

SIMULTANEOUS VOLTAGE SCALING AND VOLTAGE DOMAIN PARTITIONING FOR CHIP MULTI-PROCESSOR

Abstract

In this paper, we study the leakage-aware voltage scaling problem for Chip Multi-Processor (CMP) architecture with minimum power supply cost, and formulate the problem as a simultaneous Voltage Scaling and Voltage domain Partitioning (VSVP) problem subject to constraints on performance and power. 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 simulation, our analytical model has a high fidelity and an average error of 4%. Considering a CMP-based network processor for Voice over IP (VoIP) applications with Quality-of-Service (QoS) guarantee, we find that a single voltage domain with homogeneous voltage scaling is sufficient to reduce dynamic power. However, the ever-increasing leakage power necessitates multiple power domains and heterogeneous V_{dd} cross domains. Such leakage-aware VSVP solution consumes 22.2% less power compared to the leakage-oblivious VSVP solution, when leakage power is considered during power calculation for both solutions.

1. INTRODUCTION

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 [1]. Meanwhile, Chip Multi-Processor (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 [2].

Performance and power aware CMP design has been studied recently. [3] developed a performance model based on the M/D/1 queue model for CPU utilization rate and a design space exploration method for CMP-based network processors considering performance, power, and area. [4] proposes a simulation-based power-aware architecture exploration for multiprocessor system-on-chip. No specific power reduction techniques were proposed in either [3] or [4]. A number of studies have considered dynamic voltage scaling (DVS) for power reduction in multiprocessor systems. Considering traditional multiprocessor systems, [5, 6] targeted on hard real-time task deadlines and [7] studied QoS guarantee for soft real-time systems. Leakage power was not considered in [5]-[7]. [8] considered leakage power in the study of voltage scaling for multiprocessor system-on-chip. [5]-[8] all focused on task scheduling and resource allocation. In addition, [5]-[8] all assumed that each processor may have a customized supply voltage (V_{dd}), and therefore for CMP, an individual voltage domain containing a voltage regulator module (VRM) and a separated power routing network is needed for each processor. This assumption

may lead to a prohibitively high cost for the system power supply.

Considering the ever-increasing leakage power in nanometer technology, we study in this paper the leakage-aware voltage scaling problem for CMP with minimum power supply cost. We illustrate our methodology using a CMP-based network processor system for VoIP with QoS guarantee [19]. Our primary contributions include:

- We develop an analytical performance model for CMP considering on-chip communication overhead and heterogeneous V_{dd} for processor cores. Compared to cycle-accurate simulation, our model is extremely efficient with high fidelity. Its average error is 4%, and the maximum error is 8%.
- We formulate the leakage-aware voltage scaling problem with minimum power supply cost as a simultaneous Voltage Scaling and Voltage domain Partitioning (VSVP) problem. Subject to the constraints on performance and system power, the VSVP problem decides the discrete voltage level for each processor core (or PE) and the voltage domain partition for the CMP in order to minimize power supply cost. We obtain VSVP solutions by leveraging our analytical performance model and simulated annealing to efficiently explore the multi-dimensional solution space.
- Our experiments show that a single voltage domain with homogeneous voltage scaling is sufficient to reduce dynamic power. However, the ever-increasing leakage power necessitates multiple power domains and heterogeneous V_{dd} cross domains. Such leakage-aware VSVP solution consumes 22.2% less power compared to the leakage-oblivious VSVP solution, when leakage power is considered during power calculation for both solutions.

To the best of our knowledge, this paper is the first in-depth study on leakage-aware voltage scaling for CMP considering both voltage scaling and voltage domain partitioning to minimize power supply cost.

The rest of this paper is organized as follows. Section 2 presents system architecture and problem formulation. Section 3 introduces the CMP performance model. Section 4 studies cost-optimal VSVP solutions. We conclude in Section 5.

2. SYSTEM OVERVIEW AND PROBLEM FORMULATION

2.1. 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 cache. A memory controller is in charge of off-chip memory accesses. It receives the memory requests from

PEs, performs necessary read or write operation to off-chip main memory, and returns the responses of memory requests to PEs. PEs and the memory controller communicate with each other via on-chip communication mechanism by a common bus. Furthermore, there may be multiple on-chip voltage domains. Each voltage domain may supply V_{dd} to one or multiple PEs, and all PEs supplied by the same voltage domain always have the same V_{dd} .

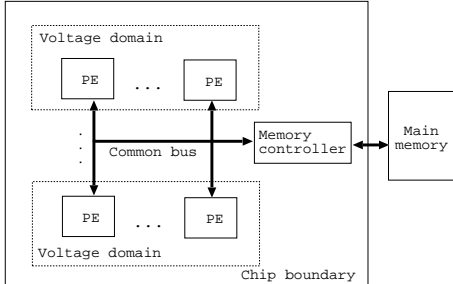


Figure 1: System architecture.

2.2. Voltage Scaling and Voltage Domain Partitioning

Voltage scaling in multiprocessor systems has more flexibility since we can apply different V_{dd} to different PEs. Specifically, we explore the following two design spaces for voltage scaling in CMP: First, we consider discrete V_{dd} levels. Each PE can only be supplied with limited choices of V_{dd} . We call each available V_{dd} as one V_{dd} level and voltage scaling with a total of X available V_{dd} as X -level scaling. More V_{dd} levels enables finer-grain voltage scaling and larger power reduction. However, fewer V_{dd} levels simplifies the design of voltage regulator module (VRM) and reduces the synchronization overhead for communication between different system modules (e.g. PE-to-PE or PE-to-memory controller) for voltage level conversions. Clearly, there is a trade-off to explore.

Secondly, a CMP can have multiple voltage domains composed by PEs and the associated power supply source. More domains provide larger flexibility for voltage scaling in CMP, but lead to a higher cost as more VRMs are required. In our work, we call the voltage domain partitioning with a total of Y voltage domains in a CMP as Y -domain partitioning. When the whole chip has only one voltage domain (1-domain partitioning), voltage scaling for CMP degrades to the traditional voltage scaling for uniprocessor.

It is easy to see that we can achieve maximum power saving when one voltage domain contains only one PE (i.e. N -domain partition where N is the total number of PEs) and the number of V_{dd} levels is maximized. We call this case *ideal case* as it provides the upper bound of power reduction by voltage scaling and voltage domain partitioning.

2.3. Problem Formulation

We assume that there are n total PEs in our CMP. In our CMP, each PE has a supply voltage V_{dd} depending on voltage scaling and voltage domain partitioning, and the clock frequency F and power consumption P associated with V_{dd} . We choose instruction throughput as the metric for performance of each PE, which is equal to the product of F and IPC . The total system throughput of a CMP is the sum of throughput of all PEs.

We formulate our co-optimization problem as follows:

Formulation 1 simultaneous Voltage Scaling and Voltage domain Partitioning (VSVP) problem: Given a total of n available PEs, system throughput requirement PF , maximum number of V_{dd} level M , power under ideal case P_{ideal} , r percentage of power overhead of P_{ideal} , and cost models for voltage domains and V_{dd} levels, find the solution with X -level scaling, Y -domain partitioning and a set of $\{V_i\}$, $1 \leq i \leq n$, with minimum total cost of power supply, and subject to

$$\sum_{i=1}^n P_i \leq (1 + \frac{r}{100})P_{ideal} \quad (1)$$

$$\sum_{i=1}^n (F_i \times IPC_i) \geq PF \quad (2)$$

where P_i and F_i are functions of V_i .

Due to bus contentions, the IPC for individual PEs may be less than that when the same PE is in a uniprocessor system. In Formulation 1, IPC_i not only depends on the microarchitecture of each PEs, but also depends on the number n because the more PEs, the more memory requests and responses between PEs and the memory controller, and therefore the longer average memory access latency for each PE. Furthermore, different memory access patterns are presented when different PEs have different clock frequencies F_i ¹. Such patterns also affect the average memory latency. Hence, IPC_i also depends on the selection of F_i . We will discuss the IPC model in Section 3.

In realistic hardware implementations, we can pre-calculate the solutions of voltage assignment $V_i, 1 \leq i \leq n$ for each PE, under different performance requirement scenarios, and store those solutions into a look-up table. The appropriate voltages to assign at runtime can simply be obtained from this look-up table.

2.4. Network Packet Processing Platform

We choose CMP-based network processors targeting at Internet packet processing for Voice over IP (VoIP) applications as the platform to perform our study. Since packet processing is a highly parallel workload (there is little or no dependency between packets in most cases), network processors implemented as Chip Multi-Processors (CMPs) or multithreaded processors are a natural design point for achieving both high performance and high efficiency (in silicon area and power) [11]. Soft real-time application such as VoIP, can tolerate a certain level of packet loss, introducing the need for an appropriate Quality-of-Service (QoS) guarantee as the performance requirement [19].

In this paper, we assume all PEs in a CMP network processor execute the same application. It takes A instructions for each PE to process one packet. Packets will be dropped if they cannot be processed promptly to meet their deadline. The performance requirement to satisfy QoS guarantee can be derived as

$$QoS_requirement = \frac{outgoing_packet_rate}{incoming_packet_rate} \quad (3)$$

$$PF = A \times QoS_requirement \times incoming_packet_rate \quad (4)$$

where $QoS_requirement$ and $incoming_packet_rate$ are decided by system specification.

¹Voltage scaling is only applied to PEs but not memory controller.

3. PERFORMANCE MODEL WITH VOLTAGE SCALING

In this section we study the CMP performance model. Our model includes an analytical IPC model to estimate the IPC of each PE considering bus contentions and different clock frequencies for different PEs due to heterogeneous V_{dd} . We first present the IPC model assuming uniform clock frequency for all PEs, and then extend the models to consider heterogeneous clock frequencies.

3.1. IPC Model

Assuming all PEs run at the same clock frequency, we propose the IPC model based on the queuing theory. Our CMP has n PEs, one shared bus and one shared memory module. For each PE to process one packet, it needs to execute A instructions, issue M memory requests and spend C cycles. C can be further broken down into two parts: (1) C_p cycles for computation and cache accesses, and (2) C_m cycles for memory accesses. We assume each PE halts (in-order microarchitecture for each packet processor) when it has any unsatisfied memory request, so C_p and C_m do not overlap. C_m can be presented as the product of M and the average memory latency T_a . Therefore, the IPC for each PE is

$$IPC = \frac{A}{C_p + M * T_a} \quad (5)$$

The average memory request rate R is defined as

$$R = \frac{M}{C_p} \quad (6)$$

When the number n increases, C_p and M stay constant but T_a will increase due to bus and memory contention. So IPC will decrease. We assume all n PEs are running the same applications and each application is assigned a separate address space, so there is no memory coherence problem and all the above parameters are the same for all PEs. Hence, the total memory request rate to memory may be between 0 (when all PEs are waiting for unsolved memory requests) and $n * R$ (when all PEs issue memory requests at the same time).

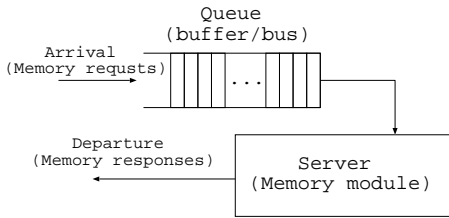


Figure 2: Queuing model for bus and memory structures. The queue includes the bus and the memory request buffer inside the memory controller. The server models the memory module.

The bus, memory controller, and memory structures can be modeled as a queuing system as shown in Figure 2, where bus and memory request buffer in the memory controller are modeled as the queue and the off-chip main memory module are modeled as the server. The overall system latency is the average memory latency T_a observed by each PE. Since each PE is either running or waiting for memory responses, this queuing system is best modeled as a finite source queue such as that in the machine repair

problem [12] instead of a general M/D/1 queue². Therefore, for n PEs, the probability for exactly i memory requests to reside in the system is given in [12] as

$$p_i = \frac{n!}{(n-i)!} * r^i * p_0 \quad (7)$$

$$p_0 = \frac{1}{\sum_{i=0}^n i * r^i * \frac{n!}{(n-i)!}} \quad (8)$$

where p_0 is the probability for the case when no memory request in the queuing system. 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 [12], the total system latency, i.e., the total memory access latency T_a , can be calculated as

$$T_a = \frac{L}{R * (n - L)} \quad (9)$$

Finally, the system IPC can be calculated by (5).

3.2. Consideration of Heterogeneous Clock Frequencies

Our IPC model in Subsection 3.1 can be easily extended to consider different clock frequencies for different PEs. In this case, we can no longer use cycle as the universal time unit as in Subsection 3.1. Instead, we use *second* as our time unit. The per-second memory access rate is represented as R^s for the whole system and R_i^s for PE number i . The memory access latency is T_a^s seconds. We only consider voltage scaling to PEs, but do not scale voltage of bus, memory controller, and memory modules. Therefore, T_a^s stays constant regardless of each PE's clock frequency F .

We keep the form of the analytical formulas in Subsection 3.1, except for two changes: (1) assuming each PE has an access rate R_i^s , we use the maximum rate $R_{max} = \max_{i=1}^n (R_i^s)$, $\forall 1 \leq i \leq n$ as the access ratio R in (9); and (2) we use the *equivalent total PE number* $n_{eq} = \frac{\sum_{i=1}^n R_i^s}{R_{max}}$ to replace the total PE number n in Subsection 3.1. After that, T_a^s can be calculated following the same approach in Subsection 3.1. Finally, we calculate the throughput of each PE as $\frac{A}{C_p + M * T_a^s}$, and the system throughput is the sum of the throughput from all PEs.

3.3. Model Verification

We verify our model by cycle-accurate simulation. We use SimpleScalar/ARM [13] toolset for ARM architecture [14] 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 [15] 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. In total we have eight PEs in our CMP. Note that given the large number of PEs on the same chip, the complexity of this CMP is no less than the state-of-the-art microprocessor designed today, although each PE has very simple microarchitecture. Our memory module has a latency of 128 cycles but can handle up to 8 requests at the same time due to internal pipelining and subbanking. During simulation, we create multiple instances of PE simulators and all of them communicate with the bus-memory simulator via UNIX interprocedure calls.

²Infinite number of sources are assumed in general M/D/1 queues

Benchmark	<i>crc</i>	<i>md5</i>	<i>nat</i>	<i>route</i>	<i>tl</i>
A	14772	21656	7089	616	197
M	101	690	315	11	6
C_p	27722	44630	12960	1380	446

Table 1: Average memory accesses M and core cycle C_p for one PE to process one packet.

We choose netbench suite [16] as our benchmarks and use packet traces available in public domain from [17]. During each round of simulation, all PE simulators execute the same benchmark binary as we assume all PEs are running the same application. 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 1 lists the profiles of all benchmarks we choose from the netbench suite. 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.

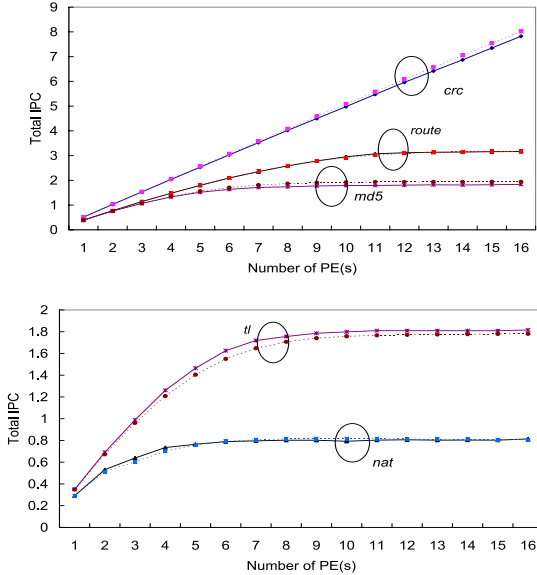


Figure 3: The comparison between simulation and our analytical formula for total sum of IPC among all PEs. In the figures, the solid lines indicate results from simulation and the dotted lines are obtained from our performance model. Five benchmarks from netbench are chosen: *crc*, *md5*, *nat*, *route*, and *tl*.

Assuming all PEs have a uniform clock frequency, we may use the total sum of IPC to represent the total throughput. Figure 3 presents the comparison between simulation and our analytical model. We try five benchmarks in netbench: *crc*, *md5*, *nat*, *route*, and *tl*, and choose PE number from 1 to 16. From Figure 1 it is easy to see that our analytical model correctly tracks changes of throughput for different PEs with a small error bound. Compared to cycle-accurate simulation, our model is extremely efficient with a high fidelity, and achieves an average error of 4% and the maxi-

imum error of 8%.

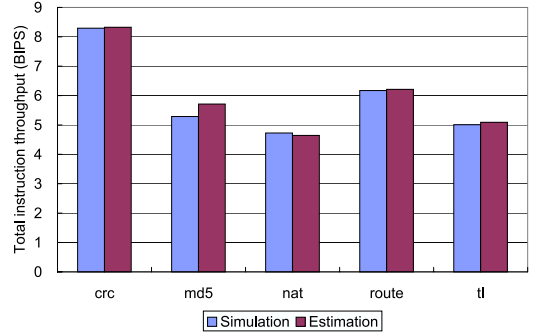


Figure 4: The comparison between simulation and our model for total throughput over eight PEs. Five benchmarks are selected from netbench: *crc*, *md5*, *nat*, *route*, and *tl*.

PE	1	2	3	4	5	6	7	8
F (GHz)	3.09	2.75	2.42	2.11	1.81	1.53	1.27	1.02

Table 2: Clock frequencies for eight PEs in Figure 4

We further modify the PE simulator to handle different clock frequencies for different PEs and verify our models. Figure 4 compares the total instruction throughput of eight PEs between simulation and our model. The clock frequency for each PE is given in Table 2. Overall, our model correctly predicts the system throughput with a high fidelity. The average error is 4% and the maximum error is 8%.

4. COST-OPTIMAL VSVP SOLUTION

In this section we study the cost-optimal VSVP solution. We first present our power model, cost model, and the V_{dd} - F relationship model, then show the system specification and experimental settings, and finally discuss leakage-oblivious and leakage-aware VSVP solutions. All power and cost value in this section is normalized to the ideal case.

4.1. Models

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

$$P_d = CV_{dd}^2 F \quad (10)$$

where C is the effective switching capacitance as 0.43×10^{-9} for 70nm technology [9]. Leakage power becomes important in deep-submicron semiconductor design. In our work, we choose the leakage power model from [18], 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 (11) where I_{sub} is the subthreshold leakage current given by (12):

$$P_s = L_g(V_{dd}I_{sub} + |V_{bs}|I_j) \quad (11)$$

$$I_{sub} = K_3 e^{K_4 V_{dd}} e^{K_5 V_{bs}} \quad (12)$$

where L_g , I_j , K_3 , K_4 and K_5 are technology constants given in Table 3 for 70nm technology from [18]. When a PE is processing

a packet, it consumes both P_d and P_s . When a PE is idle waiting for incoming packet, it only consumes leakage power P_s .

Var	Value	Var	Value	Var	Value
K_3	5.38×10^{-7}	K_4	1.83	K_5	4.19
I_j	4.8×10^{-12}	L_g	4×10^6		

Table 3: Technology constants in our power model for 70nm technology [18].

For a PE with given V_{dd} , we choose the formulas from [18] to determine its clock frequency F , as shown in (13) where V_{th} is the threshold voltage given by (14):

$$F = \frac{(V_{dd} - V_{th})^\alpha}{L_d K} \quad (13)$$

$$V_{th} = V_{th1} - K_1 * V_{dd} - K_2 * V_{bs} \quad (14)$$

where $\alpha = 1.5$, $V_{th1} = 0.244$, $K_1 = 0.063$, $K_2 = 0.153$, $L_d = 37$ and $K = 5.26 \times 10^{-12}$ [18]. We choose $V_{bs} = -0.7V$ for 70nm technology [9].

Our cost model includes the cost ratio between one voltage domain and one V_{dd} level. The cost of voltage domains is decided by VRMs and power supply routing network. The cost of V_{dd} levels is decided by fine-tuning existing VRMs and off-chip L-C components [20]. Obviously the cost of one additional voltage domain is orders of magnitude larger than the cost of one additional V_{dd} level. For simplicity, we assume that the cost ratio between one voltage domain and one V_{dd} levels is 1000:1. However, any cost model for power supply can be applied to our methodology.

4.2. System Specifications and Experimental Settings

We assume our CMP can handle up to 4000 network channels where each channel has an incoming packet rate of 100 per second. Such packet rate is sufficiently large for VoIP applications [19]. The total incoming packet rate for the whole CMP is the product of channel number and the packet rate per channel. This rate is distributed to PEs proportionally to their clock frequencies. In our experiments, we choose the QoS requirement as 95%. We set our target power overhead as less than 20% ($r = 20$) of P_{ideal} .

Number of V_{dd} levels	V_{dd} setting
1	1.0V
2	0.5V and 1.0V
3	0.5V, 0.75V and 1.0V
6	from 0.5V to 1.0V with step of 0.1V
11	from 0.5V to 1.0V with step of 0.05V

Table 4: Selection of V_{dd} levels and corresponding V_{dd} in our experiments.

For voltage scaling, we adjust V_{dd} between 0.5V and 1.0V. The clock frequencies for this range of V_{dd} are between 390MHz and 3.09GHz according to (13) and (14). Different selections of V_{dd} level are presented in Table 4. Furthermore, we also explore the following voltage domain partitioning: 1-domain, 2-domain, 4-domain, and 8-domain. According to our cost model, we derive the normalized cost for each level-domain combination as shown in Table 5.

	Number of V_{dd} levels				
	1	2	3	6	11
1-domain	0.1249	0.1251	0.1252	0.1256	0.1262
2-domain	0.2498	0.2499	0.2500	0.2504	0.2510
4-domain	0.4994	0.4996	0.4997	0.5001	0.5007
8-domain	0.9988	0.9989	0.9990	0.9994	1.0000

Table 5: Cost.

In the study of VSVP problem, we keep the same PE microarchitecture and benchmark settings as those in Section 3. For each combination of voltage levels and power domain partitioning in Table 5, we use simulated annealing to decide the optimal V_{dd} for each PE for power minimization while satisfying the performance requirement.

4.3. Leakage-Oblivious VSVP Solution

Table 6 shows the power consumption considering only dynamic power. From this table we can see that voltage scaling and voltage domain partitioning both can effectively reduce the total system power. With 1-level scaling, increasing the number of voltage domain can reduce power to as little as 13% more than the ideal case. Similarly with 1-domain partitioning, increasing the number of V_{dd} levels can reduce power to about 11% more than the ideal case. Clearly, multiple V_{dd} levels are as efficient as multiple voltage domains in terms of power reduction. However, the design with multiple V_{dd} levels is favored due to the higher cost of additional voltage domains compared to additional V_{dd} levels. According to our target power overhead (20%) and the cost model in Table 5, the optimal leakage-oblivious solution is 3-level and 1-domain.

	Number of V_{dd} levels				
	1	2	3	6	11
1-domain	2.11	2.07	1.17	1.24	1.11
2-domain	1.59	1.58	1.07	1.06	1.06
4-domain	1.31	1.30	1.07	1.05	1.02
8-domain	1.17	1.17	1.04	1.01	1.00

Table 6: Power consumption considering only dynamic power. All value are the average over five benchmarks from netbench: *crc*, *md5*, *nat*, *route* and *tl*.

4.4. Leakage-Aware VSVP Solution

Table 7 shows the power consumption considering both dynamic and leakage power. From Table 7 it is easy to see that multiple voltage domains become more effective in power reduction than multiple V_{dd} levels when leakage power is taken into account. As leakage power becomes significant, shutting down PEs has large impact on power reduction; therefore, the flexibility of additional domains is beneficial. Compared to the ideal case, 1-domain partitioning with maximum number of V_{dd} levels (11 in our experiments) consumes 43% more power with both leakage and dynamic power considered, in contrast to merely 11% (see Table 6) overhead when leakage is ignored. Clearly, a single voltage domain may be sufficient to reduce dynamic power, but with the ever increasing leakage power in the deep sub-micron design era, multiple voltage domains are necessary for efficient power reduction in CMP.

	Number of V_{dd} levels				
	1	2	3	6	11
1-domain	3.49	2.83	1.53	1.59	1.43
2-domain	2.02	1.85	1.19	1.16	1.14
4-domain	1.37	1.33	1.16	1.04	1.02
8-domain	1.13	1.12	1.07	1.02	1.00

Table 7: Power consumption considering both dynamic and leakage power. The settings are the same as Table 6.

Domain	First	Second	Third	Fourth
V_{dd}	0	0.8V	0.9V	0.9V

Table 8: V_{dd} for each voltage domain under the leakage-aware optimal solution (3-level and 4-domain). The benchmark is *md5*.

With respect to the target power overhead and the cost model in Table 5, the leakage-aware cost-optimal solution is 3-level and 4-domain. Table 8 shows the V_{dd} assignment results under the leakage-aware optimal solution. Compared to the results under the leakage-oblivious optimal solution (3-level and 1-domain), where all PEs have the same V_{dd} 0.75V, it is easy to see that leakage-aware VSVP solution tends to use heterogeneous V_{dd} because shutting down PEs helps to reduce leakage power, but leakage-oblivious VSVP solution favors turning on more PE and evenly reducing their V_{dd} . Table 9 compares the power consumption under leakage-oblivious and leakage-aware VSVP solutions³ when both dynamic and leakage power are considered. The leakage-aware solution consumes 22.2% less power compared to the leakage-oblivious solution. This result clearly shows that optimal VSVP solutions are only achieved by considering both dynamic and leakage power.

Leakage-oblivious (3-level and 1-domain)	Leakage-aware (3-level and 4-domain)	Decrease
1.53	1.19	22.2 %

Table 9: Power consumption under cost-optimal leakage-oblivious and leakage-aware VSVP solutions, when both leakage and dynamic power are considered.

5. CONCLUSIONS

We have studied the simultaneously voltage scaling and voltage domain partitioning for Chip Multi-Processor (CMP) architecture, subject to performance and system power constraints (VSVP problem). We have developed the analytical performance model for CMP with voltage scaling to explore the large multi-dimensional solution space. Considering a CMP-based network processor for Voice over IP (VoIP) applications with Quality-of-Service (QoS) guarantee, we have found that a single voltage domain with homogeneous voltage scaling is sufficient to reduce dynamic power. However, the ever increasing leakage power necessitates multiple power domains and heterogeneous V_{dd} cross domains. Such leakage-aware VSVP solution consumes 22.2% less power compared to the leakage-oblivious VSVP solution, when leakage power is considered during power calculation for both solutions

In this study we assume the performance constraint is constant (i.e. constant incoming packet rate and QoS requirement). Our

³We verify that performance required by QoS is satisfied in both solutions by cycle-accurate simulation.

future work will consider the dynamic behavior of the incoming packet rate and QoS requirement and study the dynamic voltage scaling considering the transient overhead of voltage change.

6. REFERENCES

- [1] V. Agarwal and et al, "Clock rate versus IPC: The end of the road for conventional microarchitectures," in *ISCA*, 2000.
- [2] T. Takayanagi and et al, "A dual-core 64b ultraSPARC microprocessor for dense server applications," in *DAC*, 2004.
- [3] M. A. Franklin and T. Wolf, "Power considerations in network processor design," in *Proc. of Network Processor Workshop in conjunction with HPCA-9*, Feb 2003.
- [4] F. Menichelli, M. Olivieri, L. Benini, M. Donno, and L. Bisdounis, "A simulation-based power-aware architecture exploration of a multiprocessor system-on-chip design," in *DATE*, 2004.
- [5] 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*, April 2001.
- [6] 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.
- [7] J. L. Wong, G. Qu, and M. Potkonjak, "Power minimization in QoS sensitive systems," *TVLSI*, vol. 12, pp. 553–561, June 2004.
- [8] A. Andrei and et al, "Simultaneous communication and processor voltage scaling for dynamic leakage energy reduction in time-constrained systems," in *ICCAD*, Nov 2004.
- [9] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *DAC*, June 2004.
- [10] W. J. Dally and B. Towles, "Route packets, net wires: on-chip interconnectoin networks," in *DAC*, 2001.
- [11] W. H. Mangione-Smith and G. Memik, "Network processing: Applications, architectures and examples," in *Tutorial at International Symposium on Microarchitecture*, Dec 2001.
- [12] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*. John Wiley & Sons, Inc, 1998.
- [13] "SimpleScalar LLC," in <http://www.simplescalar.com/>.
- [14] ARM Ltd, *ARM architecture Reference Manual*, 1996.
- [15] 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.
- [16] G. Memik, W. H. Mangione-Smith, and W. Hu, "Netbench: A benchmarking suite for network processors," in *ICCAD*, Nov 2001.
- [17] "NLNANR network traffic packet header traces," in <http://moat.nlanr.net/Traces>.
- [18] 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.
- [19] U. Black, *Voice over IP*. Prentice Hall, 2002.
- [20] T. Kuroda and et al, "Variable supply-voltage scheme for low-power high-speed cmos digital design," *JSSC*, vol. 33, pp. 454–462, 1998.