solution in $O(|Q|^2)$.

During runtime of the algorithm area consumption of various configurations are tested and compared as follows: based on a chosen pattern set $V \subseteq C$, the algorithm introduced in [19] is executed to find a good binary representation of V. The encoding results are then used to estimate area of the total circuit using the following Equation:

$$\begin{aligned} Total\_Area''(S, V) = 2^n((1 - d_{E(V)})H(X_{E(V)}) + \\ (1 - d_{G2(S \setminus V)}H(X_{G2(S \setminus V)}))) + \\ 2^{n_{D(V)}}(1 - d_{D(V)})H(X_{D(V)}). \end{aligned} \tag{6}$$

It is derived from Equation 5 by removing constant k. V is the set of output states that shall be associated with the encoder/decoder part while S is the entire output set. Note that values $d_{E(V)}$, $X_{E(V)}$, $n_{D(V)}$, $d_{D(V)}$ and $X_{D(V)}$ depend on the results obtained from encoding.

The actual algorithm is shown in Figure 6. It takes the candidate set C as a parameter and returns the final set Q to be assigned to the

PROCEDURE FindLowPowerSet(S, C)
  Input: randomly ordered output set S and candidate set C
  Output: low power set Q
  Initialization: $T = \emptyset$, $Q = \emptyset$
BEGIN
  WHILE $C \neq \emptyset$ **LOOP**
    select $s \in C, s \neq \emptyset$ so that $Total\_Area''(S, T \cup \{s\}) = \min$;
    $T = T \cup \{s\}$;
    $C = C \setminus \{s\}$;
    IF $Q = \emptyset$ **OR**
      $Total\_Area''(S, Q) > Total\_Area''(S, T)$ **THEN**
      $Q = T$;
    END IF;
  END LOOP;
  RETURN Q;
END PROCEDURE;

Fig. 6. Algorithm to select final set Q from C (phase 2)

encoder/decoder part. The algorithm uses two internal sets: T is a temporary set of pattern while Q holds the best pattern set found so far.

In each iteration of the while loop a pattern $s \in C$ is selected so that the total area of the circuit is minimal. This is done by applying the encoding algorithm introduced in [19] on each combination $T \cup s$ and using Equation 6 to *estimate* the area of the total circuit (i.e., area of $Encoder + Decoder + Group_2$). From these estimation results the best (smallest total area) is selected and the corresponding s is added to T and removed from C. Hence, in each iteration of the while loop a single pattern is removed from C and added to T until C is empty. Finally, in each iteration the area consumption of T is compared to the area value of Q to determine the best pattern set among all iterations executed so far.

In order to explain the algorithm in more detail a short example is given: suppose we have a candidate set $C = \{s_1, s_2, s_3, s_4\}$ that has been extracted from the output pattern set S (I.e., $C \subset S$). During the first iteration a pattern s is determined that gives us a minimum area value. E.g., we determine $Total\_Area''(S, s_1)$, $Total\_Area''(S, s_2)$, $Total\_Area''(S, s_3)$ and $Total\_Area''(S, s_4)$ in order to find the best pattern. Let us assume that $Total\_Area''(S, s_3)$ is the minimum among all these values. Hence, T^1 is set to $\{s_3\}$ and s_3 is removed from C (i.e., $C^1 = \{s_1, s_2, s_4\}$; the exponent denotes the iteration number). Next we compare $Total\_Area''(S, \{s_3, s_1\})$, $Total\_Area''(S, \{s_3, s_2\})$ and $Total\_Area''(S, \{s_3, s_4\})$. Suppose that $Total\_Area''(S, \{s_3, s_1\})$ is the minimal area among these three values. As a result, we get $T^2 = \{s_3, s_1\}$ and $C^2 = \{s_2, s_4\}$. Let us assume that in the next iteration pattern s_4 is selected giving $T^3 = \{s_3, s_1, s_4\}$ and $C^3 = \{s_2\}$. Finally, in the next iteration the last remaining pattern in C is added giving $T^4 = \{s_3, s_1, s_4, s_2\}$ and $C^4 = \emptyset$. Hence, we get four different configurations $T^1 = \{s_3\}$, $T^2 = \{s_3, s_1\}$, $T^3 = \{s_3, s_1, s_4\}$ and $T^4 = \{s_3, s_1, s_4, s_2\}$. Then, the set with a minimum area is returned as the final result. Note that the final result is not necessarily T^1. I.e., adding a pattern to T^1 may *decrease* the complexity of the encoder/decoder part due to the fact that among the decoded/encoded value the selection signal SEL must be also generated by the encoder.

Complexity Analysis

The first phase of the ENPCO algorithm starts with a sort operation, which requires $O(q \log q)$ if an efficient sorting algorithm is applied (see Figure 5). Then, the following while loop is executed q times. This gives an overall complexity of $O(q \log q)$ for the first phase.

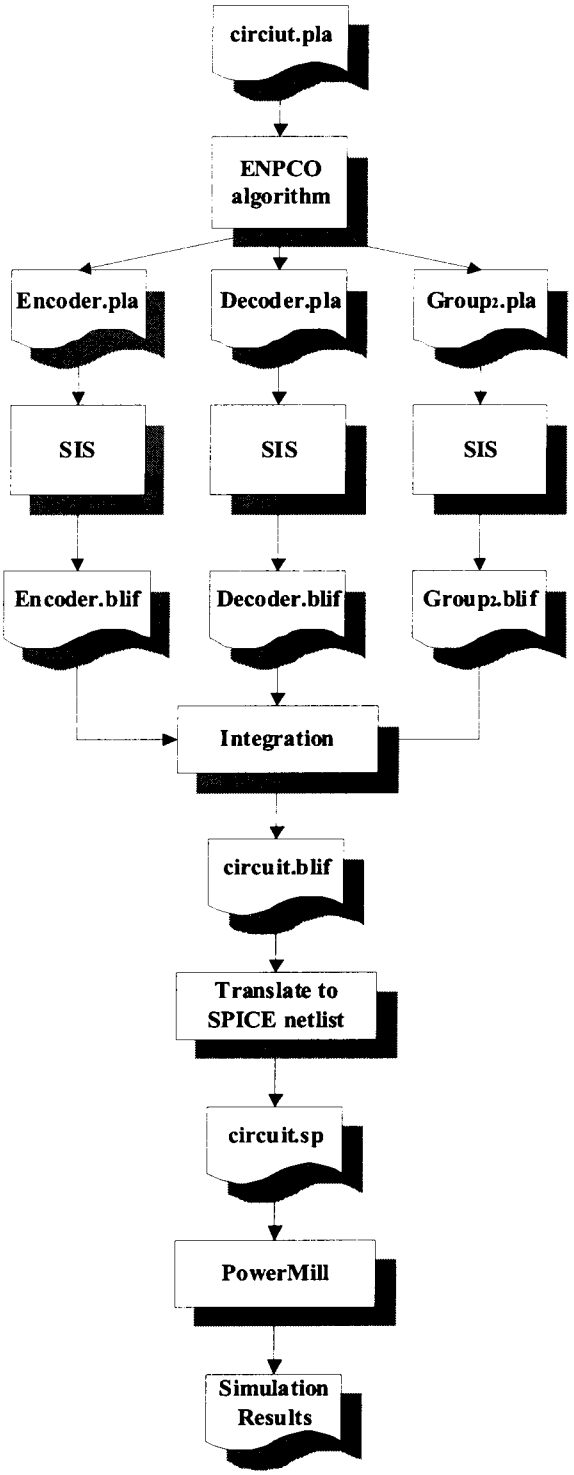


Fig. 7. Simulation Flow

Applying the encoding algorithm from [19] during phase 2 requires $O(|C|^3)$ steps, where $|C|$ is the size of the candidate set. Further, the encoding algorithm is executed $O(|C|^2)$ times as the while loop from procedure FindLowPowerSet (see Figure 6) iterates $|C|$ times and $O(|C|)$ configurations are tested in each iteration. Hence, the overall complexity of phase 2 is $O(|C|^5)$.

V. Simulation Environment and Results

A. Simulation Environment

The ENPCO algorithm has been implemented in C++ on a SUN Sparc station and several MCNC benchmark circuits were used to test it. The rugged script of SIS [20] was used to synthesize *Encoder*, *Decoder*, and $Group_2$. Power dissipation estimation was done by EPIC PowerMill. We used 5v supply voltage, and a clock frequency of 20MHz.

In our experiments, we selected the primitive standard cells which were provided by CCL [2] in 0.8μm technology. From the data book of CCL 0.8μm standard cell library, we extracted the propagation delays and driving capabilities of every standard cell. These numbers were used to construct a technology file in genlib format to be used by SIS. Further, the standard cells in SPICE sub-circuit format were converted into the EPIC format using EPIC utility *gentech*. So, as the final synthesized results are in EPIC format, EPIC PowerMill could be used to estimate power consumption of the pipelined circuits. The overall flow is shown in Fig. 7.

The MCNC benchmarks that are given in two-level PLAs are bipartitioned and encoded into three PLAs, *Encoder*, *Decoder* and $Group_2$. A synthesis script, *script.rugged* of SIS was used to optimize each PLA. Then the library binding program *map* of SIS was used to generate a corresponding gate-level file in BLIF format (Berkeley Logic Interchange Format). A BLIF to SPICE format converter was used to convert the BLIF gate-level description to SPICE transistor-level according to a layout-extracted netlist file. Finally, the utility *spice2e* of EPIC converted the pipeline circuits from SPICE format to EPIC format.

B. Experimental Results

We assumed uniform probability distribution for the primary inputs of the pipelined circuit, but this assumption is not restrictive. For example, in a pipelined circuit the input probability distribution can be computed from the output probability of the previous stage. The registers of the output part are unchanged in our architectures. Hence, we do not consider the effect of these registers on area and power dissipation. The area unit and power unit are 128 μm^2 and μW in our experiments.

The area and power dissipation of the original, single-phase and ENPCO algorithms are tabulated in Table I. The "Original" column shows the number of inputs ($\#I$), outputs ($\#O$), area $Area$ and power dissipation $Power$ of each circuit. The "single-phase algorithm" and "ENPCO algorithm" columns show $Area$, $Power$, percentage of area increase $AI\%$ computed as $100(Area_{bipartition} - Area_{original})/Area_{original}$, and power reduction $PR\%$ computed as $100(Power_{original} - Power_{[single-phase|ENPCO]})/Power_{original}$. The column DN shows the number of output vectors that were assigned to the codec structure.

The ENPCO algorithm is capable of reducing the area of combinational parts (i.e. *Encoder*, *Decoder* and $Group_2$). However, area overhead (i.e., two latches, two AND gates and multiplexers as well as the additional registers) may become significant if the original circuit is *small*. Further, these additional components also consume power. As a result, for small circuits, the area overhead introduced by our architecture may become significant while saving only little or no power.

In Fig. 8 and Fig. 9, we show the comparison of power reduction and area increase. In Fig. 8, the curve labeled *Power increase* is obtained by subtracting ENPCO algorithm's PR% from single-phase algorithm's PR%. Similar, *Area reduction* of the Fig. 9 is obtained by subtracting ENPCO algorithm's AR% from single-phase algorithm's AR%. As shown in Fig. 8 and Fig. 9, ENPCO algorithm have almost the same power saving results compared to the single-phase algorithm but requires less area overhead. Fig. 10 shows the AI% and PR% values obtained by ENPCO algorithm for comparison.

Table II compares the average area and power reduction of previous schemes with our proposed algorithm. The $precomputation$ column represents the circuits implemented by precomputation-based method [6]. The *Choi's algorithm* and *Choi's area constraint* columns represent the circuits implemented by Choi's algorithm and Choi's algorithm with area constraint, respectively [8]. The data shown in these three columns are cited from [17]. The $Single-phase$ column represents the circuits implemented by a single-phase algorithm introduced in [1]. Our proposed algorithm is shown in column $ENPCO$. Obviously, ENPCO algorithm obtained more power reduction and less area overhead than Choi's algorithm with area constraint. Moreover, ENPCO algorithm achieves almost the same power saving as the single-phase algorithm, however, it adds significant less area overhead. Although precomputation and Choi's area constraint algorithms can achieve power reduction and reduce the area marginally, their power reductions are limited. In summary, our proposed architecture obtained significant power saving with less area overhead compared to previously published techniques.

Our approach reduces power consumption from the circuits with few high-probability output vectors. We prove the bipartition-codec circuit consumes less power than the original one. The power dissipation of the input registers is also reduced for the sake of less switching of input variables. Further, the codec architecture not only reduces the switching activity of the input registers but also the internal switching activity of *Decoder*.

However, there are circuits that are not suitable for bipartition-codec methodology. Consider the benchmark circuits in Table I. We observed that for some circuits such as *sao2*, *misex1* and *cmb* the ENPCO algorithm reduces power up to 75.2%. However, for other circuits such as *rd73*, it provides only limited power reduction. Moreover, the area overhead introduced by the architecture is significant. It seems that these circuits are not suitable for this approach.

Hence, if we find a method to detect these kind of circuits in advance we could avoid applying our technique on these cases saving synthesis runtime. Table III displays the output patterns for a subset of the benchmark circuits. While for models *sao2* and *cmb* few output patterns are activated by many different input patterns, models *rd73* show a more balanced output pattern probability. Hence, for model *sao2* and *cmb* good power reduction results could be achieved while our approach fails for *rd73*. I.e., our approach is promising for models where most input patterns are mapped to a small set of output patterns. For other circuits the power reduction that can be achieved is often small while the area overhead is significant. Hence, for these kind of circuits it is better to avoid applying our algorithm.

In order to detect these circuits, we introduce an extra check in the proposed algorithm. Consider a set of q output pattern $W = \{s_1, s_2, \cdots, s_q\}$ where each of these pattern appear at least once on the output. Then, $Y = \{y_1, y_2, \cdots, y_q\}$ is the frequency vector of W, where y_i is the number of appearences of pattern s_i at the output. Y

[2] CCL stands for Computer and Communication Research Labs and is one of the members of Industrial Technology Research Institute in Taiwan.

meets the following two conditions (n is the number of input pins):

$$y_i \in \{1, 2, \cdots, (2^n - q + 1)\} \tag{7}$$

and

$$\sum y_i = 2^n. \tag{8}$$

Hence, the mean value of Y is $\overline{Y} = \frac{2^n}{q}$. In order to rate a circuit we calculated the normalized deviation ($C.V.$):

$$C.V. = \frac{\sqrt{\frac{1}{q}\sum_{i=1}^{q}(y_i - \frac{2^n}{q})^2}}{S_{max}}, \tag{9}$$

where the numerator is the standard deviation of Y and S_{max} is the maximum standard deviation value among all possible sets Y that meet Equations 7 and 8. Hence, $C.V.$ holds $0 \leq C.V. \leq 1$. The deviation of Y is maximal if all except one y_i are set to 1. E.g., y_1 may be set to $y_1 = 2^n - q + 1$ and all remaining y_i are set to $y_i = 1$. As a result, S_{max} is

$$S_{max} = \sqrt{\frac{q-1}{q}\left(1 - \frac{2^n}{q}\right)^2 + \frac{1}{q}\left(2^n - q + 1 - \frac{2^n}{q}\right)^2}. \tag{10}$$

In the following we prove that Equation 10 is actually the maximum value.

Proof: For the proof it is sufficient to show that $S'(Y') = \sum_{i=1}^{q}(y_i - \frac{2^n}{q})^2$ is maximum for $Y' = (\hat{y}, 1, 1, \ldots, 1)$, where $\hat{y} = 2^n - q + 1$. Subtracting a value δ with $0 < \delta \leq \hat{y} - 1$ from y_1 and adding it to y_2 creates a new vector Y'':

$$Y'' = (\hat{y} - \delta, 1 + \delta, 1, \ldots, 1).$$

Note that δ must not be greater than $\hat{y} - 1$ to ensure that all elements of Y'' are greater or equal to 1. Calculating the difference $S'(Y') - S'(Y'')$ gives

$$S'(Y') - S'(Y'') = 2\delta(\hat{y} - 1 - \delta) \geq 0. \tag{11}$$

Hence, $S'(Y')$ is greater or equal to $S'(Y'')$. I.e., decreasing y_1 and increasing y_2 by the same value does *not* increase S'. Based on the previous analysis we also conclude that decreasing y_2 and increasing y_3 of Y'' by δ' $(0 < \delta' \leq \delta)$ also does *not* increase S'. Consequently, $S'(Y''') \leq S'(Y')$ is valid for each vector

$$Y''' = (\hat{y} - \delta_1, 1 + \delta_1 - \delta_2, 1 + \delta_2 - \delta_3, \cdots, 1 + \delta_{q-1}),$$

with $\hat{y} - 1 \geq \delta_1 \geq \delta_2 \geq \cdots \geq \delta_{q-1} \geq 0$. As for each valid Y an appropriate δ_i-set can be determined such that $Y = Y'''$, S' as well as the standard deviation are maximal for $Y = Y'$.

In order to characterize a circuit we evaluate Equation 9. If $C.V.$ is large (suppose > 0.8), it implies most of the inputs are mapped to some few outputs. Hence the bipartition-codec methodology is capable of reducing power consumption significantly. If $C.V.$ is small (say ≤ 0.8), the occurrence of each output pattern is uniformly distributed. This implies that our bipartition-codec methodology will probably achieve only little power saving but will introduce a significant area overhead. The threshold value (here: 0.8) can be changed to match different needs and applications. I.e., if low power is the dominant consideration disregarding area overhead, we can use a smaller threshold value. If area and power reduction are both our optimization targets, we can use a larger threshold value to filter out the unsuitable circuits.

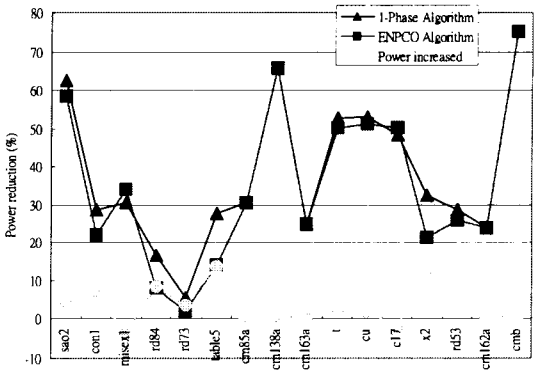


Fig. 8. Power Reduction rate (%) for single-phase and ENPCO algorithms applied to MCNC benchmarks

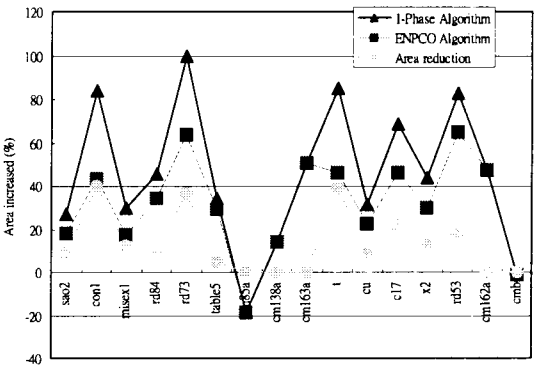


Fig. 9. Area increased rate (%) for single-phase and ENPCO algorithms applied to MCNC benchmarks

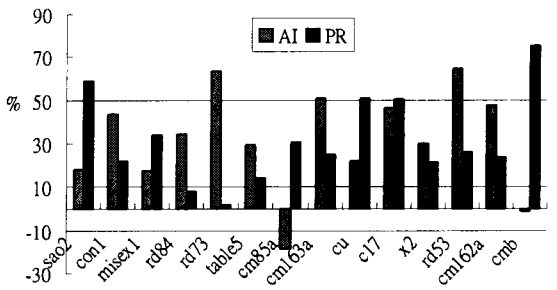


Fig. 10. Area increase (AI%) and power decrease (PR%) comparison for ENPCO algorithms applied to MCNC benchmarks

TABLE I
SIMULATION RESULTS OF ORIGINAL CIRCUIT AND BIPARTITION ARCHITECTURES

Circuits	Original				single-phase Algorithm [1]				ENPCO Algorithm				
	#I	#O	Area	Power	Area	AI%	Power	PR%	DN	Area	AI%	Power	PR%
sao2	10	4	571	3361	725	27.0	1255	62.7	2	675	18.2	1392	58.6
con1	7	2	209	1506	384	83.7	1072	28.8	2	300	43.5	1176	21.9
misex1	8	7	340	2238	441	29.7	1555	30.5	2	399	17.4	1474	34.1
rd84	8	4	531	3176	774	45.8	2649	16.6	2	714	34.5	2920	8.1
rd73	7	3	370	2362	740	100.0	2230	5.6	3	605	63.5	2316	1.9
table5	17	15	3203	7946	4312	34.6	5748	27.7	2	4145	29.4	6831	14.0
cm85a	11	3	383	2535	312	-18.5	1757	30.7	3	312	-18.5	1757	30.7
cm163a	16	5	475	3573	716	50.7	2688	24.8	1	716	50.7	2688	24.8
t	5	2	132	1007	244	84.8	477	52.6	2	193	46.2	501	50.3
cu	14	11	485	3144	637	31.3	1470	53.2	3	593	22.3	1538	51.1
C17	5	2	132	1005	223	68.9	519	48.3	1	193	46.2	500	50.2
x2	10	7	371	2442	533	43.7	1646	32.6	3	482	29.9	1919	21.4
rd53	5	3	216	1499	395	82.9	1067	28.8	1	356	64.8	1112	25.9
cm162a	14	5	444	3109	654	47.3	2369	23.8	3	654	47.3	2369	23.8
cmb	16	4	440	3265	435	-1.1	811	75.2	1	435	-1.1	811	75.2

TABLE II
AVERAGE AREA AND POWER COMPARISON AMONG DIFFERENT METHODOLOGIES

	Precomputation	Choi's algorithm	Choi's area constraint	Single-phase [1]	ENPCO
Area (%)	-0.5	99	-0.3	45.3	31.8
Power (%)	-9.6	-32	-11.6	-38.0	-34.9

TABLE III
OUTPUT PATTERNS LIST AND ITS CORRESPONDING OCCURRENCES ON SOME BENCHMARK CIRCUITS.

sao2(58.6%)		cmb(34.1%)		rd73(1.9%)	
Output patterns	Occurrences	Output patterns	Occurrences	Output patterns	Occurrences
0000	513	0110	65489	110	35
0010	257	0011	16	001	35
0011	219	0100	15	100	21
0100	8	1110	15	011	21
1000	7	1100	1	010	7
1100	6			101	7
0001	5			000	1
0101	4			111	1
1001	3				
1101	2				

Scaling Problem: In [2], the authors indicate that according to the simulation results, the total power consumption of a design is similar among different technologies. From a theoretical viewpoint, a scaling factor SC defines the technology change in a certain physical parameter (e.g., gate oxide thickness) from one technology generation to another [21]. For instance, changing the channel length from 0.35 to 0.25 μm gives SC the value 0.71.

The ideal scaling for deep-submicrometer of power dissipation and area is SC^2. For the computation of AI% and PR% in our experiments, they are the same in different technologies. Therefore, the results we obtained and the conclusion we draw in this paper apply to other technology levels as well.

VI. Conclusion and Discussion

In this paper, we have proposed an effective ENPCO algorithm to minimize power dissipation and area overhead for pipelined circuits. We bipartition the circuits into two groups: one group contains patterns that occur often at the outputs which implies higher activity, the other group contains the remaining output patterns. We apply a codec structure to the high active group for power saving. Then we estimate the area of combinational blocks (Encoder, Decoder and Group_2) by using an entropy based approach. Finally we choose the configuration with minimal estimated area overhead as our final synthesized result. We compared the single-phase and ENPCO algorithms by power and area. The experiments show that the ENPCO algorithm achieves a good trade-off between area overhead and power dissipation. The power dissipation benefit of the bipartition-codec architecture synthesized by our ENPCO algorithm comes from the following three reasons:

1) The lengths of the registers, which are used to store the output of each stage, are reduced after encoding.
2) The Hamming distance of the register values is smaller than before.
3) The circuit switching activity of the combinational block is reduced.

The circuit will benefit from our architecture if a small number of output vectors dominate most of the circuit behavior (e.g., see models *sao2* and *cmb* in Table III). Nevertheless, circuits that are characterized by a uniform output vector probability distribution are not suited to our architecture. For example, the output pattern distribution of *rd73* is more uniform than those of the others models (see Table III). As a result, power reduction is less significant than that of *sao2* and *cmb*.

Compared to precomputation architecture, precomputation only disables the partial input pins for reducing the switching activity of combinational logic. Hence, the remainder input signals may also incur redundant switching activity in the entire combinational logic. Furthermore, precomputation does not account for the power dissipation of pipeline registers. Conversely, bipartition-codec architecture not only separates the combinational logic to ensure that they will not influence each other but also reduces power dissipation of the pipeline registers by applying a codec structure.

References

[1] S.-J. Ruan, R.-J. Shang, F. Lai, S.-J. Chen, and X.-J. Huang, "A Bipartition-Codec Architecture to Reduce Power in Pipelined Circuits," in *Proceedings of IEEE/ACM Int'l. Conf. on Computer Aided Design*, pp. 84–89, Nov. 1999.

[2] W. Ye and M. J. Irwin, "Power Analysis of Gated Pipeline Registers," in *Proceeding of Twelfth Annual IEEE International ASIC/SOC Conference*, pp. 281–285, 1999.

[3] L. Benini and G. D. Micheli, *Dynamic Power Management: Design Techniques and CAD Tools*. Kluwer Academic Publishers, 1998.

[4] M. Pedram, "Power Minimization in IC Design: Principles and Applications," *ACM Tran. Design Automation of Electronc Systems*, vol. 1, pp. 3–56, Jan. 1996.

[5] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design*. New York: John Wiley, 2000.

[6] M. Alidina, J. Monterio, S. Devadas, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-Based Sequential Logic Optimization for Low Power," *IEEE Trans. on VLSI Syst.*, vol. 2, pp. 426–436, Dec. 1994.

[7] G. Lakshminarayana, A. Raghunathan, K. Khouri, N. Jha, and S. Dey, "Common-Case Computation: A High-Level Technique for Power and Performance Optimization," in *Proceeding Design Automation Conf.*, pp. 56–61, 1999.

[8] I.-S. Choi and S.-Y. Hwang, "Circuit Partition Algorithm for Low-Power Design Under Area Constraint Using Simulated Annealing," *IEE Proc. Circuit Devices Syst.*, vol. 146, pp. 8–15, Feb. 1999.

[9] S.-J. Ruan, J.-C. Lin, P.-H. Chen, F. Lai, K.-L. Tsai, and C.-W. Yu, "An Effective Output-Oriented Algorithm for Low Power Multipartition Architecture," in *IEEE Int'l. Conf. on Electronics, Circuits and Systems*, pp. 609–612, 2000.

[10] L. Benini and G. D. Micheli, "Automatic Synthesis of Low-Power Gated Clock Finite-State Machine," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 630–643, Sep. 1996.

[11] J. C. Monterio and A. L. Oliveira, "Finite State Machine Decompostion For Low Power," in *Proceeding Design Automation Conf.*, pp. 758–763, 1998.

[12] S.-H. Chow, Y.-C. Ho, T. Hwang, and C. L. Liu, "Low Power Realization of Finite State Machines— A Decompostition Approach," *ACM Tran. Design Automation of Electronc Systems*, vol. 1, pp. 315–340, July 1996.

[13] A. Lioy, E. Macii, M. Poncino, and M. Rossello, "Accurate Entropy Calculation for Large Logic Circouts Based on Output Clustering," in *in Proceedings. Seventh Great Lakes Symposium on VLSI*, pp. 70–75, 1997.

[14] M. Nemani and F. N. Najm, "High-Level Area and Power Estimation for VLSI Circiuts," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 697–713, June 1999.

[15] E. Macii, M. Pedram, and F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1061–1079, Nov. 1998.

[16] K.-T. Cheng and V. D. Agrawal, "An Entropy Measure for the Complexity of Multi-output Boolean Functions," in *proceedings of 27th ACM/IEEE Design Automation Conference*, pp. 302–305, 1990.

[17] S.-J. Ruan, R.-J. Shang, F. Lai, and K.-L. Tsai, "A Bipartition-Codec Architecture to Reduce Power in Pipelined Circuits," *IEEE Trans. Computer-Aided Design*, pp. 343–348, Feb. 2001.

[18] G. Yeap, *Practical Low Power Digital VLSI Design*. Kluwer Academic, 1998.

[19] L. Benini and G. D. Micheli, "State Assignment for Low Power Dissipation," *IEEE J. Solid-State Circuits*, vol. 30, pp. 258–268, Mar. 1995.

[20] "SIS: A System for Sequential Circuit Synthesis," tech. rep., Electronic Research Lab. in Department of EE and CS, University of California, Berkeley, May 1992.

[21] D. Sylvester and C. M. Hu, eds., *Analytical Modeling and Characterization of Deep-Submicrometer Interconnect*, vol. 89, Proceeding of IEEE, May 2001.

Power Analysis of Bipartition and Dual-Encoding Architecture for Pipelined Circuits *

Shanq-Jang Ruan
Dept. of EE
National Taiwan University, Taiwan
stj@orchid.ee.ntu.edu.tw

Edwin Naroska
Computer Engineering Institute
University of Dortmund, Germany
edwin@ds.e-technik.uni-dortmund.de

Chia-Lin Ho
Dept. of CSIE
National Taiwan University, Taiwan
hcl@orchid.ee.ntu.edu.tw

Feipei Lai
Dept. of CSIE & Dept. of EE
National Taiwan University, Taiwan
flai@cc.ee.ntu.edu.tw

Abstract

In this paper, we propose a bipartition dual-encoding architecture for low power pipelined circuit. Pipelined circuits consist of combinational logic blocks separated by registers which usually consume a large amount of power. Although the clock gated technique is a promising approach to reduce switching activities of the pipelined registers, this approach is restricted by the placement of the registers and the additional control signals that must be generated. Thus, we propose a technique for optimizing power dissipation of a pipelined circuit addressing registers and combinational logic blocks at the same time. Our approach modifies the registers using bipartition and encoding techniques. In our experiments power consumption were reduced by 72.9% for pipelined registers and 30.4% for the total pipelined stage on average.

1 Introduction

In modern processor designs, pipelining is the most popular fashion to increase overall performance. Since [1] first applied pipelined circuits for lowering power, a number of work have been done and published on synthesis of low power pipelined circuits. We can categorize these previous work as follows:

- The first category covers approaches where pipelined registers are immovable. Techniques like precomputation (e.g.,[1]), gated pipelined registers (e.g.,[13, 5]) and guarded evaluation (e.g.,[12, 7]) belong to this type of approaches. Precomputation disables the parts of the pipelined registers which do not affect the output value [1]. The authors of [13, 5] reduce switching activities of pipeline registers by using control signals to gate the clock. In [12, 7], the authors isolate those operands that result in redundant operations to save unnecessary power dissipation. Unfortunately, as registers are immovable further improvements to these approaches is limited. Another limitation is that they require additional control logic.

- In the second category pipelined registers are movable. Approaches like retiming (e.g., [6]) and repositioning of registers in datapaths (e.g., [11]) belong to this category. Monteiro *et al.* selects a set of nodes, which, by having a flip-flop placed at their output, leads to the minimization of switching activities in the network [6]. Schimpfle *et al.* computes "switch weight" for each node so that each circuit can be retimed by using switching activity instead of delays [11]. However, no effort is made to modify the combinational logic blocks.

- Approaches belonging to the third category partition circuits to reduce the size of active registers and logic blocks in order to decrease power consumption [4]. In [4], Choi and Hwang partition the combinational logic block of a pipelined circuit into multiple sub-circuits using the recursive application of Shannon expansion with respect to the selected input variables [4]. Furthermore, Ruan *et al.* showed that partitioning circuits into more than two sections does not always save power consumption due to the overhead of duplicated

*This work was sponsored in part by Synopsys, Inc. and the National Science Council of Taiwan under Grant NSC 90-2213-E002-108

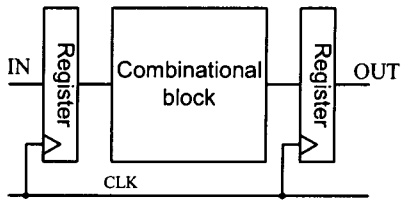


Figure 1. A combinational pipelined circuit

input latches and output multiplexers [9].

Some preliminary work in combining techniques of partitioning and retiming have been done in [10, 3], but the inability to extract the most active portion may make this approach inapplicable in the real world. In this paper we take advantage of partitioning and encoding techniques towards optimizing power in pipelined circuits. As in [10, 3], we consider a single-phase, edge-triggered design method (see Fig. 1) that is used in a large number of ASIC applications.

In this paper, we propose a bipartition dual-encoding architecture to achieve the goal of lowering power consumption in a design. We first bipartition a given circuit by using Shannon expansion in terms of minimizing the number of different outputs of two sub-circuits. Secondly, we encode both partitions to reduce the switching activities of the pipelined registers and logic block. To validate the results, we employ an accurate transistor-level power estimator[1] to estimate power dissipation.

The paper is organized as follows. In Section 2, we describe the power distribution of pipelined circuits and how we encode them for minimal Hamming distance. In Section 3 we present bipartition single-encoding and dual-encoding architectures and discuss their characteristics. Section 4 proposes a synthesis algorithm for bipartition dual-encoding architecture and Section 5 shows the experimental results to verify and prove the feasibility of our algorithm. Finally, we conclude with a summary in section 6.

2 Preliminaries

2.1 Power Distribution

In this paper, we adopt a single-phase edge-triggered pipelined circuit (see Fig. 1) that is used in a large number of ASIC applications. As shown in Fig. 1, pipelined circuits have no state feedback lines because they do not need information of the previous cycles for running the present cycle.

[1]EPIC PowerMill was developed by EPIC Design Technology, INC.

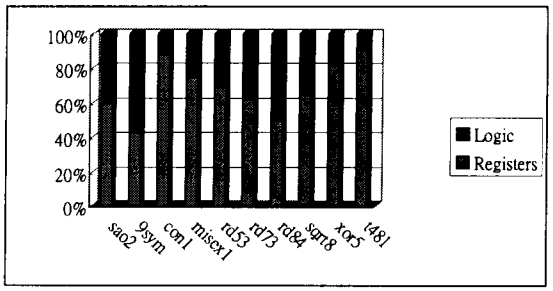


Figure 2. Power dissipation for several MCNC benchmarks

In order to compare power dissipation results obtained by the bipartition method, we began our power analysis with the original benchmark circuits. We then implemented the circuits presented in Fig. 1 using the primitive standard cell technology provided by CCL[2] (0.8μm technology). Power dissipation were estimated using EPIC PowerMill. Fig. 2 shows that the pipelined registers take a large fraction of total power dissipation for most of the circuits. On average they account for 64.6% of the total power budget. Note that even when the input data presented to a flipflop does not change, it consumes power during a clock cycle. This obviously indicates the registers are the largest contributor to the power budget of a pipelined circuit. Hence, we can say that pipelined registers should be taken into account when optimizing a circuit design for low power. The interested reader may refer to [13] for more detailed treatment.

2.2 Encoding for Low Power

As in a pipelined circuit the total power dissipation mainly depends on the switching activity of the pipelined registers we use the following cost function [8]:

$$\sum_{S_i,S_j \in S} tP_{ij} H(S_i, S_j), \qquad (1)$$

where tP_{ij} is the global state transition probability from S_i to state S_j, and $H(S_i, S_j)$ is the Hamming distance between the encoding of the two states. Further, reducing switching activity usually decreases the complexity/area of a logic block, which in turn has positive effects on the power consumed by the logic portion of the circuit.

[2]CCL stands for Computer and Communication Research Labs and is one of the members of Industrial Technology Research Institute in Taiwan.

3 Architectures

In this section, we describe two different architectures and discuss their characteristics in terms of their impacts on power dissipation.

3.1 Bipartition Single-Encoding Architecture

Fig. 3 is a bipartition single-encoding architecture based on extracting the most frequent outputs and its corresponding inputs to form a small sub-circuit. Then, the output is encoded in order to reduce the switching activity of pipelined registers and the combinational portion. Note that k is the number of different outputs of the extracted sub-circuit.

The architecture works as follows: the input data are processed by the Encoder which produces an encoded output as well as a single bit signal SEL. SEL is used to activate either the upper part (Encoder/Decoder) or the lower part (Subcircuit_2) of the architecture. The upper part will take over the most frequent output pattern while the lower part is responsible for the less frequent output pattern. The first (left) latch shown in Fig. 3 is required to prevent glitches from accidently activation R_1 or R_2 while the second (right) is needed to select the appropriate output of the pipeline stage.

Example 1: Taking the benchmark sao2 in MCNC as an example, the number of inputs is 10 and the number of outputs is 4. If each possible input pattern (2^{10}) is assigned exactly once to the circuit the output pattern frequency count of sao2 will be:

output pattern	count
0000	513
0010	257
0011	219
0100	8
1000	7
1100	6
0001	5
0101	4
1001	3
1101	2

In this example, if we cluster pattern $\{0000\}$ and $\{0010\}$, and encode them as $\{0\}$ and $\{1\}$, the number of output pins of the encoder is 2 (including signal SEL).

Power reduction is obtained during operation as the part that corresponds to to $SEL = 1$ is active most of the time. Hence, register R_1 will also be active most of the time. As the size of R_1 is usually less than the size of the register in the original circuit the registers in the bipartition single-encoding architecture will consume less power.

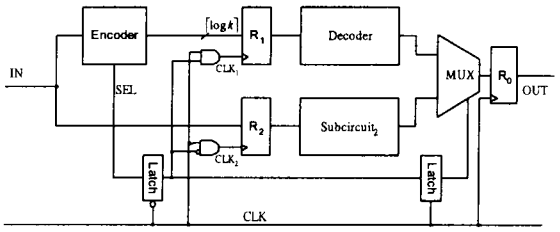


Figure 3. A bipartition single-encoding architecture.

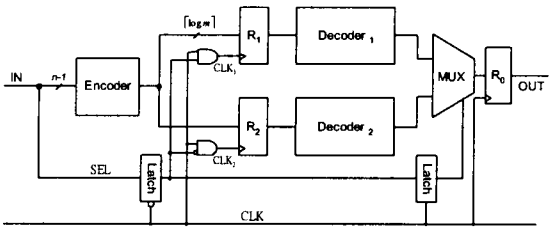


Figure 4. A bipartition dual-encoding architecture.

However, this approach only works if the distribution of the output pattern probability is not uniform. Otherwise, the circuits will not benefit from the architecture. The interested reader may refer to [10] for a further analysis on the topic.

3.2 Bipartition Dual-Encoding Architecture

In the architecture of Fig. 4, we partition the circuit based on the Shannon expansion. The criterion of selecting a partition variable is to minimize the pipelined registers. Then we encode both partitions with minimal Hamming distance for reducing the switching activity. m is the maximal number of different outputs of the encoder.

The architecture works similar to the single-encoding approach. However, here the data that is stored either in R_1 or R_2 are both produced by the Encoder. Further, signal SEL is directly taken from the input vector without any pre-processing. The latches shown in the dual-encoding architecture are introduced due to the same reasons as in the single-encoding architecture.

Example 2: In Fig. 5, we select three different input variables to partition the truth table based on the Shannon expansion. If we select x_2 as the partition variable, the output vector set will be $\{00, 10\}$ when $x_2 = 0$, and it will be $\{11, 10\}$ when $x_2 = 1$. Hence the number of different output vectors in both partitions is 2. However, if we select x_1

x_1	x_2	x_3	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

x_1

x_1	x_2	x_3	f_1	f_2
0	0	0	0	0
0	0	1	1	0
1	0	0	0	0
1	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	1	0	1	0
1	1	1	1	0

x_2

x_1	x_2	x_3	f_1	f_2
0	0	0	0	0
0	1	0	1	1
1	0	0	0	0
1	1	0	1	0
0	0	1	1	0
0	1	1	1	1
1	0	1	1	0
1	1	1	1	0

x_3

Figure 5. Partition Example.

(x_3) to partition the same table, the number of different output vectors for both partitions are 3 and 2 corresponding to $x_1 = 0$ and $x_1 = 1$ ($x_3 = 1$ and $x_3 = 0$), respectively.

Power reduction in Fig. 4 is obtained because of the reduced switching activity of registers and both decoders compared to the register of the original circuits and its combinational block. In this architecture, the partition variable is chosen first, and then encoding for minimal Hamming distance of the inputs to $R_1/Decoder_1$ and $R_2/Decoder_2$ is performed. The algorithm to perform these operations are described in the next section.

4 Synthesis of the bipartition dual-encoding architecture

The synthesis for the bipartition dual-encoding architecture contains two phases: given a circuit described in PLA format, we first bipartition the PLA into two sub-PLAs in terms of minimizing the number of different outputs. In the second phase, encoding is applied on both sub-PLAs in terms of minimizing Hamming distance.

4.1 Bipartition Algorithm

As bipartition algorithm we use a brute force approach by trying out each possible partition variable. This approach is acceptable as only n configurations must be analyzed.

In order to find the best suitable configuration each variable is selected as partition variable and the PLA is partitioned accordingly. Based on the partitioned PLA table the numbers of different output pattern are determined for partition 1 and partition 2. The pattern number are denoted as OP_1 and OP_2 in the following. Based on OP_1 and OP_2 the configuration is rated according to

$$W = \lceil \log_2(OP_1) \rceil + \lceil \log_2(OP_2) \rceil.$$

The configuration which gives a minimal rating W is selected as the final bipartitioning result.

4.2 Encoding Algorithm

From the original PLA table, two sub-PLAs PLA_1 and PLA_2 are built after bipartitioning: all lines in the original PLA that are associated with an input pattern having the partition variable set to 0 are moved to PLA_1 while the remaining lines form PLA_2.

Next, the number of different output patterns of both sub-PLAs are determined. The output pattern number of PLA_1 and PLA_2 are denoted as OPN_1 and OPN_2 in the following. Then, the output bit width of the encoder is set to

$$\max(\lceil \log_2 OPN_1 \rceil, \lceil \log_2 OPN_2 \rceil)$$

as at least $\lceil \log_2 OPN_1 \rceil$ ($\lceil \log_2 OPN_2 \rceil$) bits are required to encode the output pattern of PLA_1 (PLA_2).

The PLA with maximum number of different outputs (OPN) is denoted as PLA_x in the following. Finally, we adopt the heuristic algorithm introduced in [2] to encode PLA_x. The algorithm minimizes the Hamming distance of the output (encoded) pattern. The other PLA (denoted as PLA_y) is encoded using the output pattern of PLA_x as follows: the (not encoded) output pattern of both PLAs are sorted in increasing order of their occurrences. The sorted pattern are denoted as $sort(PLA_x)$ and $sort(PLA_y)$. Further, the encoded values are also sorted according to $sort(PLA_x)$. Finally, encoding of PLA_y is determined by assigning the leftmost pattern of $sort(PLA_y)$ to the leftmost pattern from the sorted encoding list, and so on. Assignment continues until all pattern from $sort(PLA_y)$ are processed.

Example 3: Assume the following truth table for the original combinational block:

input $x_0x_1x_2$	output $y_0y_1y_2$
000	000
001	000
010	001
011	001
100	111
101	111
110	111
111	010

If we choose x_0 as partition variable SEL then we get the following truth tables for PLA_1 and PLA_2:

PLA_1		
SEL	x_1x_2	$y_0y_1y_2$
0	00	000
0	01	000
0	10	001
0	11	001

PLA_2		
SEL	x_1x_2	$y_0y_1y_2$
1	00	111
1	01	111
1	10	111
1	11	010

Note that the partition variable is shown in a *separate* column. From the truth tables we determine the encoder output width to be 1 bit (each sub-PLAs use only two different output patterns). From PLA_1 and PLA_2 we determine the output frequency of each pattern:

PLA_1	
$y_0y_1y_2$	frequency
000	2
001	2

PLA_2	
$y_0y_1y_2$	frequency
111	3
010	1

As both PLAs are having the same number of different outputs (2) we randomly choose PLA_1 to be encoded first. Assume that pattern output 000 is encoded as 0 and 001 is encoded as 1. As a result, we get the following assignments:

PLA_1		
$y_0y_1y_2$	frequency	encoding
000	2	0
001	2	1

PLA_2		
$y_0y_1y_2$	frequency	encoding
111	3	0
010	1	1

Note that the encoding for PLA_2 is obtained by simply copying the *encoding* column of PLA_1.

Hence, the truth table for the encoder is

SEL	x_1x_2	encoded output
0	00	0
0	01	0
0	10	1
0	11	1
1	00	0
1	01	0
1	10	0
1	11	1

As a result, the truth tables for $Decoder_1$ and $Decoder_2$ are

$Decoder_1$	
encoded input	$y_0y_1y_2$
0	000
1	001

$Decoder_2$	
encoded input	$y_0y_1y_2$
0	111
1	010

5 Experimental Results

The bipartition algorithm has been implemented in C++ on a Pentium III desktop PC. We used SIS[3] to synthesize our partition results and estimated power dissipation by PowerMill. For our experiments, we assumed 5V supply voltage and a clock frequency of 20MHz. The benchmark circuits taken from LGSynth91 were used to demonstrate our algorithm. The rugged script of SIS was used to optimize the benchmarks. As the registers of the output part are unchanged in our architectures we did not take the effect of these registers into account. The area and power units are $128\mu m^2$ and μW in our experiments, respectively.

[3]SIS is a sequential circuit synthesis systems developed by U.C. Berkeley.

Table 1. Register power dissipations of the original circuit and the bipartition dual-encoding architecture

Circuits	Orig_Reg	Bi_dual	PF%
sao2	2133.3	269.1	87.4%
9sym	1905.0	166.5	91.3%
con1	1387.1	388.6	72.0%
misex1	1611.8	529.4	67.2%
rd53	1034.8	479.4	53.7%
rd73	1441.3	534.8	62.9%
rd84	1615.2	596.6	63.1%
sqrt8	1608.7	655.5	59.3%
xor5	992.2	211.6	78.7%
t481	3082.3	182.9	94.1%
Average			72.9%

Table 1 shows that the average pipelined registers power reduction is 72.9% (see the PF% column). The results are consistent with earlier discussion that claims the proposed method reduces register switching activities. Table 2 shows power and area numbers for the original circuits as well as for the bipartition dual-encoding architecture. The columns labeled with PR% and AI% represent the percentage of power reduction and area increment, respectively.

Table 3 compares the average area and power reduction of bipartition single-encoding to the bipartition dual-encoding architecture. The data of the single-encoding architecture is cited from [4]. The columns in this table have the same meaning as in Table 1 and Table 2. As shown in the table, bipartition dual-encoding architecture obtains the significant power saving in pipelined registers (PF%), however, the overall power saving is a little less than single-encoding architecture. This is due to the fact that the *Encoder* of dual-encoding architecture consumed more power than that of single-encoding. Nevertheless, the dual-encoding architecture obtains almost the same power saving as the single-encoding architecture while introducing significant less area overhead.

Table 2. Simulation result of original circuit and bipartition dual-encoding architectures

Circuits	Original		Bi_dual			
	Power	Area	Power	Area	PR%	AI%
sao2	3606.2	637	2284.2	647	36.7%	1.6%
9sym	4513.8	820	2294.9	531	49.2%	-35.2%
con1	1587.4	104	1336.5	251	15.8%	141.3%
mixex1	2157.6	340	1878.3	457	12.9%	34.4%
rd53	1511.4	215	1370.0	352	9.4%	63.7%
rd73	2319.8	370	2040.6	505	12.0%	36.5%
rd84	3235.5	577	2699.1	604	16.6%	4.7%
sqrt8	2490.0	393	2337.9	581	6.1%	47.8%
xor5	1218.8	149	851.3	139	30.2%	-6.7%
t481	3388.4	450	1016.5	225	70.0%	-50.0%
Average	2602.9	405.5	1810.9	429.5	30.4%	5.8%

Table 3. Average area and power comparison between single and dual encoding.

	bi_single	bi_dual
AI%	29.6%	5.8%
PF%	63.0%	72.9%
PR%	31.6%	30.4%

6 Conclusion

We have proposed a low power bipartition dual-encoding architecture in this paper. We bipartition a given circuit in terms of minimal different outputs and then encode both partitions with minimal Hamming distance. The results show that power dissipation of pipelined registers can be reduced by up to 94.1% and the overall power dissipation can be reduced by up to 70.0%. Compared to bipartition-single encoding architecture, the dual-encoding provides almost the same power saving while the area overhead is significantly reduced.

References

[1] M. Alidina, J. Monterio, S. Devadas, S. Devadas, A. Ghosh, and M. Papaefthymiou. Precomputation-Based Sequential Logic Optimization for Low Power. *IEEE Trans. VLSI Syst.*, 2(4):426–436, Dec. 1994.

[2] L. Benini and G. D. Micheli. State Assignment for Low Power Dissipation. *IEEE J. Solid-State Circuits*, 30(3):258–268, Mar. 1995.

[3] P.-H. Chen, S.-J. Ruan, K.-P. Wu, D.-X. Hu, F. Lai, and K.-L. Tsai. An Entropy-Based Algorithm to Reduce Area Overhead for Bipartition-Codec Architecture. In *Proceedings of IEEE Int'l. Symp. on Circuits and Systems*, pages V–49–V–52, 2001.

[4] I.-S. Choi and S.-Y. Hwang. Circuit Partition Algorithm for Low-Power Design Under Area Constraint Using Simulated Annealing. *IEE Proc. Circuit Devices Syst.*, 146(1):8–15, Feb. 1999.

[5] H. Kapadia, L. Benini, and G. D. Micheli. Reducing Switching Activity on Datapath Buses with Control-Signal Gating. *IEEE J. Solid-State Circuits*, 34(3):405–414, March 1999.

[6] J. Moterio, S. Devadas, and A. Ghosh. Retiming Sequential Circuits for Low Power. In *Proceedings of IEEE/ACM Int'l. Conf. on Computer Aided Design*, pages 398–402, 1993.

[7] M. Munch, B. Wurth, R.Mehra, J. Sproch, and N. Wehn. Automating RT-Level Operand Isolatoin to Minimize Power Consumption in Datapaths. In *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pages 624–631, 2000.

[8] K. Roy and S.Prasad. Syclop: Synthesis of CMOS Logic for Low Power Application. In *In proceeding, Int'l Conf. of Computer Design*, pages 464–467, 1992.

[9] S.-J. Ruan, J.-C. Lin, P.-H. Chen, F. Lai, K.-L. Tsai, and C.-W. Yu. An Effective Output-Oriented Algorithm for Low Power Multipartition Architecture. In *IEEE Int'l. Conf. on Electronics, Circuits and Systems*, pages 609–612, 2000.

[10] S.-J. Ruan, R.-J. Shang, F. Lai, S.-J. Chen, and X.-J. Huang. A Bipartition-Codec Architecture to Reduce Power in Pipelined Circuits. In *Proceedings of IEEE/ACM Int'l. Conf. on Computer Aided Design*, pages 84–89, Nov. 1999.

[11] C. V. Schimpfle, S. Simon, and J. A. Nossek. Optimal Placement of Registers in Data Paths for Low Power Design. In *Proceedings of IEEE Int'l. Symp. on Circuits and Systems*, pages 2160–2163, 1997.

[12] V. Tiwari, S. Malik, and P. Ashar. Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design. *IEEE Trans. Computer-Aided Design*, 17(10):1051–1060, Oct. 1998.

[13] W. Ye and M. J. Irwin. Power Analysis of Gated Pipeline Registers. In *Proceeding of Twelfth Annual IEEE International ASIC/SOC Conference*, pages 281–285, 1999.