Bipartitioning and Encoding in Low-Power Pipelined Circuits

SHANQ-JANG RUAN
National Taiwan University of Science and Technology
KUN-LIN TSAI
National Taiwan University
EDWIN NAROSKA
University of Dortmund
and
FEIPEI LAI
National Taiwan Univerisity

In this article, we present a bipartition dual-encoding architecture for low-power pipelined circuits. We exploit the bipartition approach as well as encoding techniques to reduce power dissipation not only of combinational logic blocks but also of the pipeline registers. Based on Shannon expansion, we partition a given circuit into two subcircuits such that the number of different outputs of both subcircuits are reduced, and then encode the output of both subcircuits to minimize the Hamming distance for transitions with a high switching probability. We measure the benefits of four different combinational bipartitioning and encoding architectures for comparison. The transistor-level simulation results show that bipartition dual-encoding can effectively reduce power by 72.7% for the pipeline registers and 27.1% for the total power consumption on average. To the best of our knowledge, it is the first work that presents an in-depth study on bipartition and encoding techniques to optimize power for pipelined circuits.

Categories and Subject Descriptors: B.6.1 [**Logic Design**]: Design Styles—*Combinational logic*; B.6.3 [**Logic Design**]: Design Aids—*Automatic synthesis*; J.6 [**Computer-Aided Engineering**]: *Computer-aid design (CAD)*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Low-power design

This work was partially supported by NSC92-2218-E-011-066.
Authors' addresses: S.-J. Ruan, Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan 106; email: sjruan@et.ntust.edu.tw; K.-L. Tsai, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106; email: kunlin@orchid.ee.ntu.edu.tw; E. Naroska, Computer Engineering Institute, University of Dortmund, 44221 Dortmund, Germany; email: edwin@ds.e-technik.uni-dortmund.de; F. Lai, Department of Computer Science and Information Engineering & Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106; email: flai@cc.ee.ntu.edu.tw.

1. INTRODUCTION

In modern processor designs, pipelining is the most popular fashion to increase overall performance. Since Alidina et al. [1994] first applied precomputation on sequential logic to achieve low power, much work has been published on synthesis for low-power pipelined CMOS circuits. We categorize this previous work into three groups.

The first group covers approaches where pipeline registers are immovable. Techniques like precomputation [Alidina et al. 1994], gated pipeline registers [Ye and Irwin 1999; Kapadia et al. 1999], and guarded evaluation [Tiwari et al. 1998; Munch et al. 2000] belong to this category. Unfortunately, as for these approaches registers are immovable, further improvements are limited. Another limitation is that they require additional control logic.

In the second category, pipeline registers are movable. Approaches like retiming [Moterio et al. 1993] and repositioning of registers in datapaths [Schimpfle et al. 1997] belong to this category. However, no effort is made to modify the combinational logic blocks.

Approaches of the third category reduce the size of active registers and logic blocks using partition techniques [Choi and Hwang 1999]. Choi and Hwang [1999] partitioned the combinational logic block of a pipelined circuit into multiple subcircuits by recursively applying Shannon expansion with respect to the selected input variables. Furthermore, Ruan et al. [1999, 2001] showed that partitioning circuits into more than two sections does not always save power due to the overhead of duplicated input registers and output multiplexors.

Some preliminary work on combining techniques of bipartitioning and retiming has been done in Ruan et al. [1999] and Chen et al. [2001] , but the inability to extract the most active portion may make this approach inapplicable in the real world. In this article, we take advantage of bipartitioning and encoding techniques toward optimizing power consumption of pipelined circuits. As in Ruan et al. [1999] and Chen et al. [2001], we consider a pipeline architecture where combinational logic blocks are separated by edge-triggered registers that are driven by a single clock signal. We propose a bipartition dual-encoding architecture to decrease power consumption of pipelined CMOS designs. Our approach is based on the observation that the pipeline registers take a large fraction of total power dissipation for most of the circuits. Table I shows that in our experiments pipeline register account for 64.6% of the total power budget on average. In order to address this issue, we first bipartition a given circuit by using Shannon expansion to minimize the number of different outputs of both subcircuits [Micheli 1994]. Second, we encode both partitions to reduce the switching activities of the pipeline registers and logic blocks. To validate the results, we employ the accurate transistor-level power estimator EPIC PowerMill[1] to estimate power dissipation.

The rest of the article is organized as follows. In Section 2, we present bipartition and bipartition single-encoding architectures and discuss their characteristics. Further, our new bipartition dual-encoding architecture is

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

Table I. Power Dissipation of Registers for Several MCNC Benchmarks

Circuits	sao2	9sym	con1	misex1	rd53	rd73	rd84	sqrt8	xor5	t481	Ave.
Reg.%	59.2	42.2	87.4	74.7	68.5	62.1	49.9	64.6	81.4	91.0	64.6

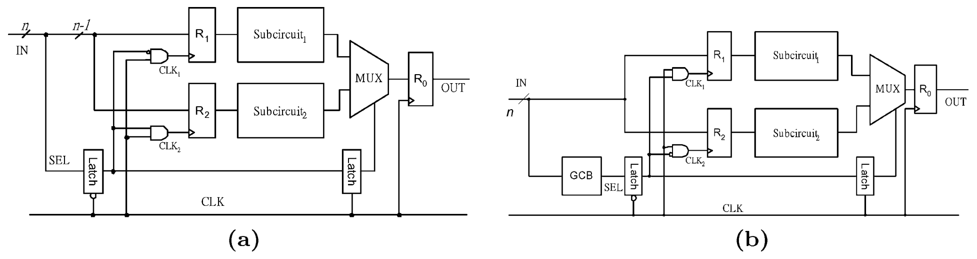


Fig. 1. First bipartition architecture (a) and bipartition architecture based on output extraction (b).

presented. The synthesis algorithms for the proposed architecture are presented in Section 3. The experimental results and conclusions are given in Sections 4 and 5.

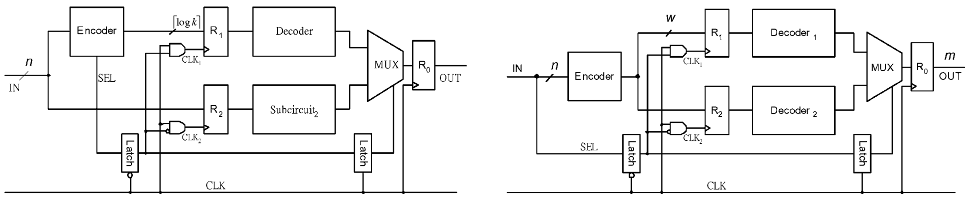
2. BIPARTITIONING AND ENCODING ARCHITECTURES

In Figure 1(a), a bipartition architecture based on Shannon expansion [Choi and Hwang 1999] is shown. Depending on the value of *SEL*, only one of the subcircuits is active while the other is disabled. Power saving is achieved if each of the two subcircuits consumes less power than a direct implementation. The disadvantage of this architecture is that duplicated registers always increase the area overhead and limit power saving.

Figure 1(b) shows the second bipartition architecture. Compared to Figure 1(a), the *SEL* signal is generated by *GCB* (global control block). The basic concept of this approach is to assign a few but frequently occurring outputs to form a (small) module $Subcircuit_1$ while the remaining (less frequently occurring) outputs are moved to $Subcircuit_2$ (see Figure 1(b)). Depending on the activity of the outputs, the architecture can have significant power reduction, even though the duplicated registers and the selection logic (GCB) may incur area and power overheads.

Starting from Figure 1(b), we now replace the highly active $Subcircuit_1$ with an encoder-decoder (codec) architecture to reduce the internal switching activity of R_1 and logic block (Figure 2(a)). Notice that *Encoder* not only encodes the *k* frequently occurring output pattern with minimal Hamming distance but also generates the selection signal (*SEL*). The interested reader may refer to Ruan et al. [1999] for a further analysis on this topic.

Based on the previous architectures, we propose a new bipartition dual-encoding approach for lowpower pipelined circuits. The main idea of this approach is to partition circuits using Shannon expansion for simplifying selection logic. Then we encode both subcircuits to reduce size as well as switching activity of registers. In Figure 2(b), the bipartition dual-encoding architecture



(a) (b)

Fig. 2. A bipartition single-encoding (a) and dual-encoding (b) architecture.

is shown. Only one of the input signals is selected as partition variable *SEL* but all input signals feed into the *Encoder*, which encodes the output with minimal register size and Hamming distance. Depending on *SEL*, either R_1 and $Decoder_1$ or R_2 and $Decoder_2$ are activated.

3. SYNTHESIS OF THE BIPARTITION DUAL-ENCODING ARCHITECTURE

3.1 Bipartition Algorithm

The choice of partition variable *SEL* is critical in the bipartition dual-encoding architecture. To find a suitable solution, we use a brute force approach to test and rate all (i.e., n, where n is the number of input pins) possible configurations. This approach is acceptable as n is usually small (less than 100).

In detail, each variable is selected as partition variable and the PLA is partitioned accordingly. Based on the partitioned PLA table, the number of different output pattern are determined for $partition_1$ and $partition_2$. These numbers are denoted as OP_1 and OP_2, respectively, in the following. Next, the configuration is rated according to $w = \max(\lceil \log_2(OP_1) \rceil, \lceil \log_2(OP_2) \rceil)$. Finally, the configuration which gives a minimal rating w is selected as the final bipartitioning result.

3.2 Encoding Algorithm

As total power dissipation mainly depends on the switching activity of the pipeline registers, we try to encode the output pattern so that the hamming distance between pattern with a high transition probability is minimal. Usually this will also have a positive effect on the power consumed by the combinational logic blocks $Decoder_1$ and $Decoder_2$.

The encoding problem consists of choosing codes for the outputs of both sub-PLAs that minimize the switching probability. As described in Section 3.1, two sub-PLAs, PLA_1 and PLA_2, are obtained after bipartitioning. The number of output patterns of PLA_1 and PLA_2 are denoted as OP_1 and OP_2, respectively. Therefore, the output bit width of the encoder is $\max(\lceil \log_2 OP_1 \rceil, \lceil \log_2 OP_2 \rceil)$. The *PLA* with the maximum number of different outputs is denoted as PLA_x in the following. Next, we adopt the heuristic algorithm introduced in Benini and Micheli [1995] to encode PLA_x. The other *PLA* (denoted as PLA_y) is encoded using the output pattern of PLA_x as follows: the (not encoded) output patterns of both *PLA*s are sorted in decreasing order of their occurrence frequency.

The sorted patterns 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 first pattern of *sort*(PLA_y) to the first pattern from the sorted encoding list, and so on.

Example. Consider an original combinational block with the following truth table:

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

If we choose x_0 as partition variable SEL, PLA_1 and PLA_2 become:

PLA_1

SEL ($= x_0$)	x_1x_2	$y_0y_1y_2$
0	00	000
0	01	000
0	10	001
0	11	001

PLA_2

SEL ($= x_0$)	x_1x_2	$y_0y_1y_2$
1	00	111
1	01	111
1	10	111
1	11	010

From the truth tables we determine the encoder output width to be 1 bit (each sub-PLA uses 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 *PLA*s 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 column for PLA_2 is obtained by simply copying the *encoding* column of PLA_1. Hence, the truth table for the encoder is

$x_0x_1x_2$	000	001	010	011	100	101	110	111
encoded outp.	0	0	1	1	0	0	0	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

Compared to an optimal output (state) assignment approach, the bipartitioning technique may deliver additional switching reduction. For example, assume that there are four outputs a, b, c, and d with probability $p_a = 0.7$, $p_b = p_c = p_d = 0.1$ (output transitions takes place randomly and independent from the current output value). Optimal assignment requires 2 output bits, for example, with assignments $a = 00$, $b = 01$, $c = 10$, and $d = 11$. As a result, on average 0.64 state bits will flip from clock cycle to clock cycle (note that a "transition" from d to d is also permitted).

Now assume that we bipartition the output space so that a and b are assigned to partition P_1 and c and d are assigned to partition P_2. P_1 as well as P_2 now can be encoded with 1 bit. We assume that a and c are assigned value 0, b and d are assigned 1. Note that we need a register bit to store which of the partitions is active. Hence, we actually have three register bits: one for P_1, one for P_2, and one to identify the active partition. As a result, the total number of bits flipping for all three registers will sum up to 0.595 on average. Compared to the previous result (0.64) the difference is due to the fact, that P_1 stores value a (probability 0.7) most of the time (i.e., the register stores 0). If there is a transition from state c or d (belongs to P_2) to a (belongs to P_1), then there will be a good chance that the register of P_1 already stores 0 ($= a$) and, hence, does not flip.

4. EXPERIMENTAL RESULTS

We implemented the algorithm and applied it to some MCNC benchmark circuits. We used the SIS[2] standard script *script.rugged* to obtain a multilevel implementation of *Encoder*, *Decoder*$_1$, and *Decoder*$_2$ for TSMC[3] 0.25-μm technology. The modules were then integrated by adding control elements latches, registers, AND gates, and multiplexers. The integrated implementations were simulated at transistor level (EPIC PowerMill) applying equally distributed random patterns to the inputs. Supply voltage and clock frequency were set to 2.5 V and 20 MHz. The area unit and power unit were μm^2 and μW, respectively, throughout the section. The power reduction rate and area increase rate were computed as $100(P_{orig} - P_{proposed})/P_{orig}$ and $100(A_{proposed} - A_{orig})/A_{orig}$, respectively, throughout the experiments.

Power dissipation of pipeline registers for original, bipartition architecture and bipartition dual-encoding architecture are named as Orig, Bipart, Bi_dual in Table II. The columns PF_B% and PF_Dual represent the power reduction of bipartition (based on Shannon expansion; Figure 1(a)) and bipartition dual-encoding architectures, respectively. Experiments showed that our bipartition algorithm dissipated 23.9% fewer power compared to the original architecture in pipeline registers. Further, we obtained a significant power reduction of 72.7% by using the bipartition dual-encoding architecture.

Table III presents the performance of our bipartition algorithm (Figure 1(a)). The "Original" columns show the power dissipation of combinational block "P_{block}" and total area "Area." In the "Bipartition architecture" columns, the power dissipation of modules *subcircuit*$_1$, *subcircuit*$_2$, and multiplexors (see Figure 1(a)) are denoted as "Subc$_1$," "Subc$_2$," and "Mux," respectively. The power dissipation of Clock covers both AND gates and latches. Finally, power improvement and area increase are labeled as "PR%" and "AI%," respectively. The

[2]SIS: A system for sequential circuit synthesis implemented by the Electronic Research Laboratory in the Department of EE and CS, University of California, Berkeley, on May 4, 1992.
[3]TSMC stands for Taiwan Semiconductor Manufacturing Company.

Table II. Register Power Dissipations of the Original Circuit and the Bipartition Dual-Encoding Architecture

Circuits	Orig	Bipart	Bi_dual	PF_B%	PF_Dual%
sao2	548.7	207.9	81.7	62.1	85.1
9sym	489.5	424.6	51.6	13.3	89.5
con1	348.8	276.0	115.4	20.9	66.9
misex1	407.3	254.4	152.2	37.6	62.6
rd53	263.9	200.5	134.5	24.0	49.0
rd73	368.0	247.4	152.4	32.8	58.6
rd84	408.6	350.1	171.4	14.3	58.0
sqrt8	407.9	355.1	182.9	12.9	55.1
xor5	248.5	192.7	64.2	22.5	74.2
t481	767.9	733.8	56.3	4.44	92.7
Average	452.9	324.2	116.3	23.9	72.7

Table III. Simulation Result of Original Circuit and Bipartition Architectures Based on Shannon Expansion

	Original		Bipartition Architecture							
Design	P_{block}	Area	$Subc_1$	$Subc_2$	Clock	Mux	Total	Area	PR%	AI%
sao2	406.0	2954.9	178.1	0.0	124.2	24.6	534.8	2995.2	44.0	1.4
9sym	694.3	3553.9	166.6	167.2	129.6	9.4	897.3	4389.1	24.2	23.5
con1	58.4	929.3	28.2	14.9	119.7	19.5	458.2	1799.0	−12.5	93.6
mixex1	153.7	1509.1	33.7	49.0	125.6	68.5	531.2	2580.5	5.3	71.0
rd53	135.9	956.2	43.4	56.0	114.8	34.0	448.7	1877.8	−12.2	96.4
rd73	246.9	1658.9	114.8	113.3	128.1	46.8	650.4	3162.2	−5.8	90.6
rd84	447.3	2494.1	242.4	122.4	131.8	56.5	903.2	4026.2	−5.5	61.4
sqrt8	254.7	1687.7	42.9	68.2	126.9	39.8	632.8	2603.5	4.9	54.3
xor5	70.6	691.2	27.7	14.7	109.6	16.2	360.9	1336.3	−13.1	93.3
t481	97.1	1791.4	36.2	58.1	166.0	11.0	1005.1	4039.7	−16.2	125.5
Average	256.8	1822.7	91.4	66.4	127.6	32.6	642.3	2881.0	5.9	58.1

results show that the bipartition architecture suffers from the power dissipated by Clock and Mux.

Table IV shows power and area numbers for the original circuits as well as for the bipartition dual-encoding architecture. The columns in this table have the same meaning as in Table III, except for the second column which shows total power dissipation of the original circuit. The columns "Enc," "Dec_1," and "Dec_2" stand for the power dissipated by the corresponding blocks Encoder, $Decoder_1$, and $Decoder_2$, respectively, from Figure 2(b). The results show that the power saving effects could be increased from 5.9% to 27.1% while the area overhead was reduced from 58.1% to 3.4% compared to the plain bipartition technique indicated in Table III.

Table V shows the average area increase and power reduction of bipartition based on Shannon expansion (Bipart-c [Ruan et al. 2001]), bipartition based on output clustering (Bipart-s), bipartition single-encoding (Bipart-single [Ruan et al. 2001]), and bipartition dual-encoding architecture (Bipart-dual) for comparison. The data of the first and third columns are cited from Ruan et al. [2001]. The columns "AI%," "PF%", and "PR%" represent the area increase,

Table IV. Simulation Result of Original Circuit and Bipartition Dual-Encoding Architectures

	Original		Bipartition dual-encoding architecture								
Design	Power	Area	Enc	Dec_1	Dec_2	Clock	Mux	Total Power	Area	PR%	AI%
sao2	954.7	2954.9	422.4	13.8	0.0	113.0	22.9	653.8	2849.3	31.5	−3.6
9sym	1183.8	3553.9	474.4	0.0	0.0	64.9	37.9	628.8	2327.0	46.9	−34.5
con1	407.2	929.3	105.0	5.3	5.6	108.3	23.3	362.9	1146.2	10.9	23.3
mixex1	561.0	1509.1	117.7	13.1	36.4	118.1	65.6	503.1	2016.0	10.3	33.6
rd53	399.8	956.2	72.7	15.8	9.8	106.5	29.6	369.0	1561.0	7.7	63.3
rd73	614.9	1659.0	230.9	17.5	13.7	111.3	34.1	559.9	2181.1	9.0	31.5
rd84	855.9	2494.1	400.7	8.4	14.2	114.6	36.0	745.3	2663.0	12.9	6.8
sqrt8	665.3	1687.7	253.7	19.7	40.8	77.6	79.3	654.2	2488.3	1.7	47.4
xor5	319.1	691.2	53.9	1.0	0.8	95.3	9.9	225.1	616.3	29.5	−10.8
t481	865.0	1791.4	113.2	0.0	0.0	98.3	10.1	278.0	996.4	67.9	−44.4
Average	682.7	1822.7	224.5	9.5	12.1	100.8	34.9	498.0	1884.5	27.1	3.4

Table V. Average Area and Power Comparison Between Single and Dual Encoding

	Bipart-c	Bipart-s	Bipart-single	Bipart-dual
AI%	44.4	58.1	29.6	3.4
PF%	26.0	23.9	63.0	72.7
PR%	9.7	5.9	31.6	27.1

power reduction of pipeline registers, and total power reduction, respectively. As shown in this table, bipartition dual-encoding architecture obtains the significant power saving in pipeline registers (PF%); however, the overall power saving is a little less than that of the single-encoding architecture. This is due to the fact that the *Encoder* of dual-encoding architecture consumes more power than the corresponding module of a single-encoding architecture. Nevertheless, the dual-encoding architecture obtains almost the same power saving as the single-encoding architecture while introducing significant less area overhead.

5. CONCLUSION AND DISCUSSION

In this article, we proposed a new bipartition dual-encoding architecture for low-power pipelined circuits. The proposed scheme exploits a bipartition approach as well as encoding techniques in a pipeline stage to reduce power dissipation not only of combinational logic blocks but also of the pipeline registers. We bipartition a given circuit described by PLA into two sub-PLAs such that the number of different outputs of both PLAs are minimal. We then encode the outputs of both sub-PLAs to minimize the Hamming distance of register values with high transition probability.

The differences between precomputation and bipartition, bipartition single-encoding and bipartition dual-encoding architectures can be summarized as follows: compared to precomputation architecture, the precomputation approach only disables some of the input pins to reduce the switching activity of the combinational logic. However, 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 and bipartition single/dual-encoding architectures separate the combinational logic to ensure they will not influence each other. In addition, bipartition signal/dual-encoding architectures also take the power dissipation of pipeline registers into account by applying a codec structure, which significantly reduces power dissipation.

Our accurate transistor-level simulations demonstrate the practical impact of the partition and encoding approaches in lowering the power of pipelined circuits. Up to 92.7% and 67.9%, for register and overall power reduction, respectively, and 72.7% and 27.1% on average for pipeline registers and total power consumption, respectively, can be obtained by the bipartition dual-encoding architecture.

REFERENCES

ALIDINA, M., MONTERIO, J., DEVADAS, S., DEVADAS, S., GHOSH, A., AND PAPAEFTHYMIOU, M. 1994. Precomputation-based sequential logic optimization for low power. *IEEE Trans. VLSI Syst. 2*, 4 (Dec.), 426–436.

BENINI, L. AND MICHELI, G. D. 1995. State assignment for low power dissipation. *IEEE J. Solid-State Circ. 30*, 3 (March), 258–268.

CHEN, P.-H., RUAN, S.-J., WU, K.-P., HU, D.-X., LAI, F., AND TSAI, K.-L. 2001. An Entropy-based algorithm to reduce area overhead for bipartition-codec architecture. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. V-49–V-52.

CHOI, I.-S. AND HWANG, S.-Y. 1999. Circuit partition algorithm for low-power design under area constraint using simulated annealing. *IEE Proc. Circ. Dev. Syst. 146*, 1 (Feb.), 8–15.

KAPADIA, H., BENINI, L., AND MICHELI, G. D. 1999. Reducing switching activity on datapath buses with control-signal gating. *IEEE J. Solid-State Circ. 34*, 3 (March), 405–414.

MICHELI, G. D. 1994. *Synthesis and optimization of digital circuits*. McGraw-Hill, New York, NY.

MOTERIO, J., DEVADAS, S., AND GHOSH, A. 1993. Retiming sequential circuits for low power. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*. 398–402.

MUNCH, M., WURTH, B., R.MEHRA, SPROCH, J., AND WEHN, N. 2000. Automating RT-level operand isolatoin to minimize power consumption in datapaths. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. 624–631.

RUAN, S.-J., SHANG, R.-J., LAI, F., CHEN, S.-J., AND HUANG, X.-J. 1999. A bipartition-codec architecture to reduce power in pipelined circuits. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*. 84–89.

RUAN, S.-J., SHANG, R.-J., LAI, F., AND TSAI, K.-L. 2001. A bipartition-codec architecture to reduce power in pipelined circuits. *IEEE Trans. Comput.-Aided Des. 20*, 2, 343–348.

SCHIMPFLE, C. V., SIMON, S., AND NOSSEK, J. A. 1997. Optimal placement of registers in data paths for low power design. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. 2160–2163.

TIWARI, V., MALIK, S., AND ASHAR, P. 1998. Guarded evaluation: Pushing power management to logic synthesis/design. *IEEE Trans. Comput.-Aided Des. 17*, 10 (Oct.), 1051–1060.

YE, W. AND IRWIN, M. J. 1999. Power analysis of gated pipeline registers. In *Proceedings of the Twelfth Annual IEEE International ASIC/SOC Conference*. 281–285.

Received January 2003; revised August 2003; accepted March 2004