

# Asynchronous Datapath with Software-Controlled On-Chip Adaptive Voltage Scaling for Multirate Signal Processing Applications

Yee William Li<sup>†</sup>, George Patounakis<sup>†</sup>, Anup Jose<sup>‡</sup>, K. L. Shepard<sup>†</sup>, and S. M. Nowick<sup>‡</sup>

<sup>†</sup>Columbia Integrated Systems Laboratory, Columbia University, New York, NY

<sup>‡</sup>Department of Computer Science, Columbia University, New York, NY 10027

## Abstract

*Adaptive control of the power supply is one of the most effective variables to achieve energy-efficient computation. In this work, 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 600 mV using an on-chip dc-dc conversion system that combines linear regulators and switched-capacitor power supplies. Because of the asynchronous design style, the processor operates continuously during the voltage scaling transitions.*

## 1 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[1, 2, 3] and is being implemented commercially as one of the most effective means of achieving energy-efficient design. It is well-known that a given computation proceeds in the most energy-efficient manner when the supply voltage is scaled to the point of “just-in-time” operation. Nearly all of these studies or commercial systems are based on clocked operation in which clocks (or clock domains) must be stopped during voltage transitions and new clock frequencies established to support different voltages. In [4], dynamic voltage scaling is applied to a simple self-timed system in which data is supplied to a self-timed pipeline from a FIFO buffer. The “fullness” of the buffer is monitored to determine when the datapath voltage must be servoed to achieve higher or lower speed operation. Our design instead uses software to specify datapath pipeline throughput requirements; an on-chip control systems automatically scales the voltage to just achieve these requirements. Because of the asynchronous design style, the datapath operates continuously during the voltage scaling transitions.

Most DVS systems are based on the idea that multiple power grids are available to be “tapped into” to support multiple voltage operation, which comes at the cost of additional complexity and area[5]. Entire design methodologies have been developed around such a concept of voltage islands[6]. An alternative to a set of externally-generated fixed voltage supplies that are switched into on-chip voltage domains is to provide for dynamic dc-dc conversion, which would allow for continuous scaling and negate the need for multiple global power grids. The most efficient techniques

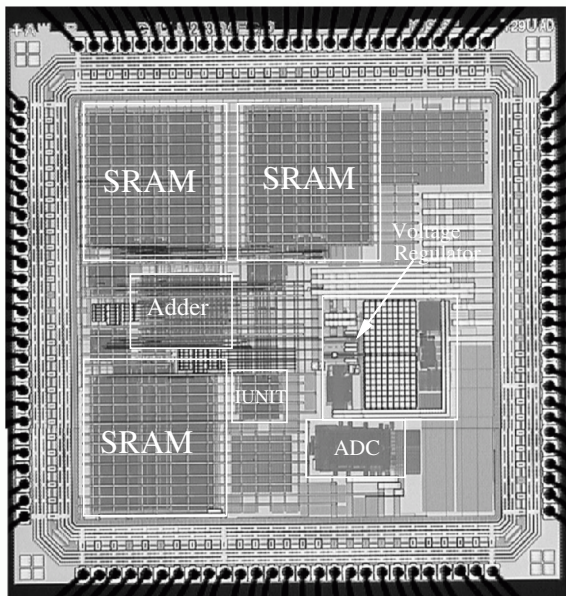
for dc-dc downconversion are based on *buck* converters, which essentially filter a pulse-width modulated (PMW) signal through an LC network to achieve a down-converted dc voltage[7, 8, 9]. Efficiencies of 80 to 90 % can be easily achieved in these systems, although off-chip inductors and (usually) off-chip capacitors are needed. In this work, we deliberately explore an on-chip voltage regulation system that combine linear regulators and switched-capacitor power supplies, achieving lower efficiencies than systems based on buck converters but using only on-chip components.

Multirate signal processing applications, such as software radio[10], provide the ideal vehicle 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. In wireless applications, sample rates can change by a factor of 100 or more during processing, potentially requiring the same pipelined datapath component to adapt to dramatically different throughput requirements but still requiring very high performances at the highest voltages. For our design, we have tried to exploit pipelines with the very highest throughputs (lowest circuit cycle times) at full supply.

The self-resetting CMOS (SRCMOS) design style[11, 12, 13] has been widely recognized as the approach to achieve the highest performance in digital logic by exploiting skewed dynamic logic gates and an asynchronous reset. The pulse-mode nature of the signalling requires careful delay matching of the leading evaluate edge to ensure that the pulses align as evaluation proceeds through the logic. Resets are pipelined with the number of “reset pipeline” stages determined by the circuit cycle time required. Static random access memories (SRAMs) can exploit the careful delay control of SRCMOS design to also wavepipeline the evaluate, allowing the design of memories with access times that are multiples of the cycle time[11]. Consequently, SRCMOS SRAMs have become commonplace in high-performance microprocessors[14]. In datapath applications, logic evaluation may also be wavepipelined, but the lack of interlocking does not allow the pipeline to stall or elastically respond to slow or fast environments (as will naturally occur with DVS) and leaves the functionality vulnerable to process, voltage, and temperature variations.

There has also been considerable work in exploiting the inherent latching properties of dynamic logic to build fine-grained asynchronous micropipelines[15, 16]. This is part of a greater body of literature on asynchronous micropipelines[17, 18, 19, 20]. Unlike wavepipelined designs, control signals (and interlocking) are introduced. In this paper, we leverage these approaches to develop asynchronous fine-grained micropipelined structures that allow one to exploit the inherent latching properties of dynamic logic but mix static and dynamic circuits. This allows the selective exploitation of static logic in cases in which dual rail distribution would be area inefficient or power-hungry. A “bundled data” approach is employed; the controller exploits self-resetting techniques to achieve high performance but introduces robust interlocking to allow for slow environments, to function in the presence of aggressive adaptive voltage

\*This work was supported by the National Science Foundation under grant CCR-00-86007.



**Figure 1: Die photo of the asynchronous DVS prototype chip in a TSMC 0.25 $\mu$ m process**

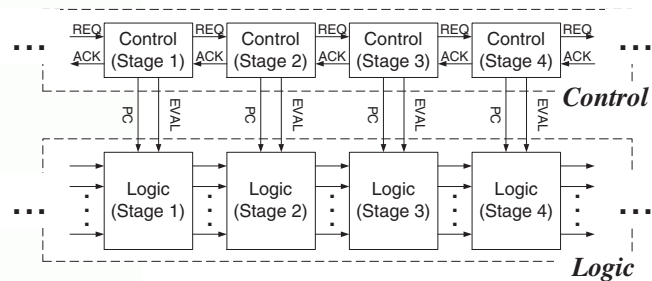
scaling, and to handle level conversion between different voltage domains.

In this paper, we describe our prototype chip, fabricated in a TSMC 0.25 $\mu$ m process, which combines aggressive asynchronous micropipelines, SRCMOS SRAMs with asynchronous extended-burst-mode controllers for address generation, and on-chip dc-dc converters for software-controlled adaptive voltage scaling. In Section 2, we present the overall architecture of the chip. The pipelined datapath circuits and their timing constraints are introduced in Section 3, while Section 4 considers the unique issues in the voltage domain interfaces. Section 5 describes the power management circuits. Power and performance results are presented in Section 6. Section 7 concludes.

## 2 Overall chip architecture

Figure 1 is the die photo of the 9mm<sup>2</sup> chip as fabrication in the TSMC 0.25 $\mu$ m mixed-signal process. The chip is packaged in a 108-pin ceramic PGA, and the associated test board includes FPGAs to interface the chip to the serial ports of a PC for testing.

The design contains three custom SRAMs operating at an unscaled 2.5 V supply. Each SRAM is 1K-by-16 bits; the SRAMs are self-resetting, low-power SRAMs with pulsed word-line decoding[21]. Each SRAM, which contains about 100,000 transistors and is about 1.2 mm<sup>2</sup> in area, is controlled by an address-generation unit that consists of an address-generation datapath and an asynchronous burst-mode controller designed using generalized C-elements (gC)[22]. This asynchronous control unit generates addresses for the SRAM unit from a specified starting address to specified ending address for array reads and writes. Address generation and array access are pipelined such that the array can supply the datapath with operands without limiting throughput.



**Figure 2: High level view of the asynchronous micropipelines. The control circuits run as a modular “spine” down the side of the datapath.**

The prototype datapath in this testchip is a simple 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 with a uniform design for the pipeline controls. The pipeline circuits, described in more detail in Section 3, 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, described in Section 4, are used to perform the voltage level conversions from the scaled datapath supply to the SRAM 2.5 V supply.

For maximum testability, scannable latches are included in each pipeline stage. When the pipeline is stalled<sup>1</sup>, the latches can be used to sample the data in each stages; this data can be subsequently scanned out for debug. In addition, we have placed 10 $\mu$ m  $\times$  10 $\mu$ m pads on the critical signals of the pipeline to allow time-domain “picoprob” of the waveforms on the testchip.

An instruction unit broadcasts an instruction word to each of the units. In this simplified testchip, this instruction word consists of starting and ending addresses for each of the SRAMs and the required throughput performance for the datapath unit. In execution, streams of data are pumped from two SRAMs with the result stored in the third. The power management system, described in Section 5, scales the supply for the datapath to meet the performance requirement specified in the instruction word. An on-chip flash analog-to-digital converter (ADC) allows noninvasive transient monitoring of the power supply to test the power management system functionality.

## 3 Asynchronous micropipelines

Figure 2 shows several stages of a linear pipeline; the top half of the figure contains the control circuits (or local “clock” generators) 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 acknowledge (ACK) signals. PC and EVAL control signals are sent to the stages of the pipeline.

For now, we assume that the stages are implemented as conventional domino logic (we consider mixing static and domino logic later in this section) with a precharge pFET device clocked by PC and an evaluate foot device clocked by EVAL. Following Reference [16], such a decoupling defines three functional “phases” for the domino stage, precharge, evaluation, and hold. Each stage cycles through these three phases; after evaluation completes, the stage “self-resets” into the hold stage. When the successor stage

<sup>1</sup>In practice, pipeline stalls are accomplished by setting an ending address for the write SRAMs that is less than the ending address of the read SRAMs.

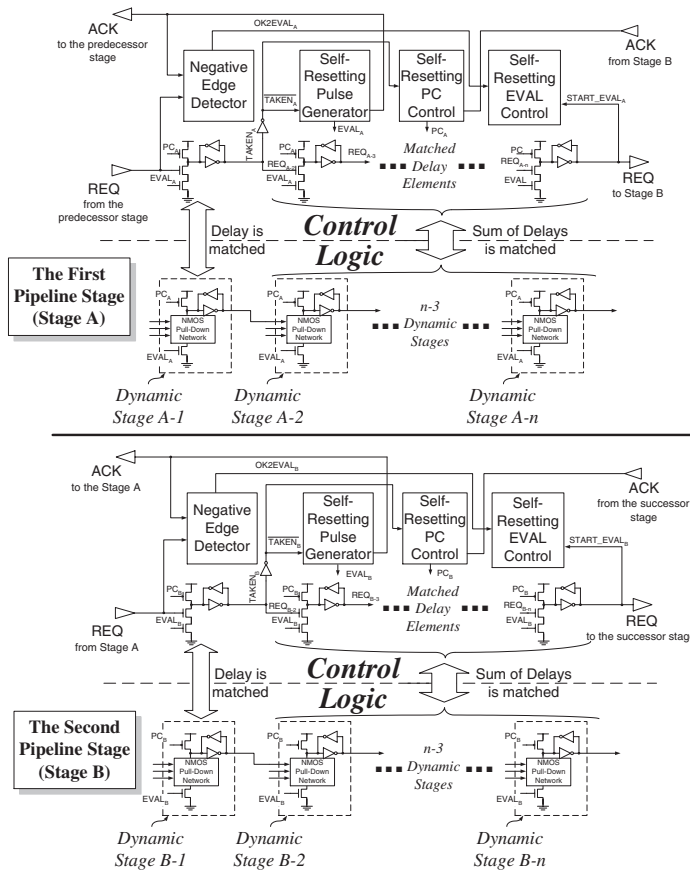


Figure 3: Two self-resetting asynchronous pipeline stages.

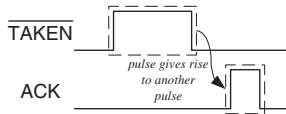


Figure 4: Functional encapsulation of the self-resetting pulse generator.

evaluates, the current stage is triggered to precharge and then subsequently “self-resets” into the evaluation state. The high-throughput protocol is similar to that of References [16, 23]; the evaluation of a given stage triggers the predecessor stage to complete its entire next cycle: precharge, evaluation, and hold of a new data item. This provides high concurrency and reduced cycle time, allowing a stage to evaluate before successors have begun precharging. This design improves on that of References [16, 23] by implementing a high-performance controller with only low-logical-effort circuit structures. Moreover, the controller does so while adding the additional interlocking necessary to ensure pipeline functionality with widely disparate intrinsic performance differences between stages that can occur (at least transiently) across voltage domains.

### 3.1 Pipeline implementation details

Figure 3 shows two pipeline stages in detail; one micropipeline stage consists of  $n$  ( $n \geq 1$ ) dynamic pull-down networks. For optimal performance, the pipeline should be balanced with  $n$  generally

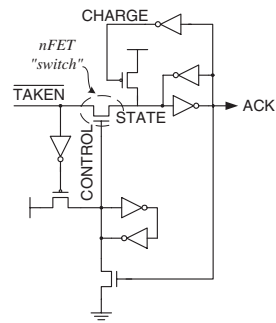


Figure 5: Self-resetting pulse generator circuit.

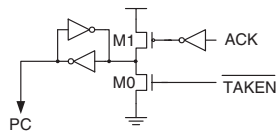


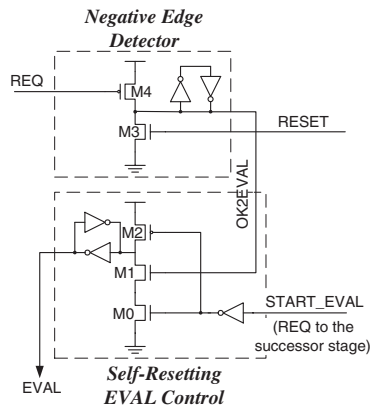
Figure 6: Self-resetting PC control circuit of pipeline controller.

chosen to be the same across pipeline stages. The controller has  $n$  domino buffers which are sized to match the 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 acknowledgement from the successor stage are processed by four modules within the controller (as shown in Figure 3) independently described below.

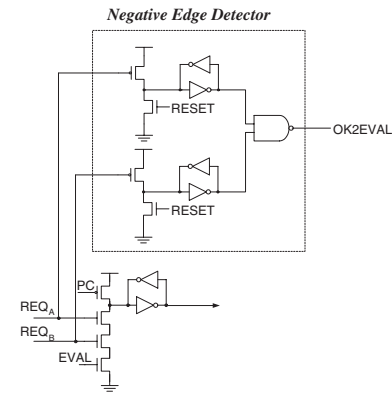
**Self-resetting pulse generator.** This circuit acts on the **TAKEN** signal, converting a  $0 \rightarrow 1 \rightarrow 0$  event on **TAKEN** into a pulse which constitutes the **ACK** signal back to the predecessor stage as shown in Figure 4. Logically, the self-resetting pulse generator detects that the current stage has been precharged and subsequently the new data token from the previous stage has been successfully captured by the first domino stage. The pulsed acknowledge informs the previous stage that it can alter its output (precharge).

This functionality is realized very elegantly (and with comparatively low logical effort) by the switch circuit shown in Figure 5. introduced in the context of the IPCMOS pipelines by Schuster, et al.[19]. To understand the operation of the switch, let us assume initially that both the **STATE** node and **TAKEN** node are at logic one and logic zero, respectively. When the **TAKEN** rises to one, the nFET switch closes and the **STATE** node stays at one. Now, when the **TAKEN** node returns to zero, the **STATE** node is pulled to zero and the **ACK** signal goes to one. **ACK** going to one opens the nFET switch. The **STATE** node is charged to one and **ACK** returns to zero. In summary, a positive pulse at **ACK** is generated when a positive pulse is observed at **TAKEN**, which is the desired functionality.

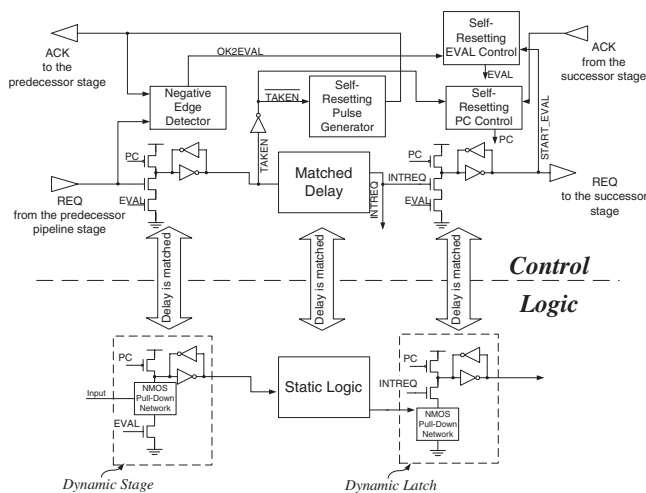
**Self-resetting PC control.** This circuit acts as a “pulse-catcher” for the **ACK** signal from the successor stage and is implemented as shown in Figure 6. The current stage (in its hold phase) is waiting for the acknowledgement from the successor stage before precharging. As such, both **PC** and **TAKEN** are logic one. Once the pulse on **ACK** arrives, **PC** will be pulled to zero and the precharge phase begins. When precharge completes, **TAKEN** is set to one and the **PC** signal is deasserted, “self-resetting” the stage into the evaluation phase. Note that a maximum pulse-width constraint exists for **ACK** to avoid short-circuit power dissipation. The pulsed **ACK** must be reset to zero before **TAKEN** is deasserted. The (func-



**Figure 7: Negative edge detector and self-resetting EVAL control circuits of pipeline controller.**



**Figure 9: Pipeline joins (only modified control spine is shown)**

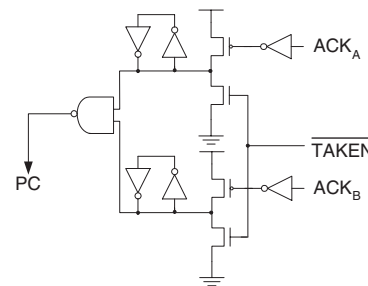


**Figure 8: Pipeline stage with static logic**

tionally) more important minimum pulse width requirement is considered below.

**Self-resetting EVAL control.** Logically, this circuit must deassert **EVAL**, putting the stage into the hold phase once evaluation has completed (**REQ** to the successor stage has gone high). A simple inverter for this function may lead to failure if the previous stage is slow, for example, if the previous stage is running at a significantly lower voltage and its precharge is slow to complete. In this case, the **REQ** from the predecessor stage will still be one, since it has not precharged, when the current stage has been reenabled for evaluation. As a result, the current stage will falsely evaluate the same data token twice. Furthermore, an extra **ACK** pulse will be sent to the predecessor stage. To avoid this problem, additional interlocking is required to check that the previous stage has precharged and assertion of the **REQ** signal signifies a new datum at the input, a function performed by the negative edge detector block.

**Negative edge detector.** The output of the negative edge detector (the signal **OK2EVAL**) is combined with the **REQ** signal to the successor stage to produce the **EVAL** control, as shown in Figure 7. This prevents the circuit from entering the evaluation phase unless the previous stage has new data. The negative edge detector block



**Figure 10: Modification to the self-resetting PC control to support pipeline forks**

is reset using the **ACK** pulse being sent to the predecessor stage, since this indicates the time point from which one must detect the falling edge of **REQ**.

### 3.2 Incorporating static logic

For  $n \geq 3$ , static logic can be easily incorporated into a pipeline by ending the stage with the latch circuit shown in Figure 8 and beginning each pipeline stage with at least one domino logic stage. Beginning the pipeline stage with domino logic prevents the corruption of data when the predecessor is precharged. The ending domino latch gives the static stage a signalling protocol to the fan-out stages that is identical to a domino stage. When the successor stage sends back an **ACK** pulse, the latch precharges. The **INTRREQ** signal (for “internal request”) ensures that the latch does not “open” until the static logic has stably evaluated.

For  $n = 2$  pipelines, the beginning domino stage can be eliminated since the predecessor will not be able to precharge before valid data is successfully captured in the latch.

### 3.3 Pipeline forks and joins

Pipeline joins can be easily supported with an additional nFET input in the first stage of the control spine with the associated enhancement to the negative edge detection circuit as illustrated in Figure 9. The additional complexity in the negative edge detection circuit does not affect the circuit cycle time. Pipeline forks can be supported by a simple modification of the self-resetting PC control circuit as shown in Figure 10.



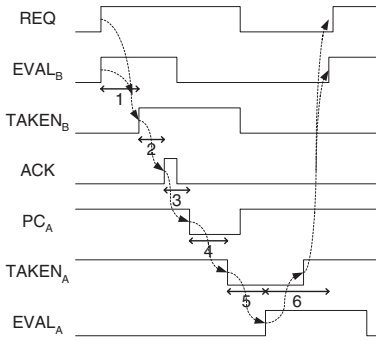


Figure 11: Timing diagram

### 3.4 Pipeline circuit cycle time and throughput

One can easily determine the components of the circuit cycle time for this pipeline by “simulating” the interaction of two pipeline stages, as shown in Figure 3. To begin this simulation, we assume the initial condition that the first pipeline stage has completed its evaluation and the second pipeline stage is about to enable evaluation (poised to assert its **EVAL** signal). With this choice of starting point, the cycle time will be defined as the time between the start of two successive evaluations of a pipeline stage. (In this simulation, we will be referencing the second pipeline stage for the cycle time determination.) With reference to the timing diagram of Figure 11, the events defining one cycle are:

1. Evaluation of dynamic stage B-1 ( $t_{eval\_b1}$ )
2. Generation of the **ACK** pulse ( $t_{ack\_pulse}$ )
3. Assertion of the **PC\_A** signal ( $t_{assert\_pca}$ )
4. Precharge of the first pipeline stage ( $t_{pc}$ )
5. Assertion of the **EVAL** signal ( $t_{assert\_eval}$ )
6. Evaluation of dynamic stage A-1 to A- $n$ . ( $t_{eval\_a}$ )<sup>2</sup>

leading to a circuit cycle time given by:

$$t_{cycle} = t_{eval\_b1} + t_{ack\_pulse} + t_{assert\_pca} + t_{pc} + t_{assert\_eval} + t_{eval\_a} \quad (1)$$

Note that the **OK2EVAL** assertion is not in the critical path as it happens concurrently with events 5 and 6.

Pipeline performance can also be described in terms of the forward (or data) latency and the reverse (or hole) latency. When the number of data items in the pipeline is small, the throughput ( $T$ ), defined as the number of data items processed by the pipeline per unit time, is said to be data-limited and given by the expression:

$$T_{data-limited} = \frac{K}{GT_f} \quad (2)$$

where  $K$  is the average number of data tokens in the pipe and  $G$  is the number of pipeline stages.  $T_f$ , the forward latency, is defined as the time it takes one data token to move from one stage to its successor. In terms of the functional phases of the pipeline, it is defined as the time from the beginning of evaluate of a stage to the beginning of evaluation of the successor stage:

$$T_f = t_{eval\_a} \quad (3)$$

<sup>2</sup>We can readily see that  $t_{eval\_a1} \approx \frac{1}{n} t_{eval\_b1}$

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:

$$T_{hole-limited} = \frac{G - K}{GT_r} \quad (4)$$

$T_r$ , the reverse latency, is the time it takes a hole to move from one stage to its predecessor, or the time from the completion of precharge of a stage to the completion of precharge of the successor stages:

$$T_r = t_{eval\_b1} + t_{ack\_pulse} + t_{assert\_pca} + t_{pc} + t_{assert\_eval} \quad (5)$$

Maximum throughput is determined by the condition in which the throughput of equations 2 and 4 are equal, which defines the “optimal” pipeline filling:

$$K = \frac{G}{1 + \frac{T_r}{T_f}} \quad (6)$$

as well as the maximum throughput<sup>3</sup>:

$$T = \frac{1}{T_r + T_f} = \frac{1}{t_{cycle}} \quad (7)$$

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 dissipated 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.

### 3.5 Timing constraints

Correct operation of the pipeline depends on a number of straightforward timing constraints.

- **Minimum pulse width requirement on ACK.** If the **ACK** pulse is not wide enough, the self-resetting PC control circuit of the predecessor stage will not be able to capture it and the **PC** signal of this stage will not be correctly asserted. The predecessor stage will not be triggered to precharge and the pipeline will stall indefinitely. To avoid this, the following must be true:

$$t_{assert\_pc} < t_{ack\_pulse\_width} \quad (8)$$

The pulse width can be easily tuned to meet this constraint by tuning or changing the number of inverters between the **ACK** and **CHARGE** signals in the self-resetting pulse generator circuit of Figure 5.

- **Time between precharge completion and start of evaluation.** Immediately after the precharge cycle, the self-resetting loop in the self-resetting PC control circuit will deassert the **PC** signal. Concurrently, the self-resetting EVAL control circuit will assert the **EVAL** signal. If the time for deassertion of **PC** is less than that for the assertion of **EVAL**<sup>4</sup>, then the

<sup>3</sup>This simple analysis, of course, assumes that the pipeline stages all have the same circuit cycle time. If this is not the case, then the pipeline stage with the slowest cycle time will become the bottleneck and limit the overall throughput performance.

<sup>4</sup>This cannot be visualized in Figure 11 as edges are assumed to have zero slew time; these difficulties arise in the case of finite slews.

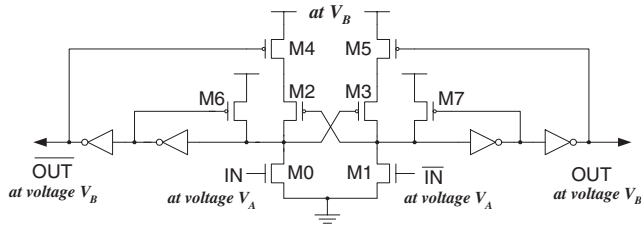


Figure 12: Enhanced voltage level-shifting circuit

stage may not have adequate time to precharge, resulting in functional failure.

$$t_{deassert\_pc} < t_{assert\_eval} \quad (9)$$

This timing constraint can be met by padding extra delay between **REQ** to the successor stage and **START\_EVAL**.

While not explicitly timing constraints, there are two other important sizing and delay matching issues that deserve attention:

- *Switch point for logic in self-resetting loop.* The self-resetting nature of the **PC** and **EVAL** signals may, in extreme cases, lead to failure. In the self-resetting PC control circuit of Figure 6, the precharge of the stage may not be complete but the **TAKEN** signal may be low enough to deassert **PC**, leading to functional failure. Similarly, in the self-resetting EVAL control circuit of Figure 7, the stage may not have finished evaluation but the **REQ** signal is high enough to deassert **EVAL**, leading to functional failure. To avoid this, we deliberately skew the self-resetting control circuits, weakening the nFET M0 in the self-resetting PC control circuit (see Figure 7) and pFET the M2 in the self-resetting EVAL control circuit.
- *Driving capability of PC and EVAL.* In Figure 3, we have assumed that **PC** and **EVAL** have sufficient drive to drive all the bits of the associated datapath “slice.” In practice, this is not the case and one or both of two solutions must be pursued. One can make the whole control spine bigger to provide larger driving capability. Alternately, one could buffer **PC** and **EVAL** to drive the large capacitive load. Buffering adds skew between the control circuits and the buffered versions of the **PC** and **EVAL** signals reaching the load. This skew does not affect the datapath functionality as long as this skew is balanced across the pipeline stages; that is, each pipeline stage sees the same skew.

## 4 Voltage interface

Integrating pipeline stages running at different voltages is a difficult design challenge. Not only must these *voltage interface* circuits between voltage domains translate voltage levels with minimal added latency, they must robustly maintain the pipeline protocol, even in the presence of (potentially) vastly disparate circuit delays across the interface.

### 4.1 Level conversion

The circuit in Figure 12 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  $V_B$ ; the entire circuit is operated at a supply voltage of  $V_B$  in this case. This circuit differs

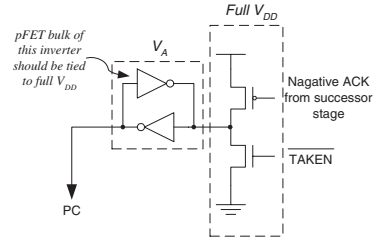


Figure 13: Self-resetting PC control for slow pipeline stage.

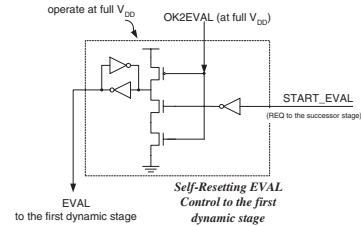


Figure 14: Self-resetting EVAL control for the first domino stage of *B* in extreme environment.

from the “traditional” level shifting circuit[24] 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 **OUT/OUT** is changing as a result of the switching of **IN/IN**, reducing the latency of the level converter. Low latency voltage conversion is necessary to prevent the voltage interfaces from becoming a throughput bottleneck.

### 4.2 Pipeline controls between voltage domains

Consider the case in which a pipeline stage *A* running at supply  $V_A$  is feeding data to a pipeline stage *B* running at supply  $V_B$  in Figure 3.  $V_{DD}$  denotes the unregulated full supply. The controller of *A* runs at the  $V_A$  supply and the controller of *B* runs at the  $V_B$  supply except for the following enhancements to achieve robust operation:

- The request of *A* to *B* is voltage converted from  $V_A$  to  $V_{DD}$  through the circuit of Figure 12.
- The negative edge detector and self-resetting pulse generator of *B* are run at  $V_{DD}$ . This requires an additional level conversion in the *B* controller to convert **TAKEN** to a  $V_{DD}$  reference.
- The self-resetting PC control of *A* is modified as shown in Figure 13 to capture the **ACK** pulse at  $V_{DD}$ . The latch of this circuit may still operate at  $V_A$ ; in this case, the feedback pFET must have its body tied to  $V_{dd}$  to avoid forward biasing its drain-body junction.
- The **EVAL** signal of the first domino stage of *B* is provided by the *additional* self-resetting EVAL control circuit shown in Figure 14. This allows *B* to quickly enter the hold phase from evaluation in the case that *B* is much slower than *A*.

Without these changes, the  $V_A > V_B$  case becomes problematic. These changes ensure that the **ACK** pulse runs at full supply with invariant pulse width. Without these adjustments, the **ACK** pulse to *A* becomes long, resulting in short-circuit currents in the self-resetting PC control circuits of *A*. Furthermore, without the

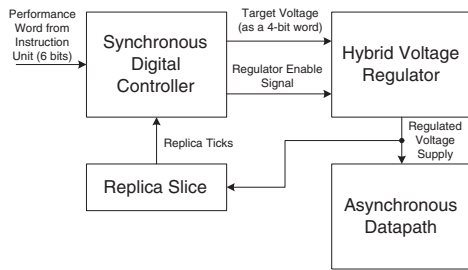


Figure 15: Power management system.

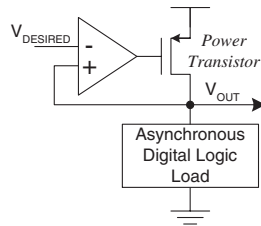


Figure 16: Linear voltage regulator.

last change, if  $A$  is running much faster than  $B$ , it is possible for  $A$  to cycle from precharge to evaluation *before*  $B$  is able to enter the hold phase, disrupting the pipeline protocol.

## 5 Power management system

The power management system outlined in Figure 15 is responsible for efficiently scaling the supply voltage for the datapath to just meet the performance target specified in the instruction word. A synchronous state machine accomplishes this by a monotonic search starting from the voltage established for the previous instruction. The search direction is determined by comparing the previous performance target with the current one; the state machine stops searching once it has reached the required performance. If the ideal target voltage lies between two discrete values, the greater of the two is chosen to guarantee a minimum circuit cycle time. The voltage-to-performance conversion is achieved via a replica slice of the unit being regulated and a counter to capture the number of replica “ticks” during a controller clock cycle. The replica is a ring structure and is initialized so that the number of data tokens captured in the ring matches the “maximum throughput” filling factor defined in Equation 6. The replica consumes 2.4% of the area and power of the associated datapath. This approach proves superior to continuous-time monitoring of the performance, which would introduce another feedback loop (and hence greater chances for instability) in the system. This controller burns little power and has a small area footprint of  $0.026\text{ mm}^2$ .

An equally important aspect of the power management systems is the design of efficient dc-dc converters to generate the required supply voltages. Dc-dc downconversion from a  $V_{dd}$  supply can be generally accomplished in one of three ways: “buck” converters, switched capacitor dividers, and linear regulators. In theory, the “buck” converters can achieve 100% efficiency if all the components are ideal. Partially integrated “buck” converters have achieved 80-95% efficiency [7, 9, 8]. Unfortunately, the inability to integrate large inductors with high  $Q$  on-chip leads to the necessity to build the LC filter of the buck converter with off-chip components. This increases the pin requirements, reduces efficiency, and makes fine-grain voltage domains impractical and expensive.

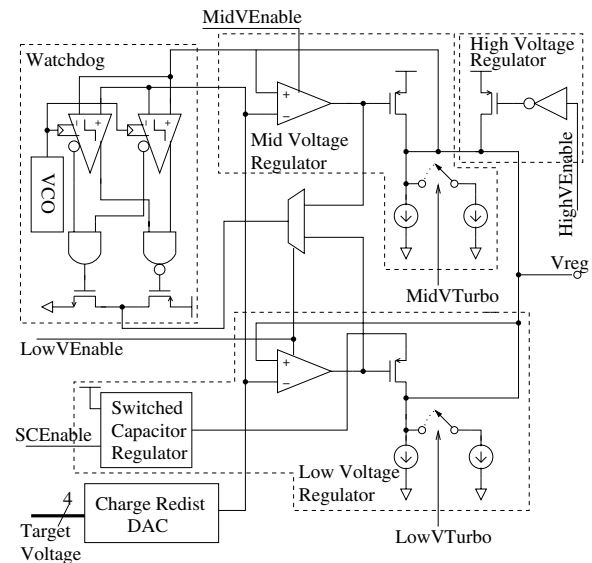


Figure 17: “Hybrid” voltage regulator.

Linear regulators (see Figure 16) are the most easily integrable dc-dc converters because they consist of only transistors, but they have poor efficiencies at low output voltages. Linear regulators have found applications in low-power digital design [25, 26, 27, 28]. Conceptually, the linear regulator is a voltage controlled resistor that forms a resistive voltage divider with the load. The variable resistance is controlled by an operational amplifier (op amp) that monitors the output voltage and compares it to the desired voltage. Therefore, higher voltage drops across the linear regulator’s power transistor result in more power being dissipated without doing “useful” work. Furthermore, a linear regulator’s op amp requires quiescent current that must be considered when the load is drawing little current. The bias current of the linear regulator must be increased if a fast response time is required. The design of linear regulators is also complicated by the wide range of loading characteristics a digital circuit produces during operation.

Switched capacitor voltage dividers (SCVDs) can trade efficiency for integrated chip area and can achieve higher efficiencies than linear regulators at low voltage. The efficiency of an *ideal* SCVD is inversely proportional to the output voltage ripple; therefore, it is proportional to the size of the switching capacitors and frequency of switching for a fixed load current. Real SCVDs incur a power dissipation overhead due to real CMOS switches and implementation details of the on-chip switching capacitors. Real switches have a finite conductance when on and need charging/discharging currents to control them. Therefore, there exists a frequency beyond which the efficiency begins to decrease due to the dynamic power dissipation. Increasing the values of the switched capacitors increases the efficiency only at the cost of increased area. SCVDs have been applied to low power medical implants [29] and inductorless high power density dc-dc conversion [30, 31].

A possible approach to the second goal of the power management system is to use a hybrid voltage regulator scheme (as shown in Figure 17) to trade off area for power efficiency. The maximum core voltage of 2.5V is supplied by a large PFET power transistor with the gate tied to 0 V. Voltages down to 1.0 V are supplied by linearly regulating down from 2.5 V. The lowest voltage range (990 mV down to 650 mV) is supplied by linearly regulating from a 1.25V supply generated by a switched capacitor regulator. Control signals to select the appropriate regulator and allow for smooth transitions between regulator boundaries are generated by the same

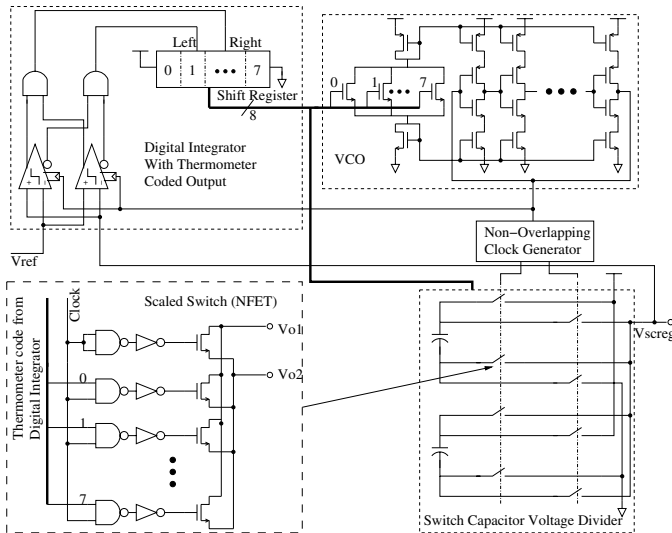


Figure 18: Switched capacitor voltage regulator.

synchronous state machine that determines the target voltage. Furthermore, the state machine controls the frequency and magnitude of the pulses from the “watchdog” unit. At lower voltages, the digital logic is running at a much slower cycle time and, therefore, does not need to be monitored as frequently. The closed loop stability of the digital feedback is guaranteed by setting the gain of the “watchdog” appropriately for each frequency.

The switched capacitor regulator (SCR) block diagram is shown in Figure 18. Scaling techniques were employed in the SCR to minimize the overhead of switching the capacitors. Further energy savings were attained through low parasitic on-chip metal-insulator-metal (MIM) capacitors instead of the more dense MOS capacitors. Simulation results predicted a two- to three-fold increase in efficiency when using MIM capacitors as compared to MOS capacitors for an eight-fold area penalty. Switching frequency scaling was also employed in the SCR by monitoring the minimum output voltage on the 1.25V supply. A clocked comparator followed by a thermometer-coded digital integrator was used to implement this function. The digital integrator proved more energy efficient than its continuous time analog counterpart and provided the proper driving signals directly. The “height” of the thermometer code output of the digital integrator is a measure of the current demanded by the load circuit. Therefore, the minimum frequency and switch width necessary to support a range of load currents can be determined *a priori* through simulation. The thermometer-coded output of the digital integrator also eliminates the need for a glitch-free decoder which is necessary when using a binary representation. The simulated efficiency of producing approximately  $\frac{V_{dd}}{2}$  using this method is greater than 60% under most loads and thus more efficient than the ideal efficiency of a linear regulator (50%). Generating more than one supply from an SCR proved to lower the efficiency due to the overhead of the extra switches and clock phases necessary for multiple outputs.

Approximately 25 pF of explicit thin-oxide on-chip decoupling capacitance on the supply node of the datapath is adequate to filter out most of the current fluctuations under normal pipeline operation. The asynchronous nature of the pipeline helps in this regard by “spreading” out the current demands of the digital logic. This means that the bandwidth of the linear regulators can be kept low (with low quiescent current) for maximum power efficiency. This low-bandwidth regulator, however, does have difficulties with the current transients associated with “turning on” or “turning off” the

datapath; that is, transients associated with beginning to pump data into the pipeline at startup or draining data from the pipeline at completion. These current transients are managed with the addition of a digital “watchdog” circuit which samples the regulated output voltage and drives the gate of the power transistor in the continuous time loop such that the output voltage is within 100 mV of the target voltage. This approach provides for a more power-efficient design than increasing the large signal bandwidth (and thus the quiescent current) of the op amp in the regulator. The complete system including linear regulators, switch capacitor regulator, digital watchdog, and digital controller occupies  $0.4 \text{ mm}^2$ .

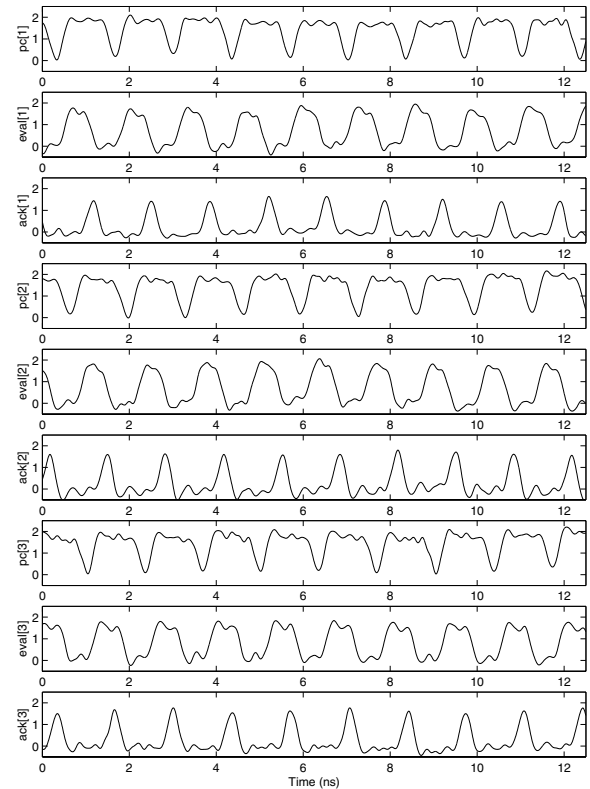


Figure 19: Full supply performance of the datapath adder.

## 6 Results

We present results on the measured full-supply performance of the datapath, performance-supply scaling, and regulator efficiency.

### 6.1 Performance

In Figure 19, 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 Model 34A pico-probes. Ringing in the signal is actually due to the relatively long ground wire of the probe. The signals were captured when the internal supply voltage was at 2.48V, showing a cycle time of 1.3ns.

Figure 20 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. Between instructions, the datapath is reset and the pipeline stops “ticking.”



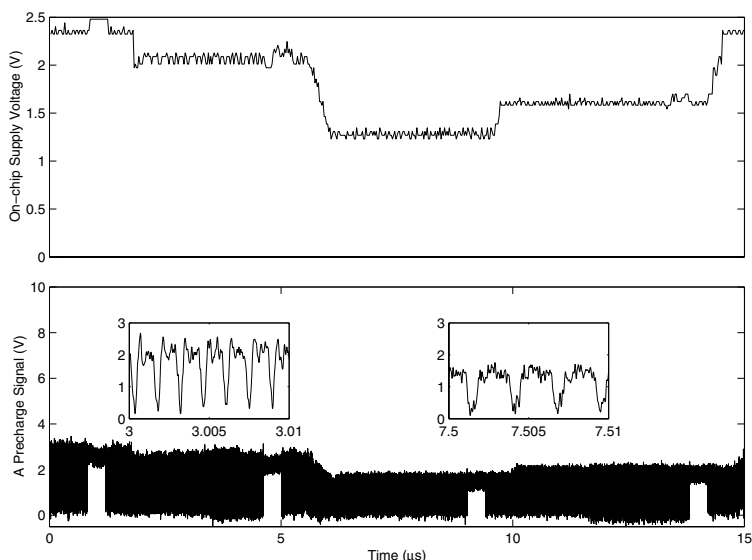


Figure 20: Datapath functionality with scaling supply.

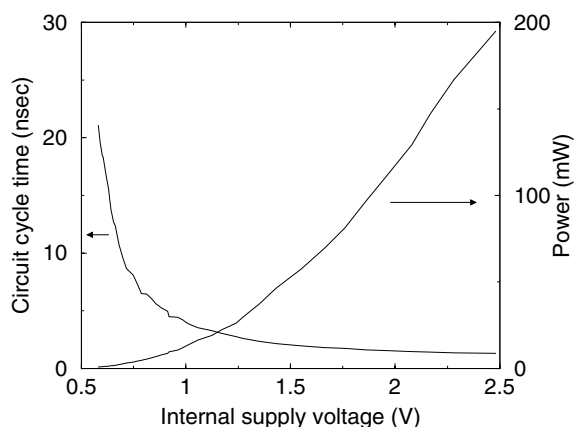


Figure 21: Performance-power scaling of the asynchronous datapath.

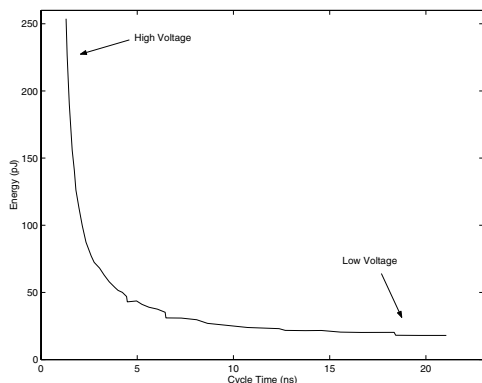


Figure 22: Energy versus Cycle-Time

## 6.2 Dynamic voltage scaling performance

Figure 21 shows the power and performance of the processor with voltage scaling of the datapath. The supply voltage is that measured

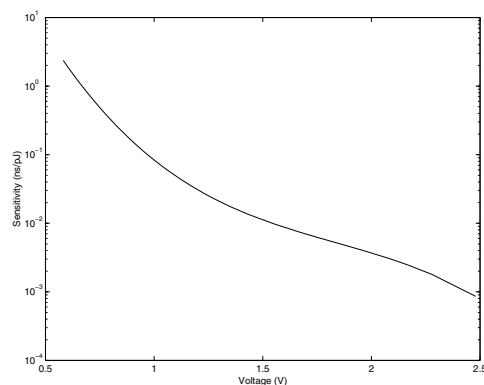


Figure 23: Sensitivity  $\frac{(\frac{\partial D}{\partial V})}{(\frac{\partial E}{\partial V})}$

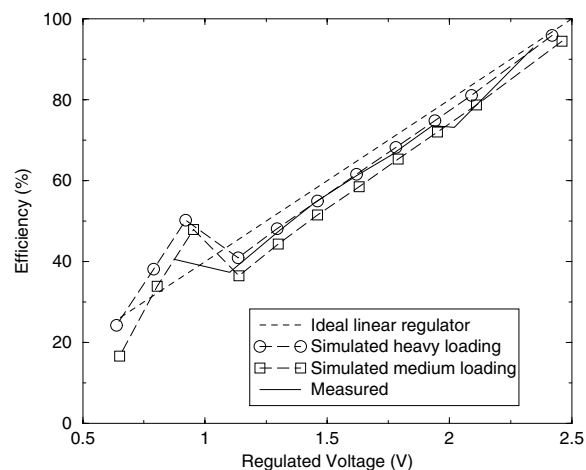


Figure 24: Efficiency of the power management system.

by the on-chip ADC. At the full supply of 2.480V, the datapath runs at 1.3ns (770MHz) and burns 195mW. At the supply of 660mV, the circuit cycle time is about 21.06ns (47.5MHz) and power consumption is 850μW.

Figure 22 shows the energy-cycle-time tradeoff with voltage scaling. The system automatically achieves delay-constrained energy optimization with respect to power supply. In Figure 23, we plot the sensitivity  $\frac{(\frac{\partial D}{\partial V})}{(\frac{\partial E}{\partial V})}$  (Lagrange multiplier) as a function of supply voltage. If other design parameters in addition to power supply were available for tuning, the system could achieve lower energy dissipation at the specified performance and supply voltage by making the sensitivities with respect to these new parameters the same as that shown in Figure 23[32][33][34].

## 6.3 Regulator efficiency

In Figure 24, we show the simulated and measured efficiency of the power management system. Below 1.0 V, the switched-capacitor power supply is engaged to provide an efficiency “boost” at the lowest supplies. The heavy-loading curves are simulated with a large diode connected NMOS transistor. The medium-loading curves are simulated with the same type of load of about half the strength. The measured results reflect the actual load of the datapath; the efficiency “boost” due to the switched-cap supply below 1.0 V is evident.

## 7 Conclusion

In this paper, we have described the design of a high-performance asynchronous micropipelined datapath that provides robust interfaces across voltage domains, performing appropriate voltage level conversions and operating between domains 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 600 mV using an on-chip dc-dc conversion system that combines linear regulators and switch-capacitor power supplies. Because of the asynchronous design style, the processor operates continuously during the voltage scaling transitions. This system was developed to explore the feasibility of such a dynamic voltage scaling system for multirate signal processing applications.

## References

- [1] G. Qu and et al. Energy minimization of system pipelines using multiple voltages. In *Proc. IEEE International Symposium on Circuits and Systems*, 1999.
- [2] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonese, S. Dwaradasas, and M. L. Scott. Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In *Proc. Intl. Symp. on High Performance Computer Architecture*, 2002.
- [3] T. D. Burd and R. W. Brodereson. Design issues for dynamic voltage scaling. In *Intl. Symp. on Low Power Electronics and Design*, 2000.
- [4] L. S. Nielsen, C. Niessen, J. Sparso, and C. H. van Berkel. Low-power operation used self-timed circuits and adaptive scaling of the supply voltage. *IEEE Transactions on VLSI Systems*, 2(4):391 – 397, 1994.
- [5] K. Usami and M. Horowitz. Clustered voltage scaling technique for low-power design. In *Proc. Workshop on Low Power Design*, 1995.
- [6] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, S. W. Gould, and J. M. Cohn. Managing power and performance for system-on-chip designs using voltage islands. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 195 – 202, 2002.
- [7] A. P. Dancy, R. Amirharajah, and A. P. Chandrakasan. High-efficiency multiple-output dc-dc conversion for low-voltage systems. *IEEE Transactions of Very Large Scale Integration (VLSI) Systems*, 8(3), June 2000.
- [8] A. J. Stratakis, S. R. Sanders, and R. W. Brodereson. A low-voltage CMOS dc-dc converter for a portable batter-operated system. In *Power Electronics Specialists Conference*, 1994.
- [9] G.-Y. Wei and M. Horowitz. A fully digital, energy-efficient adaptive power-supply regulator. *IEEE Journal Solid-State Circuits*, pages 520 – 528, April 2000.
- [10] James E. Gunn, Kenneth S. Barron, and William Ruczyk. A low-power DSP core-based software radio architecture. *IEEE Journal on Selected Areas in Communication*, 17(4):574 – 590, 1999.
- [11] T. I. Chappell, B. A. Chappell, S. E. Schuster, J. W. Allen, S. P. Klepner, R. V. Joshi, and R. L. Franch. A 2-ns cycle, 3.8 ns access 512-kb CMOS ECL SRAM with a fully pipelined architecture. *IEEE Journal of Solid-State Circuits*, 26(11):1577 – 1585, 1991.
- [12] R. A. Haring, M. S. Milshtein, T. I. Chappell, S. H. Dhong, and B. A. Chappell. Self resetting logic register and incrementer. In *Proceedings of the Symposium on VLSI Circuits*, pages 18 – 19, 1996.
- [13] G. Hinton, M. Upton, D. J. Sager, D. Boggs, D. M. Carmean, P. Rousel, T. I. Chappell, T. D. Fletcher, M. S. Milshtein, M. Sprague, S. Samaam, and R. Murray. A 0.18  $\mu$ m CMOS IA-32 processing with a 4-GHz integer execution unit. *IEEE Journal Solid-State Circuits*, 36(11):1617 – 1627, 2001.
- [14] P. F. Lu, S. P. Kowalczyk, A. P. Pelella, W. V. Huott, U. Bakhru, J. Rawlins, P. Patel, Y. H. Chan, and K. A. Jenkins. A 15Kb 1.5 ns access on-chip tag SRAM. In *Proceedings of International Symposium on VLSI Circuits*, pages 208–212, 1997.
- [15] T. Williams and Mark Horowitz. A zero-overhead self-timed 160ns 54b cmos divider. *IEEE Journal Solid-State Circuits*, 26:1651–1661, November 1991.
- [16] 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 1.3GHz. In *Proceedings of the IEEE International Symposium on Asynchronous Circuits and Systems*, 2002.
- [17] I. E. Sutherland. Micropipelines. *Communications of the ACM*, 1989.
- [18] C.E. Molnar, I.W. Jones, W.S. Coates, J.K. Lexau, S.M. Fairbanks, and I.E. Sutherland. Two fifo ring performance experiments. *Proceedings of the IEEE*, pages 297–307, February 1999.
- [19] S. Schuster, W. Reohr, P. Cook, D. Heidel, M. Immediato, and K. Jenkins. Asynchronous interlocked pipelined CMOS circuits operating at 3.3–4.5GHz. In *Digest Technical Papers, International Solid-State Circuits Conference*, pages 292–293, 2000.
- [20] Ivan Sutherland and Scott Fairbanks. Gasp: A minimal fifo control. In *Proceedings of the International Symposium on Asynchronous Circuits and Systems*, pages 46 – 53, 2001.
- [21] B. S. Amrutur and M. A. Horowitz. A replica technique for wordline and sense control in low-power SRAMs. *IEEE Journal Solid-State Circuits*, 33(8):1208 – 1219, 1998.
- [22] K. Y. Yun. Automatic synthesis of extended burst-mode circuits using generalized c-elements. In *Proceedings of the European Design Automation Conference*, pages 290 – 295, 1996.
- [23] J. Tierno, A. Rylyakov, S. Rylov, M. Singh, S. M. Nowick, M. Immediato, and S. Gowda. A 1.3 GSAMPLE/s 10-tap full-rate variable-latency self-timed FIR with clocked interfaces. In *Digest Technical Papers, International Solid-State Circuits Conference*, 2002.
- [24] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanzawa, M. Ichida, and K. Nogami. Automated low-power technique exploiting multiple supply voltages applied to a media processor. *IEEE Journal Solid-State Circuits*, 33(3):463 – 472, 1998.
- [25] G. W. den Besten and B. Nauta. Embedded 5v-to-3.3v voltage regulator for supplying digital IC's in 3.3v CMOS technology. *IEEE Journal of Solid-State Circuits*, 33(7), July 1998.
- [26] S. Jou and T. Chen. On-chip voltage down converter for low-power digital system. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(5), May 1998.
- [27] L. R. Carley and A. Aggarwal. A completely on-chip voltage regulation technique for low power digital circuits. In *Proceedings of the International Symposium on Low-Power Electronics and Design*, 1999.
- [28] N. Dragone, A. Aggarwal, and L. R. Carley. An adaptive on-chip voltage regulation technique for low-power applications. In *Proceedings of the International Symposium on Low-Power Electronics and Design*, 2000.
- [29] J. G. Ryan, K. J. Carroll, and B. D. Pless. A four chip implantable defibrillator/pacemaker chipset. In *Proceedings of the Custom Integrated Circuits Conference*, 1989.
- [30] S. V. Cheong, H. Chung, and A. Ioinovici. Inductorless dc-to-dc converter with high power density. *IEEE Transactions on Industrial Electronics*, 41(2), April 1994.
- [31] D. Maksimovic and S. Dhar. Switched-capacitor dc-dc converters for low-power on-chip applications. In *Power Electronics Specialists Conference*, 1999.
- [32] Robert W. Brodersen, Mark A. Horowitz, Dejan Markovic, and Vladimir Stojanovic Borivoje Nikolic. Methods for true power minimization. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 35 – 42, 2002.
- [33] Victor Zyuban and Philip Strenski. Unified methodology for resolving power-performance tradeoffs at the microarchitectural and circuit levels. In *Intl. Symp. on Low Power Electronics and Design*, pages 166 – 171, 2002.
- [34] H. Peter Hofstee. Power-constrained microprocessor design. In *Proc. IEEE International Conference on Computer Design*, 2002.