Circuit Modeling for Practical Many-core Architecture Design Exploration

Dean N. Truong
University of California, Davis
Electrical and Computer Engineering
hottruong@ucdavis.edu

Bevan M. Baas
University of California, Davis
Electrical and Computer Engineering
bbaas@ucdavis.edu

ABSTRACT

Current tools for computer architecture design lack standard support for multi- and many-core development. We propose using circuit models to describe the multiple processor architecture to motivate a circuit based design approach applied to a computer architecture problem. A test chip with DVFS capable processors is used to explore our ideas through software implementation.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Performance of Systems—*Modeling techniques*

General Terms

Experimentation, Design, Theory, Performance

Keywords

many-core, dynamic frequency and voltage scaling, control

1. INTRODUCTION

Research on modeling multi-core architectures has focused on cycle-accurate models (e.g. Verilog RTL), event-based models (e.g. System-C), feedback control models [1, 2, 3], and architectural simulators [4, 5] but they can quickly overwhelm the designer while obscuring qualitative insight. Integrated circuit design requires an appreciation of the physics behind each element and understanding when to select models based on appropriateness while avoiding unnecessary accuracy. By using circuit modeling of architecture parameters, this intuitive design method can also be applied to many-core architecture development.

We will test this approach on a basic model of a queue-based many-core architecture where processor design parameters (e.g. CPI, clock frequency, FIFO-depth, etc.) can be swapped with basic circuit element analogues. Initial verification of the methodology is done through software implementation and execution on a many-core chip [6].

2. PROPOSED CIRCUIT MODEL

Figure 1(a) shows an abstract system dataflow or graph model of a queue-based (i.e. FIFO-based) consumer and producer architecture. Multi-core architectures that are targeted at stream computing will have processors communicate in such a manner.

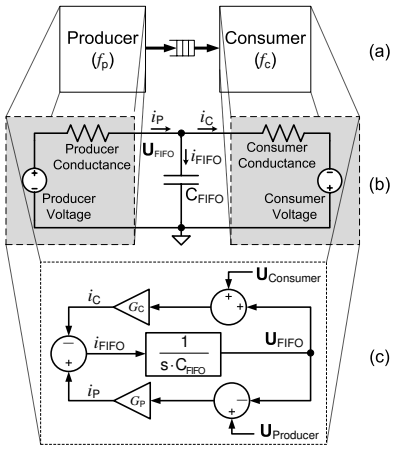

Figure 1: (a) Producer and consumer architecture with a FIFO communication link, (b) equivalent circuit model, and (c) equivalent control system model.

The movement of data words (*samples*) between two processors is analogous to electrical current (I) so we can represent the flow of data over time as *samples/sec* (i.e. data rate). The voltage $\mathbf{U}$ is analgous to the sample production or consumption efficiency of a processor. In other words, the number of instructions (IC) and the number of cycles per instruction (CPI) it takes a processor to produce or consume a data word on its I/O ($cycles/sample$) is comparable to $voltage^{-1}$. Finally, we analogize the clock frequencies ($cycles/sec$) of the producer f_p and consumer f_c by conductance (G) since frequency has a proportional effect on the flow of data. Through Ohms Law, $I = V \times G$, we can now confirm our analogy: $data\_rate = (CPI \times IC)^{-1} \times frequency$, with the units: $\frac{samples}{sec} = (\frac{cycles}{sample})^{-1} \times \frac{cycles}{sec}$.

The proposed equilvalent circuit model and control system model are shown in Figs. 1(b)(c). Let i_P and i_C be the rates of production and consumption of data over time (i.e. current); G_P and G_C are the frequencies (i.e. conductance); and $\mathbf{U}_{Producer}$ and $\mathbf{U}_{Consumer}$ are the computational efficiencies (i.e. voltage) of the producer and consumer, respectively. The FIFO is modeled with a capacitor, where its current and voltage are equal to $i_{FIFO} = i_P - i_C$ and $\mathbf{U}_{FIFO}$, respectively. When $i_{FIFO} = 0$ the flow of data does not stall the processor(s) because the FIFO does not fill or empty. In contrast, $i_{FIFO} \neq 0$ causes $\mathbf{U}_{FIFO} > 0$ or $\mathbf{U}_{FIFO} < 0$ meaning that the producer or consumer will stall due to a FIFO full or empty state, respectively.

DAC 2010, June 13-18, 2010, Anaheim, California, USA.

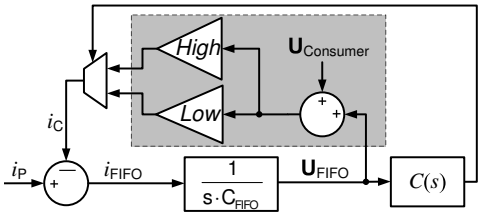

Figure 2: The feedback control model for the consumer processor when using DVFS.

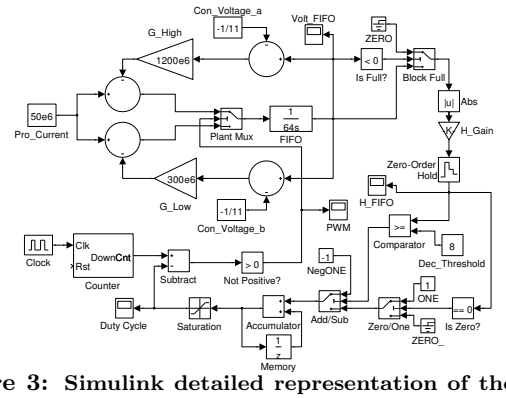

Figure 3: Simulink detailed representation of the consumer's control system (Fig. 2). Note: FIFO size = 64 samples, a fixed producer i_P = 50 MSamples/sec, two clock frequencies: 1.2 GHz and 300 MHz, for operating at Vdd_{High} and Vdd_{Low}, respectively, and 11 instructions per sample consumption (with $CPI = 1$).

3. CASE STUDY: DVFS CAPABLE MANY-CORE ARCHITECTURE

For a Dynamic Voltage and Frequency Scalable (DVFS) architecture targeted for real-time applications, finishing a task before a deadline while being energy efficient requires operating at the slowest possible frequency and lowest possible supply voltage to meet the deadline exactly [7].

Intuition can guide us to find the best DVFS policy. For instance, the circuit network in Fig. 1(b) form a resistive-capacitive network where the FIFO acts as a low pass filter that filters out quick changes in the data rate, which influences our design choices. In fact, the capacitor's voltage, $\mathbf{U}_{FIFO}$, determines the efficiency of the system since the most efficient state is when $\mathbf{U}_{FIFO} = 0$. To achieve this goal the clock frequency of the two processors must be tuned to exactly meet a real-time throughput constraint. A DVFS controller that changes a processor's frequency is analogous to a resistance change, which is the same as a variable gain; thus, the control system is a non-linear adaptive system. We can simplify the model by proposing a controller that acts on two possible "plants" (or "processes") where there are two operating points each with a voltage and frequency: Vdd_{High} & $Freq_{High}$, and Vdd_{Low} & $Freq_{Low}$. Operating at an optimal energy point requires finding its equivalent optimal supply voltage switching duty cycle (i.e. voltage dithering [8]). The simplified control system concept is shown in Fig. 2, where we have a consumer processor receive data from a static producer (e.g. when receiving data from an ADC).

From this circuit-to-control model we developed an algorithm through Simulink. The algorithm and its parameters are then estimated and further refined using the actual chip to verify the model. Figure 3 shows the Simulink schematic of our simple example (Fig. 2). Because of the difficulties of accurately modeling discrete time and continous time elements some estimates are made to keep the model tractible without sacrificing too much of the system (or hardware) intuition. (For example, because the period between two dithering events is much larger than the rate at which the FIFO status sensor samples data, we can assume that the sensor is a continous time system.) Given a fixed static rate of production, i_P, we can find the two cases representing the worst and best cases in terms of energy efficiencies while meeting the producer's data rate. The measured results of those and our DVFS algorithm are shown in Table 1. Figure 4 shows the final state of the DVFS algorithm.

Worst Case ($Vdd_{High} = 1.3\ V$)	46.5 mW
Ideal Case ($Vdd_{High} = 0.9\ V$)	12.0 mW
DVFS ($Vdd_{High} = 1.3\ V$ & $Vdd_{Low} = 0.8\ V$)	15.6 mW

Table 1: Measured average power of two boundary cases and our DVFS algorithm.

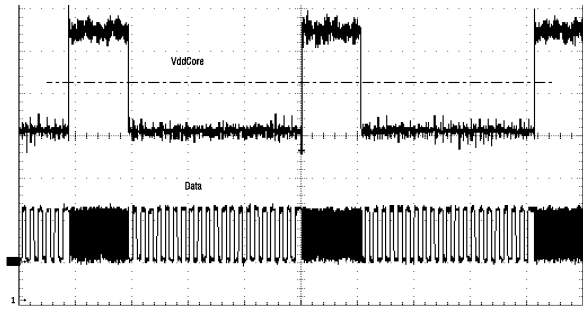

Figure 4: Measured waveforms showing the final dithering state of 25.6% Vdd_{High} and 74.4% Vdd_{Low} over a dither period of 4 us reached by our DVFS algorithm (along with the changing data rate (i.e. frequency)).

4. CONCLUSION

Simple circuit element analogies were described for a simple queue-based many-core architecture. Through the use of a circuit model to control system based design methodology, we show that the intuition of architectural behavior as seen through circuit behavior can create a heuristically derived control system model that preserves the physical essense of the architecture. Such a method can help spur the use of circuit based simulation for a computer architecture targeted EDA.

5. REFERENCES

[1] S. Garg et al. Technology-driven limits on DVFS controllability of multiple voltage-frequency island designs: a system-level perspective. In *DAC*, pages 818–821, 2009.

[2] K. Niyogi and D. Marculescu. System level power and performance modeling of gals point-to-point communication interfaces. In *ISLPED*, August 2005.

[3] A. Alimonda et al. Non-linear feedback control for energy efficient on-chip streaming computation. In *IES*, October 06.

[4] Simplescalar. http://www.simplescalar.com.

[5] K. Samadi et al. ORION 2.0: A fast and accurate noc power and area model for early-stage design space exploration. In *GSRC*, September 2008.

[6] D. N. Truong, W. H. Cheng, et al. A 167-processor computational platform in 65 nm CMOS. *IEEE Journal of Solid-State Circuits (JSSC)*, 44(4):1130–1144, April 2009.

[7] M. Pedram and J. M. Rabaey. *Power Aware Design Methodologies*. Springer-Verlag, 1st edition, 2002.

[8] V. Gutnik and A. P. Chandrakasan. Embedded power supply for low-power DSP. *TVLSI*, 5:425–435, December 1997.