Simultaneous Voltage Scaling and Voltage Domain Partitioning for Chip Multi-Processors

Weiping Liao and Lei He
Electrical Engineering Department
University of California, Los Angeles 90095
{wliao, lhe}@ee.ucla.edu

Abstract

In this paper, we study the power-optimal Voltage Scaling and Voltage domain partitioning (VSVP) problem for Chip Multi-Processing (CMP) architecture, subject to constraints on performance and the area overhead of on-chip dc-dc converters. To efficiently explore the large multi-dimensional solution space, we develop an analytical performance model for CMP considering on-chip communication contentions and heterogeneous V_{dd} for processor cores. Compared to cycle-accurate simulations, our analytical model has a high fidelity and an average error of 4%. Considering a CMP with voltage scaling capability and Quality-of-Service (QoS) guarantee, we show that with the consideration of on-chip dc-dc converters, the optimal voltage domain number may not be the maximum domain number available. Such a result clearly shows that the assumption in existing low power task and voltage scheduling methods for multi-processor systems that a system always contains the maximum number of voltage domains (one processor core per domain) may not lead to optimal power for CMP with on-chip dc-dc converters. We also show that multiple voltage domains can effectively reduce both dynamic power and leakage power by 12% and 16%, respectively. Furthermore, we show that on-chip dc-dc converters can consume up to 16.06% total power in our CMPs, indicating that the power overhead by dc-dc converters may become a severe problem for CMPs.

This paper is partially supported by NSF CAREER award CCR-0401682, SRC grant 1116, a UC MICRO grant sponsored by Mindspeed, and a Faculty Partner Award by IBM. We used computers donated by Intel and SUN Microsystems. Address comments to lhe@ee.ucla.edu.

Simultaneous Voltage Scaling and Voltage Domain Partitioning for Chip Multi-Processors

I. Introduction

Power consumption has gained a growing importance for modern integrated circuits and systems. A number of studies have considered voltage scaling for power reduction. [1] proposed task scheduling and a voltage allocation algorithm assuming continuously variable voltage. Considering discrete variable voltage scaling, [2] studied the static voltage scaling and proves that voltage scaling with at most two voltages for each single task minimizes the energy consumption under any time constraint when only a number of discretely variable voltages are available. [3] proposed a heuristic voltage scheduling algorithm considering transition overhead and voltage level discretization. [4] proposed a voltage allocation technique to achieve optimal processor energy consumption, considering multiple discrete supply voltages and arbitrary task deadline constraints. [2]- [4] all focused on the impact of discrete voltage levels on voltage scheduling algorithms, but do not consider the cost to generate these voltage levels. Furthermore, [2] - [4] all focused on uniprocessor systems and do not consider leakage power.

Over the past several years, performance improvement for the traditional monolithic uniprocessor architecture by increasing clock rate and instruction per cycle (IPC) has resulted in diminishing returns [5]. Meanwhile, Chip Multi-Processing (CMP) architecture has become increasingly attractive as it can execute multiple tasks on different on-chip processor cores simultaneously, and therefore effectively improve the system performance [6]. As we integrate multiple processors into one single chip, power consumption becomes a more important problem because of the increased integration level and larger power density. Considering traditional multiprocessor systems, [7], [8] targeted voltage scaling of hard real-time task scheduling and [9] studied power minimization with QoS guarantee for soft real-time systems. [10] considered leakage power in voltage scaling for multiprocessor system-on-chip. [7]- [10] all focused on task scheduling and resource allocation, and assumed that each processor may have a customized supply voltage (V_{dd}), i.e., an individual voltage domain containing a voltage supply module for variable V_{dd} is needed for each processor. However, the impacts of area overhead and power efficiency of the voltage supply modules are ignored in [7]- [10], which is valid only with off-chip voltage supply modules.

In this paper we consider on-chip voltage supply modules by forms of dc-dc converters for the CMP variable-voltage supply because on-chip dc-dc converters can provide higher power efficiency and have become the trend for future variable-voltage System-on-Chip designs. With on-chip dc-dc converters, the impacts of area overhead and power efficiency of dc-dc converters can no longer be ignored. Considering these impacts, we study in this paper the power-optimal Voltage Scaling and Voltage domain Partitioning (VSVP) problem for CMP. Our primary contributions include the followings:

- We formulate the power-optimal Voltage Scaling and Voltage domain Partitioning (VSVP) problem for CMP considering the impact of on-chip dc-dc converters. Subject to the constraints on performance and area overhead by dc-dc converters, the VSVP problem decides the voltage domain partition and the voltage levels for each voltage domain in the CMP in order to minimize total system power as the sum of power consumed by all processor cores and all dc-dc converters.
- In order to efficiently explore the VSVP solution space, we develop an analytical performance model for CMP considering the on-chip communication overhead and heterogeneous V_{dd} for processor cores. Compared to cycle-accurate simulation, our model is extremely efficient with high fidelity and an average error of 4%. We explore the solution space of the power-optimal VSVP problem by simulated annealing leveraging our analytical performance model.
- Our experiments show that with the consideration of on-chip dc-dc converters, the optimal number of voltage domains may not be the maximum domain number available. Such a result clearly shows that the assumption in existing low power task and voltage scheduling methods for multi-processor systems that a system always contains maximum number of possible voltage domains (one processor core per domain) may not lead to optimal power for CMP with on-chip dc-dc converters. We also show that multiple voltage domains can effectively reduce both dynamic power and leakage power by 12% and 16%, respectively. Furthermore, we show that on-chip dc-dc converters can consume up to 16.06% total power in our CMP, indicating that the power overhead by dc-dc converters may become a severe problem for CMPs.

To the best of our knowledge, this study is the first in-depth study on voltage scaling and voltage domain partitioning for CMP considering the impact of on-chip dc-dc converters.

The rest of this paper is organized as follows. Section II presents system architecture and problem formulation. Section III introduces the CMP performance model. Section IV discusses our models and exploration methodologies. Section V presents the experimental results. We conclude in Section VI.

II. System Overview and Problem Formulation

A. System Architecture

The overall structure of our CMP is shown in Figure 1. There are multiple processor cores on the same chip. We call each processor core a Processing Element (PE). Each PE is a fully functional microprocessor with local caches. A memory

controller is in charge of off-chip memory accesses. It receives the memory requests from PEs, performs necessary read or write operations to off-chip main memory, and returns the data of memory requests to PEs. The PEs and the memory controller communicate with each other via an on-chip communication mechanism by shared buses. Furthermore, there may be multiple on-chip voltage domains. Each voltage domain includes one or multiple PEs, and one on-chip voltage supply module in the form of a dc-dc converter, which provides the variable-voltage supply to all PEs in the same domain. Although the V_{dd} provided by a dc-dc converter can be changed from time to time, at any moment all PEs in the same voltage domain always have the same V_{dd}.

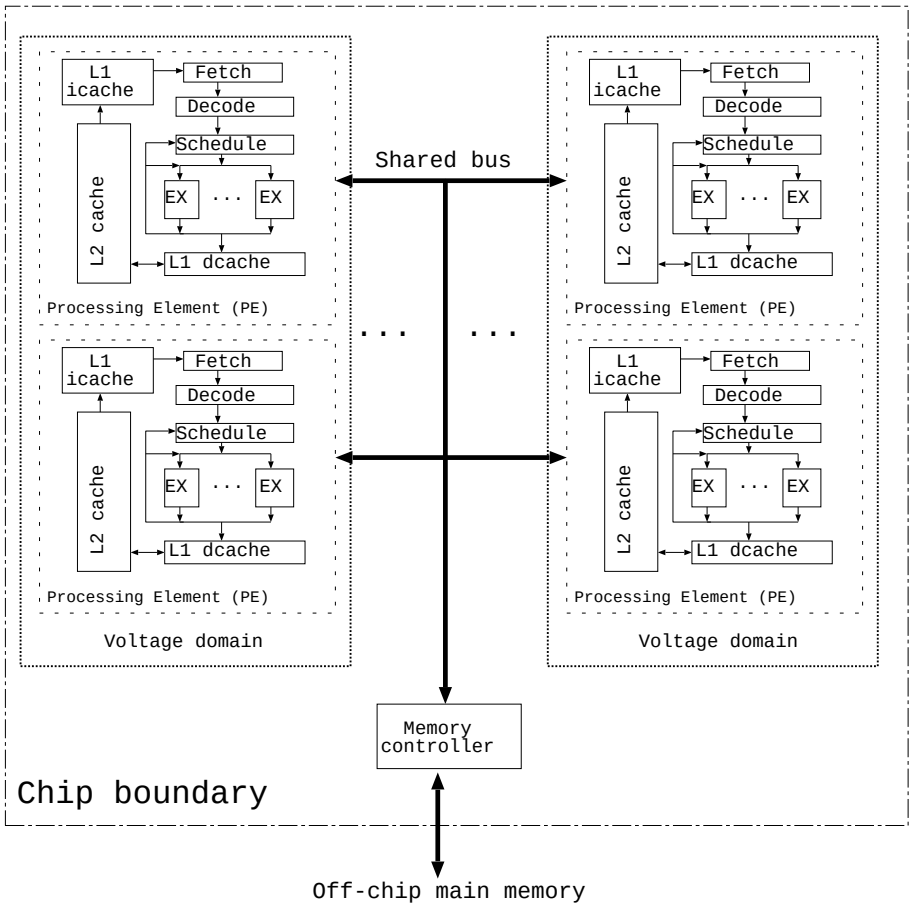


Fig. 1. System architecture.

B. On-Chip DC-DC Converters and Voltage Domain Partitioning

We target CMPs with voltage scaling capability where V_{dd} of PEs can be adjusted according to different workloads. The dc-dc converters are in charge of adjusting V_{dd} for PEs in every domain. In this study we focus on on-chip integrated dc-dc converters. Traditional off-chip dc-dc converter designs lead to significant parasitic impedances of the interconnects between the dc-dc converter and the processors. Such impedances consume additional energy and reduce the power efficiency of off-chip dc-dc converters [11]. Integrating a dc-dc converter with a microprocessor can reduce the parasitic impedances by reducing the interconnect length and utilizing advanced fabrication technologies with low parasitic impedances [12]. Therefore, on-chip dc-dc converters have higher power efficiency compared to off-chip ones. In our study, the total system power is the sum of power consumed by all PEs, and the power consumed by all dc-dc converters. The total system power P_{sys} can be calculated as:

$$P_{sys} = P_{PE} + P_{DC} = \frac{P_{PE}}{efficiency} \tag{1}$$

where P_{PE} is the total power consumed by all PEs, P_{DC} is the total power consumed by all dc-dc converters, and $efficiency$ is the power efficiency of dc-dc converters [1].

We adopt the buck converter in [12] as our on-chip dc-dc converter. As shown in Figure 2, the pulse width modulator controls the on and off of power transistors P1 and N1, and generate a waveform with a certain duty cycle at the output node.

[1]We assume all domains have the identical dc-dc converter design. So the power efficiency of all dc-dc converters is the same.

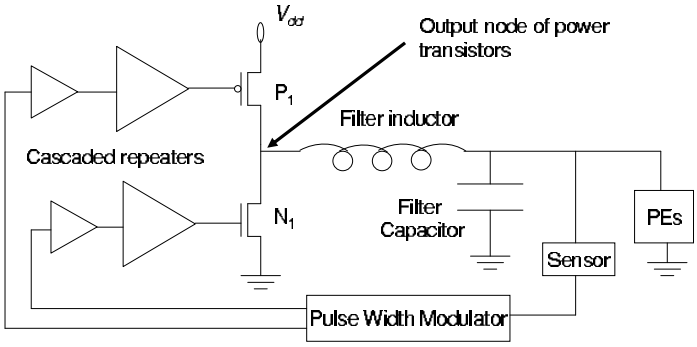


Fig. 2. Schematic of the on-chip dc-dc converters we adopt from [12].

Such a waveform is transferred to a stable V_{dd} level by a filter inductor and a filter capacitor. Output voltage level is adjusted by changing the duty cycle. The sensor and pulse width modulator provide feedback control to adjust the V_{dd} level. They can be built off-chip, so the area of a dc-dc converter is dominated by the filter capacitor. The power efficiency of a dc-dc converter increases monotonically with respect to the area of the dc-dc converter.

The integration of on-chip dc-dc converters does come with a negative impact: area overhead. Given a fixed area overhead for total dc-dc converters, there is a trade-off between different voltage domain partitioning choices: On one hand, with more voltage domains we gain larger flexibility for task-level voltage scaling in CMP because more PEs can be scheduled with independent V_{dd}. Such flexibility enables more potential of power reduction. However, more voltage domains require more dc-dc converters, which result in a smaller area for each dc-dc converter under the fixed total area overhead, and therefore reduce the power efficiency of each dc-dc converter.

On the other hand, with fewer voltage domains we have smaller flexibility for the task-level voltage scaling in CMP. However, in this case each dc-dc converter can have a larger area and therefore higher power efficiency. Such an increase in power efficiency may be able to compensate the lack of voltage scaling flexibility. Above all, the trade-off between the voltage domain partitioning schemes is non-trivial and requires detailed exploration.

C. Problem Formulation

We formulate our co-optimization problem as follows:

Formulation 1: **Simultaneous Voltage Scaling and Voltage domain Partitioning (VSVP) problem**: Given a CMP with a number of available PEs, system throughput requirement, and total area overhead constraint, find the voltage domain partition and V_{dd} for each domain to minimize total system power consumption, while subject to (1) total system throughput is no less than the given throughput requirement, and (2) total area of dc-dc converter is within the area overhead constraint. □

In our CMPs, each PE has a supply voltage V_{dd} as well as the clock frequency F and power consumption P associated with that V_{dd}. We choose instruction throughput as the metric for performance of each PE, which is equal to the product of F and Instruction-per-Cycle (IPC). The total system throughput of a CMP is the sum of throughput of all PEs.

There are three important aspects of our VSVP problem: the first one is the satisfaction of the performance requirement, the second one is the PE power model and power efficiency model for dc-dc converter, and the third one is the exploration for a VSVP solution. We will address them in Sections III, IV-A, and IV-B, respectively.

Note our VSVP problem is for static voltage scaling as it should be solved during design time. Although the VSVP solution can be further extended to facilitate runtime power reduction techniques such as Dynamic Voltage Scheduling (DVS), it is beyond the scope of this chapter and will be studied in our future work.

III. Performance Estimation

The CMP performance is affected by the contention of the shared buses, which depends on the interaction between memory access rates and therefore the V_{dd} setting of all PEs. Hence, the CMP performance is not a simple sum of performance for all PEs as individual uniprocessors, but depends on the V_{dd} setting for all PEs. For a CMP with a total n PEs and M possible V_{dd} for each PE, the number of possible solutions to the VSVP problem is on the order of M^n. Given such a large design space, it is extremely inefficient, if not impractical, to obtain performance by cycle-accurate simulation of CMP.

[13] developed an analytical performance model based on the M/D/1 queue model for CPU utilization rate. As we will discuss in this section, the M/D/1 queue model assumes an infinite number of request sources and does not fit the CMP scenario well. More importantly, the performance model in [13] does not consider and cannot be readily extended to consider multiple heterogeneous clock frequencies in CMP, which is vital in our study. To efficiently explore the VSVP solution space, we develop an analytical performance model to estimate the throughput of a CMP, considering bus contentions and heterogeneous clock frequencies on different PEs.

A. Model for Memory Access

Based on the finite source queuing theory, we develop a new analytical model for off-chip memory accesses. Our CMP has one memory controller and two shared buses: one for memory requests and the other for memory responses. There is only one shared memory module off-chip, but the memory module can accommodate multiple requests at the same time due to internal pipelining and subbanking.

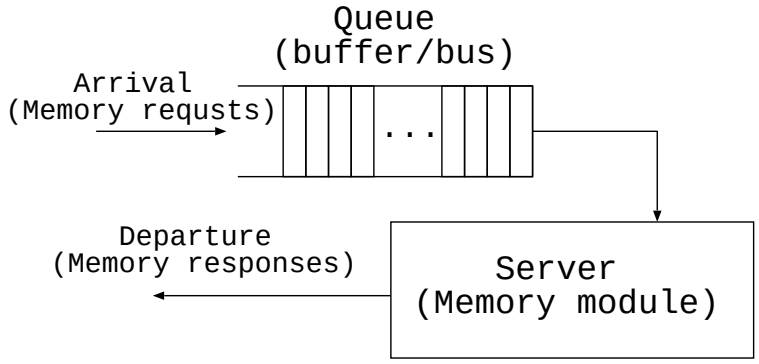


Fig. 3. Queuing model for bus and memory structures. The queue includes the bus and the memory request buffer inside the memory controller. The server models the memory module.

The bus, memory controller, and memory structures can be modeled as a queuing system as shown in Figure 3, where the bus and memory request buffer in the memory controller are modeled as the queue and the off-chip main memory module is modeled as the server. The overall system latency is the average memory latency l_m observed by each PE [2]. For in-order microprocessors such as XScale [14], the pipeline will stall whenever there is a main memory access due to a cache miss. For high-performance out-of-order microprocessors running at a few GHz clock frequencies such as the Intel Pentium IV [15], the average memory latency is on the order of hundreds of cycles, and cannot be covered by out-of-order execution. Above all, we assume that any PE will eventually halt under cache misses before the data comes back from the main memory, and no more memory access can be generated from that PE. For this reason, we model the queueing system as a finite source queue such as that in the machine repair problem [16] instead of a general M/D/1 queue where an infinite number of sources are assumed. Furthermore, if the main memory can handle c memory accesses simultaneously (due to internal pipelining and subbanking), we apply a total of c servers in our queueing system.

First, we assume all PEs have the same clock frequency F and run the same benchmark (such constraints will be removed later). Under such an assumption, all PEs have the same average memory access rate $\lambda = (\frac{M}{C_p} \cdot F)$, where M is the total number of main memory accesses and C_p is the number of cycles spent on computation and cache accesses. The average response rate of one server is $\mu = 1/T_M$, where T_M is the main memory access latency. For a CMP with n PEs, assuming that all PEs run identical benchmarks (such an assumption will be removed later), the utilization rate of the queueing system is $r = \lambda/\mu$ [16], and the probability that exactly i memory requests reside in the system is given in [16] as

$$p_i = \begin{cases} \frac{n!/(n-i)!}{i!} r^i p_0 & (1 \le i < c) \\ \frac{n!/(n-i)!}{c^{i-c} c!} r^i p_0 & (c \le i \le n) \end{cases}$$

where p_0 is the probability of the case when no memory request is in the queueing system, given as (2):

$$p_0 = \frac{1}{1 + \sum_{i=1}^{c-1} \frac{n!/(n-i)!}{i!} r^i + \sum_{i=c}^{n} \frac{n!/(n-i)!}{c^{i-c} c!} r^i} \tag{2}$$

The average number of memory requests in the system L is given by $\sum_{k=1}^{n} (k * p_k)$. According to Little's formula [16], the average number of memory requests in the system is equal to the total system latency (i.e., the average memory access latency

[2]We only consider voltage scaling to PEs, but do not scale voltage of bus, memory controller, and memory modules. Therefore, l_m is independent of each PE's clock frequency.

l_m) times the average rate memory requests arrive when the system is in equilibrium. The latter is equal to $\lambda * (n - L)$. Therefore, l_m can be calculated as

$$l_m = \frac{L}{\lambda * (n - L)} \quad (3)$$

After the derivation, we remove the restriction that all PEs run identical benchmarks. Therefore, each PE has its individual access rate as $\lambda_i = (\frac{M_i}{C_p^i} \cdot F)$ for the i^{th} PE. When we consider heterogeneous clock frequencies on different PEs, the access rate of the i^{th} PE becomes $\lambda_i = (\frac{M_i}{C_p^i} \cdot F_i)$, where F_i is the PE's clock frequency. In these cases, since the memory access rates of each PE are different, we make two changes to our previous model: (1) we use the maximum rate $\lambda_{max} = max_{i=1}^{n}(\lambda_i)$ to replace access ratio λ; and (2) we use the *equivalent total PE number* $n_{eq} = \frac{\sum_{i=1}^{n} \lambda_i}{\lambda_{max}}$ to replace the total PE number n. After that, l_m can be calculated following the same approach as discussed above. Note that the n_{eq} may be different from the total number of PEs n. In fact, l_m is only determined by the contentions of memory requests and the latency of memory module. Such contentions are independent of the sources of the requests, and should be quantified by λ_{max} and n_{eq}, but not n.

B. Model for Throughput

We target the multi-programming environment where different benchmarks have separate address spaces and there is no direct communication between PEs. Therefore, the parameters such as I, t_p, and M are instruction level characteristics of each benchmark, and independent of either n or clock frequencies of other PEs. Once we obtain the l_m, the throughput of the PE can be easily calculated as (4) for PEs with in-order execution:

$$throughput = \frac{I}{t_p + M * l_m} \quad (4)$$

where I, t_p and M can be obtained by offline profiling. For out-of-order SuperScalar PEs, l_m can be fed to a first-order analytical performance model such as [17] to calculate the PE's throughput. Finally, the total CMP system throughput is the sum of throughput of all PEs.

C. Model Verification

We verify our model by cycle-accurate simulation. We use the SimpleScalar/ARM [18] toolset for ARM architecture [19] as the PE simulator, and develop additional programs to simulate the bus, memory controller, and memory module. We configure each PE simulator similar to the StrongARM microprocessor [20] as an in-order, single issue, RISC microprocessor supporting ARM instruction set. Each PE also has two separate 4KB direct-mapped caches with 32-byte linesize for instruction and data, respectively. Our memory module has a latency of 40 ns.

Benchmark	*crc*	*md5*	*nat*	*route*	*tl*	*url*
A	14772	21656	5185	616	197	810175
M	59	691	300	12	7	14851
C_p	27525	40589	9149	1329	411	1536633

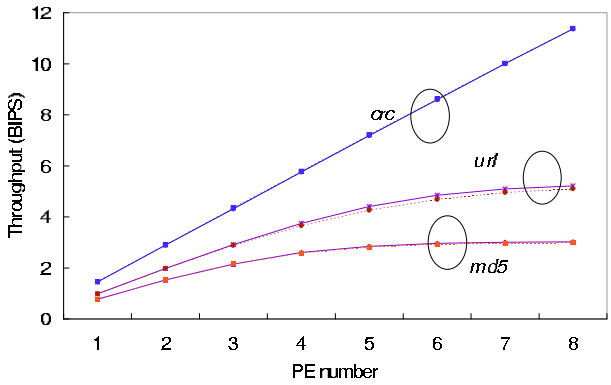
TABLE I

THE INSTRUCTION COUNT A, NUMBER OF MEMORY ACCESSES M AND CYCLES SPENT ON COMPUTATION AND CACHE ACCESSES C_p DURING THE PROCESS OF ONE PACKET BY ONE PE.

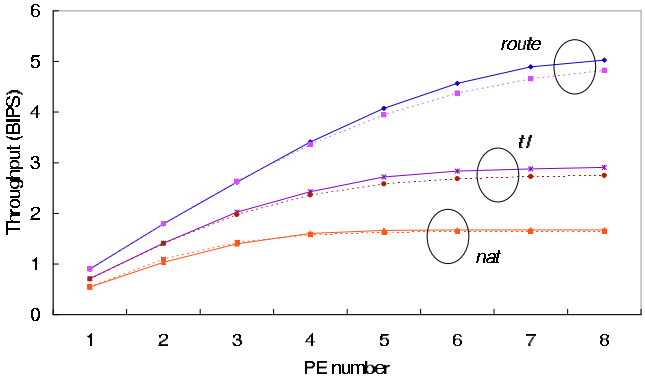
We choose NetBench suite [21] as our benchmarks and use packet traces available in the public domain from [22]. For each benchmark on every PE simulator, we always fastforward instructions to process 500 packets, and then collect simulation results for instructions to process another 500 packets. Table I lists the profiles of all benchmarks we choose from the NetBench suite. As the packet processing procedures are identical for different packets, our profile is based on the information collected when one packet is processed. Although we assume statistics for each benchmark binary can be gathered offline and fed into our controller for system optimization, our approach is easily extensible to mixed-application and dynamic-variation within a single application.We can either store the profile information in a fixed table for each connection/packet type, or capture profiles at runtime with periodic profiling for each connection/packet type.

Since we target StrongARM microprocessors with in-order execution, we use Equation (4) to calculate the throughput. We first assume all PEs have a uniform clock frequency at 3.09GHz, and run identical benchmarks. Figure 4 presents the comparison between simulation and our analytical model. We choose the PE number from 1 to 8. From Figure 4 it is easy to see that our analytical model correctly tracks changes of throughput for different PEs with a small error bound.

We further modify the PE simulator to handle different clock frequencies and different benchmarks on different PEs and verify our models. Table III compares the total throughput for a CMP with six PEs, where the clock frequencies and benchmarks on PEs are listed in Table II. The difference between our analytical model and cycle-accurate simulations is only 1.70%. Overall,



(A) Homogeneous benchmarks *crc*, *md5*, and *url*.



(B) Homogeneous benchmarks *nat*, *route*, and *tl*.

Fig. 4. The comparison between cycle-accurate simulation and our analytical performance model for total throughput among all PEs for homogeneous benchmarks. The solid lines are from simulation and the dotted lines are from our performance model.

PE	1	2	3	4	5	6
F (GHz)	3.09	2.75	2.42	2.11	1.81	1.53
Benchmark	crc	md5	nat	route	tl	url

TABLE II

CLOCK FREQUENCIES AND BENCHMARKS FOR A SIX-PE CMP.

Simualtion	3.20
Anlytical model	3.25
Error	1.70%

TABLE III

TOTAL SYSTEM THROUGHPUT FOR THE SIX-PE CMP IN TABEL II UNDER CYCLE-ACCURATE SIMULATIONS AND THE ESTIMATION FROM OUR ANALYTICAL MODEL.

compared to cycle-accurate simulations, our model is extremely efficient with a high fidelity, and achieves an average error of 4% and a maximum error of 8%.

It can be observed from Figure 4 that CMP has diminishing return when PE number approaches eight, due to the contention of the shared bus. In our experiments in Section V, we limit our CMP to have only six PEs.

IV. VSVP METHODOLOGIES

A. Models for Power, Clock Frequency and DC-DC Converter Power Efficiency

In our power model, we consider both dynamic power and leakage power. The dynamic power is given as

$$P_d = C{V_{dd}}^2 F \tag{5}$$

where C is the effective switching capacitance as 0.43×10^{-9} for 70nm technology [23]. Leakage power becomes important in deep-submicron semiconductor design. In our work, we choose the leakage power model from [24], which includes the subthreshold and the reverse bias leakage power. For a given supply voltage V_{dd}, the leakage power P_s is given by (6) where I_{sub} is the subthreshold leakage current given by (7):

$$P_s = L_g(V_{dd}I_{sub} + |V_{bs}|I_j) \tag{6}$$

$$I_{sub} = K_3 e^{K_4 V_{dd}} e^{K_5 V_{bs}} \tag{7}$$

where V_{bs}, L_g, I_j, K_3, K_4 and K_5 are technology constants given in [24] for 70nm technology. When a PE is processing a task, it consumes both P_d and P_s. When a PE is idle waiting for incoming task, it only consumes leakage power P_s. The validation of the power model can be found in [23], [24].

For a PE with given V_{dd}, we choose the formulas from [24] to determine its clock frequency F, as shown in (8) where V_{th} is the threshold voltage given by (9):

$$F = \frac{(V_{dd} - V_{th})^\alpha}{L_d K} \tag{8}$$

$$V_{th} = V_{th1} - K_1 * V_{dd} - K_2 * V_{bs} \tag{9}$$

where α, V_{th1}, K_1, K_2, L_d and K are all given in [24], and V_{bs} = -0.7V for 70nm technology [23].

Area (mm^2)	0.126	1.26	12.6
Efficiency (%)	74.7	82.8	88.4

TABLE IV
POWER EFFICIENCY AND AREA FOR DC-DC CONVERTERS WITH 5MV OF OUTPUT VOLTAGE RIPPLE [12].

We derive the model for dc-dc converter power efficiency by extrapolation. As measured in [12], Table IV lists the area of a dc-dc converter with different power efficiency[3] with 5mV of output voltage ripple. In our study, the area of a dc-dc converter is always larger than 1.26 mm^2. Therefore, we focus on the data for area between 1.26 mm^2 and 12.6 mm^2, and derive the power efficiency as a function of area as following:

$$efficiency = 0.056 \cdot log_{10}(\frac{Area}{1.26}) + 0.828 \tag{10}$$

The total system power consumption is given as total PE power divided by the power efficiency of dc-dc converter according to (1).

B. Exploration Methodologies

In our experiments, we use the *workload* to represent a group of incoming task streams simultaneously applied to the processor. Workloads may contain different numbers of streams. Each stream has two properties: one is the specific benchmark to process that stream, which is called the *type* of the stream; the other is the task rate of that stream. For a given workload, tasks from one stream can be processed by multiple PEs, but each PE is limited to executing only one type of stream all the time to avoid large context switch overhead. Therefore, the total number of streams in one workload is no more than the number of PEs.

In this paper, we assume any unfinished task on a PE will be discarded if a new task arrives at that PE. Therefore, the rate of discarded tasks can be used as a criterion of system performance requirement, which can be represented as the requirement of the Quality-of-Service (QoS).

In this work we focus on non-preemptive static scheduling and each task stream is an independent task with arbitrary arrival times and deadlines. Such scheduling on a variable voltage uniprocessor is an NP-complete problem [25]. Therefore, for a variable voltage CMP with M PEs and M voltage domains, i.e., PE number equal to the domain number, such scheduling problem is also an NP-complete problem since the uniprocessor system is a special case of such CMP. In our VSVP problem, we study the same scheduling problem for a variable voltage CMP with N PEs and M domains ($N \geq M$), which include the special case when $N = M$. Therefore, the VSVP problem is also NP-complete. Given the number of voltage domains, the

[3]We replace the capacitances of filter capacitance in the original table in [12] with the corresponding total area of dc-dc converters given in [12].

value of all available V_{dd} and a workload, we use Simulated Annealing (SA) to determine (1) V_{dd}, (2) task type (benchmark to run) in each PE, and (3) task rate in each PE such that the sum of PE power is minimized. The constraint is that all streams must have their QoS satisfied. In our SA procedure, each state is identified by a combination of V_{dd} levels, task types and rates for all PEs, and PEs within the same voltage domain always have a same V_{dd} level. Initially all PEs have a maximum possible V_{dd} and streams are assigned to PEs randomly. There are four types of moves in our SA procedure: (a) change a PE's V_{dd}; (b) shift one stream from one PE to an idle PE; (c) split one stream to two substreams and assign the two substreams to two PEs. When one stream is processed by multiple PEs, the task rates assigned to the PEs are proportional to the clock frequencies of the PEs; and (4) merge two streams with the same type but processed by different PEs into one stream and assign it to a PE. After each move, we compute the system throughput by our analytical performance model, and power by our power model for the new state. If the system throughput cannot satisfy the QoS requirement, then no move will be made, otherwise, a move to the new state may be accepted depending on the power consumption of the new state and SA algorithm.

V. Experimental Results

A. Experimental Settings

In the study of the VSVP problem, we assume a CMP with six PEs. All PEs have the same microarchitectural configuration as described in Section III-C, but may run different benchmarks. We consider discrete voltage levels in this work. We adjust V_{dd} between 0.5V and 1.0V with step 0.05V. The clock frequencies for this range of V_{dd} are between 390MHz and 3.09GHz according to (8) and (9). Furthermore, we also explore the following voltage domain partitioning: 1-domain (6 PEs in one domain), 2-domain (3 PEs in one domain), 3-domain, and 6-domain. We set the QoS requirement such that no more than 5% of total tasks are discarded. We start the SA procedure at a high temperature of 10,000 and end it at a low temperature of 0.05.

The area of CMP includes three parts: PE area, bus area and the area for dc-dc converters. For each PE, we use the CACTI 3.0 toolset [26] to estimate cache area under 65nm technology. For the other parts such as the decoder, the register file and functional units, we take the area from original StrongARM design in 350nm technology [20], and scale it down to 65nm technology. For bus area, we estimate it as 30 mm^2 for a six-PE CMP as shown in [27]. The total area occupied by six PEs and the shared bus is 142.3 mm^2. We allow a 10% area overhead for dc-dc converters. Therefore, the total area for all dc-dc converters can be up to 14.23 mm^2. We assume all this area is occupied by dc-dc converters and all dc-dc converters have identical area.

To construct a workload, we first randomly generate the number of streams and the type of each stream that is among the six benchmarks we used in Section III-C, and then determine the task rate of each stream according to (11):

$$task\_rate = nominal\_task\_rate(type) \cdot task\_ratio \tag{11}$$

where the $nominal\_task\_rate$ is the maximum task rate each single PE can process for the given $type$ of task stream when all PEs have maximum V_{dd} 1.0V and same task type and rate. The reason to estimate the $nominal\_task\_rate$ is that we need to generate a workload with reasonable task rate such that our CMP can satisfy the QoS when all PEs are assigned the maximum V_{dd} 1.0V. The $nominal\_task\_rate$ for each stream type is a constant and decided statically before workload construction. The $task\_ratio$ is a value between 0 and 1, and is randomly generated. In our experiments, we fully construct 32 workloads and all results are based on the average of these 32 workloads.

B. Experimental Results and Discussions

Figure 5 shows the power efficiency of dc-dc converters for different domain partitioning schemes. When the domain number increases from one to six, the power efficiency of dc-dc converters drops from 89% to 83%. Such reduction of power efficiency has significant impact on total system power consumption. Figure 6 plots the normalized power consumption for the whole system (PEs + converters) and that for PEs only. It shows that by increasing the domain number we can reduce total PE power by 16%, and total system power by 8.87%. Interestingly, Figure 6 indicates that the minimum system power is achieved with 3-domain partitioning. In other words, the optimal number of domains (three) is not the maximum number of domains (six). In the literature, almost all studies on system level task scheduling for multi-processor systems assume that different PEs can always have an independent voltage domain and a independent V_{dd}, which essentially assumes the maximum number of voltage domains as it requires the domain number be equal to PE number. Such assumption may not lead to the optimal design for CMPs with on-chip dc-dc converters.

In addition to the impact of voltage domain partitioning on total power, we further present the detailed impact of voltage domain partitioning on different kinds of power consumption: the dynamic and leakage power by PEs, and the power consumed by dc-dc converters. Table V shows the power consumption for each kind. From Table V we can see, voltage domain partitioning can effective reduce both dynamic power and leakage power of PEs. As the domain number increases from one to six, the dynamic power by PEs decreases by 12% from 2.03 Watts to 1.79 Watts. Such reduction is due to the fact that more PEs can have their own lowest V_{dd} to satisfy the QoS of its workload, in other words, larger flexibility for task level voltage scaling. Voltage domain partitioning has a more significant impact on leakage power than dynamic power. As shown in Table

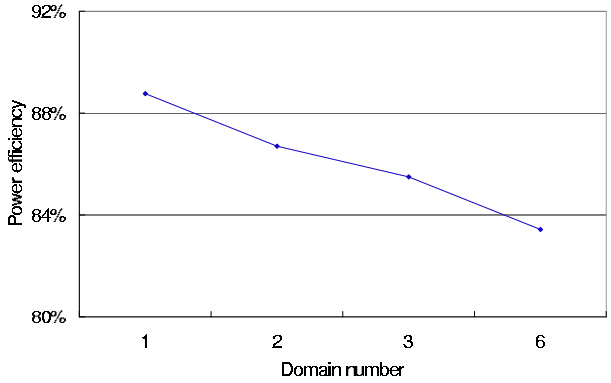


Fig. 5. Power efficiency of dc-dc converters for different voltage domain partitioning.

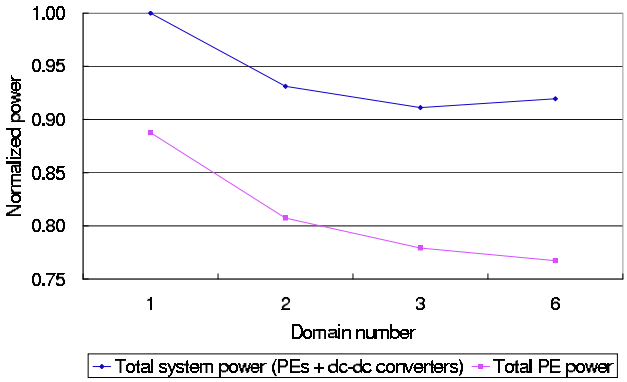


Fig. 6. Total power.

V, leakage power by PEs decreases by 16% from 2.0 Watts to 1.68 Watts. The reason is that with more domains, PEs not only can have individual low V_{dd}, but also can be individually shutdown to save leakage power. Such a phenomenon can be further illustrated by the voltage assignment patterns of PEs with different domain numbers. Table VI lists the V_{dd} assigned to all PEs under two different domain partitioning schemes as well as the workload. The ability to shut down individual PEs makes voltage domain partitioning an attractive method for leakage power reduction, as leakage power becomes increasingly significant.

Domain number	Power of PEs		Dc-dc converter power
	Dynamic	Leakage	
1	2.03 (44.44%)	2.00 (43.86%)	0.51 (11.70%)
2	1.88 (44.36%)	1.79 (42.25%)	0.56 (13.39%)
3	1.81 (43.93%)	1.72 (41.69%)	0.60 (14.38%)
6	1.79 (43.18%)	1.68 (40.76%)	0.69 (16.06%)

TABLE V

POWER CONSUMPTION IN WATT. THE VALUES IN PARENTHESIS ARE THE PERCENTAGE OF TOTAL POWER.

The negative impact of increasing the number of domains can not be ignored, as we already see that the maximum number of voltage domains does not necessarily lead to optimal power. From Table V we can see that when the domain number increases from 1 to 6, the percentage of dc-dc converter power increases 35% from 0.51 Watt to 0.69 Watt. More importantly, the dc-dc converter power has become a significant portion of total power, as the percentage increases from 11.70% to 16.06%. Such result is due to the decrease of power efficiency of dc-dc converters, which has been shown in Figure 5. As the technology keeps scaling down, we may be able to integrate a substantial number of PEs into one chip. Although our CMP contains no more than six PEs due to the scalability of the shared bus structure, CMP designs with the Network-on-Chip communication mechanism have already been able to integrate tens of PEs in the same chip. In that case, according to the trend shown in

Task type	md5	nat	route
Task rate (/s)	53173	1365414	3311055

(A) Workload.

V_{dd}	First	Second	Third	Fourth	Fifth	Sixth
One-domain	0.6V	0.6V	0.6V	0.6V	0.6V	0.6V
Six-domain	0.6V	0.7V	0.65V	0V	0.6V	0.65V

(B) V_{dd} for each PE.

TABLE VI

THE V_{dd} ASSIGNMENT PATTERN FOR TWO DIFFERENT DOMAIN PARTITION SCHEMES UNDER THE GIVEN WORKLOAD.

Table V, we can predict that the power overhead by dc-dc converters may become a severe problem.

VI. CONCLUSIONS AND DISCUSSIONS

We have studied the Voltage Scaling and Voltage domain Partitioning (VSVP) problem for Chip Multi-Processor (CMP) architecture, subject to performance and dc-dc converter area overhead constraints. We have developed the analytical performance model for CMP with voltage scaling to explore the large multi-dimensional solution space. Considering the impact of power efficiency and area of on-chip dc-dc converters, we have found that with the consideration of on-chip dc-dc converters, the optimal voltage domain number may not be the maximum domain number available. Such a result clearly shows that the assumption in existing low power task and voltage scheduling methods for multi-processor systems that a system always contains the maximum number of possible voltage domains (one processor core per domain) may not lead to optimal power for CMP with on-chip dc-dc converters. We have also shown that multiple voltage domains can effectively reduce both dynamic power and leakage power by 12% and 16%, respectively. Furthermore, we have shown that on-chip dc-dc converters can consume up to 17% total power in our CMP, indicating that the power overhead by dc-dc converters may become a severe problem for CMPs.

In this study we focus on static voltage scaling assuming the constant workload and QoS requirement. Our future work will consider the dynamic behavior of the workload and QoS requirement and study dynamic voltage scaling considering the transient overhead of voltage change.

REFERENCES

[1] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," in *IEEE FOCS*, 1995.
[2] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *ISLPED*, 1998.
[3] B. Mochocki, X. S. Hu, and G. Quan, "A realistic variable voltage scheduling model for real-time applications," in *ICCAD*, 2002.
[4] W.-C. Kwon and T. Kim, "Optimal voltage allocation techniques for dynamically variable voltage processors," in *DAC*, 2003.
[5] V. Agarwal, M. Hrishikesh, S. W. Keckler, and D. Burger, "Clock rate versus IPC: The end of the road for conventional microarchitectures," in *Proceedings of* 27^{th} *Annual International Symposium on Computer Architecture*, 2000.
[6] T. Takayanagi, J. L. Shin, B. Petrick, J. Su, and A. S. Leon, "A dual core 64b UltraSPARC microprocessor for dense server applications," in *DAC*, 2004.
[7] N. K. Bambha, S. S. Bhattacharyya, J. Teich, and E. Zitzler, "Hybrid global/local search strategies for dynamic voltage scaling in embedded multiprocessors," in *Proceedings of the Ninth International Symposium on Hardware/Software Codesign*, Apirl 2001.
[8] K. Srinivasan, , N. Telkar, V. Ramamurthi, and K. S. Chatha, "System-level design techniques for throughput and power optimization of multiprocessor soc architecture," in *ISVLSI*, 2004.
[9] J. L. Wong, G. Qu, and M. Potkonjak, "Power minimization in QoS sensitive systems," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, pp. 553–561, June 2004.
[10] A. Andrei, M. Schmitz, P. Eles, Z. Peng, and B. M. A. Hashimi, "Simultaneous communication and processor voltage scaling for dynamic leakage energy reduction in time-constrainted systems," in *ICCAD*, Nov 2004.
[11] A. Chandrakasan and R. W. Brodersen, *Low-Power CMOS Digital Design*. Kluwer Academic Publishers, 1995.
[12] V. Kursun, S. G. Narendra, V. K. De, and E. G. Friedman, "Analysis of buck converters for on-chip integration with a dual supply voltage microprocessor," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 11, pp. 514–522, June 2003.
[13] M. A. Franklin and T. Wolf, "A network processor performance and design model with benchmark parameterization," in 1^st *Workshop on Network Processors in conjunction with Ninth International Symposium on High Performance Computer Architecture (HPCA-9)*, Feb 2002.
[14] "Intel xscale technology," in *http://www.intel.com/design/intelxscale/*.
[15] G. Hinton and et al, "The microarchitecture of the pentium 4 processor," *Intel Technology Journal*, 2001.
[16] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*. John Wiley & Sons, Inc, 1998.
[17] T. Karkhanis and J. Smith, "A first-order superscalar processor model," in *ISCA*, 2004.
[18] "Simplescalar LLC," in *http://www.simplescalar.com/*.
[19] A. R. M. Ltd, *ARM architrcture Reference Manual*, 1996.
[20] J. Montanaro and et al, "A 160-MHz, 32-b 0.5-W CMOS RISC microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 31, Nov 1996.
[21] G. Memik, W. H. Mangione-Smith, and W. Hu, "Netbench: A benchmarking suite for network processors," in *ICCAD*, Nov 2001.
[22] "Nlanr network traffic packet header traces," in *http://moat.nlanr.net/Traces*.
[23] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *DAC*, June 2004.
[24] S. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for low power microprocessors under dynamic workloads," in *ICCAD*, Nov 2002.

[25] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power optimization of variable-voltage core-based systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 1702–1714, Dec 1999.
[26] P. Shivakumar and N. P. Jouppi, "Cacti 3.0: An integrated cache timing, power, and area model," in *WRL Research Report 2001/2*, 2001.
[27] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling," in *(to appear) Proceedings of 32nd Annual International Symposium on Computer Architecture*, June 2005.