# Limited switch dynamic logic circuits for high-speed low-power circuit design

W. Belluomini
D. Jamsek
A. K. Martin
C. McDowell
R. K. Montoye
H. C. Ngo
J. Sawada

*This paper describes a new circuit family—limited switch dynamic logic (LSDL). LSDL is a hybrid between a dynamic circuit and a static latch that combines the desirable properties of both circuit families. The paper also describes many enhancements and extensions to LSDL that increase its logical capability. Finally, it presents the results of two multiplier designs, one fabricated in 130-nm technology and one in 90-nm technology. The 130- and 90-nm designs respectively reach speeds up to 2.2 GHz and 8 GHz.*

## Introduction

As silicon technology moves through the 90-nm node to 65 nm and below, the power, area, and frequency benefits associated with each new process generation are decreasing. At the same time, the interval between process generations is increasing because the technology is becoming more difficult and expensive to develop. To maintain historical trends in system performance, other parts of the system stack must increase their contribution to performance. Many circuit approaches have been used to improve circuit performance beyond that of static CMOS. These circuits typically use some form of dynamic logic that goes through a precharge and evaluate phase. The most commonly used dynamic circuit form is domino logic, first presented in [1]. Many improvements and variations on domino circuits have since been presented. These can be divided into those circuits in which a signal transition controls dynamic evaluation [2–7] and those in which the clock controls the dynamic circuit evaluation [8, 9]. Signal-controlled dynamic circuits have the constraint that the logic network must be unate (no input is required in both its true and complement form; i.e., "a" and "not a" are not both part of the logic function). Since this is rarely the case in general logic, the constraint often leads to duplication of logic to make the signals dual-rail (the true and complement versions of a signal are computed and transmitted separately; i.e., there is logic to compute "a" and logic to compute "not a" instead of just inverting "a"). Dual-rail circuits are significantly larger and, in wire-delay-dominated technologies such as 65 nm and below, can eliminate the benefit of using dynamic circuits. Clock-controlled dynamic circuits do not have the unateness requirement. The dynamic circuit does not evaluate until a clock edge arrives, and therefore signals driving the dynamic circuit can take on any value before the clock edge. Clock-controlled dynamic circuits do not require dual-rail logic and are therefore a better choice when wire delays are a significant issue. The clock-controlled approach in [8] uses a latch before the dynamic gate to protect the inputs to the gate; the approach in [9] uses a delayed clock to control signal timing. This paper describes a new circuit family, limited switch dynamic logic, that uses a latch after the dynamic gate and a standard clock. This approach provides significant frequency, area, power, and voltage scalability benefits.

## Limited switch dynamic logic

Limited switch dynamic logic (LSDL) is a hybrid of static and dynamic circuits. Its goal is to combine the best properties of both. In particular, LSDL is focused on reducing the power penalty that is traditionally associated with dynamic circuits. This power penalty occurs primarily for two reasons. The first is the fact that a domino circuit must precharge the dynamic node to a high value every cycle and then transmit that (inverted) value through its output inverter to what may be a long wire and a large load. Even if the input to the domino circuit is constant, the large output load will switch every cycle if the input is such that the dynamic node discharges. This produces a large amount of wasted power. The second power issue with domino circuits is the need for dual-rail signaling. Although domino circuits require fewer transistors to implement a logic function than do static circuits, two copies of each circuit are usually required to produce a bit and its complement.



**0018-8646/06/$5.00 © 2006 IBM**

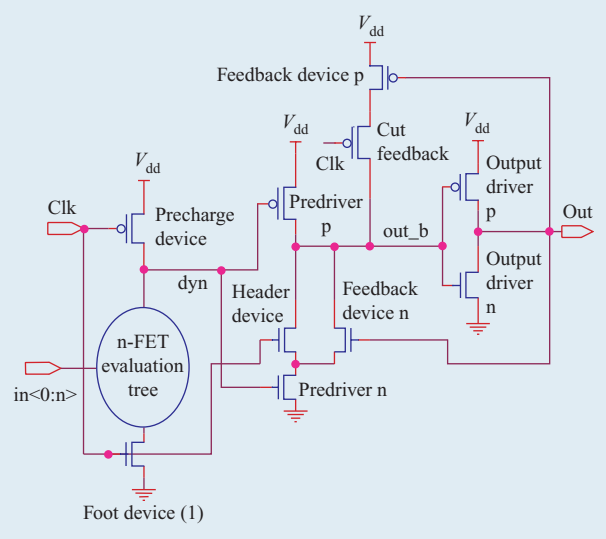


**Figure 1**

Basic LSDL circuit form.

Signals cannot simply be inverted because all inputs to a domino circuit must be low at the beginning of an evaluate cycle in order to prevent incorrect evaluation.

LSDL addresses these issues by combining a domino front end with a simple latch, as shown in **Figure 1**. When a latch is combined with a dynamic circuit, the dynamic node is isolated from long wires and large loads. It still must precharge every cycle, but the capacitance that is switching is greatly reduced. Another advantage to the latch is that the evaluation of each LSDL circuit is triggered by the rising edge of the clock. All inputs must be set up before the clock edge. This eliminates the need for dual-rail signaling, because high inputs cannot transition low later in the cycle. The next subsection describes in detail the operation of an LSDL latch and its variants.

### LSDL operation

Figure 1 shows the basic LSDL latch configuration. (Other, more complex, latch forms are discussed later.) This configuration of devices provides computation, latching, and drive. The device marked *Precharge device* is used to precharge the dynamic node *dyn*. To minimize clock power, this device is sized to be the smallest device capable of precharging the dynamic node within the allocated precharge time. The ellipse marked *n-FET evaluation tree* indicates the area in which the actual computation takes place. The device marked *Foot device (1)* functions as a cutoff device to prevent the latch from burning dynamic power during precharge. This device allows inputs to switch at any time during the precharge phase. The devices marked *Predriver p* and *Predriver n* serve two functions: They form part of the inverter pair used for latching and they provide gain to strengthen the signal before it goes to the output driver. The device marked *Header device* prevents the rising of dyn during precharge from affecting the value held in the latch. Two feedback devices (*Feedback device n* and *Feedback device p*) are used to restore charge to the latched value. These devices allow the value in the latch to be held as long as the clock is kept low. The device marked *Cut feedback* is there to prevent a conflict from occurring between feedback device p and predriver n when the out_b node switches from 1 to 0. This device is not strictly necessary but is included to provide some robustness against n/p ratio variation in the process. The devices marked *Output driver p* and *Output driver n* form one of the two inverters needed to latch the output (the other inverter is the predriver) and are also used to add gain to the signal so that it can be driven across a macro.

There is a timing race between the dynamic node pulling down and out_b beginning to fall, which can cause a glitch to occur. At the beginning of evaluation, dyn is always high, so the predriver n device is on, allowing out_b to begin to discharge. If this discharge is sufficiently large, it can cause a glitch on the output, which will burn power. In order to minimize the size of this glitch, the circuit must be sized so that the dynamic node predriver switches fast enough to keep the size of the glitch below 10% of the supply. Because sizing to prevent glitches also optimizes the circuit to respond quickly in the case in which three transitions are required to propagate data through the latch (the slow case), preventing glitches is not difficult most of the time.

### LSDL pipelines

Because of the combination of logic and latching that occurs in LSDL, the pipeline and circuits are designed concurrently. There are two basic approaches to combining LSDL latches to form a pipeline. The first is L1/L2; an L1/L2 pipeline and its timing diagram are shown in **Figure 2(a)**. In the L1/L2 configuration, the LSDL latches are used as split latches. There are two LSDL latches per clock cycle. This configuration has the advantage that it prevents short paths on which fast-moving data travels through two stages in the pipeline in one cycle. This is not possible in this configuration because two sequential latches are never open at one time. Because of this property, the L1/L2 configuration is the preferred LSDL pipeline style.

Because of the short cycle time of many modern designs, there are instances in which computation cannot be broken up in a way that allows an L1/L2 pipeline. In these cases, a *pulse latch* configuration is used. The timing diagram for the L1/L1 (pulse latch) configuration is

shown in **Figure 2(b)**. In this pipeline configuration, all of the latches are clocked at once. A signal has a full cycle to be computed and propagated to the next latch. This pipeline configuration depends on the existence of a significant delay between latches for correctness. This delay is what prevents data from propagating through two latches in a single clock cycle. The delay can occur both inside the LSDL latch and outside the latch as the signal is propagated to the next latch. It can be due to static logic placed in the path between latches or to pure wire delay. Pulse latch pipelines require additional timing checks to ensure that no signal can propagate through two latches in a cycle in early-mode operation (fast process at high voltage). However, there will be situations in which pulse latch paths are necessary. In many of these situations, the risk can be mitigated by requiring a complete half cycle for a data transition to propagate out of the LSDL latch. This contains the early-mode path inside the LSDL latch and facilitates checking for it. If it takes half a cycle to propagate out of the latch in early mode, it is guaranteed that there will not be an early-mode failure. Another way to prevent early-mode failure is to use a pulse that is less than half a cycle to clock the pulse latch. The feasibility of doing this depends on the targeted cycle time. The minimum pulse that can be reliably generated is 3FO4 to 4FO4 in length. For cycle times greater than 8FO4, using a clock that is less than half the cycle time is a good choice to increase the timing margin, although it also increases clocking complexity.

Both of the LSDL pipeline forms imply a very fine pipeline structure. With the basic LSDL form, two dynamic pull-down trees are evaluated per cycle. More complex forms of LSDL (discussed in the next section) can help increase the amount of logic that can be included in a pipeline stage. However, even with more complex LSDL forms, the pipeline for an LSDL-based processor will be quite deep. Deep, high-frequency pipelines have historically been considered well suited for floating-point intensive code, and that is a major possible application of this technology. Also, newer workloads, such as streaming media, are well suited to a high-frequency, deep pipeline. To make the deep pipeline effective, a large register file (of the order of 128 or 256 entries) is needed. Larger caches with high bandwidth to the register file are also useful in improving performance.

### *Complex output gate LSDL circuits*

While Figure 1 shows the LSDL latch in its most basic form, there are other forms that can be used to increase its computational power. These forms are often required when designers encounter the electrical limits on height and width for the n-FET tree. **Figure 3(a)** shows an example of an LSDL latch with a NAND gate built into the output latch. This circuit form overlaps additional computation with the latching function by building a gate into the predriver stage. NAND-based complex output gates can typically be used with little performance degradation because the extra delay due to stacked n-FETs in the predriver occurs only in the two transition cases (dyn1 and dyn2 remain high, out_b falls, and out rises). The three-transition case, in which the output falls, is usually the critical path, so the extra delay for the NAND gate does not increase the cycle time. **Figure 3(b)** shows an LSDL latch with a NOR gate built into the output. Unlike the NAND version, the NOR version typically adds delay to the critical path because the staked devices are p-FETs and are used in the three-transition case. However, there are many instances in which the NOR version is useful.

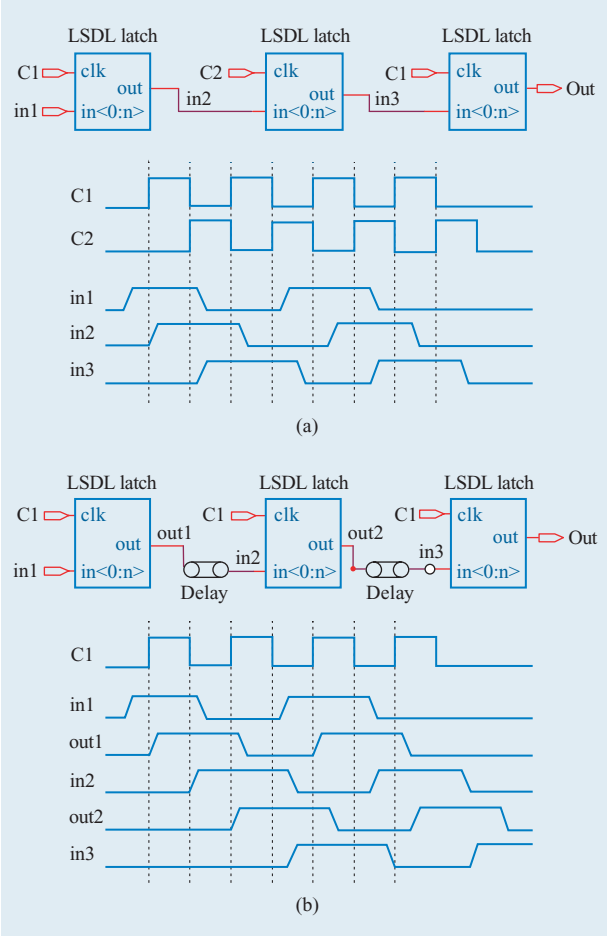


**Figure 2**

Pipeline configuration: (a) L1/L2; (b) L1/L1. ©2005 IEEE. Reprinted from [10] with permission.

### *Other LSDL circuit techniques*

**Figure 4(a)** shows an example of a domino front end to an LSDL latch using a delayed clock. This configuration can

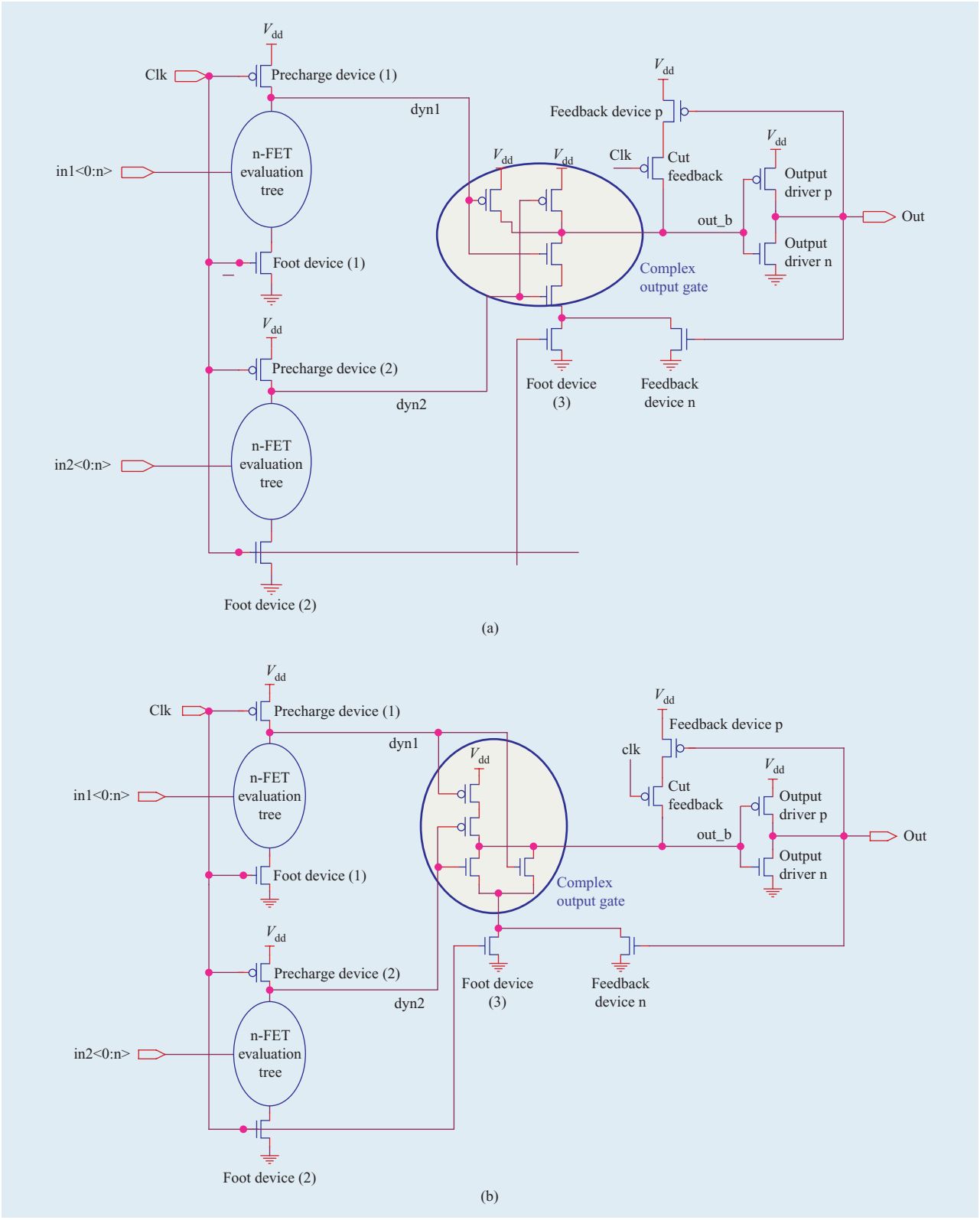


Figure 3

Complex gate forms: (a) NAND; (b) NOR.

be used either as a half cycle of computation—part of an L1 or L2 of the pipeline shown in Figure 2(a), or as a full cycle of computation—a full-cycle stage, as shown in Figure 2(b). The purpose of the delayed clock is to guarantee that the outputs of the domino stage are evaluated before the LSDL latch enters evaluate. The purpose of the secondary precharge transistor in the delayed clock domino front end is to ensure that the domino gate does not go into precharge before the LSDL latch closes. If the domino gate precharges before the latch is closed, the latch may store the wrong value. The secondary precharge transistor will not turn on until the LSDL latch is closed.

There are two benefits of the domino front-end technique. The first is the reduction in device count that occurs when an LSDL latch is converted into a domino front end. The delayed clock domino front end contains five fewer devices than an LSDL latch. The second benefit is speed. This approach has the potential to convert a full cycle of computation into half a cycle if the computation can meet timing at half a cycle, a benefit that is not available if a full-cycle computation is needed.

There are also two drawbacks to this technique. The first is power: Since the output of the domino front end is not latched, it is precharged high every cycle. Depending on the size of the output capacitance, this could consume significantly more power than an LSDL latch implementation of the front end. The second drawback is clocking complexity. Creating and distributing an accurate delayed clock adds complexity and, probably, overall clock power. Extra margin may be needed to ensure that the circuit functions properly over variation in the arrival of the delayed clock edge. The margin required depends on the transistor delay variation inherent in the technology.

**Figure 4(b)** shows a domino front end without a delayed clock. The secondary precharge device is no longer needed because both the domino gate and the LSDL latch are using the same clock signal. This configuration has a constraint that the delayed clock version does not have: The evaluation tree of the LSDL latch must not turn on when all of the incoming signals are in precharge. If all of the incoming signals are driven directly from domino gates, this is not a problem, since they will all be low in precharge. However, if there are inversions between the domino gate and the LSDL evaluation tree, some incoming signals may be high in precharge.

This case may or may not work depending on the function implemented by the evaluation tree of the LSDL latch. Another issue with this configuration is an output glitch on the LSDL latch. In the undelayed clock configuration, out_b, the internal latch node of the LSDL latch immediately begins to pull down. By the time the

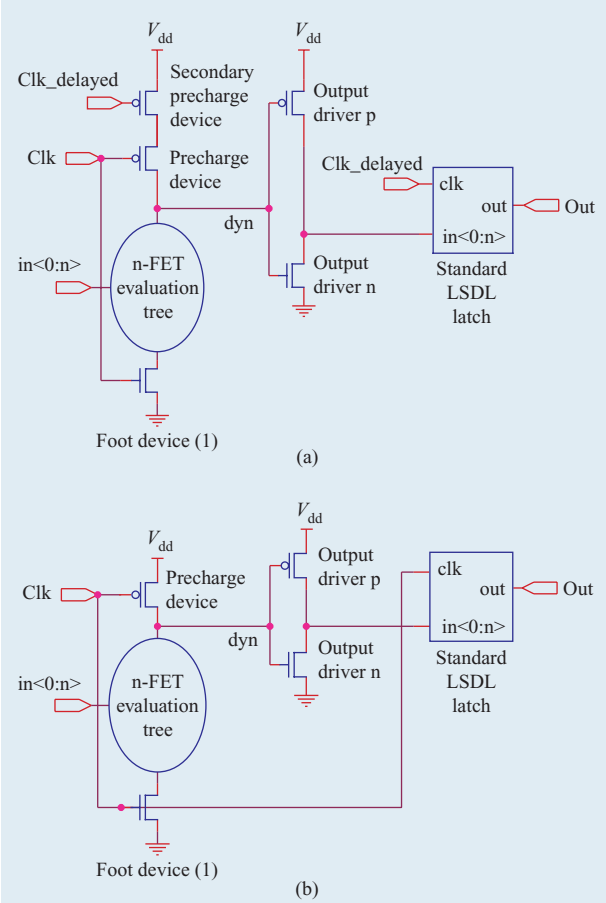


**Figure 4**

LSDL latch with domino front end with (a) delayed clock; (b) standard clock.

domino gate has driven the inputs to the LSDL latch high and the dynamic pull-down tree in the latch has pulled down, out_b may be most of the way to ground. Once the dynamic node pulls down, out_b rises again, pulling the output back down. This glitch does not affect correctness, but it does increase power. This configuration has the advantage that it does not require a delayed clock, but has the disadvantage that it can be used with only a subset of possible evaluation trees and may burn additional power owing to the output glitch.

The domino front end concept enables additional optimizations to the LSDL latch. If the evaluation tree of the LSDL latch is guaranteed to be off when the domino front end is in precharge, a foot device is not necessary. The footless LSDL evaluation tree can be used with or without a delayed clock. **Figure 5(a)** shows a version without a delayed clock. The footless implementation reduces the stack height of the evaluation tree, which

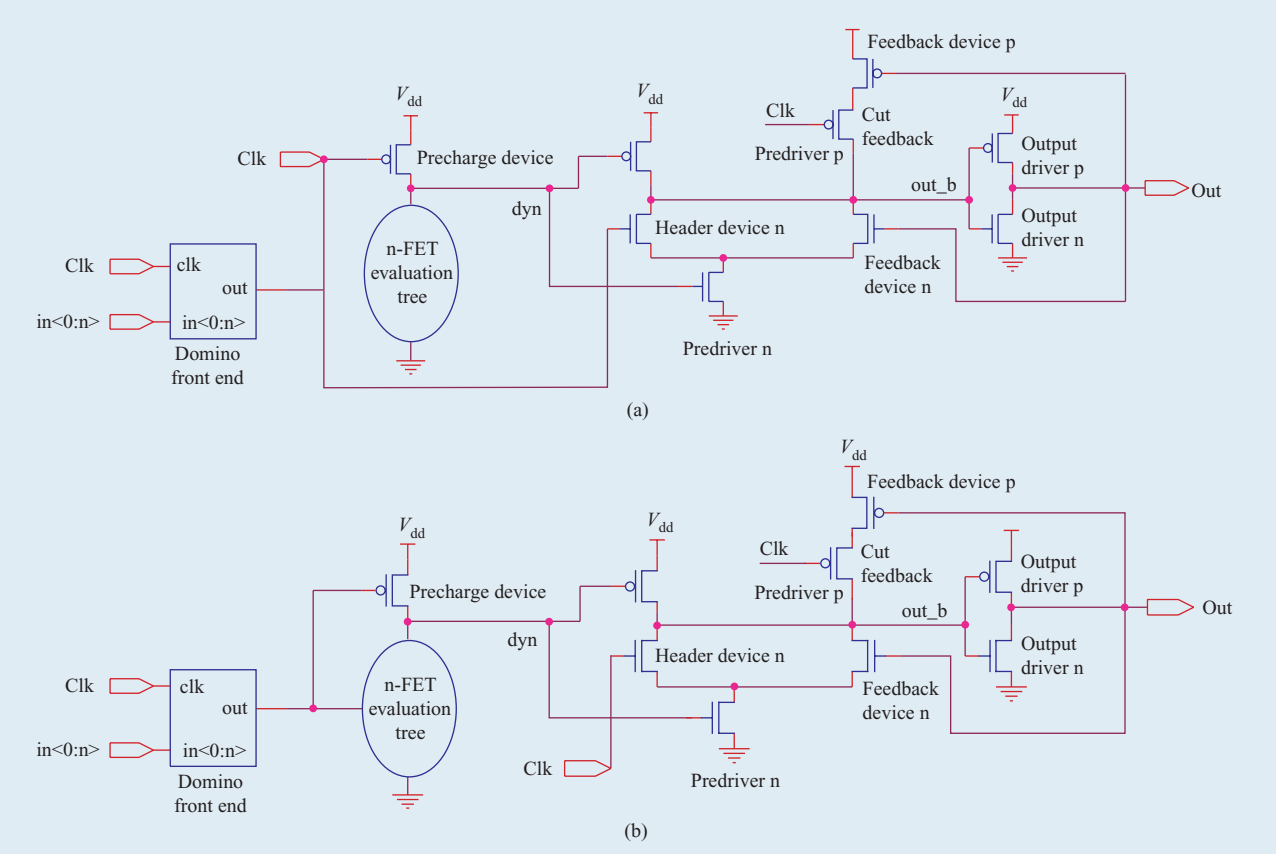


**Figure 5**

Domino front end: (a) footless; (b) data-clocked.

reduces area and power. Removing the foot device also reduces overall clock load.

**Figure 5(b)** shows a circuit that takes additional advantage of the domino front end by clocking the precharge device with an output of the domino front end. To use this approach, the LSDL evaluation tree must not pull down during the domino precharge phase, and the signal (or signals) used to time the precharge devices must be guaranteed to turn on the precharge device (or device tree) when in precharge and turn off the precharge device (or device tree) when in evaluate. This circuit eliminates even more clock load than the previous circuit described.

### *Dealing with leakage*

As technology scales, leakage becomes increasingly significant. All of the LSDL circuits described in the previous section have no mechanism to ensure that charge on the dynamic node is maintained during the evaluate cycle if the circuit does not logically evaluate. In older technologies (90 nm and larger), this configuration is workable as long as the length of the evaluate period is limited in the local clock buffer design. However, in newer technologies (65 nm and smaller), limiting evaluate time is not sufficient to ensure correct operation over all process corners. Therefore, additional devices must be added to restore charge to the dynamic node during the evaluate phase.

There are a few ways to do this. The goal is to restore the dynamic node in a way that has as little impact as possible on the overall design performance and power. This has proven difficult to do by simply adding feedback that is the same over all process conditions. Feedback that is necessary under high-leakage conditions may unduly slow down the latch during low-leakage conditions. One way to address this problem is to control the strength of the feedback with a reference voltage. This voltage allows the feedback to be turned on or off. It can also be set to an intermediate value if the chip has the capability to distribute an analog voltage. Two topologies can be used to provide feedback for simple LSDL latches: the *keeper* approach and the *minder* approach. Complex

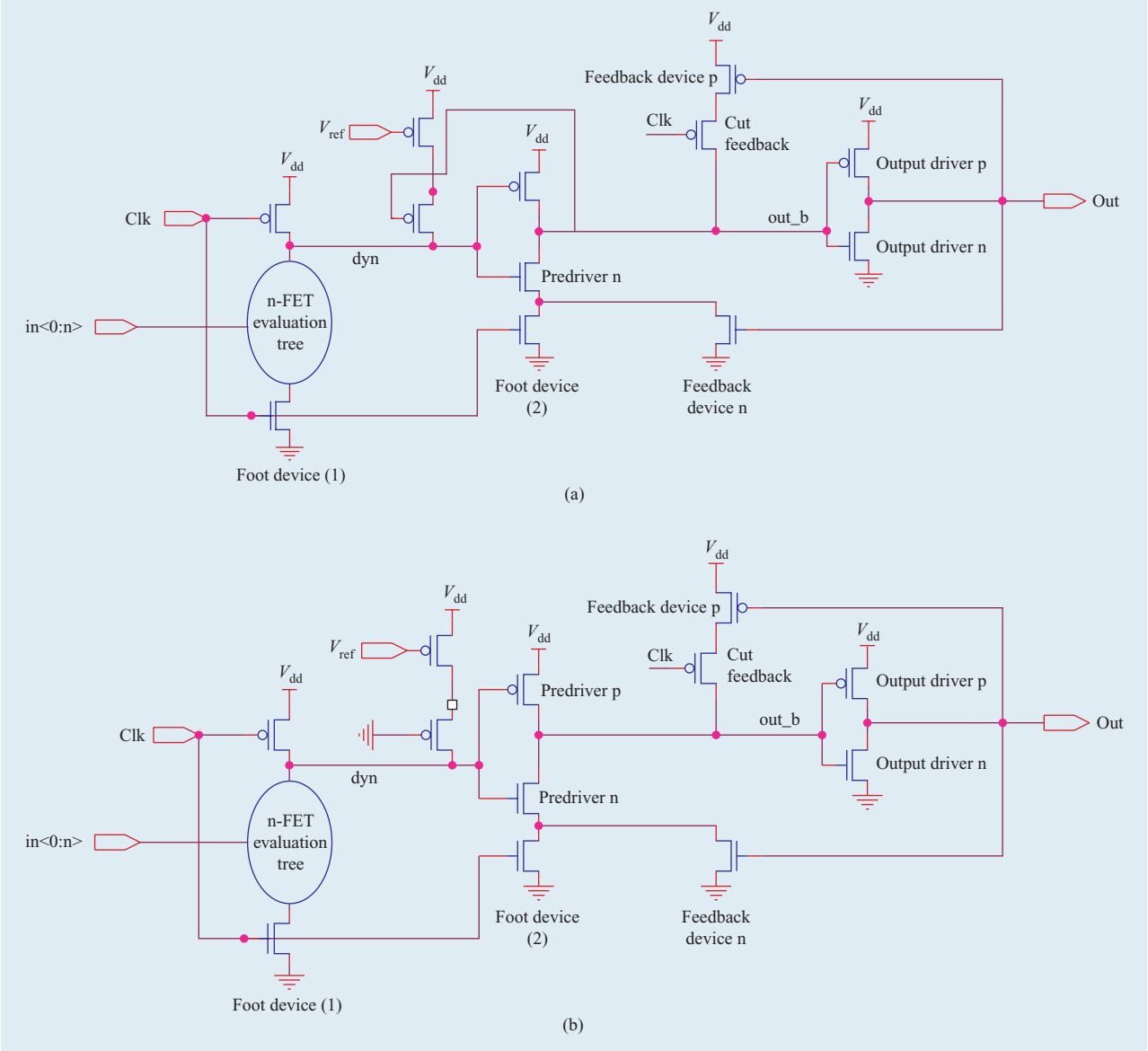


**Figure 6**

Two approaches to providing feedback for simple LSDL latches: (a) keeper approach; (b) minder approach.

output gates have somewhat different requirements, which are discussed next.

In the keeper approach, shown in **Figure 6(a)**, a two-high stack of p-FETs is used for feedback. One p-FET is attached to the reference voltage and one is controlled by the internal latch node, out_b. This stack provides charge to the dynamic node when its voltage degradation due to leakage is in danger of flipping the latch to the incorrect value. There are two cases in which a leakage-induced discharge of the dynamic node causes latch failure: out_b is low and should remain low, and out_b is high and is supposed to transition low in the current evaluation phase. In the first case, it is easy to see how the keeper protects the circuit from leakage. The node out_b is low and the keeper is on at the beginning of the evaluation. Charge that is lost by the dynamic node due to leakage will be restored. The second case is more interesting. At the beginning of the evaluation phase, out_b is high and the keeper is off. However, as soon as the clock rises, the header device and predriver n device turn on and begin pulling down on out_b. As soon as out_b gets a threshold below $V_{dd}$, the keeper turns on. If there is leakage, the

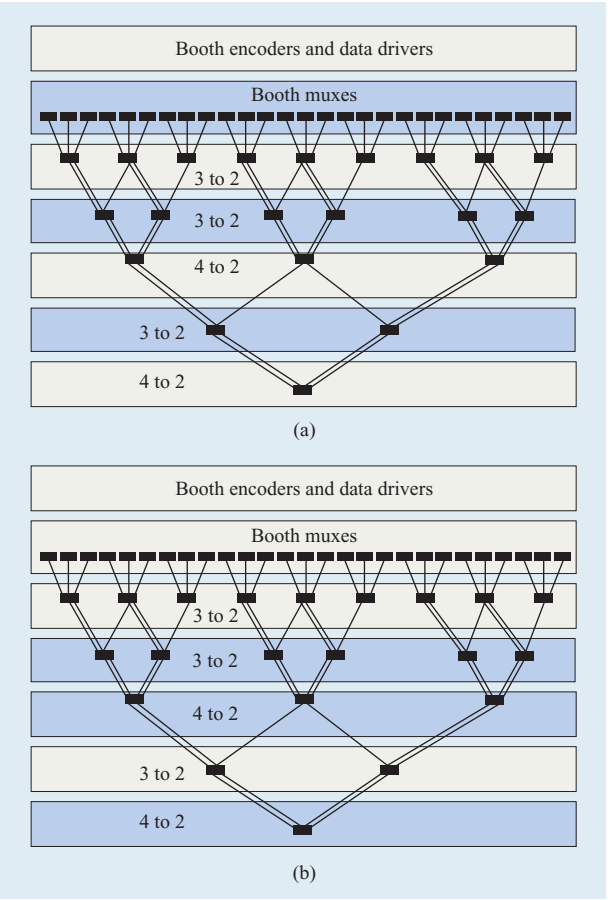


Figure 7

Multiplier microarchitecture for (a) 130-nm designs; (b) 90-nm designs. ©2005 IEEE. Reprinted from [10] with permission.

keeper will restore charge on the dynamic node. This second mode of operation depends on the property that the out_b node will always be slightly pulled down before the dynamic evaluation tree has a chance to evaluate. This is the case in almost all implementations of an LSDL latch, but it should be verified when the keeper approach is used for leakage protection.

The minder approach, shown in **Figure 6(b)**, does not use feedback; it continuously provides a small amount of charge to the dynamic node. One p-FET is tied to $V_{ref}$ and the other is tied to ground. The $V_{ref}$ voltage can either be tied to ground or be designed to be externally controllable. If it is externally controllable, it can be adjusted to provide more current in a fast (high-leakage) process and less or no current in a slow (low-leakage) process.

There are advantages and disadvantages to both approaches. The keeper has the advantage that a weaker n-stack is required to overcome the feedback and it has no dc power consumption. The minder approach has the advantage that it does not add additional load to the latch node. Complex output gates pose some difficulty for the keeper approach. In the NOR form of a complex output gate, Figure 3(b), the latch node out_b remains low throughout the evaluation if either one of the dynamic stacks pulls down. This means that for the NOR form, the keeper and minder topologies are essentially the same in that they both can remain on through the entire evaluation cycle. In the NAND form of the complex output gate [Figure 3(a)], the opposite situation occurs. The keeper will turn off for both dynamic nodes if one of them evaluates. This may allow one tree to incorrectly leak to ground. However, because the tree that evaluated flips the latch, the incorrect discharge does not affect the correctness of the latch value. Because the minder approach results in a simpler layout and less load on the latch node, it is the preferred choice in most cases. The keeper approach is preferable only when minimizing dc power is the main goal.

## Multiplier designs

There have been two fabricated LSDL multiplier designs, one in 130-nm technology [11] and one in 90-nm [10]. The 130-nm design is the first hardware implementation of LSDL circuitry. The microarchitecture for the 130-nm multiplier is shown in **Figure 7(a)**. It is a Booth-encoded Wallace tree design that produces a carry and sum result in three and a half pipeline stages (without the final add). All of the pipeline stages use the L1/L2 configuration with static logic between stages. All stages except the 4-to-2 compressor use only a single inverter between LSDL stages. The 4-to-2 compressor stage uses a static NAND gate. The size of this multiplier is 315 $\mu$m $\times$ 495 $\mu$m (0.15 mm$^2$). This is 50% of the area (scaled for technology) of the Wallace tree portion of the multiplier described in [12], which was claimed to be the smallest 54 $\times$ 54 Wallace tree multiplier published to date. The chip is measured to operate correctly at 2.2 GHz, 1.2 V, and 250°C. The power dissipation is 522 mW at an 80% switching factor.

As the first hardware implementation of LSDL, the 130-nm design left room for improvement in a number of areas. The first area was the addition of a device to cut the feedback in the latch. This device was not included in the LSDL latches in the 130-nm design. Hardware testing showed that the absence of this device caused a contention condition when the latch was switching low. This slowed down operation significantly and caused the chip to require a higher-than-expected voltage for operation. Another shortcoming detected during the testing of the 130-nm design was the clock quality. LSDL is very sensitive to clock skew and jitter. The 130-nm

design uses a delay-locked-loop (DLL) as a clock source and a very simple local clock distribution design. The 90-nm design uses a production-quality phase-locked loop (PLL) and carefully engineered local clock wiring. This results in a very-high-quality clock for the 90-nm design, with local skew of the order of 3–5 ps. The microarchitecture of the 90-nm design, shown in **Figure 7(b)**, has also been changed in order to provide increased performance. An additional half cycle has been added, resulting in four pipeline stages instead of three and a half. This half cycle is in the form of an L2/L2 pipeline stage. In Figure 7(b), the pair of consecutive colored bars shows the L2/L2 pipeline stage in the multiplier. This stage required a full cycle because it had the longest wire delay. Also, the static NAND gates used in the 130-nm design have been removed, and the 4-to-2 compressor stage has been redesigned. In the 90-nm design, the 4-to-2 compressor stage is a complex output gate LSDL latch with a five-high n-FET stack (including the clock device) in its dynamic pull-down tree. This circuit is significantly faster than the standard LSDL gate followed by a two-way static NAND that was used in the 130-nm design. The rest of the LSDL circuits in the multiplier are the basic form. The final area for the 130-nm multiplier macro is 432 $\mu$m $\times$ 288 $\mu$m (0.124 mm$^2$) in IBM 90-nm silicon-on-insulator (SOI) technology.

**Figure 8(a)** shows the measured voltage and frequency plot. The light area shows the functional frequency range, and the blue curve shows the normalized frequency of a ring oscillator that was placed near the multiplier. The measured top speed of the multiplier is 8 GHz at 1.4 V and 40°C. The macro can run at 2 GHz at voltages down to 0.75 V. Between 0.9 V and 1.2 V, the multiplier tracks very well with the ring oscillator and has linear voltage scaling.

The 8-GHz result is a factor of 3.6 improvement over the 130-nm design. The voltage/frequency characteristic of this double-precision function (mantissa multiply) is similar to those reported in [13, 14], which are 32-bit arithmetic logic unit (ALU) functions.

The total device area increase from the 130-nm design to this one was approximately 10%, adjusted for process. Most of this device area went into larger clock drivers, which were needed to improve clock quality. Minimizing this increase was critical to achieving the results, because increased device area would have resulted in a slower design due to increases in wire delay. **Figure 8(b)** shows the measured power/frequency curve of the multiplier macro. The measurements were taken at 40°C. The data plot shows total power, which includes local clock buffer power and the sector buffer that is used to distribute the clock to the local clock buffers. As expected, the power increases as the voltage increases to reach top speed. Of the measured power, 70% is dissipated in the clocking circuits, including the sector buffer and local clock buffers (but not the PLL). Despite this high percentage of clock power, the design is still quite power-efficient at the lower voltages.

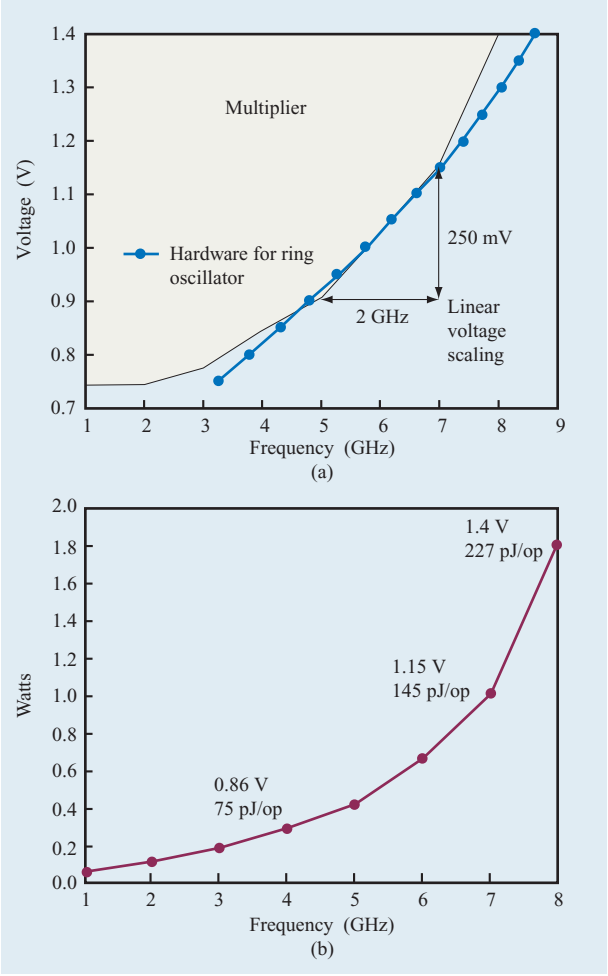


**Figure 8**

(a) 90-nm frequency results (typical hardware at 45°C). (b) 90-nm power results. ©2005 IEEE. Reprinted from [10] with permission.

## Conclusion

Hardware experiments have shown that LSDL is a fast, small, and power-efficient option for arithmetic circuits. These characteristics indicate that LSDL may provide an opportunity to continue frequency scaling trends despite the slowdown in technology scaling.

*Trademark, service mark, or registered trademark of International Business Machines Corporation.

## References


1. R. H. Krambeck, C. M. Lee, and H. Law, "High-Speed Compact Circuits with CMOS," *IEEE J. Solid-State Circuits* **17**, No. 3, 614–619 (June 1982).

2. V. Friedman and S. Liu, "Dynamic Logic CMOS Circuits," *IEEE J. Solid-State Circuits* **19**, No. 2, 263–266 (April 1984).
3. L. Heller, W. Griffin, J. Davis, and N. Thoma, "Cascode Voltage Switch Logic: A Differential CMOS Logic Family," *Proceedings of the IEEE International Solid-State Circuits Conference*, 1984, pp. 16–17.
4. C. M. Lee and E. W. Szeto, "Zipper CMOS," *IEEE Circuits & Devices* **2**, No. 3, 10–17 (May 1986).
5. F. Murabayashi, H. Yamada, T. Yamauchi, T. Ido, T. Nishiyama, K. Shimamura, S. Tanaka, T. Hotta, T. Shimizu, and H. Sawamoto, "2.5V Novel CMOS Circuit Techniques for a 150MHz Superscalar RISC Processor," *IEEE J. Solid-State Circuits* **31**, No. 7, 972–980 (July 1996).
6. T. Thorp, G. Yee, and C. Sechen, "Domino Logic Synthesis Using Complex Static Gates," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1998, pp. 242–247.
7. K. J. Nowka and T. Galambos, "Circuit Design Techniques for a Gigahertz Integer Microprocessor," *International Conference on Computer Design*, 1998, pp. 11–16.
8. J. Pretorius, A. Shubat, and C. Salama, "Latched Domino CMOS Logic," *IEEE J. Solid-State Circuits* **21**, No. 4, 514–522 (August 1986).
9. G. Yee and C. Sechen, "Dynamic Logic Synthesis," *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1997, pp. 345–348.
10. W. Belluomini, D. Jamsek, A. Martin, C. McDowell, R. Montoye, T. Nguyen, H. Ngo, J. Sawada, I. Vo, and R. Datta, "An 8GHz Floating-Point Multiply," *Proceedings of the IEEE International Solid-State Circuits Conference*, 2005, pp. 374–376.
11. R. Montoye, W. Belluomini, H. Ngo, C. McDowell, J. Sawada, T. Nguyet, B. Veraa, J. Wagoner, and M. Lee, "A Double Precision Floating Point Multiply," *Proceedings of the IEEE International Solid-State Circuits Conference*, 2003, pp. 336–337.
12. N. Itoh, Y. Naemura, H. Makino, Y. Nakase, T. Yoshihara, and Y. Horiba, "A 600-MHz 54x54-bit Multiplier with Rectangular-Styled Wallace Tree," *IEEE J. Solid-State Circuits* **36**, No. 2, 249–257 (February 2001).
13. D. Deleganes, M. Barany, G. Geannopoulos, K. Kreitzer, A. Singh, and S. Wijeratne, "Low-Voltage-Swing Logic Circuits for a 7GHz X86 Integer Core," *Proceedings of the IEEE International Solid-State Circuits Conference*, 2004, pp. 336–337.
14. S. Mathew, M. Anders, B. Bloechel, T. Nguyen, R. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit Integer Execution ALU with Dual Supply Voltages in 90-nm CMOS," *IEEE J. Solid-State Circuits* **40**, No. 1, 44–51 (January 2005).

*Received June 21, 2005; accepted for publication August 8, 2005; Internet publication March 3, 2006*

**Wendy Belluomini** *IBM Research Division, Almaden Research Laboratory, 650 Harry Road, San Jose, California 95120 (wb1@us.ibm.com)*. Dr. Belluomini is a Research Staff Member. She received a B.S. degree in computer science from the California Institute of Technology, an M.S. degree in computer science from the University of Washington, and a Ph.D. degree in computer science from the University of Utah. She joined IBM at the Austin Research Laboratory in 1999. Dr. Belluomini's research interests are high-speed circuit design and formal verification.

**Damir Jamsek** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (jamsek@us.ibm.com)*. Dr. Jamsek is currently the manager of the Verification and Analysis and Exploratory Very Large-Scale Integration (VLSI) departments at the Austin Research Laboratory. He received a B.S. degree in mathematics from the University of Dayton, and M.S. and Ph.D. degrees in computer engineering from Syracuse University. He joined the Austin Research Laboratory in 1997. Dr. Jamsek's research interests are high-speed microprocessor design and formal verification.

**Andrew K. Martin** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (akmartin@us.ibm.com)*. Dr. Martin is a Research Staff Member. He received a B.S. degree in computing and information sciences from Queen's University, Canada, and M.S. and Ph.D. degrees in computer science from the University of British Columbia. Dr. Martin joined the IBM Austin Research Laboratory in 2002; his research interests include high-level design languages for hardware design, formal verification, and synthesis.

**Chandler McDowell** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (ichandler@mac.com)*. Mr. McDowell received a B.S. degree in physics from the California Institute of Technology. In 2001 he joined the IBM Austin Research Laboratory, where he has worked on system power measurement, thermoelectric cooling, test for high-speed circuits, gate-oxide defect testing, and leakage measurement.

**Robert K. Montoye** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (montoye@us.ibm.com)*. Dr. Montoye holds a B.S. degree in physics and M.S. and Ph.D. degrees in computer science, all from the University of Illinois. Joining IBM in 1983, he designed and implemented the RS/6000* floating-point unit. After pursuing interests outside IBM from 1990 to 1995, he returned to IBM to focus on finding a lower supply circuit family with state-of-the-art performance and its impact on overall microarchitecture and architecture. He is a member of the IBM Academy of Technology. Dr. Montoye has published numerous technical papers and holds more than 20 patents.

**Hung C. Ngo** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (hcngo@us.ibm.com)*. Mr. Ngo has been with IBM for 17 years. He was a member of the first gigahertz processor prototype team, the S/390* team, and the POWER4* design team.

**Jun Sawada** *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78759 (sawada@us.ibm.com)*. Dr. Sawada received B.S. and M.S. degrees from Kyoto University and a Ph.D. degree from the University of Texas at Austin. He joined the IBM Austin Research Laboratory in 2000. His current research interests include formal specification and verification of VLSI designs.