

# A 167-Processor Computational Platform in 65 nm CMOS

Dean N. Truong, *Student Member, IEEE*, Wayne H. Cheng, *Member, IEEE*, Tinoosh Mohsenin, *Member, IEEE*, Zhiyi Yu, *Member, IEEE*, Anthony T. Jacobson, Gouri Landge, Michael J. Meeuwsen, Christine Watnik, Anh T. Tran, *Student Member, IEEE*, Zhibin Xiao, *Student Member, IEEE*, Eric W. Work, *Member, IEEE*, Jeremy W. Webb, *Member, IEEE*, Paul V. Mejia, *Member, IEEE*, and Bevan M. Baas, *Member, IEEE*

**Abstract**—A 167-processor computational platform consists of an array of simple programmable processors capable of per-processor dynamic supply voltage and clock frequency scaling, three algorithm-specific processors, and three 16 KB shared memories; and is implemented in 65 nm CMOS. All processors and shared memories are clocked by local fully independent, dynamically halttable, digitally-programmable oscillators and are interconnected by a configurable circuit-switched network which supports long-distance communication. Programmable processors occupy 0.17 mm<sup>2</sup> and operate at a maximum clock frequency of 1.2 GHz at 1.3 V. At 1.2 V, they operate at 1.07 GHz and consume 47.5 mW when 100% active, resulting in an energy dissipation of 44 pJ per operation. At 0.675 V, they operate at 66 MHz and consume 608  $\mu$ W when 100% active, resulting in a total energy dissipation of 9.2 pJ per ALU or MAC operation.

**Index Terms**—65 nm CMOS, array processor, digital signal processing, digital signal processor, DSP, DVFS, dynamic voltage and frequency scaling, embedded, GALS, globally asynchronous locally synchronous, heterogeneous, homogeneous, many-core, multi-core, multimedia, network on chip, NoC.

## I. INTRODUCTION

APPLICATIONS that require the computation of complex DSP workloads are becoming increasingly commonplace. These applications are often composed of multiple DSP

Manuscript received September 04, 2008; revised December 18, 2008. Current version published March 25, 2009. This work was supported by Intel, UC Micro, the National Science Foundation under NSF CCF Grant 430090, CA-REER Award 546907, SRC GRC Grant 1598.001, and CSR Grant 1659.001, Intelliasys, a VEF Fellowship, and SEM. Fabrication was provided by ST Microelectronics.

D. N. Truong, T. Mohsenin, A. T. Tran, Z. Xiao, P. V. Mejia, and B. M. Baas are with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616-5294 USA (e-mail: hottruong@ucdavis.edu; bbaas@ucdavis.edu).

W. H. Cheng and E. W. Work are with an early-stage startup company.

Z. Yu is with the ASIC & System State Key Laboratory, Microelectronics Department, Fudan University, Shanghai 201203, China.

A. T. Jacobson is with the Boalt School of Law, University of California, Berkeley, CA 94720 USA.

G. Landge is with the Department of Electrical and Computer Engineering, University of California, Davis, and also with Intel Corporation, Digital Home Group, Santa Clara, CA 95054 USA.

C. Watnik is with the Department of Electrical and Computer Engineering, University of California, Davis, and also with Intel Corporation, Mobility Group, Folsom, CA 95630 USA.

M. J. Meeuwsen is with Intel Digital Enterprise Group, Hillsboro, OR 97124 USA.

J. W. Webb is with the Department of Electrical and Computer Engineering, University of California, Davis, and also with Centellax Inc., Research and Development, Santa Rosa, CA 95403 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSSC.2009.2013772

tasks and are found in applications such as wired and wireless communications, audio, video, sensor signal processing, and medical and biological processing. Many are embedded and strongly energy-constrained. In addition, many of these workloads require very high throughputs, often dissipate a significant portion of the system power budget, and are therefore of high importance in system designs.

In contrast to general-purpose workloads, the targeted workloads typically comprise a collection of DSP kernels that are numerically intensive, easily parallelizable, and do not require large working data sets or large software programs. An example IEEE 802.11a/11g Wi-Fi baseband receiver block diagram is shown in Fig. 1 and illustrates how a complex DSP application can be easily partitioned into basic kernels, thus allowing a rich exploitation of task-level parallelism [1].

One-time fabrication costs for state-of-the-art CMOS designs are now several million dollars and total design costs of modern chips can easily total tens of millions of dollars. These costs are expected to continue rising in the future. In this context, programmable and/or reconfigurable processors that are not tailored to a single application or a small class of applications become increasingly attractive.

The presented many-core processing array is highly configurable and fully programmable, can compute the aforementioned complex DSP application workloads with high performance and high energy efficiency, and is well suited for implementation in future fabrication technologies [2].

This paper is organized as follows. Section II summarizes the key goals of the single-chip platform and its main features. Section III describes the processors and on-chip shared memory modules. Sections IV and V present the platform's inter-processor communication network and per-processor dynamic supply voltage and clock frequency scaling, respectively. Section VI reports implementation and chip measurements including implementations of complex applications mapped to the chip. Section VII concludes the paper.

## II. KEY GOALS AND FEATURES

Fine-grained many-core architectures have shown great promise in computing demanding complex multi-task applications with high performance and high energy efficiency, and they effectively address CMOS deep-submicron design issues such as dealing with increasing global wire delays and effectively using a larger number of transistors [3]–[5]. Moreover, many-core chips that utilize individual per-processor clock oscillators with fully independent timing requirements in a globally-asynchronous

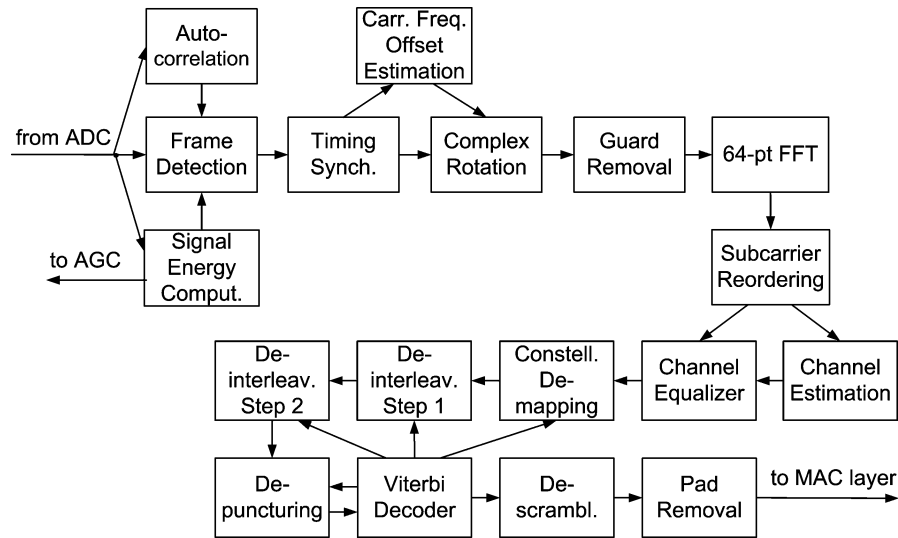


Fig. 1. Block diagram of an IEEE 802.11a/11g wireless LAN baseband receiver.

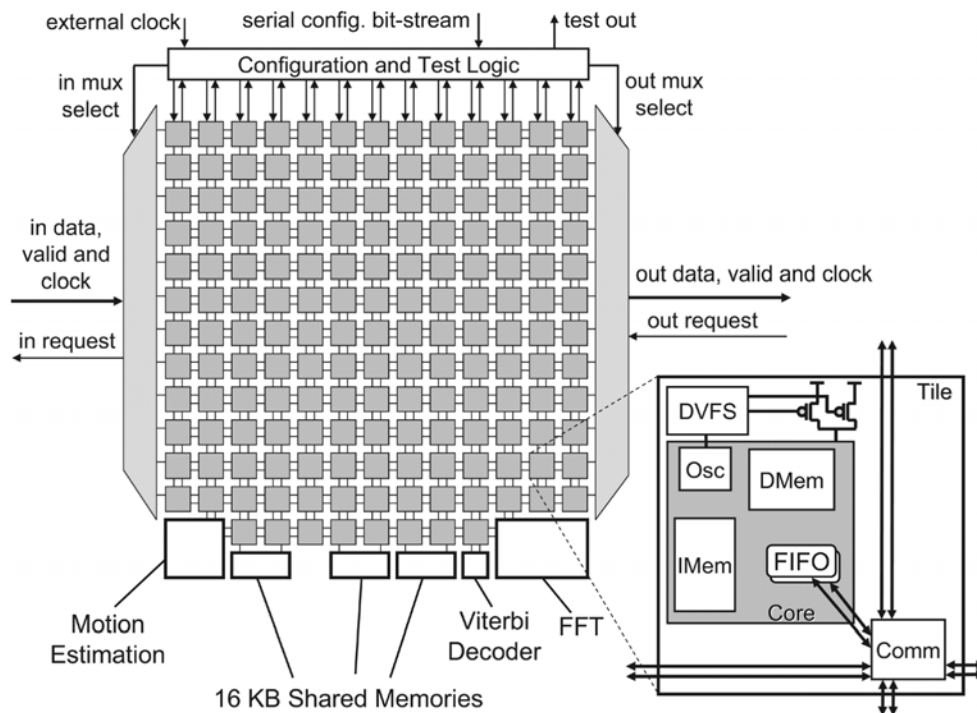


Fig. 2. Block diagram of the 167-processor computational array.

locally-synchronous (GALS) fashion can obtain high energy efficiencies through per-processor oscillator halting [6], as well as avoid many clock generation and distribution complexities, and have an increased tolerance to process variations.

However, the following goals are not well addressed by current many-core systems:

- dynamic optimization of each processing core's operating environment to reduce energy consumption of both lightly loaded and unused processors [7];
- achieving very high speeds and efficient execution for common computationally intensive tasks such as Fast Fourier Transforms (FFTs) and motion estimation for video encoding;

- large on-chip shared memories for tasks that require access to large data sets shared among multiple processors; and
- high throughput, low overhead communication between distant processors on a single chip.

To address these challenges, the presented single-chip computational array contains 164 simple, fine-grained, homogeneous programmable processors that efficiently compute DSP, embedded, and multimedia kernels; and also includes the following features [8]:

- per-processor dynamic supply voltage and clock frequency scaling (DVFS) circuits for the 164 homogeneous processors;

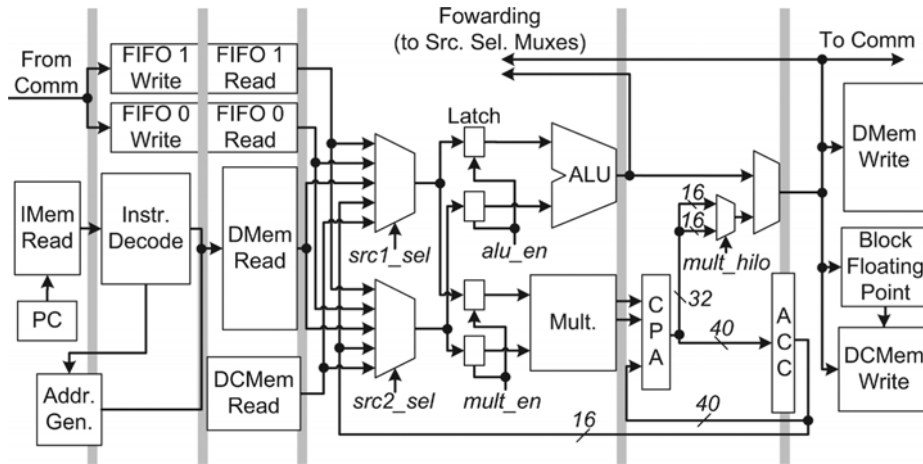


Fig. 3. Six-stage pipeline of the programmable processors. IMem = instr. mem., DMem = data mem., DCMem = dynamic config. mem., CPA = carry-propagate adder, ACC = accumulator.

- three dedicated-purpose processors to accelerate computation of the FFT, video encoding motion estimation, and Viterbi decoding;
- three 16 KB on-chip shared memories; and
- a high throughput, low area, and energy-efficient long-distance-capable inter-processor network.

Fig. 2 is a block diagram of the 167-processor computational array.

### III. PROCESSORS AND ON-CHIP SHARED MEMORIES

#### A. Programmable Processors

Each of the 164 simple programmable processors utilizes an in-order single-issue six-stage pipeline with a 16-bit fixed-point ALU and multiplier, and a 40-bit accumulator as shown in Fig. 3. The six-stage pipeline has an instruction fetch stage, instruction decode stage, memory read stage, two execution stages and a write back stage. The instruction set of the simple programmable processor adheres to a simple one-destination and two-source architecture and thus each processor contains two dual-clock FIFOs to transfer input data reliably across clock domain boundaries [9]. To save additional power, two sets of transparent latches are placed before the ALU and multiplier so that changes in source selection do not unnecessarily toggle internal nodes when the results of the ALU and/or the multiplier are unused. These circuits reduce power dissipation by 7%–21% with a program alternating between ALU and multiply-accumulate (MAC) type instructions (measured while operating at 1.3 V).

Programmable processors contain a  $128 \times 35$ -bit instruction memory,  $128 \times 16$ -bit data memory, and two  $64 \times 16$ -bit dual-clock FIFOs for asynchronous inter-processor communication. They support over 60 basic instructions and other hardware features such as four configurable address generators for memory address computation, block floating point support, conditional execution and zero overhead looping. Through the use of these features, a floating point CORDIC square root application operates with  $2.9 \times$  fewer cycles compared to the first generation AsAP platform [10].

All processors and shared memories contain their own fully independent digitally programmable local clock oscillator

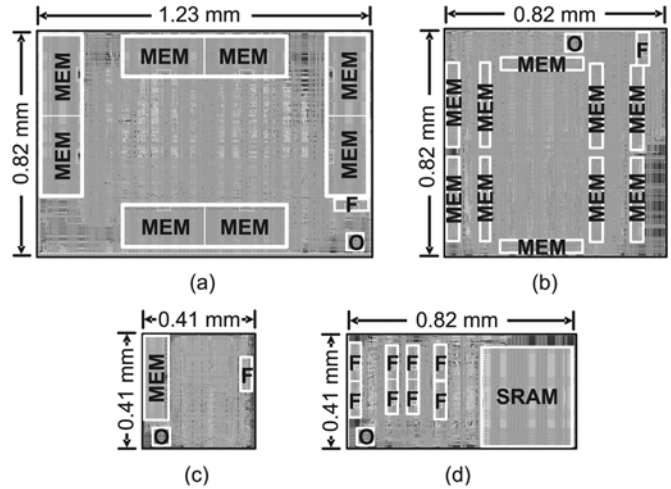


Fig. 4. Layouts of the dedicated-purpose processors and shared memory blocks (drawn to scale). (a) FFT, (b) motion estimation, (c) Viterbi, (d) shared memory. Note: “F” = FIFO, “O” = Oscillator, “MEM” = dual port SRAM, “SRAM” = 8 KWord single-port SRAM.

which requires no phase-locked loop (PLL), delay-locked loop (DLL), or crystal oscillator. Oscillators operate at their own independent clock frequencies that, in conjunction with their dual-clock input FIFOs, enable GALS clocking [11]. Oscillators are able to halt, restart, and change their frequency at any time without constraint during normal operation. This ability, along with the integration of special stall logic, allows oscillators to fully halt during periods of processor inactivity and restart in less than one clock cycle once work is available. Section V further describes details of the local oscillators.

#### B. Configurable Dedicated-Purpose Processors

To achieve even higher throughputs and energy efficiencies than what is possible with the homogeneous programmable processing array for several computationally demanding tasks, three dedicated-purpose processors are placed into the array. These processors accelerate the following algorithms: FFT, motion estimation for video encoding, and Viterbi decoding for convolutional code decoding. Layouts for the accelerators are shown in Fig. 4(a)–(c), respectively. As with the programmable

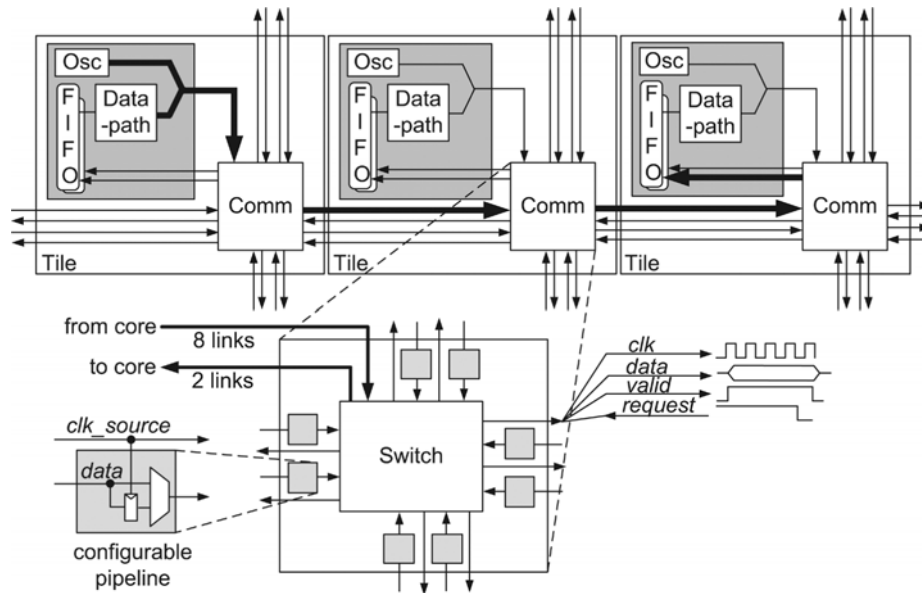


Fig. 5. Inter-processor network design highlighting an example communication path from the leftmost processor to the rightmost processor.

processors, each of the dedicated-purpose processors includes its own local dual-clock FIFO(s) and oscillator.

Integration with the 2-D mesh of programmable processors is simplified through the design of a generic interface “wrapper” that contains an input dual-clock FIFO buffer, oscillator, configuration logic, and circuit-switched communication logic. The core logic of the dedicated-purpose processors or memories must then communicate through the wrapper. When considering layout, I/O pins and power grids match across metal layers with the rest of the processing array.

The FFT algorithm is ubiquitous in many digital signal processing applications such as ones using orthogonal frequency division multiplexing (OFDM) modulation, and spectral analysis and synthesis. The FFT processor is configurable at runtime and can dynamically switch between 16 to 4096 point complex FFT and IFFT transforms. It uses a block floating-point format and its continuous flow architecture allows it to achieve a sustained throughput of one complex radix-4 or radix-2 butterfly per cycle.

Motion estimation is the most computationally intensive task in video encoding standards such as MPEG-2 and H.264. The motion estimation processor supports a number of fixed and programmable search patterns, including all of the H.264 specified block sizes (i.e.,  $4 \times 4$ ,  $4 \times 8$ ,  $8 \times 4$ ,  $8 \times 8$ ,  $16 \times 8$ ,  $8 \times 16$ , and  $16 \times 16$  pixels) within a  $48 \times 48$  pixel search range.

The Viterbi decoding algorithm is a fundamental component in many wired and wireless communication applications, as well as many storage applications (e.g., hard disks). The configurable Viterbi decoder processor can decode codes up to constraint length 10 with up to 32 different rates, including the common  $1/2$  and  $3/4$  rates.

### C. On-Chip Shared Memories

The computational platform also contains three on-chip 16 KB shared memories, whose layout is shown in Fig. 4(d).

Each shared memory supports connections with up to four programmable processors, and each contains a single-port SRAM that can range up to 64 KWords or 128 KB [12]. In the presented chip, each of the shared memories connects to two processors along the bottom of the array shown in Fig. 2, and the two unused ports are tied off. Each memory contains an  $8 \text{ KWord} \times 16\text{-bit}$  single-port SRAM, allowing each memory block to reach a peak throughput of one read or write per cycle. In addition, each port supports least-recently-served priority arbitration during times of simultaneous access by multiple processors, port priority, unique split port address/data modes, and independent programmable address generation to support a variety of addressing modes. In order to integrate the memories into the GALS array, each port contains an input and output FIFO, and the block contains a local clock oscillator.

## IV. INTER-PROCESSOR COMMUNICATION NETWORK

All programmable processors are interconnected by a “double-link” reconfigurable mesh network that has two communication links in each direction on each of the four edges of each programmable processor tile [13] for a total of eight input and eight output links as illustrated in Fig. 5. Each processor core can send data to any dynamically changing combination of the tile’s eight output links through software instructions during runtime. The core can receive data from any two of the eight input links through configurable circuit-switched muxes. These muxes route the tile’s input links to its dual-clock FIFOs and are normally configured statically. To accomplish this routing capability, the processor core has two input and eight output links to the communication switch, allowing full access to all inter-processor links.

The novel network design allows links to be configured to pass data across processor tiles in a dedicated channel without disturbing intermediate processors and without regard to the intermediate frequency or voltage domains [13]. Fig. 5 shows a long-distance communication example in which the leftmost

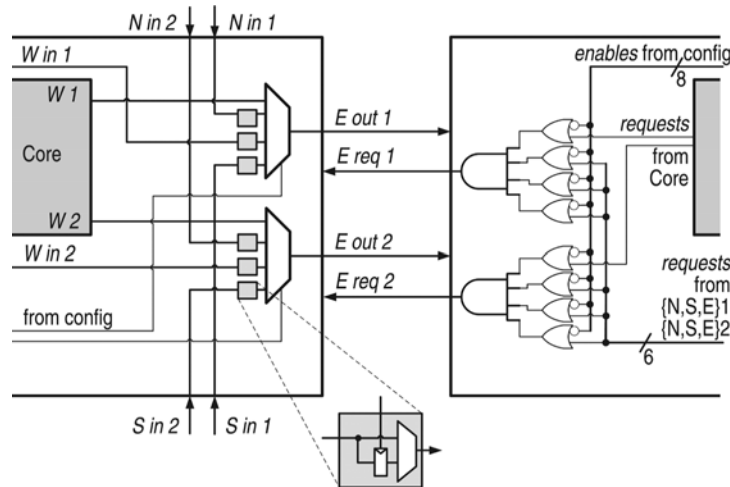


Fig. 6. Inter-tile communication link connectivity highlighting the *East Out* port logic and the “layered” architecture where, except for the core, links connect only with other links in their layer (1 or 2).

processor core communicates directly with the rightmost processor core without disturbing the middle processor core. This path is highlighted with heavier lines.

As shown in Fig. 5, asynchronous communication between processors is accomplished by sending the source processor’s clock along with its data. Data are then received by the destination processor through a dual-clock FIFO, whose write side is clocked by the sender’s clock while its read side is clocked by the receiver’s clock. This communication scheme is *source-synchronous* since the source’s clock and data are sent together over the entire path where the data is finally registered by the receiver using the source clock.

Each source-synchronous communication link contains a clock, 16-bit data bus, valid signal, and a reverse-direction request signal for flow control. Compared to clock-less handshaking interfaces, the source-synchronous links with destination FIFOs more easily obtain a throughput of one data-word per clock cycle since a round trip handshake is not required for each word transferred. Configurable pipeline registers shown in Fig. 5 are clocked by the accompanying source clock and permit pipelining of long-distance communication for increased clock rates.

The internal communication circuits are illustrated in Fig. 6 with details of only the *East Out* ports for simplicity. Each output link has an output mux—the basic block of the communication switch. The *East Out* muxes can switch between *Core*, *North*, *South*, and *West* inputs. The “two layer” architecture significantly reduces interconnect complexity while not limiting routing capability (except in rare pathological cases) by not making connections between layers (e.g., *S in 2* does not connect with *E out 1*).

Each output link has a corresponding *request in* signal that notifies the sender when the receiver is ready to accept data. Each input link has a corresponding *request out* signal that can be a combined request of multiple receivers. This gives senders the ability to broadcast information to more than one receiver through efficient circuit switches rather than dedicating processors to act as repeaters. Fig. 6 shows request signals being masked by *enable* signals and OR gates where a value of

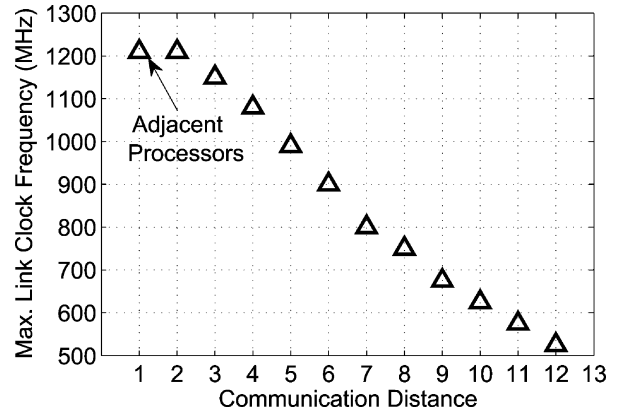


Fig. 7. Measured maximum long-distance link clock frequencies for communication between processors at the given distances from each other where adjacent processors are indicated at a Communication Distance of 1.

*enable* = 1 allows the corresponding *request* signal to pass to the port’s AND gate which allows the final *request out* (*E req 1, 2* in Fig. 6) to assert once every masked request signal is “1”—this is equivalent to saying that the sender must wait until every receiver is ready before sending data.

Fig. 7 plots measured data for maximum allowable source clock frequencies when sending data over long distance links along a straight West to East direction. Communication is possible at a maximum clock rate of 1.21 GHz for adjacent processors and processors two tiles apart. Chiefly because of clock duty cycle distortion caused by unbalanced rise and fall times through buffers and wires, as well as data crosstalk and clock jitter effects, a decrease in maximum source clock frequency is observed with longer links.

## V. PER-PROCESSOR DYNAMIC SUPPLY VOLTAGE AND CLOCK FREQUENCY SCALING

The 164 programmable processors of the many-core array are able to dynamically and independently switch their supply voltage between one of two power grids and are also able to dynamically and independently adjust their clock frequency.

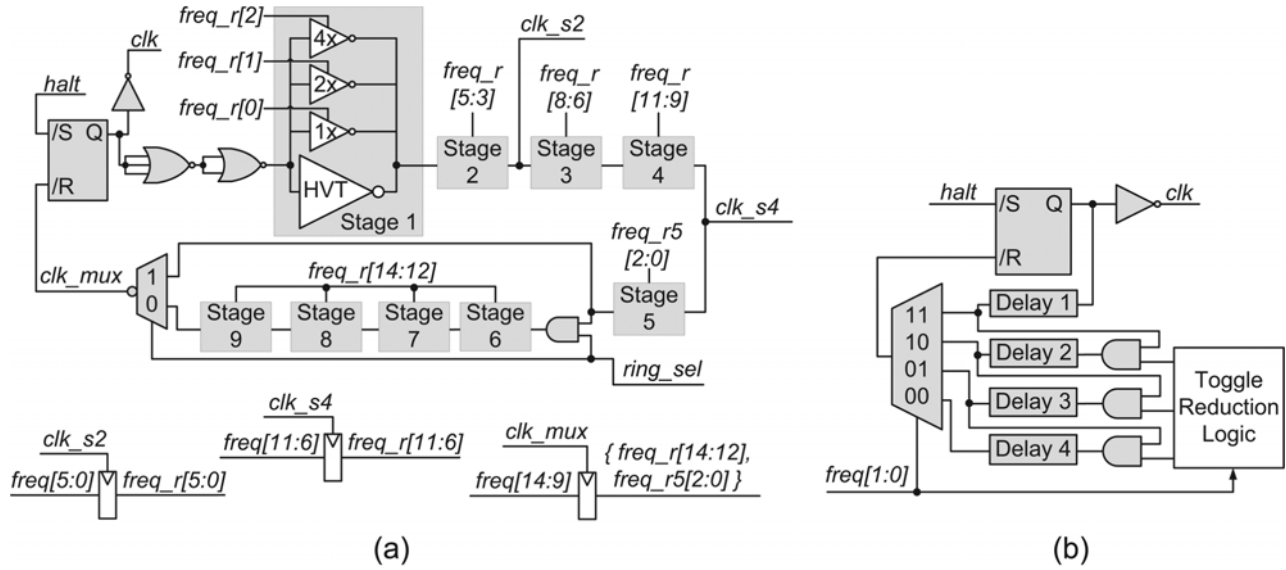


Fig. 8. Configurable (a) 5- or 9-stage ring oscillator with HVT = high threshold voltage device, and (b)  $\mu$ Watt ring oscillator.

Changes are made by a local configurable dynamic voltage and frequency scaling (DVFS) controller.

#### A. Configurable Local Clock Oscillator

As mentioned previously, local independent clock oscillators enable GALS clocking. This capability meshes very well with DVFS operation when applied on a per-core basis.

In addition, the finely adjustable and independent oscillators permit tolerance and even performance *gains* due to process, voltage, and temperature variations. For example, schemes can be devised that estimate higher performing and lower performing cores due to these variations and map heavy and light computational loads accordingly.

Each local oscillator is composed of the configurable ring oscillator shown in Fig. 8(a), the low power “ $\mu$ Watt” ring oscillator shown in Fig. 8(b), and a configurable clock divider capable of dividing the ring oscillator’s output frequency by any power of two from 1–128. The ring oscillator consists of a configurable ring of either 5 or 9 inverters. The current drive of each inverter stage can be finely tuned through digitally controllable tri-state inverters. The load capacitance of each stage is essentially the sum of the gate capacitances of the following stage, which stays constant regardless of the state of the tri-state inverters. Thus, the frequency increases roughly linearly with the current drive of a stage. Fig. 8(a) shows a detail of one inverter stage. Frequency bits 0, 1, and 2, control the  $1\times$ ,  $2\times$ , and  $4\times$  tri-states of Stage 1, respectively. The  $1\times$ ,  $2\times$ , and  $4\times$  notations signify their relative drive strengths in relation to each other. In theory, incrementing the three bits from zero (000) to seven (111) should result in a linear increase in frequency. However, increasing the inverter’s width linearly does not, in practice, cause a linear decrease of its gate delay [14]. Each stage is assigned three bits, some of which are not unique (e.g., Stages 4 and 5 are controlled by bits [11:9] (though by different clocks to save power), and Stages 6 through 9 are controlled by bits [14:12]). These assignments were guided by SPICE simulations with the design goals of minimizing the number of configuration bits and ensuring complete

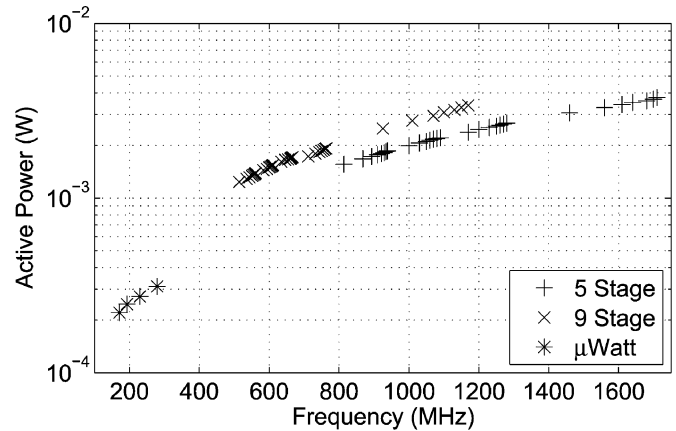


Fig. 9. Measured oscillator power versus frequency at 1.3 V for a small number of representative frequency configurations.

coverage of the entire frequency range. A total of 15 bits are used to fine tune the oscillator’s frequency.

To give a very wide frequency tuning range, the main inverters of each stage consist of high threshold voltage (HVT) inverters while the tri-state inverters are low threshold voltage (LVT) inverters. With this combination the oscillator achieves a wider tuning range with the 15 control bits than with a ring built with single threshold voltage transistors. The 5-stage oscillator generates frequencies that range between 814 MHz and 1.71 GHz, and the 9-stage oscillator generates frequencies between 583 MHz and 1.17 GHz at 1.3 V as shown in Fig. 9. Using the clock divider, the oscillator can achieve frequencies as low as 4.55 MHz at 1.3 V. In Section V-B, the oscillator will be shown to be on its own power supply, and thus the oscillator can also operate at much wider ranges by scaling its supply voltage than is noted here.

To first order, power is approximately the same for both 5-stage and 9-stage rings operating at the same frequency. An AND gate is placed before the final 4 stages to gate the propagation of the clock signal when only 5 stages are used.

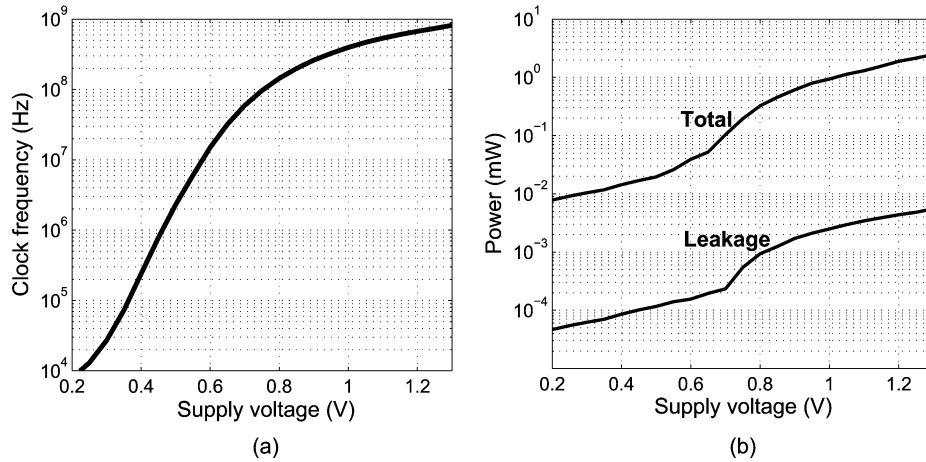


Fig. 10. Measured 5-stage ring oscillator frequency (a) and power (b) versus voltage.

The ring oscillator clocks each of its frequency control bits using the clock edge appearing at the output of the stage whose tri-state inverters are controlled by those corresponding bits. This eliminates potential glitches and duty cycle distortions when the DVFS controller changes the frequency while the processor is operating. To halt the clock cleanly without glitches, an SR latch is used to latch both the clock (just before the output buffer) and halt signal. The SR latch consists of two uniquely sized NAND gates, and two input-tied NOR gates are placed at the latch output, which shifts the switching threshold from the  $V_{dd}/2$  midpoint. Metastability is avoided because the *halt* signal is generated by logic clocked by the local oscillator and its timing is therefore guaranteed valid.

Oscillator power is significant when clocking processors at very low frequencies—e.g., on the order of a few MHz. To generate a slow clock the primary oscillator uses the clock divider to divide down an original clock of several hundred MHz. This consumes unnecessary power when a simpler ring oscillator can generate the same clock at lower power. Thus, as shown in Fig. 8(b), a simplified “ $\mu$ Watt” oscillator that is tunable using muxes and delay elements supplements the standard oscillator. This oscillator can run between 171 and 279 MHz while dissipating on average  $263 \mu\text{W}$  of active power at 1.3 V (see Fig. 9). This compares favorably with the lowest frequency setting of the 9-stage oscillator, which dissipates over 2 mW.

Fig. 10 shows the ring oscillator’s frequency, and the total and leakage powers for the 5-stage oscillator, without any of its tri-state inverters enabled, from 0.2–1.3 V. The figure illustrates the wide range of voltages the oscillator is capable of running at, including its sub-threshold operation. We can take advantage of this wide range of operable voltages of the oscillator since all oscillators are powered by their own voltage supply. This allows the oscillator supply voltage to be optimized along with the processor supply voltages to minimize total system power.

### B. Multiple Voltage Domain Architecture

Common techniques for supplying per-processor variable voltage domains to a chip containing multiple processors include the use of individual on-chip DC-DC converters, or

multiple local power grids that obtain their power from off-chip power supplies. Clearly, the overhead of both approaches is undesirable as the number of on-chip cores increases beyond a few. An alternative is to use hierarchical power grids with many local grids where each local grid is connected by switchable power gates to one of the multiple parallel global grids.

There are two well-known on-chip DC-DC conversion methods: linear regulation and switching regulation. While linear regulators are small and easy to integrate on chip, their power conversion efficiency is limited [15]. On the other hand, switching regulators have higher power efficiency but consume a relatively large amount of die area [16], [17]. With current fabrication technologies, on-chip switching DC-DC converters are not feasible for many-core platforms because they require large area devices such as capacitors and inductors for each converter. In addition, DC-DC converters can take hundreds of clock cycles to switch from one voltage level to another—assuming clock frequencies in the range of 1 GHz or higher [18]. This delay can significantly degrade system performance.

A technique using multiple global external supply voltages with hierarchical power grids has been adopted due to its simplicity and efficiency. Processors change their supply voltage by connecting their local power grid to one of several parallel global power grids. The supply voltage of each core is decided by a global and/or local voltage controller. This approach is simple, efficient, and capable of allowing a switching delay of only a few clock cycles [19].

The presented platform has two global voltages for use by the programmable processors:  $V_{ddHigh}$  and  $V_{ddLow}$ . The relative benefits of using more than two discrete voltages are small when compared to the area and complexity of the circuits needed to handle switching between more than two voltages effectively [20]. To the best of our knowledge, our chip is currently one of two fabricated many-core chips which use multiple parallel global power grids for dynamic voltage scaling. The other chip was designed by Beigné *et al.* [21].

Fig. 11 provides an overview of the various components of DVFS. Several major voltage domains are present in the programmable processor including  $V_{ddCore}$ ,  $V_{ddOsc}$ , and

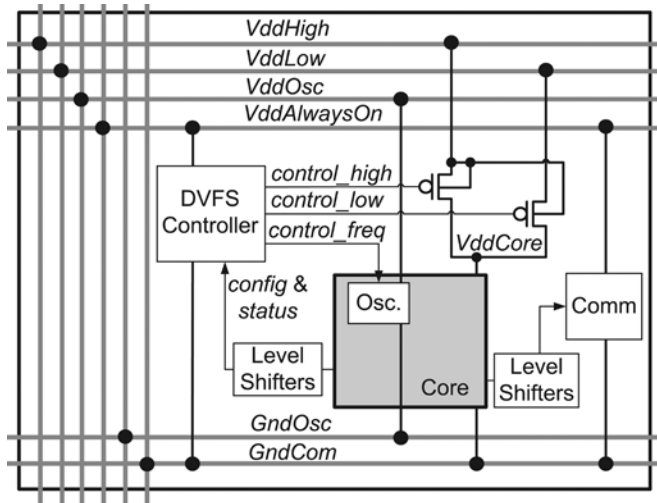


Fig. 11. Power and ground plan of the programmable processors.

*VddAlwaysOn*. The processor tile is composed in two hierarchical levels: a processor core and a wrapper surrounding it. The wrapper operates on *VddAlwaysOn*, which is normally larger than or equal to the core's supply voltage *VddCore*. This is so that level shifters are needed only for signals originating from the core and going to the wrapper. Since the communication unit is in the wrapper level it is also supplied by *VddAlwaysOn* allowing it to be independent from the voltage domain of the core as mentioned in Section IV. The wrapper also contains the DVFS controller, which controls both the core's local oscillator and the PMOS power gates used to select between *VddHigh* and *VddLow*. Both power gates can also be disabled to greatly reduce leakage. Both power gates have their substrates tied to *VddHigh*, which greatly reduces static current caused by the parasitic diode between substrate and drain [22].

Although Fig. 11 shows the two PMOS power gates as single transistors, each power gate actually consists of 48 parallel programmable PMOSs with their own individual control signals. As described later, the DVFS controller takes advantage of this flexibility when switching the *VddCore* voltage. The layout view of the programmable processor is shown in Fig. 12. On each side of the core are twelve power gate cells, each of which contains four PMOS transistors—two for each power supply. Thus, there are 48 PMOSs that connect with *VddHigh*, and 48 PMOSs that connect with *VddLow*. The area of the cell is  $253 \mu\text{m}^2$  where a single PMOS has a  $W/L$  of approximately 500.

The PMOS power gate is in the triode region when it is active (i.e., "on") and can potentially cause a non-negligible voltage droop on *VddCore* due to its on-resistance. To experimentally measure the effective on-resistance of the power gates over a range larger than what is used in operation, we set the two power grids to the same voltage, i.e.,  $VddHigh = VddLow$ , and then for a very large static current load of approximately 64 mA, we change the effective width of the effectively double-wide PMOS power gate by selectively controlling each power gate group—a total of 96 power gates—giving an effective  $W/L$  range of approximately 6000 to 48,000 in increments of 6000. Results of this experiment are shown in Fig. 13 and show a knee in the

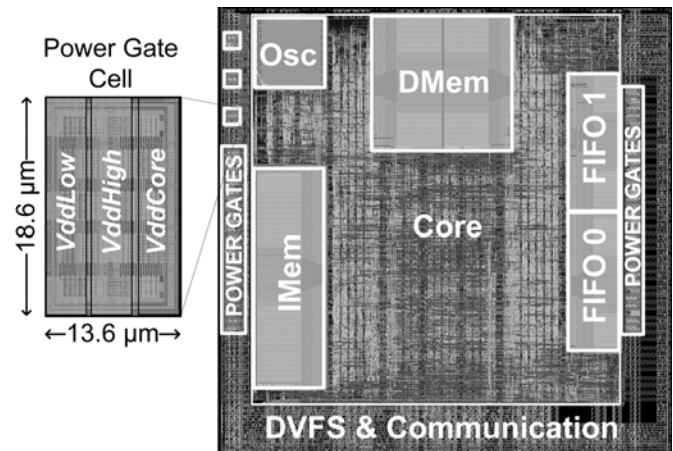


Fig. 12. Programmable processor and power gate cell layouts.

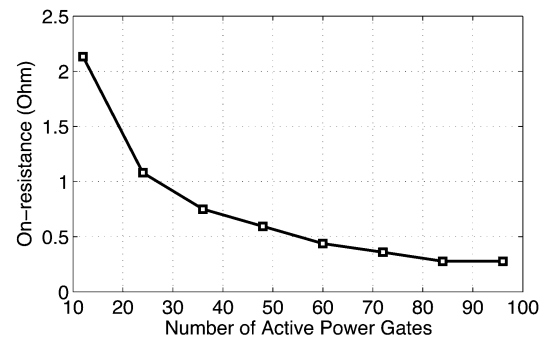


Fig. 13. Measured effective power gate width versus PMOS on-resistance found by setting  $VddLow = VddHigh$ , which effectively doubles the number of power gates from 48 to 96, and changing the number of parallel PMOS power gates that are "on".

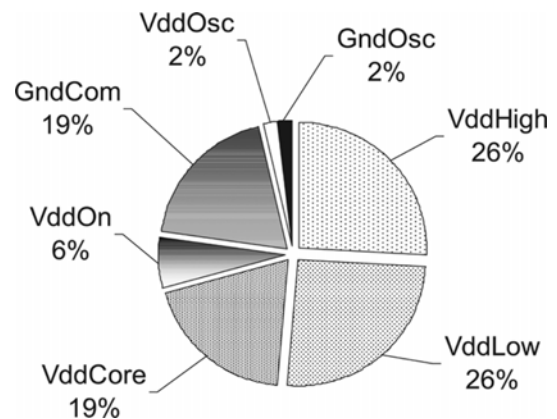


Fig. 14. Breakdown of power and ground usage of Metal 6 and Metal 7.

curve in the general range of 30–40 parallel gates, just below the chosen 48 parallel power gates.

The low resistance metals Metal 6 (M6) and Metal 7 (M7) are devoted almost entirely to power and ground distribution and thus only 3.6% of M6 and 0.5% of M7 have been utilized for signal routing. The amount of M6 and M7 used for each type of power and ground is summarized in Fig. 14. This breakdown resulted from design estimates of the worst case expected current consumed by each major block.



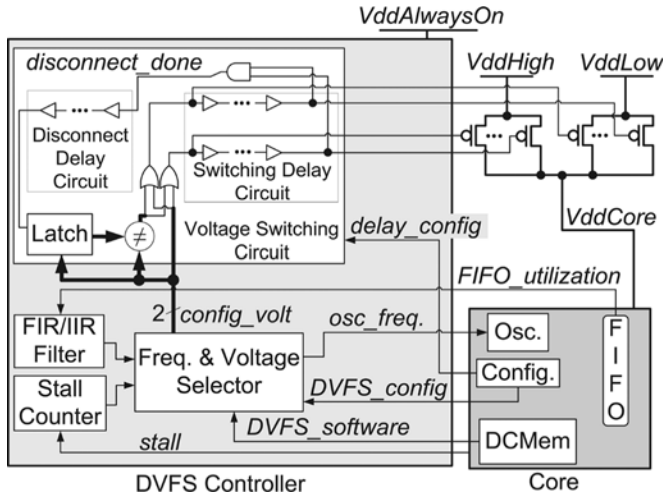


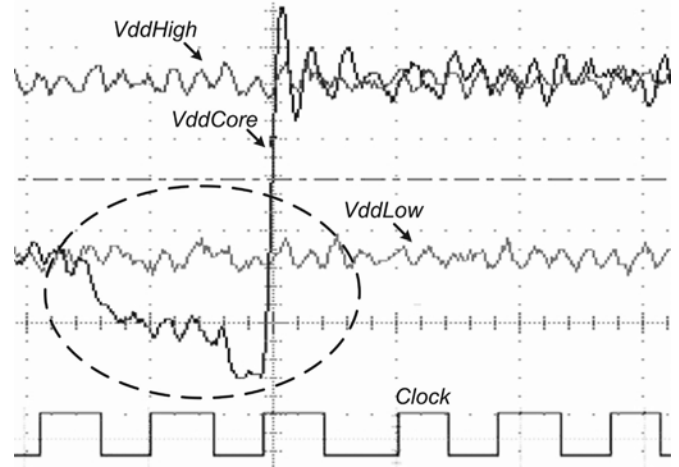
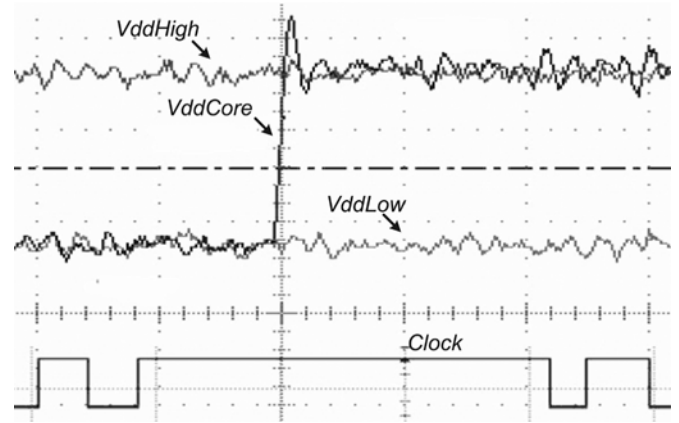
Fig. 15. DVFS controller block diagram.

### C. DVFS Controller

The DVFS controller is highly configurable and can set the voltage and frequency of programmable processor cores through three methods: 1) static configuration, 2) dynamic runtime configuration through software, and 3) dynamic control through a local hardware controller. Static configuration is useful for tasks that have a static load behavior at runtime, which is common in many DSP applications. However, some tasks have dynamic yet well-defined load behaviors. In such cases the user or compiler can take advantage of periodic or one-time activity characteristics through software configuration. Typically, applications contain a combination of the above two cases, or even a largely input-dependent load characteristic, which is common in applications with data-dependent control. An additional benefit of the local hardware controller is that it adapts for processor-specific and runtime-specific effects such as process and supply-voltage variations.

The DVFS controller is shown in Fig. 15. Static and software configuration is depicted by the *DVFS\_config* and *DVFS\_software* signals, respectively. Two other signals from the core, *FIFO\_utilization* and *stall*, are used by the hardware controller to select the core's frequency and voltage. *FIFO\_utilization* represents the current fullness of the core's input FIFO. If the FIFO is often nearly full, then the core may be sped up to increase the rate of data processing. On the other hand, an often empty FIFO can indicate that the processor is going through data more quickly than the rate that data is being sent to it, and thus the core may be slowed down. This strategy to increase energy efficiency works well for tasks with infrequent FIFO reads between large blocks of computation, which is true for many embedded applications. The *FIFO\_utilization* signal is averaged over time using configurable digital FIR or IIR filters to reduce unnecessary voltage switching, which can lead to an increase in global power grid noise, an increase in overall power consumption, and a decrease in overall performance.

An alternate hardware control method is to use the *stall* signal, which asserts whenever the oscillator halts due to processor inactivity. Stalls are caused by reading from an empty FIFO or writing to a full FIFO. Stalling is monitored by the


 Fig. 16. Measured *VddCore* switching from *VddLow* to *VddHigh* without clock halting and with an extremely early disconnect from *VddLow* that would never be used in practice, with *VddLow* = 0.9 V, *VddHigh* = 1.3 V, and minor timescale = 2 ns.

 Fig. 17. Measured *VddCore* switching from *VddLow* to *VddHigh* with clock halting during the transition (*VddLow* = 0.9 V, *VddHigh* = 1.3 V, minor timescale = 2 ns).

DVFS controller through a special externally-clocked counter that counts the number of cycles that the core stalls, and if this counter reaches a user configured threshold, then the voltage and frequency decrease accordingly.

### D. Voltage Switching

As mentioned earlier, each programmable processor's core power supply, *VddCore*, is connected to the global power grids through 48 individual PMOS transistors for *VddHigh* as well as 48 for *VddLow*. Configurability is desired due to the negative effects of voltage switching. Besides the unrecoverable energy loss incurred when the core voltage is raised from a low to high *Vdd* (ignoring energy scavenging techniques), significant noise is generated on the global power grids by the local *VddCores* switching between them. To alleviate this noise, the DVFS controller is able to change the rate at which the core switches between voltages. Fig. 15 shows a general overview of the Voltage Switching Circuit. The *config\_volt* signal is sent to a comparator, which compares whether this present voltage configuration is the same as the previous. If they are different,

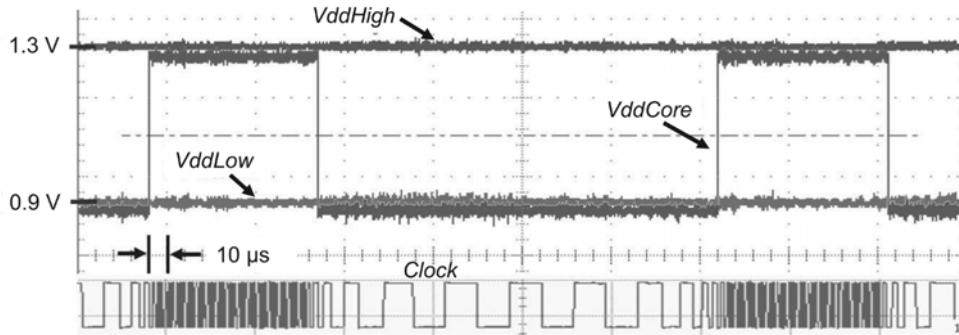


Fig. 18. Measured waveforms of a processor's clock and supply voltage dynamically changing while controlled by the local DVFS controller.

then the OR gate's output is forced to "1," which is then sent to a Switching Delay Circuit that disables every PMOS power gate in a user specified pattern (e.g., disable a variable number of power gates for every delay unit, or even switch them all off with very small delay) using configurable delays set through static configuration via the *delay\_config* signal. An AND gate is used to determine when all PMOS power gates have been switched off. When this happens the *disconnect\_done* signal is asserted and is then sent to a configurable Disconnect Delay Circuit, which leaves the core disconnected for a configurable amount of time. Once the delay circuit has settled, the latch is clocked to allow the current *config\_volt* signal to pass through the OR gate and reconnect the PMOS power gates to their new voltage configuration through the Switching Delay Circuit.

To summarize, in normal operation the switching algorithm operates with these steps: 1) the present voltage configuration is compared with the previous configuration, 2) if different, all PMOS power gates are disconnected using the configurable switching delays, 3) once all PMOS power gates have been disconnected, the controller waits using the configurable disconnect delay, and 4) finally, the new PMOS power gates are connected through the configurable switching delays.

Figs. 16 and 17 contain measured waveforms of low to high supply voltage switches with very long disconnect times while not halting and halting the core, respectively. Such long disconnect times would normally never be used and are shown for illustration purposes only. When the core is left running while a voltage switch occurs, a voltage droop on *VddCore* occurs due to core current consumption, circled in Fig. 16. However, with clock halting, *VddCore* in Fig. 17 shows no noticeable droop.

Fig. 18 shows measured waveforms of switching due to the DVFS hardware controller making decisions based on a two processor producer-consumer test case. The controller gradually increases the core's clock frequency and then switches *VddCore* from *VddLow* to *VddHigh* at a configurable trip point, and continues to increase the clock frequency up to a configurable maximum. The operations work in reverse when moving from a high clock frequency to a low clock frequency.

## VI. IMPLEMENTATION AND MEASURED RESULTS

### A. 65 nm CMOS Implementation

The presented chip design is fabricated in ST Microelectronics 65 nm low-leakage CMOS. Except for certain portions

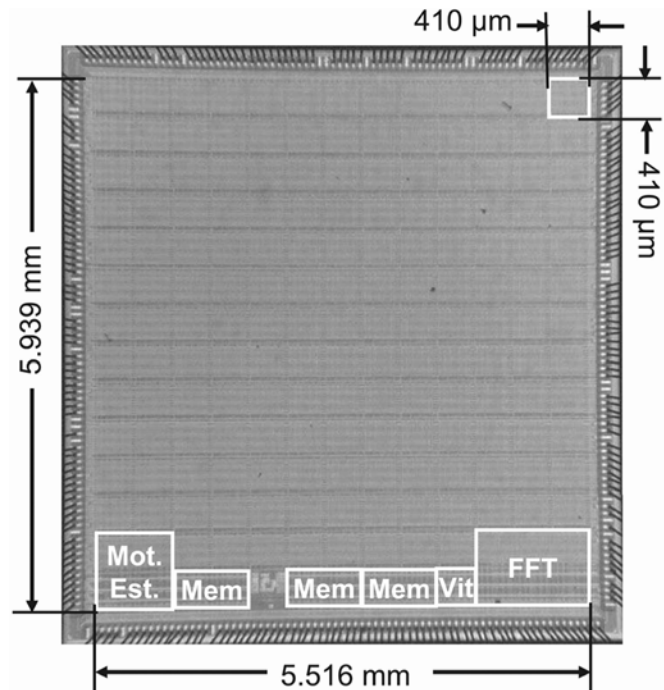


Fig. 19. Die micrograph.

TABLE I  
MEASURED KEY DATA OF DEDICATED-PURPOSE PROCESSORS AND SHARED MEMORIES OPERATING AT THE MAXIMUM SUPPLY VOLTAGE OF 1.3 V. ACTIVE POWER IS FOR OPERATION AT THE REPORTED MAXIMUM CLOCK FREQUENCY

Processor	Area (mm <sup>2</sup> )	Max. Frequency (MHz)	Active Power (mW)
FFT	1.01	866	35.0
Viterbi	0.17	894	17.6
Motion estimation	0.67	938	196.0
Shared memory	0.34	1300	4.5

of the DVFS logic, oscillator, PMOS power gates, decoupling capacitors, and chip I/O, a standard cell and automatic place and route flow was used with industry standard tools that include Synopsys Design Compiler, Cadence SoC Encounter, and Mentor Calibre.

Fig. 19 shows the die micrograph of the 167-processor array. The die occupies 39.4 mm<sup>2</sup> and contains 55 million transistors. Each programmable processor in the homogeneous array contains 325,000 transistors in a 0.17 mm<sup>2</sup> area. Data summarizing the area and preliminary measurements of the three

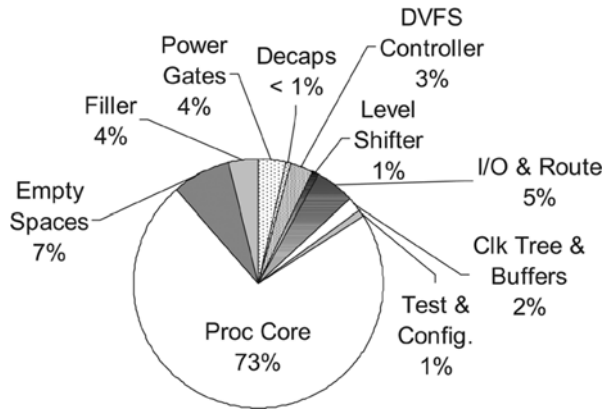


Fig. 20. Breakdown of the programmable processor tile area.

dedicated-purpose processors and the shared memory are reported in Table I.

All processors and memories are built in a modular fashion with nearly identical design flows. First, the core processor was described in Verilog RTL. They were then placed in an asynchronous wrapper interface consisting of a local independent oscillator and FIFO macro blocks (except for the programmable processors which contain their FIFOs in the core) as well as accompanying asynchronous interfacing logic. Basic modifications are done to tailor the wrapper to the specific processor including adding stalling logic to halt the clock whenever the core logic is idle, customizing the configuration logic to add hardware reconfigurability, and adding processor specific test logic to send critical signals to the chip output pads for observation. The individual processor blocks were completed and then placed into the final chip level layout with I/O, test, and configuration glue logic. As a result of maintaining consistent layouts, the power grid and inter-processor wires are kept short and straight. This reduces voltage droops, signal degradation, antenna violations, and electromigration issues. Finally, post-layout timing and functional verification, in addition to DRC and LVS, from oscillator to processor to chip level is done hierarchically.

Local power distribution metal stripes within cores are interleaved between global power metal stripes. A chip level power ring encircles the central array and redistributes the various  $V_{dd}$  and  $Gnd$  supplies from the many power and ground pads which are evenly placed around the die.

### B. Programmable Processor Analysis and Measurements

A breakdown of the programmable processor area is shown in Fig. 20. The processor's core occupies 73% of the total tile area and only 7% is devoted to DVFS related items including PMOS power gates (4%) and the DVFS controller (3%). Inter-processor communication circuits require only 7% of the tile area, which includes I/O and Route (5%) and Clock Tree and Buffers (2%). A total of 11% of the tile's silicon area is unused because of interconnect complexity caused by long distance communication, and area constraints caused by the PMOS power gates and M6/M7 power stripes. Fig. 21 shows the area breakdown of the processor core itself, and the core's logic area breakdown.

When operating at a supply voltage of 1.2 V, programmable processors run up to 1.07 GHz and dissipate an average of 47.5 mW when executing ALU and MAC instructions while 100% active, resulting in an energy dissipation of 44 pJ per operation, or 22 pJ per operation if a MAC is considered as two operations.

At a supply voltage of 1.3 V, programmable processors run at 1.2 GHz while consuming 62 mW. At this clock rate, the chip achieves a throughput of 196.8 GMACs per second not including operations by the dedicated-purpose accelerators.

At a supply voltage of 0.675 V, programmable processors run at 66 MHz while consuming 608  $\mu$ W, which results in an energy of 9.2 pJ per ALU or MAC operation, or 4.6 pJ per operation if a MAC is considered as two operations.

At a supply voltage of 0.467 V, programs that read and write to and from a small data memory made of flip-flops are fully functional over all instruction types. The maximum operating frequency is 563 Hz and is believed limited by instruction memory reads.

The maximum frequency that a processor can run an ALU or MAC instruction for a range of supply voltage levels is shown in Fig. 22. Power dissipation at the corresponding maximum frequencies and voltages are shown in Fig. 23.

A breakdown of power for various functions within a programmable processor is given in Table II. Power numbers depend strongly on factors such as data values, memory address values, and even instruction ordering so the reported power values are averaged over many cases. Leakage currents for one processor are shown in Fig. 24 for the case of all 48 power gates turned "on" and "off".

### C. Dedicated-Purpose Processor and Shared Memory Measurements

The following early measurement data are taken from one chip tested thus far. The FFT processor runs up to 866 MHz and at this frequency it obtains a throughput of 681 million complex samples per second while computing 1024-point complex FFTs. The Viterbi processor, which contains eight Add-Compare-Select (ACS) units, delivers 82 Mbps at a rate of 1/2 with  $k = 7$  codes and a 894 MHz clock rate. The video motion estimation processor can support 1080p HDTV at 30 fps while achieving a throughput of 15 billion SADs (Sums of Absolute Differences) per second at 938 MHz. The shared memories operate up to 1.3 GHz and are capable of achieving a peak throughput of 20.8 Gbps.

### D. Example Applications

The coding of many DSP, multimedia, and general tasks has been completed including filters, convolutional coders, interleaving, sorting, square root, CORDIC  $\sin/\cos/\arcsin/\arccos$ , matrix multiplication, pseudo random number generators, FFTs of lengths 32–1024, a complete  $k = 7$  Viterbi decoder, and a complete fully compliant IEEE 802.11a/11g wireless LAN baseband transmitter [23].

For a 9-processor JPEG encoder implementation [10], the average power consumption per processor is 24 mW while operating at 1.1 GHz with all processors on a 1.3 V supply. When operating with supply voltages of 1.3 V and 0.8 V, the encoder

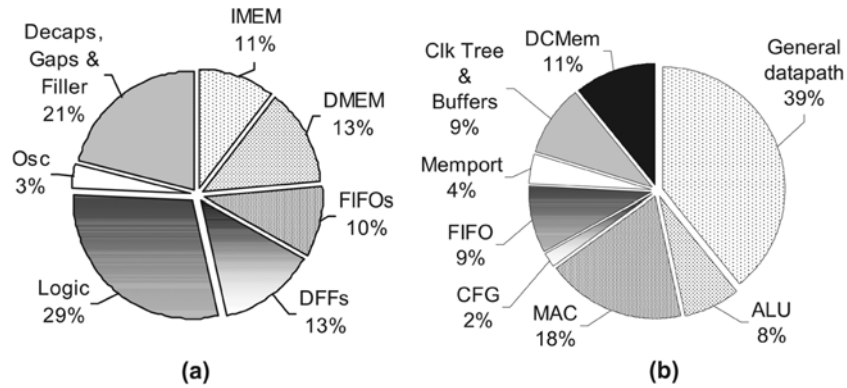


Fig. 21. Area breakdown of the (a) programmable processor's core, and (b) logic only.

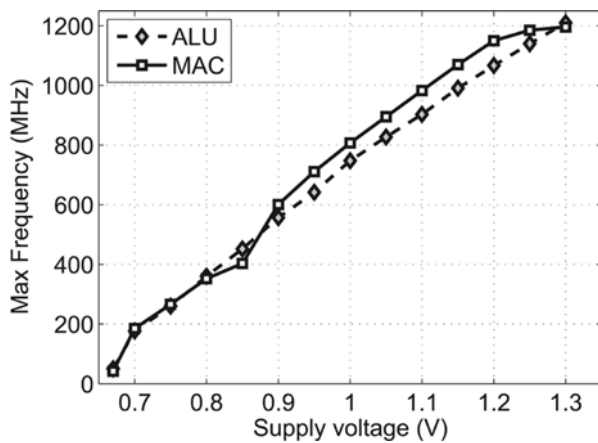


Fig. 22. Measured maximum clock frequencies over various supply voltages for ALU and MAC instructions.

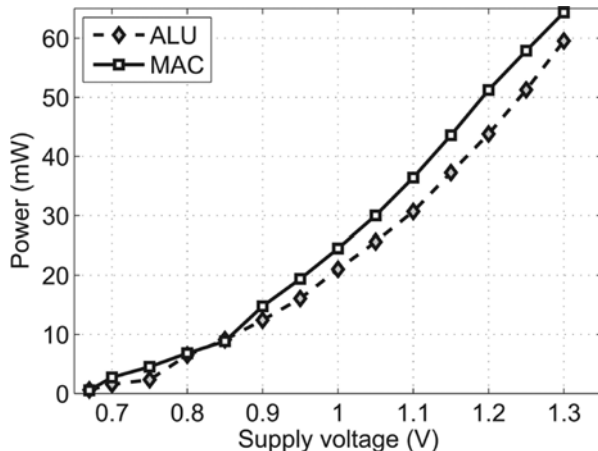


Fig. 23. Measured power dissipation at the maximum clock frequencies shown in Fig. 22 over various supply voltages for ALU and MAC instructions.

achieves a simulated 48% reduction in energy dissipation with an 8% reduction in throughput compared to the same encoder running at 1.3 V.

An H.264/MPEG-4 AVC CAVLC encoder supporting non-zero residual number prediction has been completed. Twenty processors are required when using only nearest-neighbor interconnect, but only 15 processors are required when using long-

TABLE II  
POWER CONSUMPTION OF OPERATIONS AT 1.2 GHz AND 1.3 V. MAC AND ALU NUMBERS INDICATED BY "\*" ARE FROM PROGRAMS WITH HIGH INPUT OPERAND BIT ACTIVITY, INPUT OPERANDS COMING FROM A MIX OF DMEM AND IMMEDIATE SOURCES, AND 100% ACTIVITY

Operation	Power (mW)
FIFO Read/Write	4.6
DMEM Read	13.0
DMEM Write	9.1
Nearest-neighbor comm.	5.9
Long-distance comm. one tile	12.1
Branch	27.2
MAC*	53.3
ALU*	48.5
NOP/stall	31.0

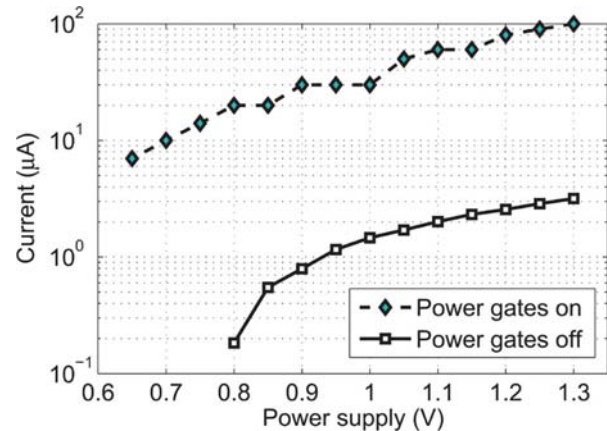


Fig. 24. Measured leakage currents for one programmable processor utilizing all 48 power gates on one power grid.

distance interconnect—a 25% reduction. With processors running at 1.07 GHz, the CAVLC encoder supports 720p HDTV at a real-time rate of 30 fps [24].

A complete and lightly optimized IEEE 802.11a/11g wireless LAN baseband receiver has been completed using 22 programmable processors plus the Viterbi and FFT dedicated-purpose processors. The receiver obtains a real-time 54 Mbps throughput and consumes 198 mW while operating at 590 MHz and 0.95 V, which includes 1.7 mW for the FFT and 6.5 mW for the Viterbi [1]. Power numbers are derived from the measured operational power along with the activity percentage of processors.

## VII. CONCLUSION

A 167-processor single-chip many-core computational platform that is well-suited for DSP, embedded, and multimedia workloads has been fabricated in 65 nm CMOS. The chip contains 164 programmable processors with per-processor dynamic clock frequency and per-processor dynamic supply voltage capabilities, and a two-layer interprocessor interconnect that is capable of direct long-distance connections, which occupies only 7% of each tile's circuit area.

To broaden the target application domain space, the processing array includes three 16 KB shared memories, an FFT processor (for general DSP applications), a Viterbi processor (for communications applications), and a video motion estimation processor (for video multimedia applications).

The per-processor dynamic clock frequency and supply voltage circuits reduce the power of a 9-processor JPEG encoder operating at supply voltages of 1.3 V and 0.8 V by 48% while reducing its performance by 8%. At a supply voltage of 1.3 V, the chip achieves 196.8 billion ALU or MAC operations per second, not considering the accelerators, while dissipating 10.2 W. At a supply voltage of 0.675 V, the programmable processors operate at 66 MHz and dissipate 608  $\mu$ W, which means 93 chips would achieve 1 Tera-op/sec at a power of only 9.2 W, or 47 chips at a power of 4.6 W if a MAC is considered as two operations.

## ACKNOWLEDGMENT

The authors gratefully acknowledge fabrication by ST Microelectronics, and thank J.-P. Schoellkopf, P. Coge, K. Torki, S. Dumont, Y.-P. Cheng, R. Krishnamurthy, K. Bowman, M. Anders, and S. Mathew.

## REFERENCES

- [1] A. T. Tran, D. N. Truong, and B. M. Baas, "A complete real-time 802.11a baseband receiver implemented on an array of programmable processors," in *Proc. Asilomar Conf. Signals, Systems and Computers (ACSSC)*, Oct. 2008, pp. MA5–6.
- [2] D. Truong, W. Cheng, T. Mohsenin, Z. Yu, T. Jacobson, G. Landge, M. Meeuwsen, C. Watnik, P. Mejia, A. Tran, J. Webb, E. Work, Z. Xiao, and B. Baas, "A 167-processor 65 nm computational platform with per-processor dynamic supply voltage and dynamic clock frequency scaling," in *Symp. VLSI Circuits Dig.*, Jun. 2008, pp. 22–23.
- [3] Z. Yu, M. J. Meeuwsen, R. W. Apperson, O. Sattari, M. Lai, J. W. Webb, E. W. Work, D. Truong, T. Mohsenin, and B. M. Baas, "AsAP: An asynchronous array of simple processors," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 695–705, Mar. 2008.
- [4] M. B. Taylor *et al.*, "A 16-issue multiple-program-counter micro-processor with point-to-point scalar operand network," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2003, pp. 170–171.
- [5] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar, "An 80-tile 1.28 TFLOPS network-on-chip in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2007, pp. 98–99.
- [6] Z. Yu and B. M. Baas, "High performance, energy efficiency, and scalability with GALS chip multiprocessors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 66–79, Jan. 2009.
- [7] S. Borkar, "Thousand core chips—a technology perspective," in *44th Annual Conf. Design Automation (DAC)*, Jun. 2007, pp. 746–749.
- [8] D. Truong, W. Cheng, T. Mohsenin, Z. Yu, T. Jacobson, G. Landge, M. Meeuwsen, C. Watnik, P. Mejia, A. Tran, J. Webb, E. Work, Z. Xiao, and B. Baas, "A 167-processor computational array for highly-efficient DSP and embedded application processing," in *HotChips Symp. High-Performance Chips*, Aug. 2008, session 2, Stanford University, Palo Alto, CA.
- [9] R. W. Apperson, Z. Yu, M. J. Meeuwsen, T. Mohsenin, and B. M. Baas, "A scalable dual-clock FIFO for data transfers between arbitrary and halttable clock domains," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1125–1134, Oct. 2007.
- [10] Z. Yu, M. Meeuwsen, R. Apperson, O. Sattari, M. Lai, J. Webb, E. Work, T. Mohsenin, M. Singh, and B. Baas, "An asynchronous array of simple processors for DSP applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2006, pp. 428–429.
- [11] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Stanford Univ., Stanford, CA, Oct. 1984.
- [12] M. J. Meeuwsen, Z. Yu, and B. M. Baas, "A shared memory module for asynchronous arrays of processors," *EURASIP J. Embedded Syst.*, vol. 2007, 2007, Article ID 86273, 13 pages.
- [13] Z. Yu and B. M. Baas, "A low-area interconnect architecture for chip multiprocessors," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2008, pp. 2857–2860.
- [14] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits—A Design Perspective*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.
- [15] P. Hazucha, G. Schrom, J. Hahn, B. A. Bloechel, P. Hack, G. E. Dermer, S. Narendra, D. Gardner, T. Karnik, V. De, and S. Borkar, "A 233-MHz 80%–87% efficient four-phase DC-DC converter utilizing air-core inductors on package," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 838–845, Apr. 2005.
- [16] J. Kim and M. Horowitz, "An efficient digital sliding controller for adaptive power-supply regulation," *IEEE J. Solid-State Circuits*, vol. 37, no. 5, pp. 639–647, May 2002.
- [17] J. Wibben and R. Harjani, "A high efficiency DC-DC converter using 2 nH on-chip inductors," in *Symp. VLSI Circuits Dig.*, Jun. 2007, pp. 22–23.
- [18] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *Proc. Int. Symp. High-Performance Computer Architecture (HPCA)*, Feb. 2008, pp. 123–134.
- [19] W. H. Cheng and B. M. Baas, "Dynamic voltage and frequency scaling circuits with two supply voltages," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2008, pp. 1236–1239.
- [20] K. Agarwal and K. Nowka, "Dynamic power management by combination of dual static supply voltage," in *Proc. Int. Symp. Quality Electronic Design*, Mar. 2007, pp. 85–92.
- [21] E. Beigné, F. Clermidy, S. Miermont, and P. Vivet, "Dynamic voltage and frequency scaling architecture for units integration within a GALS NoC," in *Proc. IEEE Int. Symp. Networks-on-Chip (NOCS)*, Apr. 2008, pp. 129–138.
- [22] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan. 2006.
- [23] M. J. Meeuwsen, O. Sattari, and B. M. Baas, "A full-rate software implementation of an IEEE 802.11a compliant digital baseband transmitter," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS 2004)*, Oct. 2004, pp. 124–129.
- [24] Z. Xiao and B. M. Baas, "A high-performance parallel CAVLC encoder on a fine-grained many-core system," in *IEEE Int. Conf. Computer Design (ICCD)*, Oct. 2008, pp. 248–254.



**Dean N. Truong** (S'07) received the B.S. degree in electrical and computer engineering from the University of California, Davis, in 2005. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California, Davis. His research interests include high-speed processor architectures, dynamic supply voltage and dynamic clock frequency algorithms and circuits, and VLSI design. Mr. Truong was a key designer of the second generation 167-processor 65 nm CMOS Asynchronous Array of simple Processors (AsAP) chip.



**Wayne H. Cheng** (M'08) received the B.S. degree in electrical engineering from the University of California, San Diego, in 2005, and the M.S. degree in electrical and computer engineering from the University of California, Davis, in 2008.

He is currently an Engineer at an early-stage startup company, and is also working towards the M.B.A. degree at San Francisco State University. His research interests include low-power and high-performance VLSI design, and dynamic supply voltage circuits.



**Tinoosh Mohsenin** (M'04) received the B.S. degree in electrical engineering from Sharif University, Tehran, Iran, and the M.S. degree in electrical and computer engineering from Rice University, Houston, TX. She is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California, Davis.

She is the designer of the Split-Row, Multi-Split, and Split-Row Threshold decoding algorithms and architectures for low density parity check (LDPC) codes. Her research interests include energy-efficient

and high-performance signal processing and error correction architectures including multi-gigabit full-parallel LDPC decoders and many-core processor architecture design.



**Zhiyi Yu** (S'04–M'07) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Davis, in 2007. While at UC Davis, he was a key designer of the 36-core Asynchronous Array of simple Processors (AsAP) chip, and was one of the designers of the 167-core second generation computational array chip, focusing on the reconfigurable dual-link inter-processor network.

Dr. Yu is currently an Associate Professor with the ASIC & System State Key Lab, Microelectronics Department, Fudan University, Shanghai, China. His research interests include high-performance and energy-efficient digital VLSI design with an emphasis on many-core processors. From 2007 to 2008, he was with IntellaSys Corporation, Cupertino, CA, where he participated in the design of the many-core SEAForth chips which utilize stack-based processors with extremely small area and low power consumption.



**Anthony T. Jacobson** received B.S. degrees in electrical engineering and mathematics from the University of Idaho, Moscow, ID, in 2004, and the M.S. degree in electrical and computer engineering from the University of California, Davis, in 2007. He is now working towards the J.D. degree at the Boalt School of Law at the University of California, Berkeley. His research contributions include the design of a continuous-flow 870 MHz 65 nm CMOS configurable complex fast Fourier transform (FFT) processor which computes complex transforms with lengths from 16

points to 4096 points.



**Gouri Landge** received the B.S. degree in electronics engineering from Pune University, India, in 1994. She is currently pursuing the M.S. degree in electrical and computer engineering at the University of California, Davis.

Her research interests include digital video processing and other digital signal processing applications, and her contributions include a 940 MHz 65 nm CMOS programmable H.264 video motion estimation processor that can compute real-time 30 fps 1080p HDTV. She is now a Staff Engineer at Intel

Corporation. She is working on micro-architecture and design of multi-million gate SoC chips, involving communication, digital signal processing and display applications.



**Michael J. Meeuwsen** received the B.S. degrees with honors in electrical engineering and computer engineering (both *summa cum laude*) from Oregon State University, Corvallis, and the M.S. degree in electrical and computer engineering from the University of California, Davis, in 2005. His research contributions include the design of a 1.3 GHz 65 nm CMOS four-port 16 KB shared memory.

He is currently a Hardware Engineer with Intel Digital Enterprise Group, Hillsboro, OR, where he works on CPU hardware design. His research interests include digital circuit design and CPU memory system architecture.

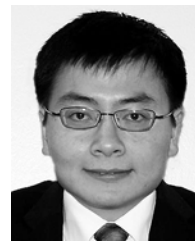


**Christine Watnik** received the B.S. degree in electrical and computer engineering from the University of California, Davis, in 1998. She is now working towards the M.S. degree in computer engineering at the University of California, Davis, where she is a member of the VLSI Computation Laboratory. Her research contributions include the design of a 900 MHz 65 nm CMOS configurable Viterbi decoder which was successfully fabricated as part of the Asynchronous Array of simple Processors (AsAP) project.

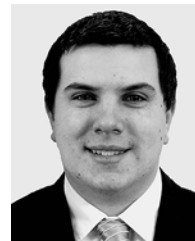
She is also a Senior Component Design Engineer at Intel Corporation where she has worked on chipset design since 1997.



**Anh T. Tran** (S'07) received the B.S. degree with honors in electronics engineering from the Posts and Telecommunications Institute of Technology, Saigon, Vietnam, in 2003. He is currently working towards the Ph.D. degree in electrical and computer engineering at the University of California, Davis. His research interests include VLSI design, multi-core architecture, on-chip interconnects, and software-defined baseband radio receivers. He has been a VEF Fellow since 2006.



**Zhibin Xiao** (S'07) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2003 and 2006, respectively, where his research focused on high-performance multimedia processor design. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California, Davis. His research interests include high-performance many-core processor architecture, parallel video encoding implementations, and scalable memory system design.



**Eric W. Work** (M'06) received the B.S. degree from the University of Washington, Seattle, in 2004, and the M.S. degree in electrical and computer engineering from the University of California, Davis, in 2007.

He is currently an Engineer at an early-stage startup company. His research interests include the algorithms and software tools for mapping arbitrary task graphs to processor networks, and other software tools for hardware design.



**Jeremy W. Webb** (M'00) received the B.S. degree in electrical and computer engineering in 2000 from the University of California, Davis, where he is currently pursuing the M.S. degree in electrical and computer engineering.

He is a Senior Digital Hardware Engineer at Centellax working on high-speed serial bit error-rate testers. He has also held design positions at Agilent Technologies and Barco-Folsom. His research interests include high-speed board design and system interfacing.



**Paul V. Mejia** (M'05) received the B.S. degree in computer engineering from the University of the Philippines, Diliman, in 2004. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California, Davis. His research interests include computer architecture and algorithms.



**Bevan M. Baas** (M'99) received the B.S. degree in electronic engineering from California Polytechnic State University, San Luis Obispo, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1990 and 1999, respectively.

From 1987 to 1989, he was with Hewlett-Packard, Cupertino, CA, where he participated in the development of the processor for a high-end minicomputer. In 1999, he joined Atheros Communications, Santa Clara, CA, as an early employee and served as a core

member of the team which developed the first IEEE 802.11a (54 Mbps, 5 GHz) Wi-Fi wireless LAN solution. In 2003, he joined the Department of Electrical and Computer Engineering at the University of California, Davis, where he is now an Associate Professor. He leads projects in architecture, hardware, software tools, and applications for VLSI computation with an emphasis on DSP workloads. Recent projects include the 36-processor Asynchronous Array of simple Processors (AsAP) chip, applications, and tools; a second generation 167-processor chip; low density parity check (LDPC) decoders; FFT processors; Viterbi decoders; and H.264 video codecs. During the summer of 2006, he was a Visiting Professor in Intel's Circuit Research Lab.

Dr. Baas was a National Science Foundation Fellow from 1990 to 1993 and a NASA Graduate Student Researcher Fellow from 1993 to 1996. He was a recipient of the National Science Foundation CAREER Award in 2006 and the Most Promising Engineer/Scientist Award by AISES in 2006. He is an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS and has served as a member of the Technical Program Committee of the IEEE International Conference on Computer Design in 2004, 2005, 2007, and 2008, and on the Program Committee of the HotChips Symposium on High Performance Chips in 2009. He also serves as a member of the Technical Advisory Board of an early stage technology company.