Minimizing Energy Consumption for High-Performance Processing

Eric F. Weglarz, Kewal K. Saluja, and Mikko H. Lipasti

University of Wisconsin-Madison
Madison, WI 53706 USA
{*weglarz,saluja,mikko*}*@ece.wisc.edu*

Abstract

Power consumption is becoming an increasingly important constraint in the design of microprocessors. This paper examines the use of multiple constrained processors running at lowered voltage and frequency to perform a similar amount of work in less time and lower power than a uniprocessor. The paper also studies the effect of reducing cache and Branch Target Buffer (BTB) sizes for further reducing power consumption while still providing adequate performance. The best configuration requiring four processors reduced energy by 56%. Reducing cache and BTB provided a further 16% savings in energy while still finishing the workload in the same amount of time as the uniprocessor.

1 Introduction

Power consumption is a major concern for the design of portable devices. These devices are usually powered by a battery of a very limited capacity. Fortunately these devices often do not require vast computational capabilities, although this has been changing in the past few years. While designing portable devices it is often very easy to make trade-offs in favor of reducing power consumption even if they have a large impact on performance.

Recently, power consumption is becoming a major factor in the design of high-performance processors. Most of the traditional power reduction techniques focus mainly on the circuit and process level [11]. Many architectural techniques that are available for reducing power consumption are not used in these devices because the impact on performance is too great. In order for a technique to be used it must show a significant improvement in power consumption while having little or no impact on performance.

One of the best techniques for power reduction is to lower the supply voltage of the processor. Unfortunately this also increases the circuit delay, and therefore, the clock speed has to be lowered. Since a processor needs to be running at full speed only a fraction of the time, a variable voltage variable frequency processor can show a significant reduction in energy consumption without significantly increasing run time [3]. For these variable voltage variable frequency processors, much effort has been exerted to find efficient scheduling techniques [3, 6, 8, 9, 12, 13]. These techniques rely on the fact that often the standard processor is idle and hence the clock frequency and processor voltage can be reduced. However an alternative method to compensate for the reduction in clock frequency is to try to process data in parallel at lower frequency and hence lower voltage.

It is easy to conceive that 200 Intel XScale processors running at 500mW can outperform a Compaq Alpha 21364 at 100W on workloads with sufficient parallelism [10]. However to the best of our knowledge, no studies exist that test or validate such an assertion. Methods to exploit parallelism fall into two categories: Simultaneous Multi-Threaded (SMT) processor or a Chip-Multiprocessor(CMP). The use of a CMP and SMT for next generation wireless technology was compared with a uniprocessor [7]. In that study the power consumption for both the SMT and CMP processors were far lower than the uniprocessor. The SMT edged out the CMP, but the CMP's resources could have been optimized to reduce its power consumption even further.

In this paper we study a workload well suited for parallelization. We perform a systematic study to quantify the impacts of voltage reduction, clock frequency and the size of the caches and BTB on the energy consumed while meeting the application deadline requirements. Our study makes the following three important contributions: 1) we establish that an arbitrary multiprocessor based architecture will not save energy, 2) a constrained multiprocessor architecture can lead to energy savings without increasing execution time, and 3) constraining cache and BTB sizes can further reduce energy consumption in a multiprocessor setting while still meeting the performance requirement.

The rest of of the paper is organized as follows. In section 2, the appropriate first order power and delay equations are examined. In section 3, the workload and experiments are characterized. In section 4, the simulation environment is explained. In section 5, the experimental results are reported. In section 6, future work is stated. Finally section 7 concludes the paper.

2 Power and Delay Calculations

The first order approximation for power consumption in a processor is

$$P = \alpha C V_{dd}^2 f \quad (1)$$

Where α is the activity, C is the capacitance V_{dd} is the supply voltage and f is the frequency. Since power is proportional to V_{dd}^2, clearly a slight reduction in the supply voltage can lead to a significant reduction in the power consumption.

Unfortunately reducing the supply voltage also increases the delays of the circuit, thus decreasing the achievable clock frequency. The first order approximation of this relationship is:

$$t = K V_{dd} / (V_g - V_t)^h \quad (2)$$

Where K is proportionality constant, V_{dd} is the supply voltage, V_g is the voltage applied to the gates, and V_t is the threshold voltage. For this study we assume h = 2. If other technologies require h to be less than 2, implying a somewhat reduced impact of $V_g - V_t$ on the delay, the results will show an even larger saving in energy and energy-delay product than is reported here.

Instead of using equation (1), this paper uses simulation to determine power consumption. Equation (2) is used for computing the supply voltage for a given clock frequency. The default processor used in this experiment has a frequency of 600MHz, V_{dd} of 2.5V, and V_t of 0.67V. This is the default technology (0.35m) provided by the simulator [1]. These numbers are somewhat dated by today's standards, but are useful for illustrating the potential power savings. These values are used to solve for K for the given manufacturing process. Then for a desired frequency, the minimum V_{dd} can be calculated. For example, for a 300MHz processor the minimum V_{dd} is found to be 1.75V.

3 Workload and Experiment Design

The workload used in this paper is an MPEG-2 encoder. It was shown that in a proposed 3G wireless device, MPEG-2 encoding is by far the dominant CPU resource utilizer [7]. In fact, for their simulation purposes, the MPEG-2 encoding is spread across four processors while the rest of the wireless device's processing needs are easily handled by a fifth processor as illustrated in Figure 1.

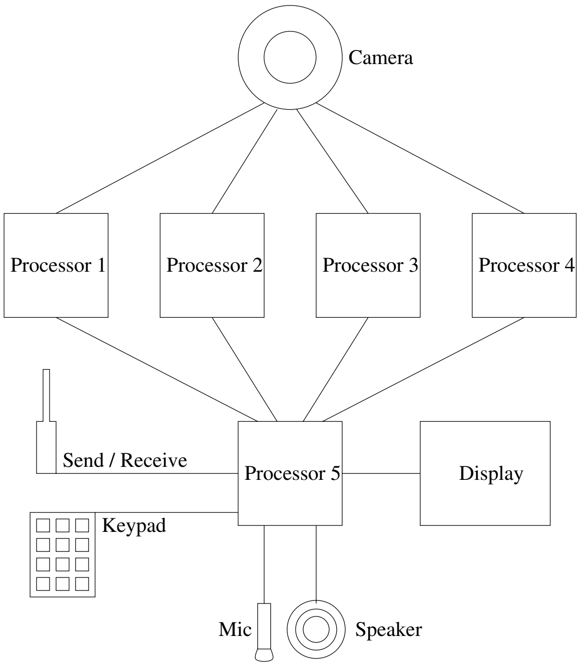

Figure 1. Workload Division.

The MPEG-2 encoder used in this paper is the public domain MPEG-2 encoder from the MPEG Software Simulation Group (MSSG) [5] . This encoder is used because of the availability of the source code. It was necessary to provide a PISA binary for the Wattch [1] simulator, so an open source encoder was necessary. The source code was compiled using the PISA cross compiler included with SimpleScalarV2.0 [2].

The raw video files used in simulation were individual YUV color space frames extracted from a sample uncompressed video clip of 300 frames in Quarter Common Intermediate Format (QCIF) format. In order to generate a workload for the multiprocessor, the parallel encoder is approximated by a quarter screen image extracted from the QCIF file [7]. These sub-images correspond to the upper-left, upper-right, lower-left and lower-right quarters of the original file as seen in Figure 2. The target bit rate for the uniprocessor case was 500kbps while each of the quarter screen compressors was 125kbps.

This encoding technique could be used in an actual 2-way communication device where the channel bandwidth is the most limiting factor. Each separate processor only encodes a quarter of the screen at approximately one quarter of the overall bit rate and all streams are combined and sent on the channel. Due to the subjective nature of gauging video quality, the resulting effects on visual quality were not considered.

Note that this workload does not show a direct relationship in the size of the image and the number of instructions required to finish the workload. The uniprocessor executes approximately 6 billion instructions while each of the multi-

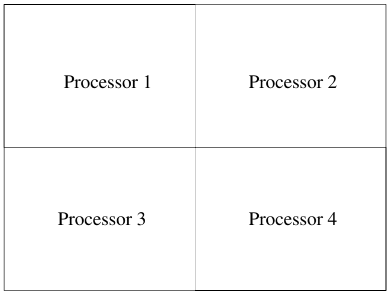

Figure 2. Workload Division.

processors executes about 2 billion instructions. Therefore, without a reduction in the supply voltage, the multiprocessor would require much more energy than the uniprocessor case since the total workload is much larger.

Two and eight processor configurations were briefly examined as discussed below. These configurations are unlikely to be superior to the four processors system for the technology examined in this paper. As a result these configurations are not studied in this paper.

A quick examination of a dual processor configuration shows that it would require at least two 300MHz processors to meet the deadline and using equation (2) we conclude that it must operate at 1.75V. Assuming perfect parallelization (no increase in number of clock cycles over uniprocessor), the results do show a significant reduction in energy consumption over the uniprocessor, but this ideal case will require a total energy of 227J which is 9Section 5.1.

Examination of an eight processor configuration also provides interesting results. An assumption is made that the total workload for the eight eight processor system is the same as that for a 4 processor system as shown in Section 5.1 . The requirements of the 8 processor system are a minimum of 94MHz processors running at 1.09V. While each of these processors require at most 4.5W, the total energy required for eight processors to complete the workload at this power level is higher than both the two and four processor configurations.

4 Simulator

The simulator used in this experiment is a modified version of the Wattch [1] simulator, a power estimator based on the popular SimpleScalar [2] simulator. Wattch is based on the SimpleScalar sim-outorder simulator, but it adds a mechanism to account for power consumption by the sub-units of the processor. The Wattch simulator is about 30% slower than SimpleScalar but provides a more accurate account of power consumption than the first order equation can. The simulation results show that while the absolute numbers provided by the Wattch simulator are not very accurate (about 10% off), they do show a strong correlation to the general trends in the processor's power consumption [1].

Since the workload is divided into completely independent sub-components, the multiprocessor is approximated by four separate instances of the Wattch simulator. This assumption means that each processor has its own individual cache and memory system and that no communication between the processors is needed. For the division of the workload that this paper uses, this is a possible implementation. The approximation technique may not accurately reflect energy consumption in a multiprocessor environment where significant cache coherence traffic is occurring.

The only modification that was needed to the Wattch simulator was a change to account for the reduced voltage and frequency. These are process related constants stated in the source code header files and were varied as described in the following section.

5 Results

The comparison of the voltage and frequency constrained multiprocessors is presented in Section 5.1. The processor is further constrained by reducing the cache sizes in Section 5.2. The BTB sizes are reduced in Section 5.3. Finally in Section 5.4 a combination of the cache and BTB size reduction is used to provide the best energy savings result. As mentioned in Section 3, the only configuration that was studied in detail was the four processor case.

5.1 Speed and Voltage Reduction

Simply applying a multiprocessor to a workload without adding some form of constraint will not result in energy savings. The workload for the uniprocessor is about 6 billion instructions while each multiprocessor executes about 2 billion instructions. Clearly there is a 33% increase in the total number of instructions executed. Figure 3 shows that a four processor multiprocessor running at 300MHz and 2.5V when compared with a uniprocessor running at 600MHz and 2.5V, consumes 18% more energy. This is due to the increase in the total number of instructions. However the multiprocessor will complete the workload in 33% less time.

The large improvement in power consumption comes from reducing the processor supply voltage. Figure 3 shows the total energy required to complete the workload. Figure 4 shows the maximum power, average power and execution time relative to the 600MHz-2.5V processor.

A conservative estimate of the supply voltage for a 300MHz processor on the same technology as a 600MHz 2.5V processor is 2.0V. With these settings there is a 23% decrease in the energy required to complete the full workload. There is also a significant decrease in runtime in using the multiprocessor setup compared to the uniprocessor.

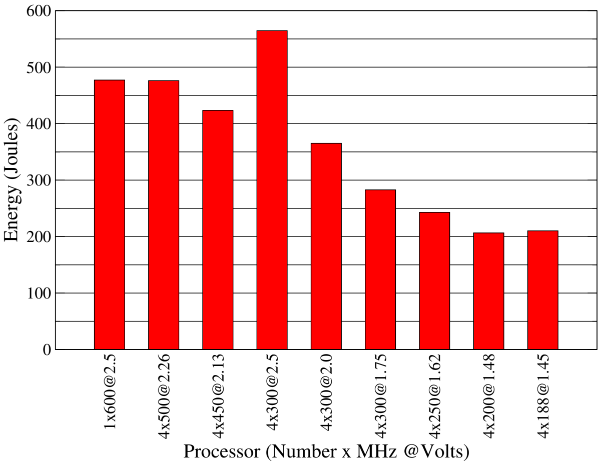

Figure 3. Energy Consumption.

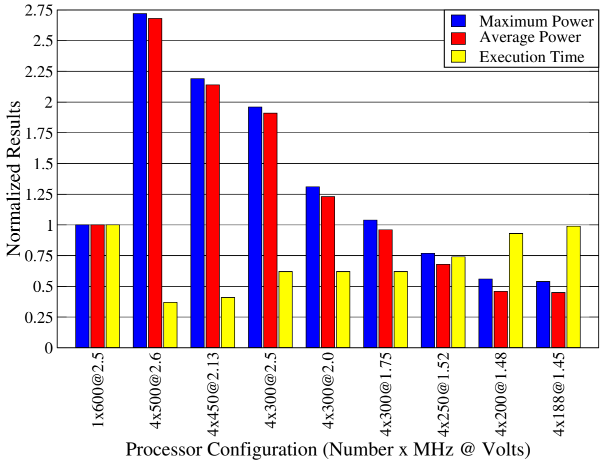

Figure 4. Normalized Power and Execution Time.

Direct calculation of the of the lowest V_{dd} shows that the supply voltage can be reduced to 1.75V. The simulation results in Figure 3 show that there is a 41% decrease in the energy required to complete the workload.

The simulation results from the quarter screen simulation are used to find the minimum frequency that the multiprocessor can run at in order to complete the workload in the same amount of time. The calculation shows that 188MHz is the desired frequency. The calculated supply voltage is 1.45V. The simulation results show a 56% decrease in the energy consumption compared to the uniprocessor.

Simulations were run at a variety of other frequencies. The simulation results show that a 200MHz processor configuration actually consumes less energy than the 188MHz configuration. These processors have average power consumption of 8.42W and 8.06W respectively. The slightly longer execution time of the 188MHz processor causes its energy consumption to be higher.

From the graphs in Figure 3 and Figure 4 it can be seen that there is a very wide range of frequencies that are able to complete the workload in less time and with less energy than the uniprocessor. For this workload, there is clearly an energy advantage with a multiprocessor from around 500MHz all the way down to 188MHz. Depending on the requirements of the application, the appropriate energy delay trade off can be made.

Since the multiprocessor system can complete the workload in less time than the uniprocessor can, additional techniques can be used to further reduce the power consumption of the multiprocessor system. Two areas explored in this paper are reduction of the cache and BTB sizes to reduce energy while still completing the workload ahead of the uniprocessor.

5.2 Cache Size Reduction

It has been shown that the largest consumers of power in a processor are the clock and on chip memories (i.e. caches, BTBs, registers, etc.) [4]. In fact it is estimated that between 50-80% of the power consumption of a microprocessor occurs in the on-chip memories and clock circuitry. As reported by Wattch, the largest power consumers on the simulated architecture are the caches (16.5% of total power) and the branch target buffer (6.58% of total power). Since the multiprocessors are able to complete the workload in a shorter time period than the uniprocessor, the multiprocessor core can be modified to try to improve power consumption. The first area to target is the caches. The baseline cache arrangement is a 16kB direct mapped L1 instruction cache, a 16kB 4-way set associative L1 data cache, and a 256kB 4-way set associative unified L2 cache. Processors with 1/2, 1/4, 1/8, and 1/16 size caches were simulated.

As can be seen in Figure 5 the energy consumption is minimal at 1/4 the cache size. The results for the 600MHz uniprocessor are not shown but follow the same trend. Examining the simulation results shows that beyond a quarter cache size there is a large jump in the number clock cycles required to complete the workload.

The average power consumption savings of the 200MHz processor at 1/4 sizes is 13.9%. Unfortunately this comes at the price of a 5% increase in execution time. Thus the resulting savings in energy is about 9.9%. The energy savings for the higher frequency processors is slightly less.

In all of these cases the multiprocessor configurations complete the workload in less time and with less energy than the uniprocessor. From the simulation results, the optimal cache size for this workload would appear to 1/4 size or 4kB

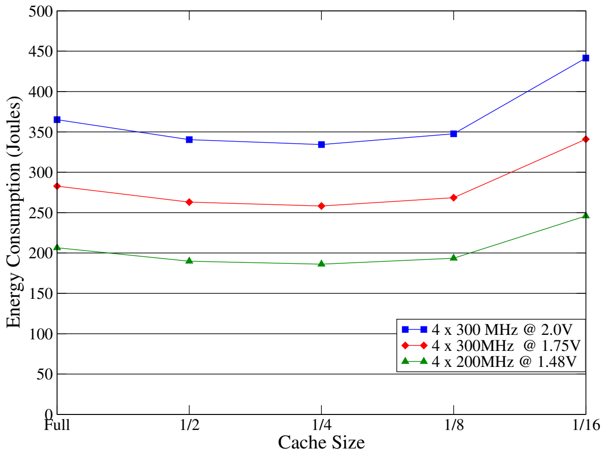

Figure 5. Cache Size Results.

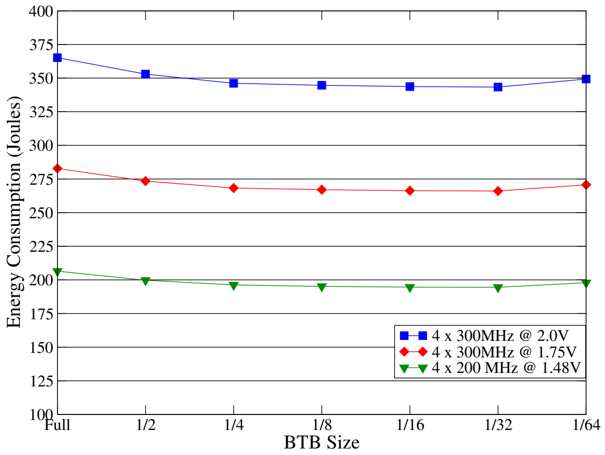

Figure 6. BTB Sizing Results.

L1 caches and a 64kB L2 cache.

5.3 BTB Size Reduction

The next target for energy reduction is the BTB. The baseline BTB has 2048 entries. Simulations were run with 1/2, 1/4, 1/8, 1/16, 1/32, and 1/64 sized BTB. Since the BTB accounts for less than 7% of the total processor power, it's impact on power consumption is not expected to be as great as that of the caches.

Figure 6 illustrates the energy savings gained by reducing the size of the BTB. The results for the 600MHz processor are not shown, but follow the same trend. There is a 5.8% reduction in the energy consumption by reducing the size of the BTB by a factor of 32. More importantly the BTB's power consumption drops from 6.5-7% to 0.8-1.2% of total processor power. At 1/32 size, the BTB only consumes around 1% of the total system power. Further reductions in the size (including removal) of the BTB will not yield much further reduction in the energy consumption. In fact reducing the BTB from 1/32 to 1/64 of the original size only drops the power consumption by less than 0.1%, but the number of clock cycles needed increases by almost 2% thus requiring more energy to complete the task.

5.4 BTB and Cache Size Reduction

The BTB and cache size reduction techniques are combined to examine the effect on energy consumption. Table 1 shows that a further 16% improvement in energy consumption is seen with only a 5% increase in the number of cycles. Note that the multiprocessor with reduced cache and BTB sizes requires only 196MHz processors to complete the workload ahead of the baseline uniprocessor. Figure 7 shows the energy, execution time, and energy delay product for the 600MHz uniprocessor, cache constrained, BTB constrained, and cache and BTB constrained multiprocessors at 200MHz. For ease of display in one graph, all results are normalized with respect to the 600MHz uniprocessor. The best energy-delay product is seen with the combined cache and BTB constrained multiprocessor.

6 Future Work

In this paper we investigated the potential energy savings of implementing a specific, though common workload using a multiprocessor. Examination of other workloads would help to validate the use of multiprocessing for power savings. Some workloads will not parallelize as well as this workload making the trade-off evaluation more complex. Future work will involve developing an analytical model that estimates the minimum amount of parallelism that is needed for a multiprocessor to save energy.

Further developments of the simulation environment will include a shared bus multiprocessor model along with an actual parallel version of the MPEG-2 encoder. Additional processor resource optimization will be explored in an attempt to further reduce energy consumption.

Another future project is to develop an analytical model to calculate near optimal cache and BTB sizes for a particular workload. This would reduce the total simulation time needed to find a power efficient architecture for a particular workload.

7 Conclusion

Multiprocessing is a promising technique for saving energy on certain workloads. A 4-way multiprocessor can have

Processor	Baseline Processor		Reduced Cache and BTB		Percentage Change	
Configuration	Energy	Clock Cycles	Energy	Clock Cycles	Energy	Clock Cycles
600MHz @ 2.5V	477.14	3,950,990,864	407.04	4,115,259,420	-14.69%	4.16%
4 x 500MHz @ 2.26V	476.17	1,226,061,571	408.23	1,287,480,059	-14.27%	5.01%
4 x 450MHz @ 2.13V	423.33	1,226,061,571	362.24	1,287,480,059	-14.43%	5.01%
4 x 300MHz @ 1.75V	282.83	1,226,061,848	240.69	1,287,480,059	-14.90%	5.01%
4 x 250MHz @ 1.62V	242.72	1,226,061,848	205.72	1,287,480,059	-15.25%	5.01%
4 x 200MHz @ 1.48V	206.52	1,226,061,848	173.58	1,287,480,059	-15.95%	5.01%
4 x 188MHz @ 1.45V	210.29	1,226,061,848	176.62	1,287,480,059	-16.01%	5.01%

Table 1. Combined Cache and BTB Results.

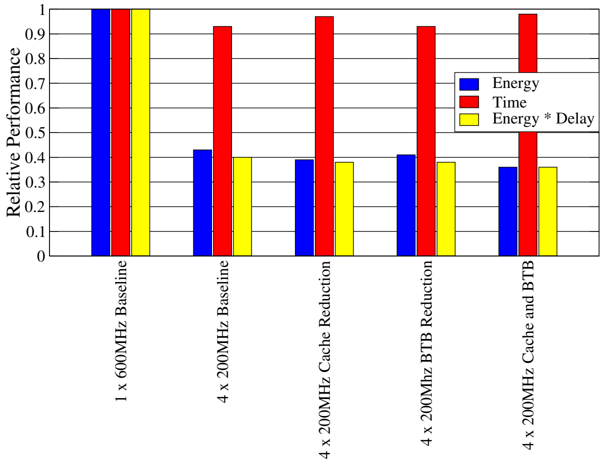

Figure 7. Comparison of baseline uniprocessor and cache and BTB constrained multiprocessors.

up to an 56% energy reduction as compared to a uniprocessor when running a CPU intensive DSP type application. This energy advantage is observed even when the multiprocessor has to execute 33% more instructions.

Along with using multiprocessing, proper sizing of the largest power consuming blocks also shows a noticeable energy saving. The two most likely candidates for size optimization are the caches and the BTB. Optimizing the cache and BTB sizes can yield a further 16% improvement in energy consumption while still out-performing a uniprocessor.

References

[1] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A Framework for Architectural-Level Power Analysis and Optimization,” in *Proc. Int. Sym. on Computer Architecture*, Jun. 2000, pp. 83–94.

[2] D. Burger and T. Austin, “The SimpleScalar Tool Set, Version 2.0,” in *University of Wisconsim-Madison Computer Sciences Department Technical Report 1342*, Jun. 1997.

[3] J. Chang and M. Pedram, “Energy Minimization Using Multiple Supply Voltages,” in *Proc. Int. Sym. on Low Power Electronics and Design*, Aug. 1996, pp. 157–162.

[4] R. Gonzalez and M. Horowitz, “Energy Dissipation In General Purpose Microprocessors,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1277–1284, Sep. 1996.

[5] MPEG Software Simulation Group, “MPEG-2 Encoder / Decoder,” in *http://www.mpeg.org/MSSG/*.

[6] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. Srivastava, “Power Optimization of Variable Voltage Core-Based Systems,” in *Proc. Conf. on Design Automation Conference*, Jun. 1998, pp. 176–181.

[7] S. Kaxiras, A. Berenbaum, and G. Narlikar, “Simultaneous Multithreaded DSPs: Scaling from High Performance to Low Power,” in *Bell Laboratories Technical Memorandum 10009639-001024-06TM*.

[8] Y. Lin, C. Hwang, and A. Wu, “Scheduling Techniques for Variable Voltage Low Power Designs,” *ACM Trans. on Design Automation of Electronic Systems*, vol. 2, no. 2, pp. 81–97, Apr. 1997.

[9] J. Lorch and A. Smith, “Reducing Power Consumption by Improving Processor Time Management in a Single-User Operating System,” in *Proc. Int. Conf. on Mobile Computing and Networking*, Nov. 1996, pp. 143–154.

[10] T. Mudge, “Power: A First Class Design Constraint for Future Architectures,” in *Proc. Int. Conf. on High Performance Computing*, Dec. 2000, pp. 215–224.

[11] E. Musoll, “Predicting the Usefulness of a Block Result: a Micro-Architectural Technique for High-Performance Low-Power Processors,” in *Proc. Int. Sym. on Microarchitecture*, Nov. 1999, pp. 238–247.

[12] T. Okuma, T. Ishihara, and H. Yasuura, “Real-Time Task Scheduling For A Variable Voltage Processor,” in *Proc. Int. Sym. on System Synthesis*, Nov. 1999, pp. 24–29.

[13] S. Raje and M. Sarrafzadeh, “Variable Voltage Scheduling,” in *Proc. Int. Sym. on Low Power Electronics and Design*, Apr. 1995, pp. 9–14.