ADAPTIVE SYSTEM ON A CHIP (ASOC): A BACKBONE FOR POWER-AWARE SIGNAL PROCESSING CORES

Andrew Laffely, Jian Liang, Russell Tessier, Wayne Burleson

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA. 01003
{alaffely, jliang, tessier, burleson}@ecs.umass.edu

ABSTRACT

For motion estimation (ME) and discrete cosine transform (DCT) of MPEG video encoding, content variation and perceptual tolerance in video signals can be exploited to gracefully trade quality for low power. As a result, power-aware hardware cores have been proposed for these video encoding subsystems. ***A****daptive* ***S****ystem-****o****n-a-****C****hip, aSoC, supports power-aware cores by providing an on-chip communications framework designed to promote scalability and flexibility in system-on-a-chip designs. This paper describes aSoC's ability to dynamically control voltage and frequency scaling through a simple voltage and frequency selection scheme. A small demonstration system is tested and shows up to 90% reduction in core power when the aSoC voltage scaling features are enabled.*

1. INTRODUCTION

The demand for portable image and video processing continues to increase in products including PDAs, hand held gaming units and cell phones. To meet the power and performance demands of these applications, many hardware architectures have been proposed for specific subsystems [1, 2]. Many recently proposed subsystems include features, which control power and performance trade-offs at run-time [3, 4]. Given the development of these intellectual property (IP) subsystems, or cores, the main challenge lies in integrating them into a single system capable of leveraging their individual flexibilities.

In this paper, Adaptive System-on-a-Chip (aSoC) is used as a backbone for power-aware video processing cores. In our previous work [5] we showed that aSoC, by nature of its statically scheduled mesh interconnect, performs up to 5 times faster than bus-based architectures. Additionally, interconnect usage for typical digital signal processing applications is under 20%. This leaves significant interconnect bandwidth to accommodate the control communications required by the power-aware features of modern cores [6].

ASoC's ability to provide dynamic voltage and frequency scaling is critical to future portable digital signal processing applications. This will allow SoC implementations to exploit the inevitable mismatches in core utilization, due to data content variations or user requirements, to reduce power consumption. A simple clock division scheme makes it possible for each core to select and switch between 8 different frequencies. Based on principles described in [7], the clock selection scheme is complemented by a voltage scaling procedure, which allows the individual core supply voltages to switch between 4 different values. System power consumption is reduced since each core operates at a voltage and frequency, that is coarsely tuned for its specific utilization. To demonstrate this, we use a partial video encoding system consisting of a motion estimation (ME) core and a discrete cosine transform (DCT) core. This simple system shows how the availability of multiple core frequencies and voltages reduces individual core power by 90%. Additionally, we describe a hardware control system that automatically selects the voltage and frequency of each core at run-time by monitoring interconnect utilization.

This paper proceeds as follows. Section 2 presents a brief overview of the aSoC architecture. Section 3 presents the power management techniques used in aSoC. The experimental approach and results for our demonstration system are described in Section 4. Section 5 concludes the paper and suggests future work.

2. SCHEDULED COMMUNICATION ARCHITECTURE

Many recently proposed SoC and Network-on-Chip (NoC) architectures use a tile-based floorplan and a point-to-point mesh interconnect structure [5, 8, 9, 10, 11, 12]. As shown

This work was supported by The National Science Foundation under Grant Numbers CCR-0081405, CCR-9988238, CCR-9875482.

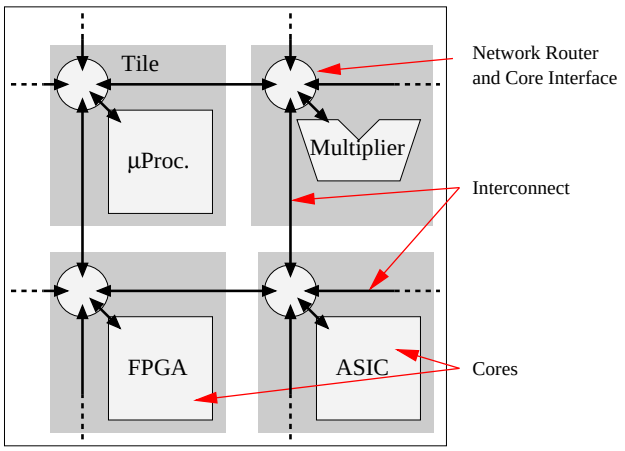

Fig. 1. Tiled Architecture

in Figure 1, each tile in these architectures includes a computational core and its interface to the network. Our approach to SoC integration, aSoC, is a tiled architecture, which supports the use of heterogeneous processing cores occupying one or more tiles [5]. ASoC connects these tiles using a statically scheduled mesh of interconnect, which assures predictable inter-core communication. Data moves between neighboring tiles in a communication pipeline, enabling fast clock rates and time sharing of interconnect resources. The interconnect is reconfigurable at run-time to allow for dynamic communication patterns.

The core interface uses a synchronized global communications schedule to manage communications through each tile. As shown in Figure 2, the instruction memory holds a list of the communication patterns required at run-time. A program counter (PC) fetches these patterns in succession and a decoder converts them into switch settings for a crossbar. The crossbar routes data between the local core and the neighboring tiles (North, East South or West). Each incoming data word can contain local interface configuration information to be sent over the *local config.* line to the controller. The *core-ports* in Figure 2 use a simple protocol to interface communications between the potentially different clock domains of the core and interconnect. Multiple input and output *core-ports* can be used depending on the core and application requirements. During normal operations, the controller simply loops through the communications schedule.

3. DYNAMIC POWER MANAGEMENT FOR ASOC

Dynamic power management exploits run-time variations in data content and operational requirements to minimize one or more of the terms in the VLSI power equation [7], shown in Equation 1. Our previous work showed how aSoC

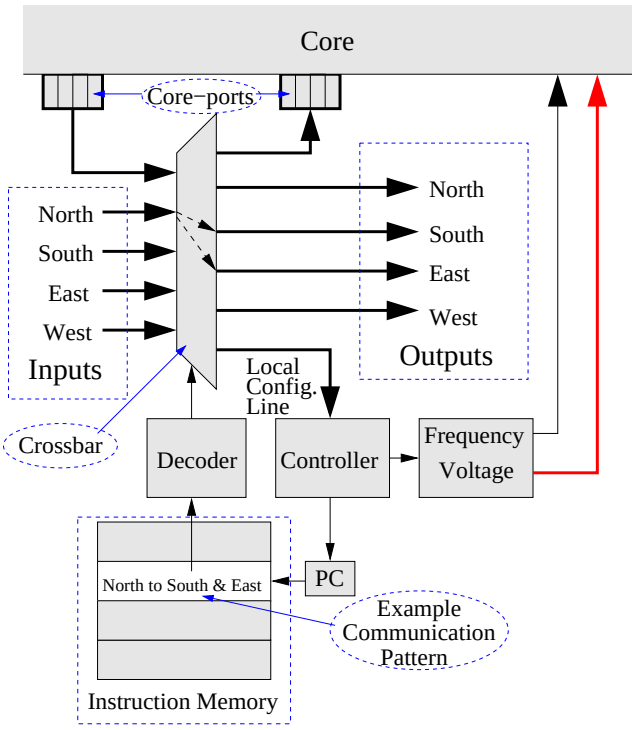

Fig. 2. Core and Communication Interface

supports the reduction of effective capacitance (C_{eff}) by enabling core adaptation [6]. This paper presents a simple frequency (f) and voltage (V_{dd}) scaling system and procedure.

$$P_{ave} = C_{eff} \times V_{dd}^2 \times f \quad (1)$$

To meet critical path requirements, independently developed heterogeneous cores may require independent clock and voltage domains. Additionally, reconfigurable IP cores may require the reconfigurability of both the clock and supply voltage. As a result, much of the overhead for adaptive clock and supply selection already exists in heterogeneous SoC.

Figure 3 shows our approach to coupled frequency and voltage scaling. At each core, frequency and voltage are automatically adjusted using a four part system. The first subsystem, *Data Rate Measurement*, uses up/down counters to track the data transfer rate between core and interconnect. Blocked or unsuccessful transfers cause the count to increase, while successful transfers decrease the value. If the core input port is blocked consecutively, the core is running too slowly with respect to its predecessors. If the core output port is consecutively blocked, the core is running too quickly for its successors. In either case, these counters send trigger signals to the core configuration unit to increase or decrease the core clock. To change the clock, the *Clock Selector* selects a different frequency. Eight different fre-

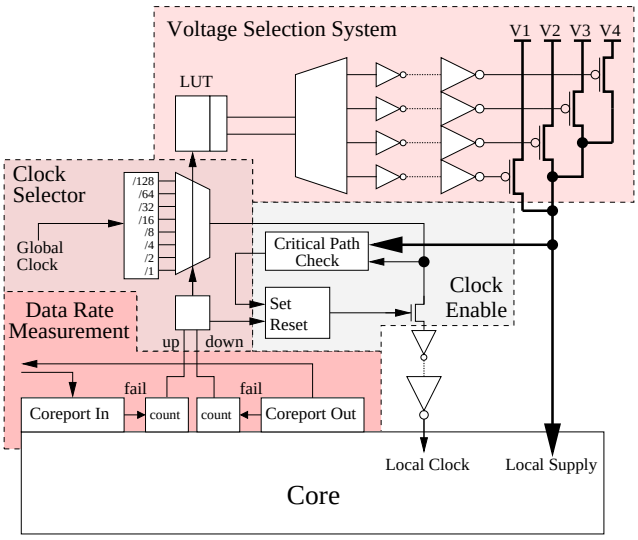

Fig. 3. Dynamic Voltage Selection

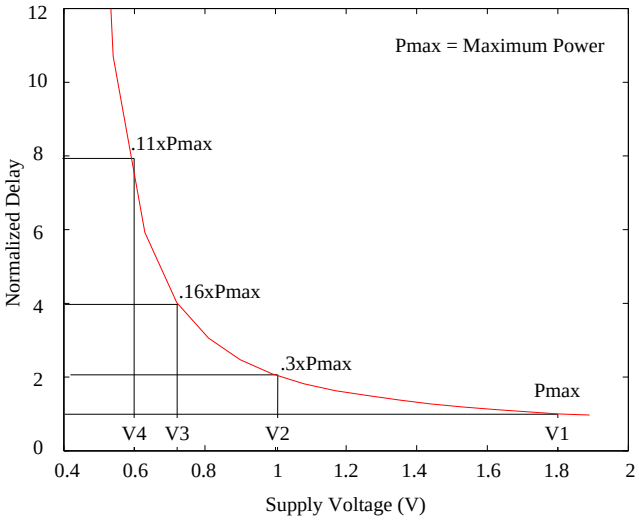

Fig. 4. Voltage Values Using Delay/Voltage

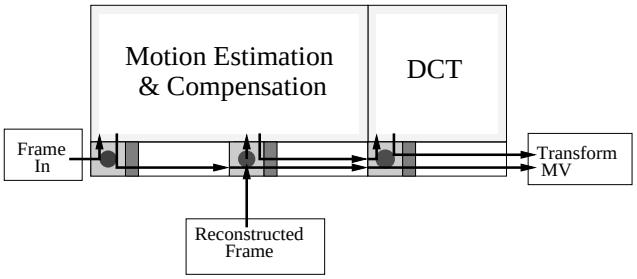

Fig. 5. Test System

quencies are made available by successively dividing a high frequency global clock signal by multiples of two. Using a small look up table (LUT) the new frequency setting automatically selects a new supply voltage from the available global supply values in the *Voltage Selection System*. Only four voltage values, $V1$ to $V4$, are used to save system overhead. During the transition the local clock is disabled until test data can successfully pass through a reconstructed core critical path [13] in the *Clock Enable* system. When the clock is changed, a latch is reset blocking the transmission of the new clock signal to the core. In the *Clock Enable* system, the *Critical Path Check* models the critical path of the core. When a bit of data can successfully pass through the *Critical Path Check*, the latch is set and the new clock can propagate to the core. This prevents data loss in the core during voltage and frequency changes.

4. METHODS AND RESULTS

Schematic and Layout-level models of voltage scaling subsystems are developed, extracted and evaluated using 0.18 micron CMOS models from the Berkeley Predictive Technology Models [14]. The voltage levels $V1$ to $V4$ in Figure 3, are chosen based on the available clock frequencies and the approximated critical path of the processing cores as shown in Figure 4.

The *Normalized Delay* in Figure 4 shows the relative changes in critical path delay for the changes in voltage. This is done by dividing the critical path delay by the delay when a supply of 1.8 V is applied to the system. With this information the values of the four selectable voltages, $V1$ to $V4$, can be chosen based on the desired relative clock frequencies. The core can be driven by the supply voltage $V2$ when the core processing delay can be twice the minimum possible delay. When slower frequencies can be used, operating at the voltages, $V2 = 1V$, $V3 = 0.72V$, and $V3 = 0.6V$ reduces core power by 70%, 84% and 90% respectively.

A major issue with voltage scaling is its overhead in system performance, power and area. Using gating transistors with multi-stage drivers, the power grid for even large (5×10^6 gates) cores can be made to switch voltages within 30ns. Switching time is on the order of 10 aSoC interconnect clock cycles. The multi-stage driver and gating transistor for any core with 10^4 to 5×10^6 gates, uses only 0.15% of the area used by the core. The energy consumption in switching voltages ranges from 2.49×10^{-12}J to 1.22×10^{-9}J per switch over the tested range of core sizes. Thus, the largest cores could switch voltages 1000 times a second before the power overhead became noticeable at approximately 1mW.

Our demonstration system, shown in Figure 5, is a simple combination of video encoding cores developed in [3, 15]. The DCT is a replicated row accumulate (RAC) unit implementation [15], which includes dynamic power sav-

Core: Mode	Optimal Frequency (MHz)	aSoC Frequency (MHz)	Power Reduction
ME:FS64x32	105	105	0%
ME:FS16x16	13.1	13.13	90%
ME:spiral	9.9	13.13	90%
ME:TSS	2.75	3.28	90%
DCT	9.6	13.13	90%

Table 1. Power for Modes and Clock Rates

ings mechanisms such as most significant bit (MSB) rejection and row column classification (RCC), as found in [4]. These mechanisms create a system throughput, which varies with data content. The ME core permits selection of several search algorithms including full, spiral and three step [1]. The optimal operating frequency for this core varies dramatically with the selected search algorithm. The second column of Table 1 shows the best average frequency for these subsystems when processing 360 by 240 pixel frames at 30 frames a second. With the global clock fixed at 105 MHz, aSoC provides the following frequencies in MHz: 105, 52.5, 26.25, 13.13, 6.56, 3.28, 1.64, and 0.82. The third column of Table 1 shows aSoC's best matching clock frequency. Setting the core frequencies based on aSoC's best match results in the subsystem power reduction shown in the last column. Notice that for these cores nearly every configuration results in the use of the lowest voltage available and the best power reduction according to Equation 1.

Finally, in a dynamic system, the addition of voltage scaling greatly reduces power consumption. In [3], we show how to use the magnitude of motion vectors to select the search range for motion estimation in the upcoming frames. With this approach we were able to evaluate motion vectors with small, 16×16 pixels, search windows nearly 70% of the time. This search window reduction saved nearly 60% of the power for motion estimation with only a 2% reduction in quality [3]. Applying dynamic voltage scaling to this system saves an additional 20% in power.

5. SUMMARY AND FUTURE WORK

This paper presents the dynamic power management capability of aSoC applied to video processing systems. A methodology has been presented to use aSoC core-port monitoring to dynamically vary both frequency and voltage individual cores. Reconfigurable clock based system balancing creates an environment of just in time computing, which can reduce overall power usage. When coupled with coarse grained voltage selection, this method can reduce core power by 90%. The overhead of this type of system was shown to be insignificant.

Presently, a C-based simulator is being modified to support the adaptive frequency and voltage mechanism. We hope to show that many real applications can benefit from this dynamic voltage scaling.

6. REFERENCES

[1] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, Norwell, MA, 1999.

[2] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards, Algorithms and Architectures, Second Edition*, Kluwer Academic Publishers, Norwell, MA, 1997.

[3] P. Jain, "Parameterized motion estimation architecture for dynamically varying power and compression requirements," M.S. thesis, University of Massachusetts, Amherst, Department of Electrical and Computer Engineering, 2001.

[4] T. Xanthopoulos and A. Chandrakasan, "Low-power DCT core using adaptive bitwidth and arithmetic activity exploiting signal correlations and quantization," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, May 2000.

[5] J. Liang, S. Swaminathan, and R. Tessier, "aSoC: A scalable, single-chip communications architecture," in *Proceedings, International Conference on Parallel Architectures and Compilation Techniques*, Oct. 2000.

[6] A. Laffely, J. Liang, P. Jain, N. Weng, W. Burleson, and R. Tessier, "Adaptive system on a chip (aSoC) for low-power signal processing," in *Proceedings, Thirty-Fifth Asilomar Conference on Signals, Systems, and Computers*, Nov. 2001.

[7] L. Benini and G. De Micheli, *Dynamic Power Management, Design Techniques and Cad Tools*, Kluwer Academic Publishers, Norwell, MA, 1998.

[8] A. Alsolaim et al., "Architecture and application of a dynamically reconfi gurable hardware array for future mobile communication system," in *Proceedings, FCCM*, Apr. 2000.

[9] E. Cheng et al., "Balancing the interconnect topology for arrays of processors between cost and power," in *Proceedings, IEEE International Conference on Computer Design: VLSI in Computers and Processors*, Sep. 2002.

[10] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proceedings, 38th Design Automation Conference*, Jun. 2001.

[11] S. Kumar, A. Jantsch, M. Millberg, J. berg, J. Soininen, M. Forsell, K. Tiensyrj, and A. Hemani, "A network on chip architecture and design methodology," in *Proceedings, IEEE Computer Society Annual Symposium on VLSI*, Apr. 2002.

[12] R. Marculescu, "Networks-on-chip: The quest for on-chip fault tolerant communication," in *Proceedings, IEEE Annual Symposium on VLSI*, Feb. 2003.

[13] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, and T. Furuyama, "Variable supply-voltage scheme for low-power high-speed cmos digital design," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, Mar. 1999.

[14] Y. Cao, T. Sato, D. Sylvester, M. Orchansky, and C. Hu, "New paradigm of predictive mosfet and interconnect modeling for early circuit design," in *Proceedings, IEEE Custom Integrated Circuits Conference*, Jun. 2000.

[15] S. Venkatraman, "A power-aware synthesizable core for the discrete cosine transform," M.S. thesis, University of Massachusetts, Amherst, Department of Electrical and Computer Engineering, 2001.