VLSI DESIGN
2001, Vol. 12, No. 3, pp. 399–406
Reprints available directly from the publisher
Photocopying permitted by license only

On Mixed PTL/Static Logic for Low-power and High-speed Circuits

GEUN RAE CHO* and TOM CHEN†

Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523

(Received 20 June 2000; In final form 3 August 2000)

We present more evidence in a 0.25 μm CMOS technology that the pass-transistor logic (PTL) structure that mixes conventional PTL structure with static logic gates can achieve better performance and lower power consumption compared to conventional PTL structure. The goal is to use the static gates to perform both logic functions as well as buffering. Our experimental results demonstrate that the proposed mixed PTL structure beats pure static structure and conventional PTL in 9 out of 15 test cases for either delay or power consumption or both in a 0.25 μm CMOS process. The average delay, power consumption, and power-delay product of the proposed structure for 15 test cases are 10% to 20% better of than the pure static implementations and up to 50% better than the conventional PTL implementations.

Keywords: VLSI; Mixed PTL/Static logic; PTL logic; Low-power; High-speed; Pass-transistor logic

1. INTRODUCTION

With power becoming more and more a limiting factor for system performance, demand for low power circuits without sacrificing circuit performance is evident and will continue to be the focus of low-power high-performance circuit design community.

Conventional static CMOS logic gates have been widely used in ASIC design today mainly due to the advantages of easy synthesis and a well developed synthesis and analysis environment. Pass-transistor logic (PTL), which was first proposed in [1, 2], has attracted a lot of attention early on. PTL's advantage as an alternative low power circuit design approach in 0.5 μm and 0.3 μm CMOS technology has been well documented [3, 4]. However, PTL's advantage in deep-submicron CMOS technology is not well understood yet. There are speculations [3] that with aggressive scaling in transistor critical dimension and supply voltage, the advantages of PTL over conventional static CMOS gates will gradually diminish. This paper presents results in a 0.25 μm CMOS process

*Tel.: 970-491-7999, Fax: 970-491-2249, e-mail: geunc@engr.colostate.edu
†Corresponding author. Tel.: 970-491-6574, Fax: 970-491-2249, e-mail: chen@engr.colostate.edu

that show PTL circuits, when mixed with static logic gates, offer up to 50% power-delay benefits compared to conventional PTL. The adoption of PTL is further exacerbated by the lack of synthesis environment to support the use of PTL gates. The well known method of synthesize PTL circuit from a set of logic equations was proposed in [5]. The use of binary decision diagrams (BDDs) [6–8] made the synthesis process easy and straightforward. To minimize delay, a buffer is inserted every fixed number of transistors in series in the PTL chain. The process of buffer insertion can also be viewed as the process of BDD decomposition as suggested in [9], where buffers are inserted at the root of each decomposed BDD. For complex logic functions, the number of buffers to be inserted in a PTL circuit can be significant. One way to alleviate this problem is to use static logic gates acting as buffer as well as performing logic functions. Mixed PTL with static logic gates has been proposed in [10], where only one circuit example is used to show the benefit of mixed PTL/Static logic structure in a 0.35 μm CMOS process.

This paper presents more evidence in a deep-submicron CMOS process to further validate the benefit of mixed PTL circuits. By eliminating explicit buffering between PTL stages, mixed PTL circuits can further improve the low power nature of PTL circuits without sacrificing performance. Our experimental results show that our proposed structure beats pure static structure and conventional PTL in 9 out of 15 test cases for either delay or power consumption or both in a 0.25 μm CMOS process. The average delay, power consumption, and power-delay product of the proposed structure for 15 test cases are 10% to 20% better of than the pure static implementations and up to 50% better than the conventional PTL implementations.

The remaining part of this paper consists of three sections. We will review some basic issues related to PTL and propose our mix PTL structure in Section 2. In Section 3, a set of experimental results is presented in detail. The concluding remarks are given in Section 4.

2. MIXED PTL/STATIC CIRCUITS

2.1. Properties of Pass-transistor Chains

The average power, P_{av}, consumed by a CMOS gate is given by

$$P_{av} = V_{dd}^2 \cdot f \cdot C_L \cdot A_i \tag{1}$$

where V_{dd} is the supply voltage, f is the operating frequency, C_L is the total load capacitance driven by the gate, and A_i is the switching activity of the gate. V_{dd}, f, and A_i are fixed. Decreasing C_L will have a positive impact on power consumption. Since the input capacitance of a PTL gate is smaller than the input capacitance of a static CMOS gate, replacing CMOS gates with PTL gate should result in a reduction in the power consumed.

We cannot arbitrarily connect PTL gates together, however, since the delay through pass-transistors is a quadratic function of the number of transistors in series. The delay in PTL can be estimated as $t_d = \tau_n \cdot N^2$ where τ_n and N are the time constant and the number of transistors in series, respectively [11]. This quadratic dependence can be seen in Figure 1 where the results from an HSPICE simulation of a 0.25 μm process are shown. The delay is plotted *versus* the number of nMOS transistors in series in Figure 1. Figure 2 plots the percentage of increase in delay over one pass-transistor against the number of transistors in series. These graphs show that the delay is unacceptably high when the number of pass-transistors is three or more. Therefore, we limited ourselves to a maximum of two pass-transistors in series without an intervening static gate for this process.

Another issue with pass-transistors is that they do not have full output voltage swing. The high output voltage is limited to $V_{dd} - V_{tn}$ for an nMOS transistor chain where V_{tn} is the nMOS threshold voltage. We had hoped to use this limited voltage swing to reduce the dynamic power consumption. However, this output voltage was not high enough to completely turn off the pMOS transistors in the

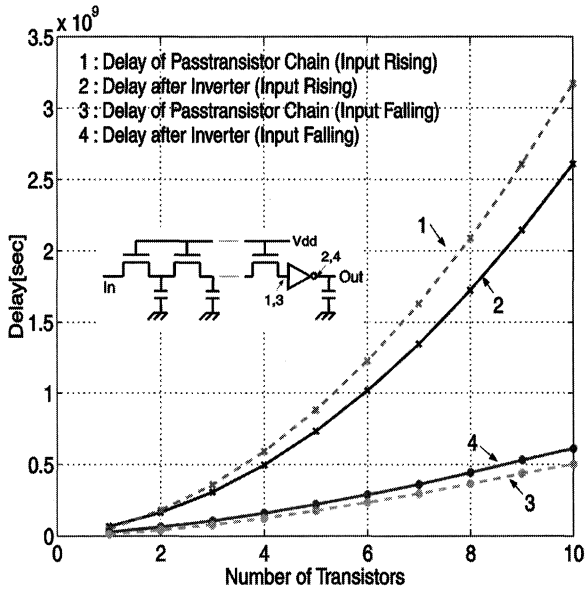

FIGURE 1 Relationship between the # of pass-transistors in a chain and delay (0.25 μm CMOS technology).

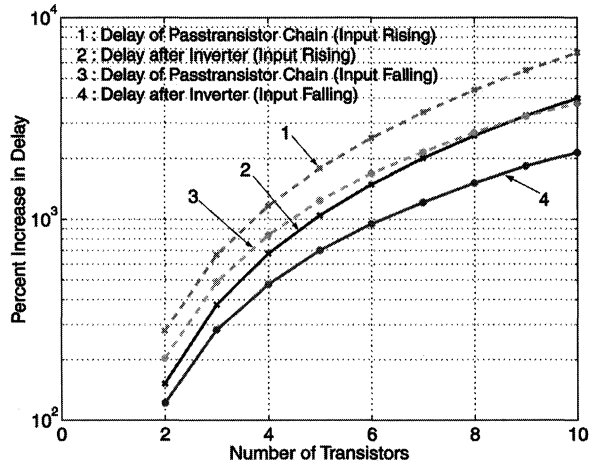

FIGURE 2 Relationship between the # of pass-transistors in a chain and percentage increase in delay (0.25 μm CMOS technology).

following static gates, and, therefore, the static current became unacceptably high. To fix this problem, an inverter and a pull-up pMOS transistor had to be added at the output of each PTL cell to restore the swing all the way to V_{dd}. The size of inverter and pull-up pMOS was chosen in such a way that the loading effect at the cell output is minimized. This arrangement is shown in the inset of Figure 5.

2.2. Conventional Static and PTL Mapping

One of our goals is to compare our proposed mixed PTL structure with the conventional PTL structure. To map a given set of Boolean functions to a conventional PTL circuit, we followed a method similar to the one described in [5]. The logic equations are first converted to a reduced and ordered BDD. This BDD is then converted to a PTL circuit simply by replacing each node with two nMOS pass transistors in a Y-shaped pattern as shown in the inset in Figure 3. This is a straight forward method, and it assures that only one path is on at a time. The final step is to add buffers after every two pass-transistors in series for the reasons outlined in Section 2.1. The buffers use pull-up transistors to restore full swing at the output of the pass-transistors.

As an example of a conventional PTL circuit, the Boolean equation

$$f = y_0' s_2' s_1' x_2' + s_1 s_0 x_2 x_1 + s_2' x_2 x_1 x_0 + y_0 s_2' s_1 s_0 x_2 \quad (2)$$

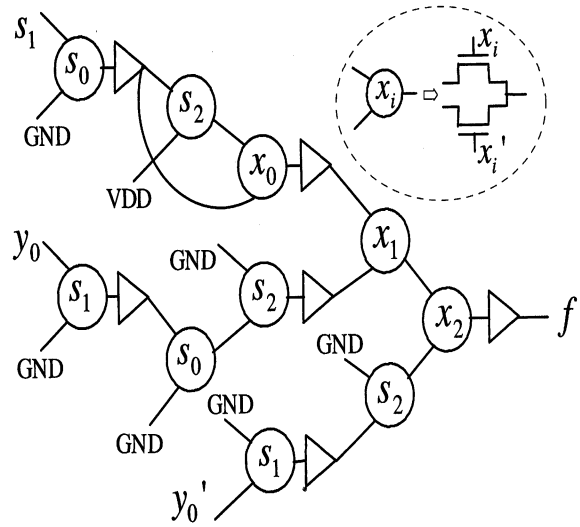

FIGURE 3 Conventional PTL logic implementation.

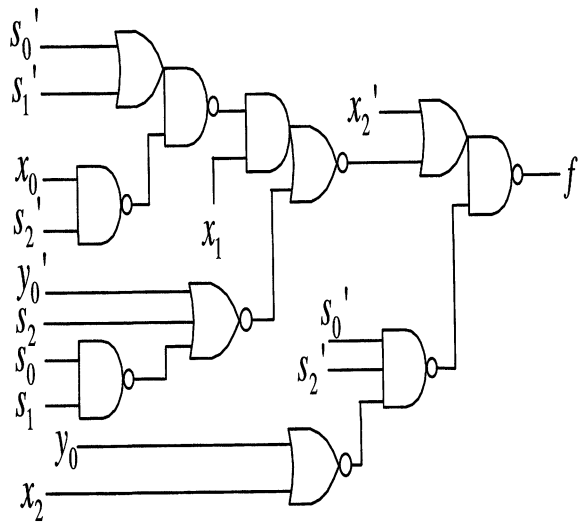

FIGURE 4 Pure Static logic implementation.

was mapped using the above process. The results are shown in Figure 3.

For the pure static mapping, we used the SIS program as described in [12, 13]. Since technology mapping is an NP-hard problem [14, 15], SIS uses a set of heuristic algorithms. The first algorithm decomposes the given Boolean function into a graph containing only simple gates, such as 2-input NAND's and inverters. In the second step, this large graph is then partitioned into smaller pieces called subject graphs [16, 17] to make the third step tractable. The third step is called covering [16–18] and consists of trying to match portions of the subject graph to graphs created from the cell library. The cell library used by SIS contains only static CMOS gates. In the final step, the algorithm chooses which library cells to use, usually based upon some sort of timing constraint.

SIS was used to implement the function f given in (2). The result is shown in Figure 4.

2.3. PTL/Static Hybrid Mapping

Our approach to the mixed PTL/Static technology mapping is similar to the method as described in SIS [12, 13]. The mapping begin by decomposing the Boolean functions into a graph containing only simple gates called base functions: 2-input AND's, 2-input OR's, and inverters in our case. The primary purpose of this step is to express all local functions as simple functions and is required to ensure that every node is covered by at least one library cell. In order to ensure the existence of a solution the library have to include the base functions.

During partitioning, the large graph is broken into subject graphs. Since the technology mapping problem is NP-complete in nature, we can reduce the problem size and the problem becomes tractable by breaking the graph into cones. We have chosen multiple fanout points as the boundary of the subject graphs.

We used a similar covering method as that in SIS, but differs from the SIS method in that three PTL gates are added to the library of static gates and the static gate is used to perform both logic functions and buffers between PTL gates. These PTL gates have a maximum of 2 pass-transistors in series. Figure 5 shows the schematic of these three cells. With the addition of the PTL gates, some new rules must be added to the conventional covering scheme to generate mapped circuit after finding possible matches at every node in the subject graph. First, all inputs to a PTL cell must be from static cells. Similarly, all outputs from a

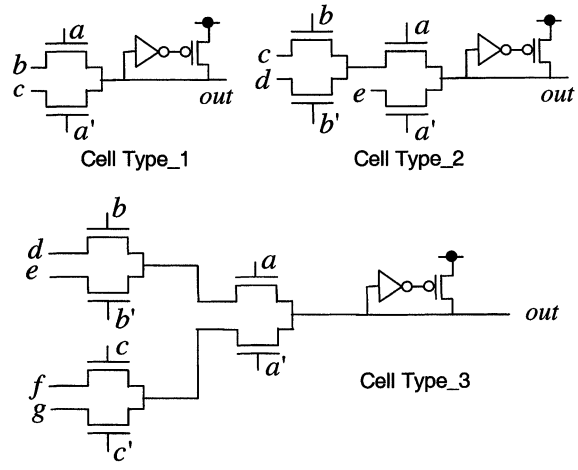

FIGURE 5 Pass-transistor logic cells.

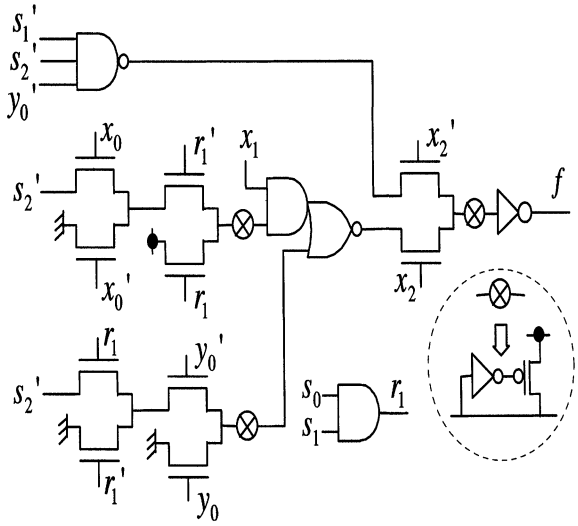

FIGURE 6 Mixed PTL/Static logic implementation.

PTL cell must be to static cells. These rules assure that two PTL cells are never connected in series, and thus no more than two pass-transistors are in series at any point in the circuit. This has to be done to keep the delay at acceptable levels as discussed in Section 2.1.

The details of the mapping process is beyond the scope of this paper. The mapping algorithm will be presented in a separate paper at a later time. A sample of our hybrid approach is shown in Figure 6 where the function f given in (2) is implemented.

3. EXPERIMENTAL RESULTS

We implemented three circuits from a microprocessor design in each of the three logic styles, conventional static, Conventional PTL, and mixed PTL/static. The characteristics of three circuits are summarized in Table I. These three circuits are implemented using a commercial 0.25 μm CMOS process with $V_{dd}=2.5$ V. For each of the three circuits, we randomly chose five paths to compare delay, power consumption (under different input setup to activate the given path), and power-delay product. The transistor size in all the circuits were determined such that rise and fall time of all the signal nodes in the circuits are in the range between 300 ps and 400 ps.

TABLE I Benchmark circuits

	No. of PIs	No. of POs	No. of Gates
Circuit 1	7	13	52
Circuit 2	32	16	107
Circuit 3	93	53	172

The HSPICE simulation results are shown in Table II, where the paths numbered 1 through 5 are from circuit 1, 6 through 10 are from circuit 2, and 11 through 15 are from circuit 3. This was the original process we designed our cells for, and the results are encouraging. In 9 out of 15 paths (path # 1, 6, 7, 8, 11, 12, 13, 14, 15), our proposed scheme outperformed the conventional static logic and the conventional PTL in either delay or power consumption. Out of these 9 paths, 6 paths outperformed both delay and power consumption. In 10 out of 15 paths (path # 1, 6, 7, 8, 9, 11, 12, 13, 14, 15), our scheme outperformed the conventional static logic and the conventional PTL in power-delay product. On average, our proposed scheme has a 30 percent improvement in delay while consuming approximately half of the power when comparing to the conventional PTL. More modest, but still substantial, gains are seen when comparing our approach to the static CMOS implementation, where there is approximately a ten percent savings in both delay and power. For power-delay product (PDP), our approach yields a 17 percent improvement over the static CMOS method.

In addition, our experimental results show that, at 0.25 μm, the advantages of conventional PTL over conventional static are disappearing. In fact, the conventional PTL scheme is significantly outperformed by the conventional static logic in all three categories of delay, power consumption, and power-delay product. It is our belief that explicit use of buffering between PTL stages in the conventional PTL added significant delay and additional power consumption to the conventional PTL circuits.

TABLE II HSPICE simulation results using 0.25 μm, $V_{dd}=2.5$ V

	Delay(sec)			Power(W)			PDP		
Path	Static	Conv. PTL	Proposed	Static	Conv. PTL	Proposed	Static	Conv. PTL	Proposed
1	4.83E−10	4.22E−10	4.86E−10	5.08E−05	8.72E−05	4.71E−05	2.45E−14	3.67E−14	2.29E−14
2	5.82E−10	5.07E−10	6.64E−10	1.04E−04	6.81E−05	8.52E−05	6.02E−14	3.45E−14	5.66E−14
3	4.95E−10	5.67E−10	6.20E−10	8.29E−05	6.07E−05	6.74E−05	4.11E−14	3.44E−14	4.18E−14
4	3.64E−10	4.27E−10	4.09E−10	3.93E−05	3.98E−05	3.95E−05	1.43E−14	1.70E−14	1.62E−14
5	2.82E−10	2.55E−10	3.70E−10	1.11E−04	7.67E−05	9.48E−05	3.14E−14	1.96E−14	3.51E−14
6	4.92E−10	6.84E−10	4.27E−10	5.48E−05	8.21E−05	5.27E−05	2.70E−14	5.62E−14	2.25E−14
7	5.56E−10	5.45E−10	3.90E−10	6.07E−05	6.77E−05	4.70E−05	3.38E−14	3.69E−14	1.83E−14
8	2.65E−10	2.50E−10	1.87E−10	4.77E−05	9.89E−05	3.88E−05	1.26E−14	2.47E−14	7.25E−15
9	3.90E−10	2.13E−10	2.16E−10	8.78E−05	1.73E−04	1.06E−04	3.42E−14	3.68E−14	2.29E−14
10	3.20E−10	6.33E−10	3.21E−10	3.97E−05	1.61E−04	4.18E−05	1.27E−14	1.02E−13	1.34E−14
11	6.85E−10	1.19E−09	5.73E−10	8.99E−05	3.01E−04	7.27E−05	6.15E−14	3.57E−13	4.16E−14
12	6.16E−10	1.15E−09	6.25E−10	8.95E−05	2.92E−04	7.92E−05	5.51E−14	3.34E−13	4.95E−14
13	5.47E−10	8.94E−10	4.63E−10	4.80E−05	1.36E−04	5.33E−05	2.62E−14	1.21E−13	2.47E−14
14	6.14E−10	5.62E−10	4.28E−10	6.17E−05	1.29E−04	5.28E−05	3.78E−14	7.26E−14	2.26E−14
15	4.30E−10	9.82E−10	3.55E−10	3.92E−05	8.12E−05	3.24E−05	1.69E−14	7.97E−14	1.15E−14
Avg.	4.75E−10	6.18E−10	4.36E−10	6.71E−05	1.24E−04	6.07E−0.5	3.18E−14	7.66E−14	2.65E−14

4. CONCLUSIONS

We presented a new pass-transistor scheme in which conventional PTL stages are mixed with static logic gates. The static logic gates not only act as buffers between PTL stages, but also perform part of the logic functions of the given block. Comparing to the conventional PTL, the elimination of explicit buffering between PTL stages significantly improved both circuit performance and power consumption. The experimental results confirms that, by using proposed scheme, over 10% improvement in delay and power consumption over the conventional static logic can be achieved, and up to 50% improvement in power consumption and up to 30% in delay over the conventional PTL can be achieved. The proposed scheme requires a new way the technology mapping is done. The problem is NP-complete in nature. We are continuing our effort to formalize the process of mapping a given logic function using the mixed PTL and static logic gates.

References

[1] Mead, C. A. and Conway, L. (1980). *Introduction to VLSI Systems*, Reading, Mass.:Addison-Wesley.

[2] Radhakrishnan, D. and Whitaker, S. R., "Formal Design Procedures for Pass Transistor Switching Circuits", *IEEE Journal of Solid-State Circuits*, **SC-20**, 531–536, April, 1985.

[3] Munteanu, M., Ivey, P. A., Seed, L., Psilogeorgopoulos, M., Powell, N. and Bogdan, I. (1999). "Single Ended Pass-Transistor Logic", In: *VLSI Systems on a Chip (Kluwer Academic Publisher)*, pp. 206–217.

[4] Yano, K., Yamanaka, T., Nishida, T., Saito, M., Shimohigashi, K. and Shimuzu, A., "A 3.8 ns CMOS 16 × 16-b Multiplier Using Complementary Pass-Transistor Logic", *IEEE Journal of Solid-State Circuits*, **25**, 388–394, April, 1990.

[5] Yano, K., Sasaki, Y., Rikino, K. and Seki, K., "Top-Down Pass-Transistor Logic Design", *IEEE Journal of Solid-State Circuits*, **31**, 792–803, June, 1996.

[6] Akers, S. B., "Binary Decision Diagrams", *IEEE Trans. on Computers*, **C-27**, 509–516, June, 1978.

[7] Bryant, R. E., "Graph-Based Algorithms for Boolean Function Manipulation", *IEEE Trans. on Computers*, **C-35**, 677–691, Aug., 1986.

[8] Drechsler, R. and Becker, B. (1998). *Binary Decision Diagrams: Theory and Implementation*, Kluwer Acedemic Publishers.

[9] Buch, P., Narayan, A., Newton, A. R. and Sangiovanni-Vincentelli, A. (1997). "Logic synthesis for large pass transistor circuits", *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*.

[10] Yamashita, S., Yano, K., Sasaki, Y., Akita, Y., Chikata, H., Rikino, K. and Seki, K. (1997). "Pass-transistor/cmos collaborated logic: the best of both worlds", In: *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 31–32.

[11] Weste, N. H. E. (1993). *Principles of CMOS VLSI Design: A System Perspective*, Addison Wesley, 2nd edn.

[12] Sentovich, E., Singh, K., Moon, C., Savoj, H., Brayton, R. and Sangiovanni-Vincentilli, A. (1992). "Sequential circuit design using synthesis and optimization", In: *IEEE Int. Conf. Comput. Design*.

[13] Sentovich, E. M., Singh, K. J., Lavagno, L., Moon, C., Murgai, R., Saldanha, A., Savoj, H., Stephan, P. R., Brayton, R. K. and Sangiovanni-Vincentelli, A. (1992). *SIS: A System for Sequential Circuit Synthesis*, Memorandom No. UCB/ERL M92/41, University of California, Berkeley, CA 94720.

[14] Hachetl, G. D. and Somenzi, F. (1996). *Logic Syntesis and Verification Algorithms*, Kluwer Academic Publisher.

[15] Garey, M. R. and Johnson, D. S. (1979). *COMPUTERS AND INTRACTABILITY: A Guide to the Theory of NP-Completeness*, W. H. FREEMAN AND COMPANY.

[16] Rudell, R. L. (1989). *Logic Synthesis in VLSI Design, Ph.D. Dissertation*, University of California, Berkeley, CA 94720.

[17] Micheli, G. D. (1994). *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, Inc.

[18] Mailhot, F. and Micheli, G. D., "Algorithms for Technology Mapping Based on Binary Decision Diagrams and Boolean Operations", *IEEE Trans. on Computer-Aided Design of Integrated Circuit and Systems*, **12**, 599–620, May, 1993.

Authors' Biographies

Geun Rae Cho worked for Semiconductor Research and Development Center at Samsung Electronics, Kiheung, KyungKi-Do, Korea. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering at Colorado State University. His research interests include VLSI design and CAD methodology, Logic synthesis for low-power applications, and Computer Architecture.

Tom Chen received the B.Sc. degree in Electronic Engineering from Shanghai Jiao-Tong University, China, and the Ph.D. degree in Electrical Engineering from the University of Edinburgh, Scotland, UK, in 1982 and 1987, respectively. From 1987 to 1989 Dr. Chen was with Philips Semiconductors in Hamburg, Germany. From 1989 to

1990 he was with the department of Electrical and Computer Engineering, New Jersey Institute of Technology. He is currently an associate professor in the Department of Electrical and Computer Engineering at Colorado State University. Dr. Chen's research interests are in the areas of VLSI design, VLSI CAD methodology, and Low-power circuit design.