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 " switchweight" 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

1063-6404/02 $17.00 © 2002 IEEE

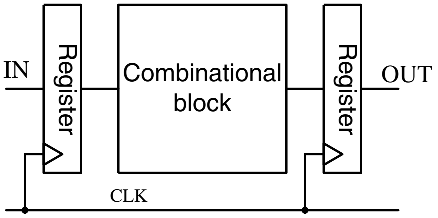

Figure 1. A combinational pipelined circuit

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.

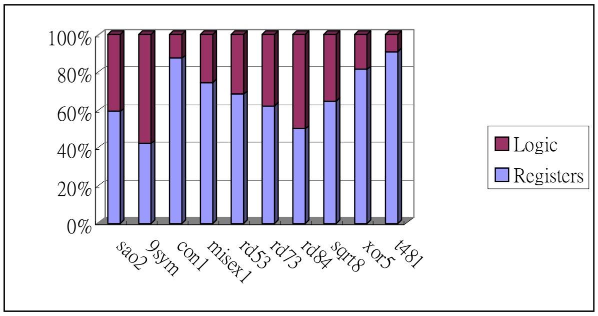

Figure 2. Power dissipation for several MCNC benchmarks

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]:

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

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.

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

[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.

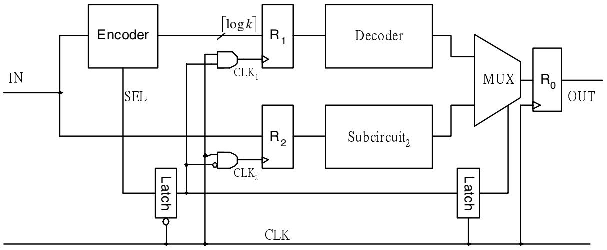

Figure 3. A bipartition single-encoding architecture.

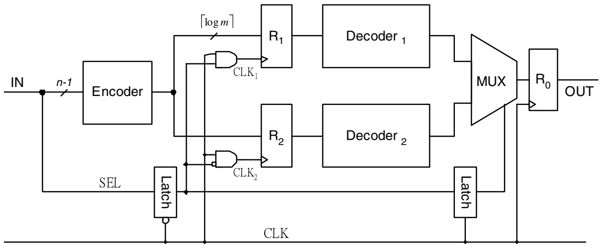

Figure 4. A bipartition dual-encoding architecture.

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	output
$x_0x_1x_2$	$y_0y_1y_2$
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.

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.

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

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.

1063-6404/02 $17.00 © 2002 **IEEE**