Brief Papers

A New Family of Semidynamic and Dynamic Flip-Flops with Embedded Logic for High-Performance Processors

Fabian Klass, Chaim Amir, Ashutosh Das, Kathirgamar Aingaran, Cindy Truong, Richard Wang, Anup Mehta, Ray Heald, and Gin Yee

Abstract— In an attempt to reduce the pipeline overhead, a new family of edge-triggered flip-flops has been developed. The flip-flops belong to a class of semidynamic and dynamic circuits that can interface to both static and dynamic circuits. The main features of the basic design are short latency, small clock load, small area, and a single-phase clock scheme. Furthermore, the flip-flop family has the capability of easily incorporating logic functions with a small delay penalty. This feature greatly reduces the pipeline overhead, since each flip-flop can be viewed as a special logic gate that serves as a synchronization element as well. The flip-flop family presented in this paper has played an integral role in meeting the cycle-time goal of the microprocessor reported in [1].

Index Terms—Clocking, CMOS digital integrated circuits, flip-flops, microprocessors, pipeline processing, pulsed latches.

Fig. 1. Single-phase pulsed flip-flop: (a) concept and (b) possible implementation.

I. INTRODUCTION

THE continual push for higher clock rates and higher performance has led microprocessor designers in recent years to build superpipelined machines with multiple functional units that can execute operations concurrently. High clock rates in these machines are often achieved with fine granularity pipelining, for which there are relatively few levels of logic per pipeline stage. One direct consequence of this design trend is that the pipeline overhead is becoming more significant. This pipeline overhead is primarily due to the latency of the flip-flop or latch used and the clock skew of the system. While the clock skew varies and in some cases can be used constructively (e.g., routing the clock in the same direction of the data), the latency of the flip-flops cannot be hidden. As an example, assuming that a flip-flop latency is three gate delays and that the clock cycle in a state-of-the-art, high-speed microprocessor is 20 gate delays, the flip-flop overhead amounts to 15% of the cycle time. This is a substantial penalty that degrades the overall performance of the system, since no useful logic operation is performed on the data when it is being latched. Another consequence of the aforementioned trend is that the number of flip-flops in the system has increased dramatically. From a few thousand flip-flops in early designs, several tens of thousands of flip-flops is not an uncommon number in current designs. Clearly, an efficient flip-flop design, where the tradeoffs among
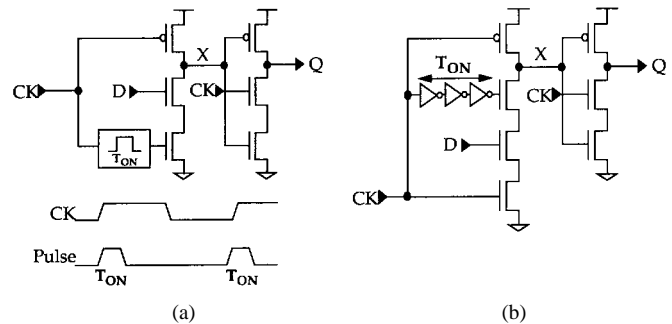
speed, area, and power are well balanced, is of fundamental importance.

In an attempt to reduce the pipeline overhead, a new family of edge-triggered flip-flops has been developed. The flip-flops belong to a class of semidynamic and dynamic circuits that can interface to both static and dynamic logic [2]. The term *semidynamic* is used here to denote circuits that internally have a precharge and evaluation phase, similar to dynamic gates. The main features of the basic design are short latency, small clock load, small area, and a single-phase clock scheme. Furthermore, this flip-flop family has the capability of easily incorporating logic functions with a small delay penalty. This feature greatly reduces the pipeline overhead, since each flip-flop can be viewed as a special logic gate that serves as a synchronization element as well. Taken together, these features make the flip-flop family presented in this paper well suited for high-performance microprocessor design.

II. A SINGLE-PHASE PULSED FLIP-FLOP

One of the primary requirements of a flip-flop for high-speed digital design, besides short latency, is to have a simple and robust clocking scheme. A family of static and dynamic latches with such characteristics is true single-phase clocking (TSPC) [3], [4]. TSPC latches can be combined in several different ways to implement edge-triggered flip-flops. While their single clock phase is advantageous, a drawback of TSPC flip-flops is long latency. A way to reduce latency is to clock a single transparent latch with a very narrow pulse, as illustrated in Fig. 1 (see [6]). Notice that in this implementation, only one device needs to be pulsed. For a sufficiently narrow pulse
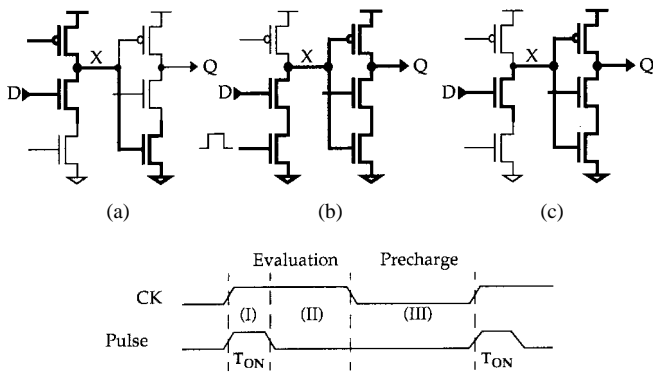
Fig. 2. Operation of a pulsed flip-flop: (a) precharge-III, (b) evaluation-I, and (c) evaluation-II.

Fig. 3. Semidynamic edge-triggered flip-flop.

width $T_{\rm ON}$, the latch behaves as an edge-triggered flip-flop. The operation of the new structure is illustrated in Fig. 2. When clock CK is low, the flip-flop is in the precharge phase. Node X is precharged to the level of the power supply, and node Q holds its previous value [see Fig. 2(a)]. On the rising edge of the clock, the flip-flop enters the *evaluation phase*. Here, two periods are distinguished. In the first period, the pulse is active and the circuit is in the *sampling* (or transparent) mode. The value of output Q is determined by the value of input D [see Fig. 2(b)]. Once internal node X is discharged, due to its precharge nature, it will stay low until the next clock cycle. In the second period, the pulse is inactive. The sampling of D is disabled, so X and Q will retain the values they acquired during the sampling period [see Fig. 2(c)]. Notice that any subsequent change at D after the sampling period will have no effect on Q.

If the pulse in Fig. 1(a) were to be generated externally, using a pulse generator [5], [6], the circuit would suffer from charge sharing. A way to prevent this is to generate the pulse locally, as depicted in Fig. 1(b) [12]. Besides avoiding charge sharing, this local pulse generation allows better control of the pulse width, so that a very narrow effective pulse can be produced. A narrow pulse is advantageous because it can reduce potential race-through problems and also improve the noise sensitivity of the circuit. On the other hand, the external generation and distribution of a narrow pulse is very hard in practice. The costs for the local generation are a slight speed penalty, due to the presence of one extra series device, and a small area penalty.

III. A Semidynamic Flip-Flop

While the structure of Fig. 1(b) is faster than TSPC, it still shares some of its shortcomings. First, internal node X is truly dynamic, i.e., it is not actively driven by any device during most of the evaluation phase. Second, output Q is in high impedance when the clock signal CK is low. Both these factors reduce noise immunity, making the design more sensitive to noise induced by the input, output, power supply, and substrate. Third, the timing of the sampling window is critical: a short $T_{\rm ON}$ delay could lead to metastability or functional failure because the sampling window is too short to correctly evaluate the input. A long $T_{\rm ON}$ delay, on
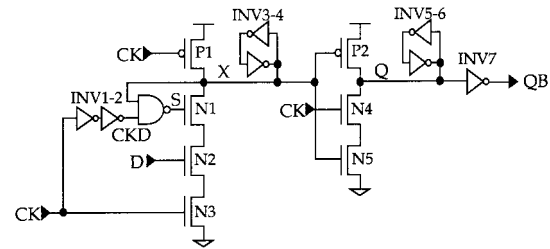
the contrary, may be detrimental to the performance of the circuit since the hold time—determined by the length of the sampling window—would be increased. To provide a more robust design, the following enhancements were introduced to the structure of Fig. 1(b): 1) back-to-back inverters were added to avoid dynamic nodes, 2) node Q was buffered to isolate it from the output, and 3) a conditional shutoff circuit was implemented in the pulse generation by replacing the last inverter with a NAND gate (this feature is discussed later). A positive triggered version of the resulting circuit is shown in Fig. 3. The structure has been denoted a *semidynamic flip-flop* (SDFF) because of its combination of dynamic and static circuits. In the embodiment shown in Fig. 3, the flip-flop samples input D and produces output QB, which is the logic complement of D. A detailed description of the operation of SDFF follows.

On the falling edge of clock CK, the flip-flop enters the precharge phase. Node X is precharged high, cutting off node Q from the input stage. The static latch INV5–6 holds the previous logic level of Q and QB. Since CKD is also low during precharge, node S remains high, holding transistor $N1$ on. The *evaluation phase* begins with the rising edge of clock CK. If input D is low—the flip-flop is latching a zero—node X will remain high, held by the INV3–4 latch. Node Q either will remain low or will be discharged through transistors N4–5, driving QB high. Three gate delays after CK rises, node S is driven low, turning transistor $N1$ off. This *shutoff* operation prevents a subsequent low-to-high transition of D from discharging node X, thus providing the flip-flop its edge-triggered nature.

If input D were high prior to evaluation—the flip-flop is latching a one—node X would be discharged through the pulldown path N1–3. The static latch INV3–4 would hold the value of X even if input D were subsequently driven low. The falling transition of X would turn transistor $P2$ on, driving Q high and output QB low. This would also force node S to stay high, preventing the shutoff of transistor $N1$, which is unnecessary after node X has been discharged.

Notice that by using a NAND gate coupled to node X and CKD, the shutoff of the pulldown path is conditioned to the state of input D. If D were high prior to evaluation, signal CKD would be blocked and no shutoff would be performed. By strongly skewing inverters INV1–2 and the NAND gate, this modification allows the reduction of the sampling window by about one inverter delay at the expense of a small speed penalty for increased loading on node X. This feature yields a shorter hold time and better input-
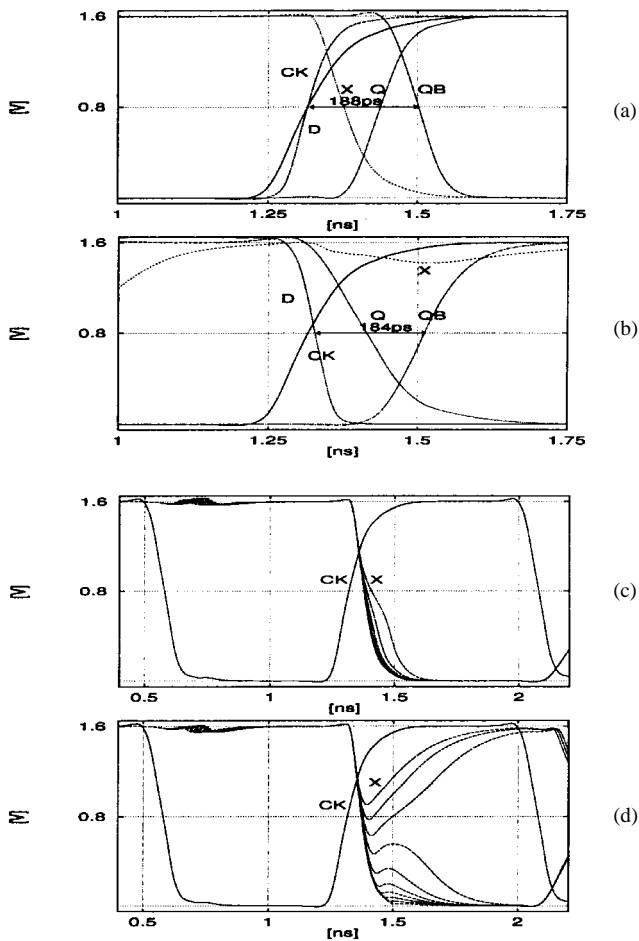
Fig. 4. SPICE waveforms for SDFF. Waveforms when latching (a) a "1" and (b) a "0." Waveforms for (c) conditional shutoff versus (d) unconditional shutoff.

Fig. 5. SDFF with embedded $(A + B)\,(C + D) + (E + F)\,(G + H)$ function.

to unconditional shutoff, conditional shutoff is less sensitive to variations of the sampling window, and therefore more robust; or that a shorter sampling window can be achieved with this scheme.

IV. EMBEDDING LOGIC FUNCTIONS

While the idea of incorporating logic functions into latches is not new [7], [8], the challenge has been to develop latch structures that can do it efficiently, in terms of both speed and area. One distinctive advantage of SDFF is that complex logic functions can be added easily. Indeed, most logic functions available in domino logic, such as wide OR functions, multiplexors, and complex gates, can also be implemented into SDFF. For the positive triggered circuit of Fig. 3, a logic gate can be built in the dynamic stage using NMOS transistors only. For an N-input function, N transistors are needed. Consequently, compact area and fast operation can be achieved if the flip-flop is converted into a gate-flip-flop structure. In particular, scan functionality can be added to the basic design with almost no degradation of performance. As an example, Fig. 5 shows an SDFF with embedded $(A + B)\,(C + D) + (E + F)\,(G + H)$ logic. Such a complex function is possible because it is built into the precharged stage of the circuit. While latency is increased, the merger allows the elimination of one or more levels of logic from the path leading to the flip-flop. The result is a reduction in the overall latency of the circuit. Another useful logic function to embed in this flip-flop is a wide AND/OR gate. When used as a multiplexor, this gate provides the additional advantage that select exclusivity is not required.

An extensive library of SDFF cells with different built-in functionality has been developed for the microprocessor reported in [1]. This has allowed designers to optimize logic by merging gates and flip-flops, as well as timing optimization by removing one or more gates from critical paths.

V. DYNAMIC FLIP-FLOPS

The flip-flops introduced in the previous sections are intended to interface with static logic. In this section, equivalent circuits are introduced that interface with dynamic logic.

Fig. 6 shows the implementation of a single-rail dynamic flip-flop (DFF) with embedded $(A + B + C + D)$ logic. The circuit operates in a way similar to SDFF (see Fig. 3) except that output Q is reset during precharge. This is required by the monotonic nature of dynamic logic. In this way, the flip-flop can directly drive dynamic gates without the penalty associated

noise rejection. Furthermore, the conditional shutoff makes the circuit less sensitive to variations of the sampling window that are due to manufacturing-process variations or to unaccounted layout parasitics.

Fig. 4(a) shows SPICE waveforms for the flip-flop of Fig. 3. Results were obtained in a 0.25-μm technology at 1.6 V, 105°C, and typical process. The latency of the flip-flop at zero setup time is 188 ps for the low-to-high input transition and 185 ps for the high-to-low input transition. Notice that the setup time is zero, and it even can be slightly negative. The worst case hold time is 130 ps.

To illustrate the operation of the conditional shutoff in SDFF (Fig. 3), a simulation was performed in which the delay between CK and CKD (INV1–2) was forced to change from 100 to 10 ps in intervals of 10 ps. This is equivalent to reducing the sampling window (T_{ON}). The same simulation was performed on an SDFF where the NAND gate was replaced with an equivalent inverter. Simulation results are shown in Fig. 4(b). Notice that the conditional shutoff [Fig. 4(c)] yields clean signal transitions and correct operation even when the CK–CKD delay is forced to 10 ps. The unconditional shutoff circuit, on the other hand, shows a noticeable signal degradation as the window gets shorter, eventually producing a complete failure [Fig. 4(d)]. This result shows that compared
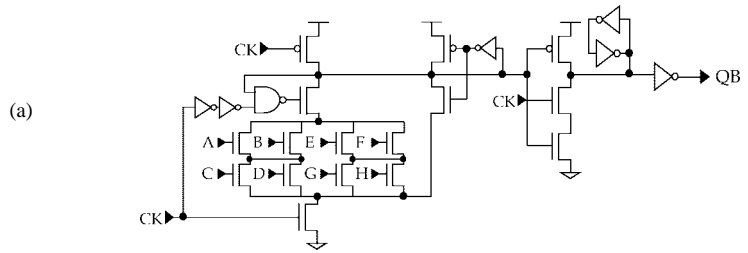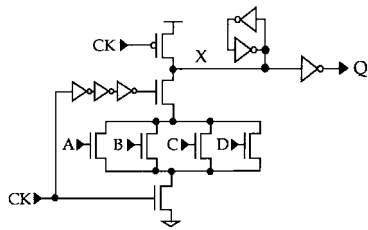
Fig. 6. Single-rail dynamic flip-flop with embedded $(A + B + C + D)$ function.

Fig. 7. Dual-rail dynamic flip-flop.

Fig. 8. Dual-rail DFF with embedded logic and common shutoff.

TABLE I
SPEED COMPARISON OF SDFF VERSUS OTHER DESIGNS

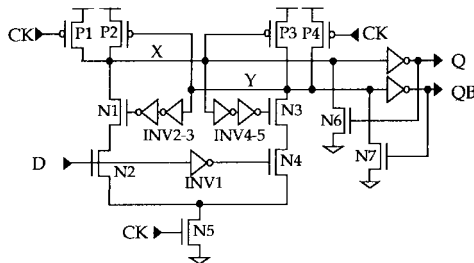|	TSPC	HLFF	SDFF	SFF
Latency	304 ps	194 ps	188 ps	300 ps
Speedup	0.99	1.55	1.6	1.0

with delayed clocking [9]. This implementation yields a very fast and simple circuit. Notice that unconditional shutoff has been used to provide minimal device count and minimum loading on node X.

When true and complement signals are needed, a dual-rail dynamic flip-flop is required. A dual-rail converter has been reported in [10], and a dual-rail flip-flop in [12]. The new circuit schematic is shown in Fig. 7. This is similar to the edge-triggered latch reported in [13], except that the outputs are precharged. The flip-flop samples input data D and produces dual-rail outputs Q and QB. Both outputs are monotonic, so the flip-flop can drive dynamic logic directly. Notice that the inversion needed to produce the complementary output QB is done by inverting input D (INV1). This adds one inverter delay to the latency of QB.

The circuit operates as follows. On the falling edge of clock CK, the flip-flop enters the precharge phase. Nodes X and Y are precharged high, while outputs Q and QB are predischarged low; transistors $N6, N7, P2$, and $P3$ are all off, while transistors $N1$ and $N3$, the *shutoff* devices, are both on. On the rising edge of the clock, the flip-flop enters the *evaluation* phase. If input D is high, node X will be discharged, causing output Q to go high, transistor $N3$ to shut off, and $P3$ to turn on. Node Y will remain high, held by transistor $P3$, which operates as a *keeper,* forcing output QB to remain low. If D goes from high to low while CK is still high, transistor $N6$ will hold node X at ground. The shutoff transistor $N3$, which is off, will prevent node Y from discharging. If input D is low, node Y will be discharged, causing output QB to go high and Q to remain low. The purpose of inverters INV2–3 and INV4–5 is to reduce the load on the critical nodes X and Y. This minimizes the flip-flop latency at the expense of a larger hold time.

Similarly to SDFF and single-rail DFF, the dual-rail DFF can also incorporate logic functions. Due to the complementary n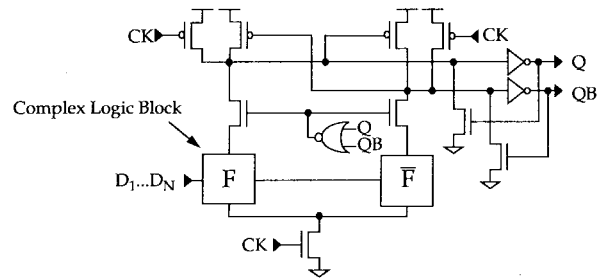ature of the circuit, in general, $2N$ devices are needed to implement a function with N inputs. In some cases, the device count can be reduced by sharing transistors. In cases where the embedded logic is too complex, for certain input combinations there exists a risk that charge stored in intermediate nodes of the logic network may affect dynamic nodes X or Y after one or the other has evaluated (back charge sharing). This is possible because one of the complementary paths in the flip-flop is always left open. A way to prevent this is to unconditionally shut off both paths, as shown in Fig. 8, after either Q or QB has evaluated.

VI. PERFORMANCE ANALYSIS

To evaluate the performance of SDFF, other designs were simulated under similar conditions. The following flip-flops were compared.

- SDFF (Fig. 3).
- TSPC [Fig. 1(a)]; to make a fair comparison, back-to-back inverters were added to internal dynamic nodes and the output was buffered.
- SFF, which is a conventional master–slave flip-flop built with transmission gates similar to the one used in [11], but with weak feedback inverters instead of clocked feedback. This is used as a baseline for comparison.
- HLFF, which is a hybrid structure reported in [12].

The results are summarized in Table I. Notice that SDFF is the fastest. This is because despite the fact that SDFF has three stages, the first stage is dynamic and the second stage is heavily skewed. Thus, these two stages are very fast compared to static logic. In terms of power, SDFF dissipates about 0.33 mW at 600 MHz. In addition to its shorter latency, SDFF is about half the size of SFF, presents about half the clock load, and requires only a single clock phase. In a study reported in [14] and [15], a set of 11 representative latches and flip-flops used in high-performance and low-power microprocessors was analyzed. SDFF was ranked first in terms of speed and third in power-delay product.

Table II shows the latency of SDFF with embedded logic for different logic functions versus the speed of a discrete

TABLE II
SPEED COMPARISON OF SDFF WITH EMBEDDED LOGIC VERSUS DISCRETE LOGIC

	D	A.B	A+B	AB+CD
Embedded	188ps	208ps	196ps	228ps
Discrete	188ps	280ps	286ps	348ps
Speedup	1.0	1.35	1.46	1.53

combination of static logic and SDFF. This table shows that the speed penalty of adding a two-input OR function $(A+B)$ is less than 10 ps, two-input AND (AB) is about 20 ps, and function $AB + CD$ is about 40 ps. This saves on the average about 100 ps compared to the discrete case. Notice also how SDFF with $AB+CD$ logic is still faster than the conventional SFF.

VII. CONCLUSION

This paper describes a new family of semidynamic and dynamic edge-triggered flip-flops, which are well suited for high-performance microprocessor design. They provide short latency and a good interface to static and dynamic logic, and can easily incorporate complex logic functions with a small delay penalty. These features contribute to reducing the pipeline overhead of the processor by allowing the elimination of one or more gate delays from a path leading to the flip-flop. These flip-flops have played an integral role in meeting cycle-time goals in [1].

REFERENCES

[1] B. Lynch and G. Lauterbach, "UltraSPARC-III: A 600 MHz 64-bit superscalar processor for 100-way scalable systems," in *Proc. HotChip Symp.*, Stanford University, Stanford, CA, Aug. 1998, pp. 17–25.
[2] F. Klass, "Semi-dynamic and dynamic flip-flops with embedded logic," in *Dig. Tech. Papers Symp. VLSI Circuits*, Honolulu, HI, June 1998, pp. 108–109.
[3] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, pp. 62–70, Feb. 1989.
[4] Y. Ji-Ren, I. Karlsson, and C. Svensson, "A true single-phase-clock dynamic CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 899–901, Oct. 1987.
[5] S. Kozu, M. Daito, Y. Sugiyama, H. Suzuki, H. Morita, M. Nomura, K. Nadehara, S. Ishibuchi, M. Tokuda, Y. Inoue, T. Nakayama, H. Harigai, and Y. Yano, "A 100 MHz, 0.4 W RISC processor with 200 MHz multiply-adder, using pulse-register technique," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 140–141.
[6] A. Shibayama, M. Mizuno, H. Abiko, A. Ono, S. Masuoka, A. Matsumoto, T. Tamura, Y. Yamada, A. Nishizawa, H. Kawamoto, K. Inoue, Y. Nakazawa, I. Sakai, and M. Yamashina, "Device-deviation-tolerant over-1 GHz clock-distribution scheme with skew-immune race-free impulse latch circuits," in *ISSCC Dig. Tech. Papers*, Feb. 1998, pp. 402–403.
[7] J. Earle, "Latched carry-save adder," *IBM Tech. Disclosure Bull.*, vol. 7, no. 10. pp. 909–910, Mar. 1965.
[8] D. W. Dobberpuhl, R. Witek, R. Allmon, R. Anglin, D. Bertucci, S. Britton, L. Chao, R. Conrad, D. Dever, B. Gieseke, S. Hassoun, G, Hoeppner, K. Kuchler, M. Ladd, B. Leary, L. Madden, E. McLellan, D. Meyer, J. Montanaro, D. Priore, V. Rajagopalan, S. Samudrala, and S. Santhanam, "A 200 MHz 64-b dual-issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1555–1565, Nov. 1992.
[9] G. Yee and C. Sechen, "Clock-delayed domino for adder and combinational logic design," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1996, pp. 332–337.
[10] N. B. Gaddis, J. R. Butler, A. Kumar, and W. J. Queen, "A 56-entry instruction reorder buffer," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 212–213.
[11] G. Gerosa, S. Gary, C. Dietz, D. Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, T. Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle, "A 2.2 W, 80 MHz superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 90, pp. 1440–1452, Dec. 1994.
[12] H. Partovi, R. Burd, U. Salim, F. Weber, L. DiGregorio, and D. Draper, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 138–139.
[13] J. Montanaro, R. Witek, K. Anne, A. Black, E. Cooper, D. Dobberpuhl, P. Donahue, J. Eno, A. Farell, G. Hoeppner, D. Kruckemyer, T. Lee, P. Lin, L. Madden, D. Murray, M. Pearce, S. Santhanam, K. Snyder, R. Stephany, and S. Thierauf, "A 160 MHz 32 b 0.5 W CMOS RISC microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 214–215.
[14] V. Stojanovic, V. Oklobdzija, and R. Bajwa, "A unified approach in the analysis of latches and flip-flops for low-power systems," in *Proc. ISLPED98*, Monterey, CA, Aug. 1998, pp. 227–232.
[15] ———, "Comparative analysis of latches and flip-flops for high-performance systems," in *Proc. IEEE Int. Conf. Computer Design*, Austin, TX, Oct. 1998, pp. 264–269.