Brief Papers

High-Throughput Asynchronous Datapath With Software-Controlled Voltage Scaling

Yee William Li, *Student Member, IEEE*, George Patounakis, *Student Member, IEEE*, K. L. Shepard, *Senior Member, IEEE*, and S. M. Nowick, *Member, IEEE*

***Abstract*—Adaptive control of the power supply is one of the most effective variables to achieve energy-efficient computation. In this paper, we describe the development of a high-performance asynchronous micropipelined datapath that provides robust interfaces across voltage domains, performing appropriate voltage level conversions and operating between stages with fanout-of-four delays differing by almost two orders of magnitude. With software-specified throughput requirements, the power supply of the datapath is scaled from 2.5 V to 650 mV using an on-chip dc–dc conversion system. Because of the asynchronous design style, the processor operates continuously during the voltage scaling transitions.**

***Index Terms*—Asynchronous circuits, dynamic voltage scaling, low-power design.**

I. Introduction

POWER consumption has become one of the most important issues in processor design, not only in portable battery-powered applications, but in high-performance desktop and server applications because of packaging and cooling requirements. Dynamic (or adaptive) voltage scaling (DVS) has been widely studied as one of the most effective means of achieving energy-efficient design [1]. Our design uses software to specify datapath pipeline throughput requirements; a control system automatically scales the voltage to just achieve these requirements. Because of the asynchronous design style, the datapath operates continuously during the voltage scaling transitions. We employ an on-chip dc–dc converter that combines linear regulators and switched-capacitor power supplies, using only on-chip components.

Multirate signal processing applications, such as software radio [2], provide the ideal vehicles for exploring performance–power tradeoffs with adaptive voltage scaling. Vector (or stream) dataflow architectures are the natural choice for such applications and benefit considerably from deep pipelining. Building on earlier work [3]–[5] in exploiting the inherent latching properties of dynamic logic to build fine-grained asynchronous pipelines, we develop pipeline circuits which exploit self-resetting techniques to achieve high performance. These circuits introduce robust interlocking to allow for fast or slow environments, to function in the presence of aggressive adaptive voltage scaling, and to handle level conversion between different voltage domains.

Manuscript received August 4, 2003; revised November 20, 2003. This work was supported by the National Science Foundation under Grant CCR-00-86007.

Y. W. Li, G. Patounakis, and K. L. Shepard are with the Columbia Integrated Systems Laboratory, Columbia University, New York, NY 10027 USA (e-mail: ywli@ieee.org; gpat@cisl.columbia.edu; shepard@ee.columbia.edu).

S. M. Nowick is with the Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: nowick@cs.columbia.edu).

Digital Object Identifier 10.1109/JSSC.2004.825246

II. Overall Chip Architecture

The design contains three custom 1K-by-16-bit self-resetting low-power SRAMs with pulsed wordline decoding [6]. Each SRAM has associated with it an address-generation unit that consists of a datapath and an asynchronous burst-mode controller designed using generalized C-elements (gC) [7]. Address generation and array access are pipelined such that the array can supply the datapath with operands without limiting throughput.

The prototype datapath in this test chip is a 16-bit carry-lookahead (tree) adder, implemented with seven (micro)pipeline stages. The basic asynchronous pipeline structure supports a mixture of static and dynamic logic, using single-rail implementation, with a uniform design for the pipeline controls. The pipeline circuits are designed to operate across all process corners from 2.5 V down to 650 mV and continue to correctly handshake with the SRAMs operating at 2.5 V. Additional pipeline stages are used to perform the voltage level conversions across voltage domains, in this case between the datapath and SRAMs. For maximum testability, scannable latches are included in each pipeline stage. In addition, we have placed pads on the critical signals of the pipeline to allow time-domain "picoprobing" of the waveforms on the test chip.

An instruction unit broadcasts the (VLIW-like) instruction word for the vector (or stream) operation to each of the units. The instruction word consists of the starting and ending addresses for each of the SRAMs and the required throughput performance for the datapath unit. The power management system scales the supply for the datapath to meet the throughput requirement specified in the instruction word. An on-chip 40-MHz 6-bit flash analog-to-digital converter (ADC) allows noninvasive transient monitoring of the power supply to test the power management system functionality. Detailed discussions on the power management system, including the dc–dc converters, can be found in [8].

III. Asynchronous Micropipelines

Fig. 1 shows a single stage of a linear pipeline; the top half of the figure contains the control circuits (or local "clock" genera-

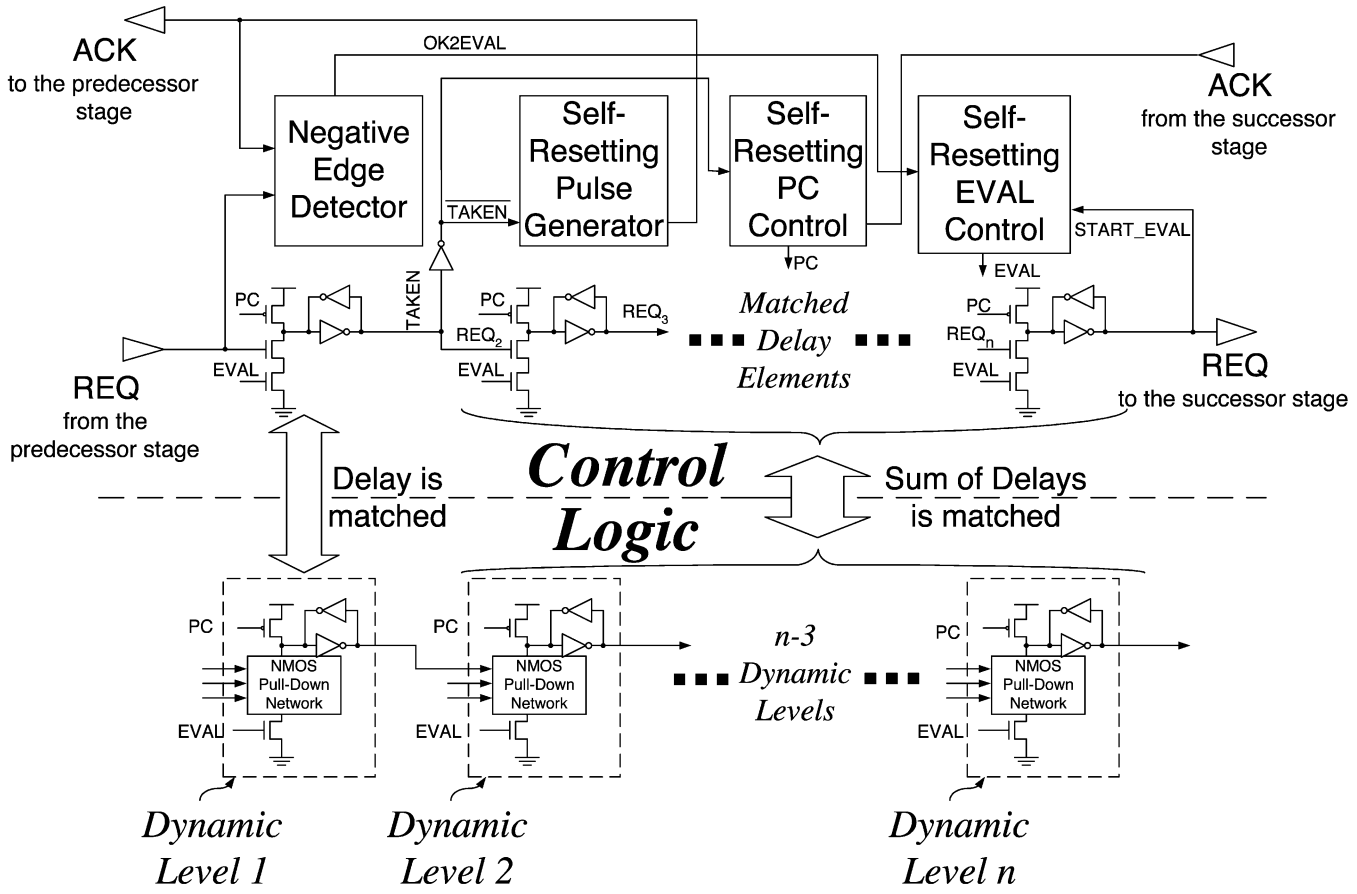

Fig. 1. One pipeline stage.

tors) for the pipeline. In the layout, this resembles a "spine" that runs down the side of the datapath with the area and power overhead of the controller amortized over an entire datapath slice. Adjacent pipeline stages are interlocked by means of the request (**REQ**) and acknowledgment (**ACK**) signals. **PC** and **EVAL** control signals are sent to the stages of the pipeline. Assertion of the **REQ** signal indicates that a new data token is ready to be evaluated by the successor stage while the pulsed **ACK** going to the predecessor stage notifies the latter that it can change its output.

Assuming the logic blocks are implemented as domino logic with their pFET clocked by **PC** and the evaluation foot device clocked by **EVAL**, such a decoupling defines three functional "phases" for the domino levels: precharge, evaluation, and hold [4], [5]. Each stage cycles through these three phases; after evaluation completes, the stage "self-resets" into the hold state. When the successor stage evaluates, the current stage is triggered to precharge and then subsequently self-resets into the evaluation state. The evaluation of a given stage triggers the predecessor stage to complete its entire next cycle: precharge, evaluation of a new data item, and hold. This approach provides high concurrency and reduced cycle time as compared with designs which do not allow a stage to precharge before the successor has finished evaluating [3].

One micropipeline stage consists of n $(n \geq 1)$ dynamic levels. For optimal performance, n should be the same across stages (i.e., balanced). The controller has n domino buffers

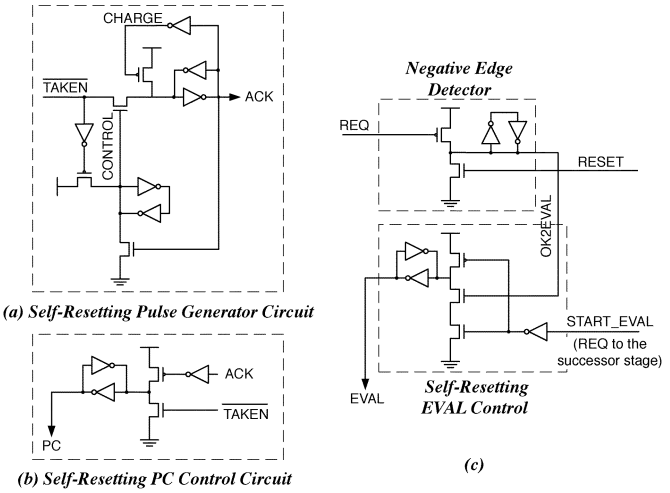

Fig. 2. Negative edge detector and self-resetting **EVAL** control circuits of pipeline controller.

which are sized to match the worst case precharge and evaluation delay of the corresponding logic stages. The outputs of the first and last of these dynamic buffers along with the request from the preceding stage and the acknowledgment from the successor stage are processed by four submodules within the controller (as shown in Fig. 1), described below. Unlike [4] and [5], the pipeline structure uses a pulse-mode protocol [9] for the acknowledgment signal.

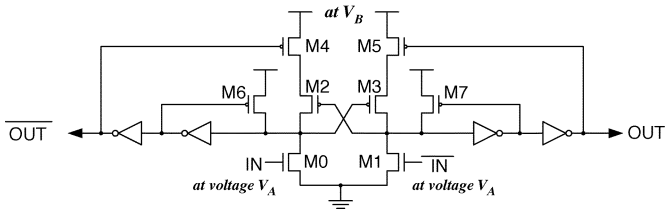

Fig. 3. Enhanced voltage level-shifting circuit.

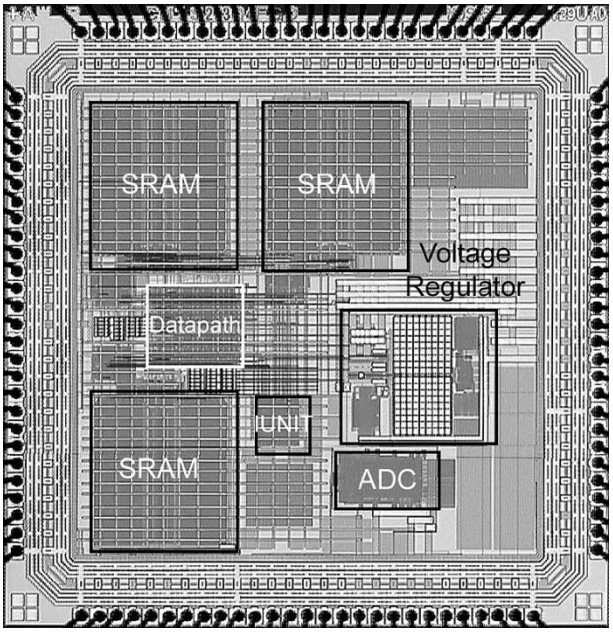

Fig. 4. Die photo of the asynchronous DVS prototype chip in a TSMC 0.25-μm process.

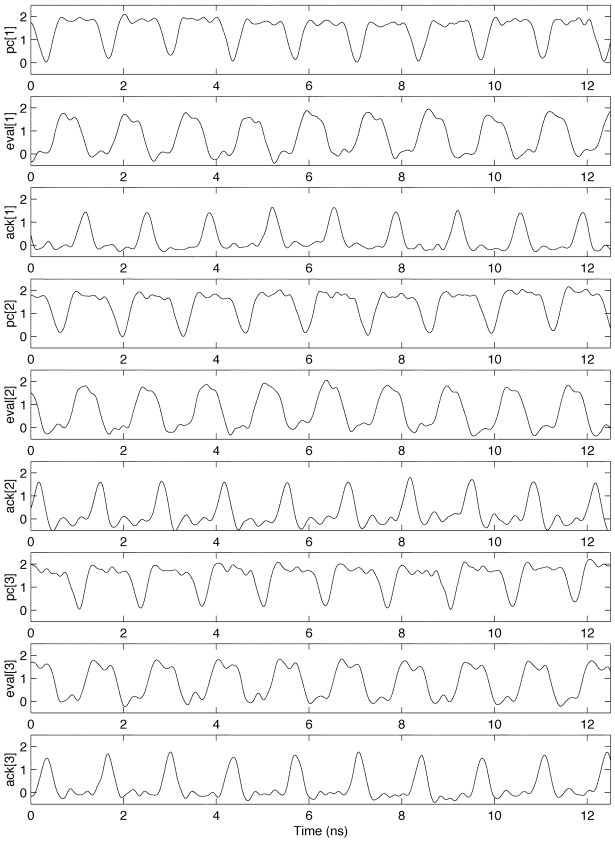

Fig. 5. Full-supply performance of the datapath adder; the measured local circuit cycle time is 1.3 ns, which corresponds to approximately eight FO4 delays.

Self-Resetting Pulse Generator: This circuit [9] acts on the $\overline{\mathbf{TAKEN}}$ signal, converting a $0 \rightarrow 1 \rightarrow 0$ event on $\overline{\mathbf{TAKEN}}$ into a pulse which constitutes the **ACK** signal back to the predecessor stage as shown in Fig. 2(a). The pulse triggers the precharge of the previous stage.

Self-Resetting PC Control: This circuit acts as a "pulse-catcher" for the **ACK** signal from the successor stage and is implemented as shown in Fig. 2(b). Precharging starts once the **ACK** pulse is captured and it is deasserted by $\overline{\mathbf{TAKEN}}$ going to logic one when precharge finishes.

*Self-Resetting **EVAL** Control and Negative Edge Detector*: The **EVAL** signal is asserted when precharge completes and is deasserted at the end of evaluation, putting the current stage in hold. Logically, this function could be implemented using an inverter. However, slow precharge of the predecessor stage would lead to false evaluation of old data. To avoid this problem, the **EVAL** should be asserted only after the predecessor stage is precharged, which is accomplished with the more complex circuit structure of Fig. 2(c).

For correct functionality, the delay blocks in the control spine must track the worst case delays of the corresponding logic stages. We achieve this by duplicating the critical path of the corresponding logic circuit, adding timing margin, and testing thoroughly after extraction across all process corners. The close proximity and equivalent layout of the matched delay element to the corresponding logic minimize the effects of intrachip process variability and ensures that the matched delay elements experience similar temperature and power-supply variations.

Static logic can replace levels 2 to $n-1$ (as shown in Fig. 1) for $n \geq 3$. However, the last level has to be modified to "clock" the evaluate foot device by the $\mathbf{REQ}_n$ signal, rather than **EVAL**, so as to maintain the monotonicity of the output signal. Beginning the pipeline stage with domino logic prevents the corruption of data when the predecessor is precharged. For $n = 2$ pipelines, the first domino level can be eliminated since the predecessor will not be able to precharge before valid data is successfully captured in the latch.

The circuit in Fig. 3 is used to provide low-latency voltage conversion, that is, to convert a digital signal with a logic one value of V_A to a signal with a logic one value of V_B. This circuit differs from the "traditional" level shifting circuit [10] by the addition of devices M4 and M5 and the associated feedback. These devices ensure the pFET pull-up networks are transiently disabled when the state of $\mathbf{OUT}/\overline{\mathbf{OUT}}$ is changing as a result of the switching of $\mathbf{IN}/\overline{\mathbf{IN}}$, reducing the latency of the level converter. Low latency voltage conversion is necessary to prevent the voltage interfaces from becoming a throughput bottleneck. Furthermore, to ensure the pipeline functions correctly across voltage domains, certain elements of the controller must always be run at the full V_{DD} supply.

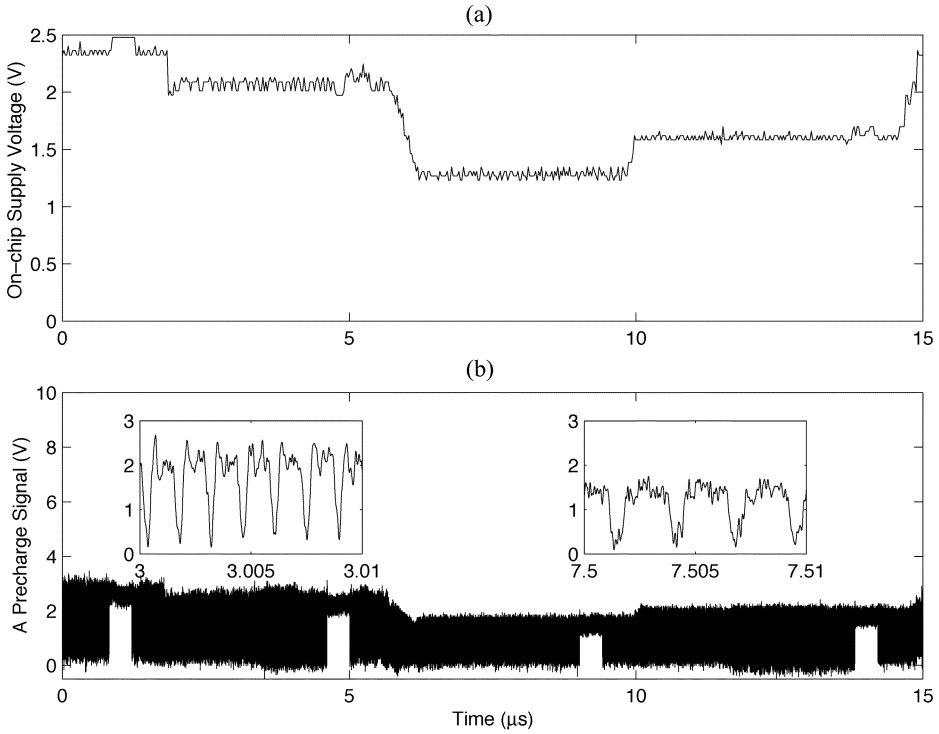

Fig. 6. Datapath functionality with scaling supply. (a) Supply voltage as measured by the on-chip ADC. (b) Picoprobed waveform for one of the **PC** control signals of the datapath.

Throughput (T) is defined as the number of data items processed by the pipeline per unit time. When there are only a few data items in the pipeline, the throughput is said to be data-limited. When the number of data items in the pipeline becomes too high, the pipeline becomes congested and the throughput is limited by the rate at which empty stages (or holes) can move from right to left. These throughput expressions are given by

$$T_{\text{data-limited}} = \frac{K}{Gt_f}, \quad T_{\text{hole-limited}} = \frac{G-K}{Gt_r} \tag{1}$$

where K is the average number of data tokens in the pipe and G is the number of pipeline stages. The forward latency t_f is defined as the time it takes one data token to move from one stage to its successor. The reverse latency t_r is the time it takes a hole to move from one stage to its predecessor.

When the equations (1) are equal, the throughput is optimal and we can calculate $K_{optimal}$ and maximum throughput as

$$K_{\text{optimal}} = \frac{G}{1+\frac{t_r}{t_f}}, \quad T = \frac{1}{t_r+t_f} = \frac{1}{t_{\text{cycle}}} \tag{2}$$

where t_{cycle} is the circuit cycle time.

Dynamic voltage scaling is used to adapt this raw throughput capability to the sample rate demands of the signal processing application. More power must be burned to accommodate high bandwidth (high sample rate) signals but the "intrinsic bandwidth" of the pipelines (as characterized by t_{cycle}) can be reduced (saving power) in the case that low-bandwidth (low sample rate) signals are being processed. Note that the ability to perform this optimization continuously and without having to stop execution is a feature of the asynchronous nature of the chip and is not easily achieved with synchronous techniques.

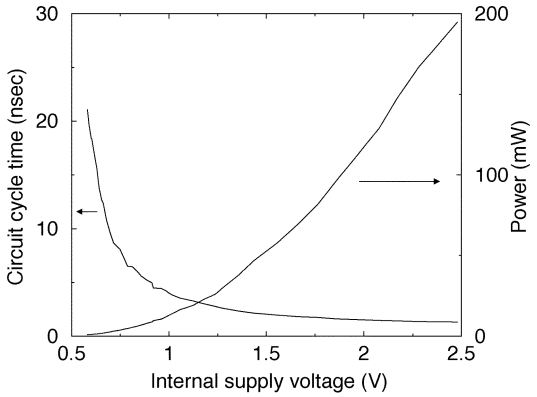

Fig. 7. Performance–power scaling of the asynchronous datapath.

IV. Results

Fig. 4 is the die photo of the 9 mm^2 chip as fabricated in the TSMC 0.25-μm mixed-signal process. We present measured results on the full-supply performance of the datapath and performance-supply scaling. Comprehensive test results for the on-chip dc–dc conversion system are presented in [8].

In Fig. 5, we show the control signals **PC** and **EVAL** and the handshaking signal **ACK** of three consecutive pipeline stages. These signals are directly measured on-chip using GGB picoprobe Model 34A. Ringing in the signal is actually due to the relatively long ground wire of the probe. This is verified by a cleaner signal when the measurement is done

using signal–ground probe. The signals were captured when the internal supply voltage was at 2.48 V, showing a cycle time of 1.3 ns.

Fig. 6 shows the supply voltage measured from the ADC output and one of the **PC** signals measured on-chip. The system is running four instructions, each specifying a different performance. The system continues to function during supply-voltage transitions and the **PC** signal amplitude and period scale accordingly.[1]

Fig. 7 shows the power and performance of the processor with voltage scaling of the datapath. The supply voltage is measured by the on-chip ADC. At the full supply of 2.48 V, the datapath adder runs at 1.3 ns (770 MHz) and burns 195 mW. At the supply of 650 mV, the circuit cycle time is about 21.06 ns (47.5 MHz) and power consumption is 850 μW.

V. Conclusion

We have described the design of a high-performance asynchronous micropipelined datapath that allows "just-in-time" energy-efficient operation through on-chip dynamic voltage scaling in a stream-based dataflow architecture. The asynchronous circuits enable this by providing robust interfaces across voltage domains, performing appropriate voltage level conversions, and allowing uninterrupted operation during voltage-scaling transients. We have demonstrated a high-throughput pipeline architecture in this context that achieves a circuit cycle time of about eight fanout-of-four delays (FO4), allowing 1 Gsample/s throughput at 2.5-V supply in a 0.25-μm CMOS process.

[1]In Fig. 6, we have chosen to "reset" the datapath between instructions, causing the pipeline to stop "ticking" at certain intervals.

Acknowledgment

The authors gratefully acknowledge the contributions of A. Jose, K. Zhang, and B. Liu to the physical design of the chip. They also acknowledge M. Singh for many helpful discussions.

References

[1] T. Burd and R. Brodersen, "Design issues for dynamic voltage scaling," in *Proc. Int. Symp. Low Power Electronics and Design (ISLPED'00)*, July 2000, pp. 9–14.

[2] J. E. Gunn *et al.*, "A low-power DSP core-based software radio architecture," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 574–590, Apr. 1999.

[3] T. E. Williams and M. A. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1651–1661, Nov. 1991.

[4] M. Singh and S. M. Nowick, "Fine-grain pipelined asynchronous adders for high-speed DSP applications," in *Proc. IEEE Computer Society Annu. Workshop VLSI (WVLSI'00)*, Orlando, FL, Apr. 2000, pp. 111–118.

[5] M. Singh, J. A. Tierno, A. Rylyakov, S. Rylov, and S. M. Nowick, "An adaptively-pipelined mixed synchronous-asynchronous digital FIR filter chip operating at 1.3 GHz," in *Proc. 8th Int. Symp. Asynchronus Circuits and Systems (ASYNC 2002)*, Manchester, U.K., Apr. 2002, pp. 84–97.

[6] B. S. Amrutur and M. A. Horowitz, "A replica technique for wordline and sense control in low-power SRAM's," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1208–1219, Aug. 1998.

[7] K. Y. Yun, "Automatic synthesis of extended burst-mode circuits using generalized C-elements," in *Proc. Eur. Design Automation Conf. (EuroDAC)*, Sept. 1996, pp. 290–295.

[8] G. Patounakis, Y. W. Li, and K. L. Shepard, "A fully integrated on-chip dc-dc conversion and power management system," *IEEE J. Solid-State Circuits*, vol. 39, pp. 443–451, Mar. 2004.

[9] S. Schuster *et al.*, "Asynchronous interlocked pipelined CMOS circuits operating at 3.3–4.5 GHz," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC'00)*, Feb. 2000, pp. 292–293.

[10] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *Proc. Workshop Low Power Design*, 1995, pp. 3–8.