Process Variation Aware Dynamic Power Management in Multicore Systems with Extended Range Voltage/Frequency Scaling

Shoumik Maiti[1], Nishit Kapadia[2], Sudeep Pasricha[2]
[1] Broadcom Corporation, [2] Colorado State University, Fort Collins, CO, U.S.A.
shoumik.maiti@broadcom.com, nkapadia@colostate.edu, sudeep@colostate.edu

Abstract- **Emerging multicore processors are increasingly power constrained and plagued by design uncertainty due to process variations. This paper proposes a novel framework that enables runtime core selection, thread-to-core mapping, and extended range dynamic voltage frequency scaling (DVFS) operating under near-threshold computing (NTC), nominal, and turbo-boost (TB) conditions. Our framework leverages the process variation profile information of each core together with dark-silicon constraints in chip multiprocessors (CMPs) to select cores, map applications, and compute the optimal voltage and frequency operating points of each core to *(i)* minimize energy under throughput constraints or *(ii)* maximize throughput under power constraints. Our experimental results motivate the need for extended range DVFS and consideration of process variation information. Our framework that supports extended range DVFS results in 15% energy savings and 14.6% higher throughput compared to a framework that uses nominal mode only DVFS. Furthermore, the process-variation awareness of our framework results in 3.7% energy savings and 11.9% improvement in throughput over prior work that does not leverage process-variation information during dynamic power management.**

1. INTRODUCTION

With the proliferation of multicore processors in all segments of computing there has been an increased demand for system-level power management techniques that dynamically control the operation of multiple cores on a die. These techniques typically aim to achieve optimal energy-efficiency and also mitigate reliability concerns due to rising thermal densities as process technology scales down. Runtime thread-to-core mapping (TCM) and dynamic voltage frequency scaling (DVFS) are levers that are widely used by such techniques to maximize chip multiprocessor (CMP) performance while meeting power/energy constraints [1].

In deep sub-micron (DSM) technology nodes there is limited headroom between the threshold voltage (V_{th}) and nominal operating voltage (V_{dd}) [2]. This has reduced the efficiency of voltage scaling, which has led to increased chip power-densities, giving rise to the *dark-silicon* phenomenon [2] – a significant fraction of the chip needs to be shut down at any given time to satisfy the chip power-budget. With the extent of dark-silicon increasing every technology generation (30%-50% at 22nm), designs are becoming increasingly power-limited. One possible solution to overcome the limited voltage scaling problem in DSM nodes is to use an extended range DVFS that encompasses near-threshold computing (NTC), nominal, and turbo-boost (TB) modes of operation. It has been shown in [4] that extended range DVFS can potentially achieve better energy-efficiency and throughput increase than using only the conventional operating range.

Furthermore, DSM nodes are plagued with increased process variations at the within-die (WID) and die-to-die (D2D) levels, which creates unpredictability in leakage power and circuit-delay on CMP dies, complicating design and verification efforts. Any DVFS and TCM scheme that does not include process variation information will not be able to make optimal decisions [3] as there are significant differences in power consumption and performance among cores due to variations in process parameters. The process variation profile information of each core can be readily obtained by measuring on-die intra-core ring oscillator frequencies [3] and DVFS/TCM frameworks should leverage this information.

In this paper, *we propose a novel framework that, for the first time, integrates process variation-awareness and dark-silicon-awareness into a runtime thread-to-core mapping (TCM) and extended range dynamic voltage frequency scaling (DVFS) framework encompassing near-threshold computing (NTC), nominal, and turbo-boost (TB) conditions.* Our framework leverages the process variation profile information of each core in a CMP to select cores to turn-on while satisfying dark-silicon goals, map threads-to-cores, and compute the optimal voltage and frequency operating points of each core for two designer-specified goals: *(i)* energy minimization under performance constraints, or *(ii)* throughput maximization under power constraints. The framework makes use of easily available data from on-die ring oscillators and performance counter information for IPC (instructions per cycle) and LLC (last level cache) misses. As a result, the framework can be applied on any multicore processor that supports fine grained DVFS control, such as in emerging AMD Opteron® and Intel Xeon® multicore processors.

2. RELATED WORK

Several recent works have explored runtime DVFS and TCM techniques for CMPs. For instance, Hanumaiah et al. [1] proposed a runtime conventional DVFS and TCM technique for dynamic thermal management. However, they do not consider process variations or extended range DVFS. Teodorescu et al. [3] and Raghunathan et al. [5] proposed process variation-aware TCM and DVFS schemes but did not consider NTC and TB modes of operation. Juan et al. [4] explored extended range DVFS with NTC and TB modes and dynamic fine-grained DVFS, but did not include process variations or solve the TCM problem in their framework. It is imperative to incorporate process variability information in a TCM and DVFS framework to avoid overly optimistic decisions, and achieve optimal performance and power management.

Our work enhances the extended range DVFS framework described in [4] in two significant ways: by integrating core-level process variation information during decision-making, and by performing active core selection and thread-to-core mapping to meet dark silicon constraints in emerging CMPs.

The remainder of the paper is organized as follow. Section 3 describes our process variation model. Section 4 presents the core level power model. Section 5 explains our proposed dynamic power management framework. Section 6 describes the experimental setup and discusses our experimental results. Lastly, Section 7 concludes the paper.

3. PROCESS VARIATION MODEL

In this section, we discuss our process variation model and assumptions. Process variation results in changes in transistor parameters beyond their nominal values due to imprecise manufacturing in DSM technology nodes. In the presence of process variations, a process parameter can be modeled as a Gaussian random variable with mean (μ) and standard deviation (σ) and its co-relation reduces with increasing distance.

To capture the process variation parameters in 16nm technology,

This research is supported by grants from SRC, NSF (CCF-1252500, CCF-1302693), and AFOSR (FA9550-13-1-0110).

we ran a 1000 point Monte Carlo (MC) Spice simulation using SPECTRE® with 3σ variations of all process parameters at nominal V_{dd} = 0.7V and T = 27C for a 7-stage fanout of 4 (FO4) ring oscillator (RO). The range of frequency of oscillations of the RO for 3σ process variation was noted and divided into equal segments called *process bins*. Candidate ROs were chosen from the MC simulations that represented the process parameter of each bin. The RO for each bin was simulated for the NTC mode [V_{dd} = 0.4V – 0.55V], Nominal Mode [V_{dd} = 0.55V – 0.7V], and Turbo Boost Mode [V_{dd}= 0.7V – 1.0V] to get frequency, dynamic power and static power characteristics in all modes of operation under process variations. The frequency, dynamic and static power of a 7 stage FO4 RO has been shown to follow a similar trend as that of a processor core [4]. The data from RO spice simulation was used to train the processor core level power model (Section 4).

It should be noted that if critical path information of a core is available, similar MC simulations can be performed on the critical path to estimate the core frequency at various process points for all modes of operation and that information can also be used to train the power model. The next section discusses our power model.

4. POWER MODEL

The key challenge in deploying an extended range DVFS framework is developing a unified power model of a processor core that works in a wide range of operation from NTC to TB mode. In this section we formulate our power model for processing cores as a function of frequency and utilization.

Changing the voltage and frequency of a core impacts its power consumption by changing the dynamic and leakage power consumption of each transistor, as well as the utilization of the core (in terms of instructions executed per second). The power consumption of a processing core as a function of operating frequency *P(F)* can be expressed as:

$$P(F) \;= P_{dyn} * F_d(F) * U_I(F) + P_{leak} * F_s(F) \qquad (1)$$

In the equation above, F is the operating frequency; $F_d(F)$ and $F_s(F)$ encapsulate the impact of voltage and frequency scaling on dynamic and static power consumption respectively, assuming the processor core utilization remains constant; $U_I(F)$ is the change in processor core utilization as a result of frequency scaling; and P_{dyn} and P_{leak} are the peak dynamic and static power consumption of each core from the design specifications.

A constraint posynomial (polynomial with positive coefficients) model as proposed in [4] is used to learn the *frequency-power* relationship over the extended NTC to TB operating range. In general, power consumption is a monotonically-increasing function of operating frequency. We approximate this monotonically-increasing function by using a constrained-posynomial function. The algebraic expression of a constrained posynomial used for $F_d(F)$ and $F_s(F)$ in the power model can be expressed as :

$$f(F) = \textstyle\sum_{i=0}^{d} \alpha_i F^d = \alpha_0 F^0 + \alpha_1 F^1 + \dots + \alpha_d F^d \qquad (2)$$

where $\alpha_i \geq 0$, F^k represents the k^{th} power of F, $d \in N$ (non-negative integers), and $F \geq 0$. Since $f''(F) \geq 0$ such models are always convex. Now the problem of learning f is converted into searching for the optimal α and d. The dataset for learning $F_d(F)$ and $F_s(F)$ are obtained from the 7-stage FO4 ring oscillator simulation in 16nm technology as described in the previous section. $F_d(F)$ and $F_s(F)$ is learnt for each process bin. The model parameters α_i are obtained by minimizing the squared error cost function using gradient descent method as used in curve fitting applications [6][7]. It was shown in [4] that d = 4 (4th order posynomial) and d = 6 (6th order posynomial) gives the minimal Root Mean Square Percentage Error (RMSPE) for $F_d(F)$ and $F_s(F)$, respectively, therefore these values of d are used in this work. With these values, a RMSPE of less than 2% was obtained for $F_d(F)$ and $F_s(F)$.

After $F_d(F)$ and $F_s(F)$, we now focus on deriving the *utilization-power* relationship. As pointed out in prior work [4], dynamic power can be fairly accurately approximated as a linear function of IPC, because IPC approximately represents the activity rate of a processor core. Therefore, utilization as a function of frequency $U_I(F)$ can be modeled as:

$$U_I(F) = c_1 * I + c_2 \qquad (3)$$

where I is the IPC, and c_1 and c_2 are fitting coefficients, with $c_1,c_2 \in N$ (non-negative integers). As $U_I(F) \geq 0$ it maintains the convexity of the overall model. The IPC (I) can be expressed as:

$$I = 1/\,[C_{comp} \,+\, (\rho.T)] \qquad (4)$$

where C_{comp} is the ideal CPI assuming an ideal last level cache (LLC), ρ is the average number of LLC misses, and T is the LLC miss penalty. Eq. (4) can also be extended to model other performance overheads, such as L1 cache misses and the difference of the core and the uncore frequency, by including additional parameters. The dataset required to learn c_1 and c_2 captures the change in dynamic power dissipation with IPC when frequency and voltage is kept constant. The Sniper [10] architectural simulator is used to collect IPC and other workload characteristics (C_{comp}, ρ and T) on the target architecture shown in Table 1, for SPLASH2 [9] and PARSEC [8] benchmarks. The dynamic power traces are calculated using McPAT [11]. A standard gradient descent linear regression method [6][7] is used to obtain values of c_1 and c_2. The model matches simulation results with 95% accuracy.

Table 1: Target Architecture

Parameters	Values
Number of Cores	24 (area/core = 5.17 mm^2)
Nominal Freq., Volt.	2.66 GHz, V_{dd} = 0.7V
Core Model	Intel®-X86 Gainestown®
L2 Caches	Private 256 KB, 4-way SA, LRU
L3 Caches	Shared 32 MB, 16-way SA, LRU
DRAM	4 GB
Technology	16 nm

Knowing $F_d(F)$, $F_s(F)$, and $U_I(F)$ and having validated them separately with good accuracy, we now focus on the overall power function *P(F)* which can be expanded from Eq. (1) as:

$$\begin{gathered} P(F) = P_{dyn} * F_d(F) * \big(U_I(F)\big) + P_{leak} * \big(F_S(F)\big) = \\ P_{dyn} * (\textstyle\sum_{i=0}^{4} \alpha_i * F^i) * (c_1 * I + c_2) + P_{leak} * \big(\textstyle\sum_{j=0}^{6} \alpha'_j * F^j\big) \qquad (5) \end{gathered}$$

This overall power model is validated using dynamic and static power traces from McPAT for the architecture in Table 1, with the nominal operating point [V_{dd}=0.7V, F = 2.66GHz, V_{th} = 0.35V] as well as other operating points from [V_{dd} = 0.55V, F = 1.25 GHz], to [V_{dd} = 1.0V, F = 4GHz]. The learned model for the typical process condition was verified and was able to predict the power of the multi-core processor from Table 1 for SPLASH2 and PARSEC benchmark workloads under various voltage, frequency and workload conditions with accuracies ranging from 90 – 98 %. The learned model is convex and standard convex optimization techniques [7] can be used to find the optimal voltage frequency operating point of a core for a given workload.

5. DYNAMIC POWER MANAGEMENT FLOW

In this section we describe our proposed power management framework. Fig. 1(a) gives a high level overview of this framework. At the beginning of each control epoch at runtime, based on the measured IPC (using performance counters) and process variation information (using per-core RO's) for all cores in the CMP, an optimal set of cores are selected that satisfy the dark silicon power budget and thread-to-core mapping is performed. Subsequently, optimal voltage/frequency operating points of the scheduled cores are computed across the extended range DVFS region. If the scheduling interval and DVFS control epochs need to be different, scheduling can also be performed less frequently, e.g., once every several DVFS control epochs.

Our framework can be applied to solve two related problems of interest to CMP architects: (P1) minimizing energy consumption under a predefined throughput constraint, and (P2) maximizing throughput while satisfying a power constraint.

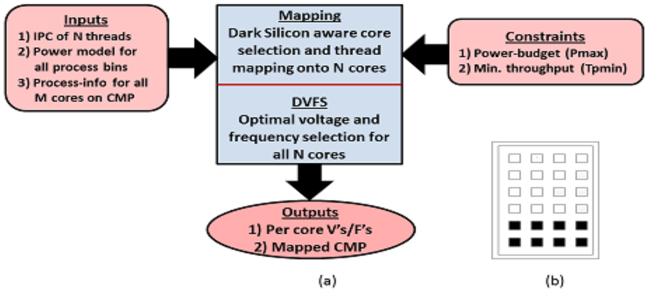
Inputs
1) IPC of N threads
2) Power model for all process bins
3) Process-info for all M cores on CMP
Mapping
Dark Silicon aware core selection and thread mapping onto N cores
DVFS
Optimal voltage and frequency selection for all N cores
Constraints
1) Power-budget (Pmax)
2) Min. throughput (Tpmin)
Outputs
1) Per core V's/F's
2) Mapped CMP
(a)
(b)

Fig 1: (a) Our proposed dynamic power management framework that performs process-variation aware and dark-silicon aware core selection, thread-to-core mapping, and extended-range DVFS; (b) 4×4 cores selected from 6×4 cores in dark silicon CMP

The following subsections discuss our core selection/thread-to-core mapping and DVFS strategies for both problems P1 and P2.

5.1 Core Selection and Thread-to-Core Mapping

In DSM technology nodes, due to process variations, each core in a homogeneous CMP behaves heterogeneously, dissipating different amounts of power and supporting different maximum frequencies (max. performance). Typically, the high performance cores dissipate more leakage power. Moreover, in a dark-silicon constrained CMP there are more cores than can be powered on at the same time for a given chip power budget. Intuitively, better results may be obtained for energy minimization or throughput maximization if process variation information is leveraged when selecting the cores to turn-on and when mapping threads to cores.

As a first step, we need to determine which cores to select (turn on) during an epoch, based on dark-silicon constraints. In our framework, we always prefer to select a contiguous rectangular region of CMP tiles to activate, as shown in Fig. 1(b). Each tile has a core and a NoC router in it. By selecting a contiguous region, we ensure that inter-core communication latency is reduced, and NoC routers needed for communication between active cores are also active. If non-contiguous regions were selected, communication between active regions would not be possible as the tiles between regions (and thus the intermediate routers) would be turned off.

We aim to find a contiguous region of cores that either dissipate the least total leakage-power or operate at the highest performance level, depending on the problem to be solved (P1 or P2; as discussed below). To this end, we perform a simple exhaustive search over all tiles on the CMP die. This step has linear time complexity (with respect to the number of tiles; details omitted due to lack of space), for a given number of cores that can be active based on the power budget, and the rectangular shape constraint.

To map threads to cores for Problem P1, we propose the following two schemes:

Scheme-1: Select a contiguous rectangular region of cores that has the minimum leakage power and map threads with highest IPC to cores with lowest leakage power. This approach works well if the leakage power is dominant, i.e., the savings from leakage power are more than the additional dynamic power required by the slow cores to run at a higher voltage to meet the throughput objective.

Scheme-2: Select a contiguous rectangular region of cores that has the highest performance and map threads with highest IPC to cores with highest performance. This approach will be better if the dynamic power savings by operating the faster cores at a lower voltage is more than the excess leakage power from the cores.

To map threads to cores for Problem P2, we propose to use *Scheme-2*. The scheme is effective for this problem P2 because high IPC threads are typically more compute bound, and mapping these high IPC threads to the fastest cores yields higher performance for a given dark-silicon power budget.

5.2 Extended Range DVFS Operating Point Selection

Once we have selected the active cores and finalized the thread-to-core mapping, we next need to select the voltage and frequency operating level for each core. We formulate per-core frequency selection in our framework as constrained optimization problems for both P1 and P2 as shown below. Once a frequency level for a core is determined, voltage selection is a trivial step: we select the lowest allowed voltage for the selected frequency level on a core.

Problem P1

Objective : arg min(F$_i$) $\sum_{i=1}^{N} P(Fi) \times Te$

Subject to : Throughput (T) = $\sum_{i=1}^{N} Ii * Fi \geq Tpmin$ (6)

Total Power $\sum_{i=1}^{N} P(Fi) \leq Pmax$ (7)

$F_{min} \leq F_i \leq F_{max}, \forall i$ (8)

The objective function here is to minimize the CMP energy. Te is the DVFS control epoch, which can be any positive value. We consider a value of 1 millisecond (ms) without loss in generality. The constraints to the problem are: the total throughput (T) should be greater than a required throughput *Tpmin* and the total power should be within the power budget of *Pmax*. The outputs are frequency values for each core.

Problem P2

Objective : arg max(F$_i$) $T(F) = \sum_{i=1}^{N} I_i \times F_i$ (9)

Subject to : Total Power $\sum_{i=1}^{N} P(Fi) \leq Pmax$ (10)

$F_{min} \leq F_i \leq F_{max}, \forall i$ (11)

The objective function here is to maximize throughput, under the constraint that total power should be less than the power budget *Pmax*. The outputs are frequency values for each core.

By leveraging the convexity of the learned model *P(F)*, as discussed in Section 4, both problems are converted into convex optimization problems. We can solve these problems in several ways, e.g., using the Interior Point, Successive Quadratic Programming or Dual methods. We use the Interior Point method [7] which is extremely fast and can finish within a control epoch.

As the outputs are continuous F levels, they further need to be converted into discrete values. Thus as a final step, the frequency output of the DVFS optimization engine is converted to discrete frequency levels for each core, based on the core's process profile. For problem P1, the next higher discrete V/F value of a core is selected so that the throughput goal is not violated. For problem P2, the nearest lower V/F value is selected such that there are no power constraint violations.

6. EXPERIMENTS

6.1 Experimental Setup

A 24 core tiled homogenous CMP as described in Table 1 with 33% dark silicon (i.e. simultaneously only 16 cores can be turned on) is used in our simulation-based analysis. A 2D mesh network-on-chip (NoC) is used to communicate between cores, with minimal path routing. Each core also has on-chip Phase Locked Loop (PLL) and Low Dropout Regulator (LDO) to control its voltage and frequency of operation. The voltage of each core can be varied from 0.4V to 1.0V in increments of 100mV and the overhead of voltage transitions is less than 9ns [4]. The operating modes for cores across the entire spread of process variation assumed in our work are follows: (1) Turbo Mode: V_{dd} = [0.7V – 1.0V] and F = [2.66GHz – 4.0GHz], (2) Nominal Mode: V_{dd} = [0.55V – 0.7V] and F = [1.25 GHz – 2.66GHz], (3) Near Threshold Computing Mode: V_{dd} = [0.4V – 0.55V] and F = [400MHz – 1.25 GHz]. The nominal V_{dd} = 0.7V, F = 2.66GHz, and V_{th} = 0.35V. As each core can run at a different voltage and frequency level, there are voltage level shifters and synchronizations FIFOs at the core interface. It is also assumed that the frequency of the system bus is always kept at the nominal value. IPC and LLC misses for each control epoch are obtained from performance counters.

For our experiments, we generated 100 random 24 Core CMP die profiles with 3σ process variations to model within-die (WID) and die-2-die (D2D) variations. We used the following PARSEC benchmarks for our simulation-based analysis: blackscholes(bs), bodytrack(bd) canneal(cn), dedup(dp), facesim(fm), ferret(ft), fluidanimate(fe), freqmine(fn), raytrace(re), streamcluster(sr), swapoptions(ss), vips(vs), and x264(x4).

6.2 Experimental Results

We compared our proposed process variation-aware core selection, thread-to-core mapping, and extended range DVFS framework (epDVFS+TCM) against two other approaches that are unaware of process variation information: *(i)* Nominal range DVFS framework with random thread-to-core mapping; and *(ii)* Extended range DVFS framework from [4] with random thread-to-core mapping (eDVFS+TCM). For a fair comparison, we utilize our dark-silicon aware core selection strategy for all three frameworks. Also, to ensure that the two process variation unaware frameworks do not violate timing or TDP constraints, they are limited to using the voltage/frequency map of the slow process and the dynamic and leakage power information of the fast process.

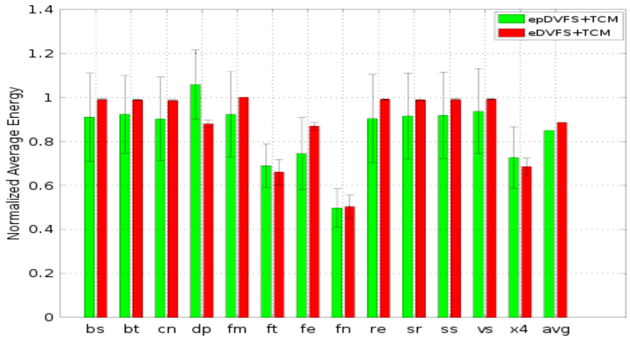

Fig 3: Energy minimization results with throughput constraint

Fig. 3 shows the results for average energy minimization with throughput constraints (problem P1) for PARSEC benchmarks run on all the above 100 random 24 core CMP die profiles. All results are normalized with respect to the nominal range DVFS framework. Note that the error-bars represent the range of values obtained across the 100 trials. In our epDVFS+TCM framework*,* *Scheme-2* was used for thread-to-core mapping in Problem P1, as it produced slightly better results than *Scheme-1*. Benchmarks such as *bs* and *bt* that have high IPC in all threads benefit from the mapping of high IPC threads to faster cores with *Scheme-2*.

It can be seen from Fig. 3 that our epDVFS+TCM framework minimizes energy more effectively than the process variation unaware eDVFS+TCM framework. This confirms that DVFS, core selection, and thread-to-core mapping should be carefully done based on the process variation aware leakage power profile on the die. On average over all the benchmarks, our epDVFS+TCM framework results in 15% energy improvement over the nominal DVFS framework and 3.7% energy improvement compared to the process-variation unaware eDVFS+TCM framework.

Fig. 4 shows the results for average throughput maximization with power constraints (problem P2) across the PARSEC benchmarks with a 95W power budget, which is the TDP of the target architecture. The results are again normalized with respect to the nominal range framework. Results for all benchmarks show significant improvement in throughput when using our epDVFS+TCM framework, which intelligently maps high IPC threads to faster cores thereby allowing the extended range DVFS to find the optimal performance/watt operating point. The process variation unaware framework eDVFS+TCM has the same throughput over all die profiles for a given power constraint, as the voltage/frequency points are computed using worst case models. Our framework shows better results for all die profiles and benchmarks. On an average eDVFS+TCM results in 14.6% improvement in throughput over nominal mode only DVFS. Our process variation aware epDVFS+TCM framework outperforms eDVFS+TCM by an additional 11.9% on average.

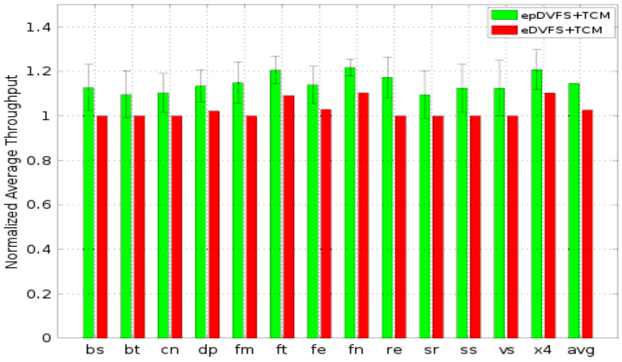

Fig 4: Throughput maximization results with 95W power constraint

7. CONCLUSION

This paper presented a novel dark-silicon aware and process variation-aware runtime core selection, thread-to-core mapping and extended range dynamic voltage frequency scaling (DVFS) framework that is capable of operating in near-threshold computing (NTC), nominal, and turbo-boost (TB) conditions. Our framework exploits process variation information to optimize energy and throughput when executing applications on a CMP die. Our proposed core selection, thread-to-core mapping and extended DVFS based framework that considers process variations results in 3.7% better energy minimization under throughput constraints and 11.9% improvement of throughput maximization under a fixed power budget on average over prior work [4] that considers extended range DVFS but ignores process variations. Thus our framework provides a very promising approach for chip designers and system architects to exploit process variations in emerging dark silicon CMPs to achieve optimal performance-per-watt.

REFERENCES

[1] V. Hanumaiah et al., “Energy-efficient operation of multicore processors by dvfs, task migration, and active cooling”, IEEE Trans. on Comp., Vol. 63, No. 2, Feb 2014.

[2] M. Shafique et al., “The EDA Challenges in the Dark Silicon Era”, DAC, 2014.

[3] R. Teodorescu et al., “Variation-aware application scheduling and power management of chip multiprocessors”, ISCA, 2008.

[4] D. Juan et al., "Learning the optimal operating point for many core systems with extended range voltage/frequency scaling", CODES+ISSS, 2013.

[5] B. Raghunathan et al., “Cherry picking: exploiting process variation in dark-silicon homogeneous chip multi-processors”, CODES+ISSS, 2014.

[6] C.M. Bishop, “Pattern recognition and machine learning”, Springer 2006.

[7] S Boyd et al., “Convex Optimization”, Cambridge Univ. Press, 2004.

[8] C. Bienna et al., "The Parsec benchmark suite: Characterization and architectural implications", PACT, 2008.

[9] S. Woo et al., "The Splash2 programs : Characterization and methodological considerations", SC, 2011.

[10] T. Carlson et al., “Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations” SC, 2011.

[11] S. Li et al., “The McPAT framework for multicore and manycore architectures: Simultaneously modeling power, area and timing.” TACO, 2011.