Process-Variation and Soft-Error Reliability-Aware Workload Mapping with Adaptive Parallelism in SoCs

Nishit Kapadia, Sudeep Pasricha
Department of Electrical and Computer Engineering
Colorado State University, Fort Collins, CO, U.S.A.
nkapadia@colostate.edu, sudeep@colostate.edu

Abstract *- With deep technology scaling, significant design challenges are projected due to process variations, soft-errors, and dark-silicon in integrated circuits. It is well known that spatial variations in process parameters introduce unpredictability in the performance and power profiles of systems-on-chip (SoCs). By intelligently mapping applications on to the best set of available cores, process-variations can potentially be used to our advantage in the era of dark-silicon and worsening soft-error reliability. In this work, we propose a novel framework that leverages the knowledge of variations on the chip to perform run-time application mapping and dynamic voltage scaling to optimize system performance and energy, while satisfying dark-silicon power-constraints of the chip as well as application-specific performance and reliability constraints. Our experimental results show average savings of 35%-80% in application service-times and 13%-15% in energy consumption, compared to the state-of-the-art.*

I. INTRODUCTION

With increasing transistor miniaturization, circuit densities have drastically increased, and the critical charge, which is the minimum charge capable of a bit-flip in a memory- or a logic-cell, has significantly decreased [1], [2]. This phenomenon has caused newer process technologies to be more susceptible to transient-faults due to the effects of radiation, e.g., alpha-particle and neutron strikes. Simultaneously, unpredictability in leakage power and circuit-delay due to variability in modern fabrication processes has become a serious concern. In emerging system-on-chip (SoC) designs, spatially correlated systematic within-die (WID) variations manifest across multiple cores, creating core-to-core (C2C) variations. At the same time, die-to-die (D2D) variations remain quite significant.

Increasing transistor counts and leakage with technology scaling has also led to a rise in chip power-densities [8], manifesting in the dark-silicon phenomenon – a significant fraction of the chip needs to be shut-down ("dark") at any given time to satisfy the chip power-budget. With the extent of dark-silicon increasing every technology-generation (30%-50% for 22nm) [9], [10], designs are becoming increasingly power-limited rather than area-limited. Run-time power-saving techniques such as dynamic voltage scaling (DVS) are thus becoming increasingly important.

Given these multiple daunting design challenges, there is a critical need for a system-level solution that can simultaneously and adaptively manage the constraints imposed by dark silicon, process variations, and soft-error reliability, while executing applications. In this paper, we address this need by proposing a novel run-time application scheduling framework that employs dynamically adaptable application degrees of parallelism (DoPs) to minimize average application service times and energy, while meeting a chip-wide dark-silicon power constraint (DS-Pc) and application performance and reliability constraints, in the presence of process variations. Our key contributions in this paper are summarized below:

- Our novel run-time application-mapping methodology is suitable for the emerging dark-silicon-constrained-design-regime, improving over traditional mapping approaches optimized for the area-constrained-design-regime;
- Our framework simultaneously manages all dynamically arriving applications while adapting application-DoPs to optimally utilize the system-power-slack (difference between DS-Pc and current system power dissipation);
- We design a novel heuristic to integrate within the application-mapping process a DVS mechanism that is constrained not just by performance but also by application-reliability requirements;
- Our combined mapping and DVS approach performs WID variation-aware mapping on to cores with optimal power/performance characteristics, and D2D variation-aware chip-wide DVS where faster chips would need lower V_{dd} levels and slower chips may run at higher V_{dd} levels.

II. PROBLEM FORMULATION

A. Reliability Modeling

The dependence of raw soft error rate, raw-SER (λ), in a hardware component (core or router) on voltage and frequency values is modeled using [4]. Prior works [1], [2] have shown that at technology nodes of 32nm and below, process variations have almost no effect on SER. Therefore, in this work, we assume no dependence of V_T-variations on SER, instead exploiting variations for speed/power benefits only.

B. Inputs and Problem Objective

We assume the following inputs to our problem:

- A SoC with a regular mesh-based 2D network-on-chip (NoC), with T tiles where each tile consists of a compute core and a NoC router;
- A set S of candidate supply voltage (V_{dd}) levels for the chip;
- Application sequence s of length ℓ, made up of η different applications, with arbitrary application inter-arrival times;
- Application task graphs for the set $P = \{P_1, P_2, \ldots P_\eta\}$ of DoPs for all applications; an application i has $|P_i|$ viable DoPs;
- Vertices of each task-graph with execution-times of compute cores and edges with inter-task communication volumes; execution time and volume values are assumed available from offline profiling;
- Energy-optimal frequency constraints $\{f_1, \ldots, f_\eta\}$ and minimum application reliability-constraints $\{Rc_1, \ldots, Rc_\eta\}$;
- A chip-wide dark-silicon power constraint (DS-Pc).

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

We make the following assumptions in our work:

- Applications are mapped contiguously in non-overlapping rectangular regions of the die, for inter-application isolation;
- A chip wide V_{dd} exists that can be scaled using DVS;
- All cores executing an application run at the same frequency;
- Variation-map data for a chip is available at run-time, in terms of the V_T distribution, from the chip-frequency-profile obtained using ring-oscillator based delay sensors [12];

Problem Objective: Given the above inputs and assumptions, our objective is to perform run-time application-scheduling and DVS on a given SoC platform such that the average application service-time and average energy are minimized, while all application-specific operating frequency- and reliability-constraints, as well as DS-Pc are satisfied.

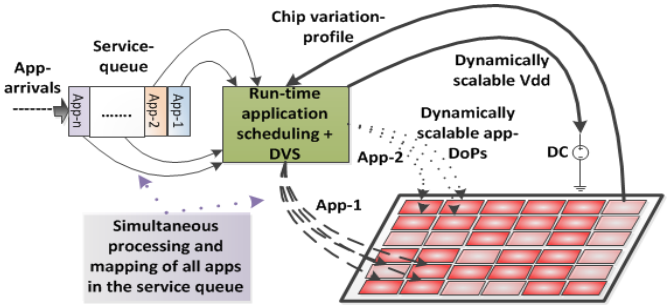


Figure 1: Overview of our application-scheduling + DVS framework

III. Overview

Figure 1 illustrates the key aspects of our proposed framework. The knowledge of the chip-variation profile is continuously utilized in the scheduling and DVS steps. Assuming equal priority for all incoming applications, the application-scheduling step consists of *(i)* determining the DoP (out of the $|P_i|$ DoPs) for each waiting application in the service-queue, and *(ii)* mapping the appropriate task-graphs on to the tiles of the SOC. For a given V_{dd}, scaling-up of application-DoPs (*app-DoPs*) is constrained by the available power-slack (difference between DS-Pc and current system-power), application-reliability constraints, and the available tiles meeting the application-frequency constraints. At any given time, the scaling-down of V_{dd} (to save power/energy) is constrained by the frequency and reliability-constraints of the applications running on the SoC, whereas scaling-up of V_{dd} (to boost *app-DoPs*) is constrained by the DS-Pc for the SoC.

Our framework is effectively executed in two nested procedures: *(i)* V_{dd}-level selection (outer loop), triggered on an arrival or a departure of any application; and *(ii)* determination of application-schedule for the current V_{dd}-level (inner loop). These procedures are discussed in detail in sections *III.A* and *III.B*, and the corresponding design-flows for the procedures are shown in figure 2(a) and figure 2(b), respectively.

A. V_{dd}-level Selection

To extract maximum performance from the applications being considered for mapping at any time instant, the first-order objective in our framework is to maximize overall DoP of the system (sum-total of all *app-DoPs*). An application typically has a maximum viable DoP, and higher DoPs can cause performance to degrade (due to high synchronization overheads) – such higher DoP configurations are ignored. Our V_{dd}-selection heuristic (figure 2(a)) selects the V_{dd}-level that yields the maximum overall DoP. As a second-order power/energy saving objective, on completion of any application, our framework reduces V_{dd} to the lowest allowable level that would not introduce any violations in frequency and reliability constraints of existing (already running) applications.

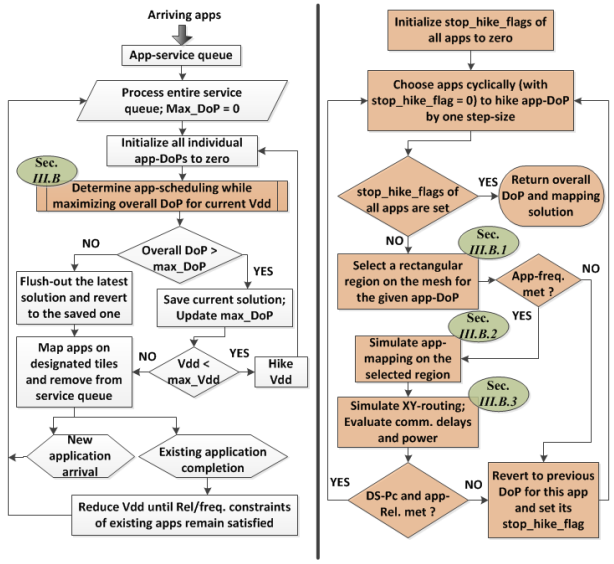


Figure 2: Design-flows for our framework: (a) V_{dd}-level selection (discussed in section *III.A*); (b) Determination of application-schedule for the current V_{dd}-level (discussed in section *III.B*).

We assume all incoming applications are buffered in a service-queue. On arrival or completion of any application, the V_{dd}-selection heuristic is triggered, which processes the entire service-queue. The V_{dd}-selection heuristic iteratively invokes the application-schedule determination procedure (discussed in section *III.B*), which produces the mapping-solution with the highest overall DoP corresponding to the current V_{dd}-level. The V_{dd}-level is hiked (in increments of 0.1V) until either the overall DoP reduces, in which case the immediately preceding solution with the highest overall DoP is reverted to, or the maximum allowable V_{dd}-level (max_V_{dd}) is reached. Note that the overall DoP may increase with increasing V_{dd}-levels, as more applications satisfy frequency and reliability constraints for higher DoPs; at the same time, the chip-power will reach the DS-Pc quicker at higher V_{dd}-levels, thereby limiting overall DoP. Therefore, our search for the optimal V_{dd}-level culminates when the increase in overall DoP is limited by the DS-Pc. Finally, the best application-mapping solution with the highest overall DoP is mapped to the SoC. The voltage supply is hiked to the selected V_{dd}-level, and the mapped applications are then removed from the service-queue.

B. Determination of Application-schedule

Given a specific execution environment for the SoC (including the V_{dd}–level, available power-slack, and variation-profile), the objective of the application-schedule determination heuristic is to maximize the overall DoP, while simultaneously considering all applications in the service-queue and satisfying application-frequency and -reliability constraints. Figure 2(b) shows the design-flow of this heuristic. Starting at the least possible DoP value of zero (DoP of zero leaves the application unmapped at the current time) applications are considered

cyclically for hiking of DoP to their next higher valid DoP-level. Here, to extract maximum performance from the SoC, we choose applications for hiking of DoP in order of their compute-intensiveness, because of the relatively smaller communication delay and power overheads for compute-intensive applications at higher DoPs. Also, we hike *app-DoPs* symmetrically across all applications because execution of an application is generally more energy-efficient at lower DoPs (due to lower parallelization- and communication-overheads).

To produce an optimal mapping for the application under consideration (with a specific DoP), three steps are performed: *(i)* rectangular-region selection on the SoC for the current DoP (section *III.B.1*); *(ii)* mapping of the corresponding task-graph to the selected region (section *III.B.2*); *(iii)* communication-flow routing and delay/power analysis (section *III.B.3*). After the above steps, the mapping is evaluated for overall power footprint and reliability of the applications. Satisfaction of the application-frequency constraints are checked during the rectangular-region-selection step. As shown in figure 2(b), DoP-hike of any application could fail due to potential violation(s) in application-frequency and –reliability constraints or DS-Pc. When an attempted DoP-hike is stalled for any application, its *stop_hike_flag* is set to preclude it from future DoP-hike consideration, and the feasible mapping with the preceding DoP is finalized for this application.

B.1 Rectangular Region Selection

We consider application-mapping on rectangular regions, of pre-determined dimensions corresponding to each possible DoP value in set *P*. All intra-application communication is contained within a rectangular region. This provides inter-application isolation, and eliminates communication cross-interference overhead. Given the V_T-map, the maximum frequency that each core can be reliably clocked at depends upon the V_T and V_{dd} values, given by:

$$f_{max} = \frac{\mu(V_{dd} - V_T)^{\alpha}}{C_0 . V_{dd}}$$

where α and μ are technology-dependent constants, and C_0 is switching capacitance of the critical path [6]. In our region selection method, we utilize knowledge of both frequency and leakage-power profiles of the chip. Our objective is to find the region on the mesh (of pre-defined dimensions) that dissipates the least total leakage-power and all cores within which satisfy the frequency-constraint of the application being mapped. To this end, we perform an exhaustive search over all tiles on the mesh as the left-upper corner of the given rectangular region. If the rectangle is not a square, then both of its orientations need to be checked to find the optimal rectangular region.

<u>*Time-complexity of region-selection*</u>: At most *T* tiles are considered for the prospective rectangular region. Note that *app-DoP* (relatively small integer *c* – treated as constant) tiles are to be evaluated for frequency and leakage-power at each of these iterations. This gives a linear-time complexity: O(2*cT*).

B.2 Application Mapping

After the rectangular region (of size equal to app-DoP) on the mesh has been selected, our mapping heuristic maps the appropriate application-task-graph on to the SoC tiles. We employ a mapping approach based on [3].

B.3 Communication Delay and Power Estimation

Similar to numerous prior works such as [11], we use the XY-routing scheme to route the communication-flows of applications. The communication-delays with congestion-overheads are calculated from the application-frequency (routers and links run at the rated application-frequency) and link BWs. Profiling of compute- and communication-delays could potentially be performed at design-time. Based on the active-times (execution-times) of routers and compute-cores, the application-reliability is computed. For our analyses, the energies and run-times of applications are calculated from component powers and active-times.

C. Complexity Analysis of Our Framework

It can be shown that the region-selection step, which has the highest theoretical time-complexity in the design-flow, finishes in linear-time, the time-complexity of our framework is linear, with respect to the SoC mesh-size, *T*.

IV. EXPERIMENTS

Our experiments were conducted using η=14 different parallel application benchmarks: seven from the SPLASH-2 benchmark suite (*cholesky, fft, lu, ocean, radix, radiosity, and raytrace*), and seven from the PARSEC benchmark suite (*vips, swaptions, fluidanimate, dedup, streamcluster, canneal, and blackscholes*). We consider DoPs that are multiples of 4, up to 16, where a DoP of 8 is considered the nominal DoP value, as a reasonable trade-off between speed and energy. The minimum reliability constraints of different applications are set in the range: 0.99 to 0.999, where application-reliability is {1-Probability of one or more soft-errors during execution}. We assume the ARM Cortex-A9 processors [5] as the baseline SoC compute cores, which support five operating voltage levels (|*S*|=5): 0.8V, 0.9V, 1.0V, 1.1V, and 1.2V. The application-specific energy-optimal core frequencies range from 1300 MHz to 1900 MHz, based on the level of compute intensity of the tasks assigned to cores. We use a 100-core mesh topology based SoC platform with cores arranged in a 10×10 mesh. The dark-silicon power-constraint (DS-Pc) is set at 100W.

To investigate the applicability of our approach to SoC dies with diverse variation-profiles, we use 1000 test-chips (*N*=1000), in our experiments. The 1000 V_T-maps are generated using an open-source tool [6]. The power values of routers and links (32-bit wide) for different voltages and frequencies at varying communication loads, for the 32 nm node are obtained from ORION 2.0 [7]. The router power values obtained are for nominal V_T, and are scaled for varying V_T values.

A. Results

We compare the results obtained from our framework with those obtained from using run-time application mapping frameworks proposed in recent prior works [10] and [11]. A variation- and dark-silicon-aware mapping technique is proposed in [10], whereas [11] advocates for a traditional area-constrained design approach. Our experiments considered two unique application-sequences (Seq-A and Seq-B) that represent an ordering of arriving application instances, with instances randomly chosen from among the 14 applications considered. For each sequence, we vary the inter-arrival times of application-instances randomly within the following ranges: 0

to 1 seconds (Seq-1A and Seq-1B), 0 to 2 seconds (Seq-2A and Seq-2B), 0 to 4 seconds (Seq-3A and Seq-3B), and 0 to 8 seconds (Seq-4A and Seq-4B). We assume ℓ=100 application-instances in any application-sequence. Our results in figure 3 and Table 1 show the mean-values across 1000 test-chips.

The prior works [10] and [11] assume fixed nominal app-DoPs. Our framework adapts app-DoPs in accordance with the application inter-arrival rates to minimize the application service-times. Observe in Table 1 that for both sequences, the average app-DoP reduces with increasing inter-arrival-rates for our framework. At higher inter-arrival rates when applications with nominal DoPs cannot be quickly serviced due to the DS-Pc constraint, our framework cuts down application wait-times significantly by reducing DoPs (as shown in figure 3 - Seq-1A,B and Seq-2A,B), although the application run-times tend to increase due to the reduction in DoPs.

On the other hand, at lower inter-arrival rates, with on average fewer applications to be serviced simultaneously, our framework opportunistically hikes the application-DoPs to minimize run-times (as shown in figure 3 - Seq-3A,B and Seq-4A,B). In comparison with [11], we obtain 27%-43% savings (**35%** on average) in average service-times. Note that maximum savings are obtained when the inter-arrival-rates are most stringent, as shown for Seq-1A and Seq-1B in figure 3. The communication-unaware framework in [10] maps applications on to large rectangular regions of non-contiguous tiles, resulting in longer run-times due to longer communication-latencies. Compared to [10], we obtain 70% - 87% savings (**80%** on average) in average service-times.

Table 1: Mean values for average DoP per application-instance

	Seq-1A	Seq-2A	Seq-3A	Seq-4A	Seq-1B	Seq-2B	Seq-3B	Seq-4B
DoP	4.5	10.3	14.1	14.5	5.1	10.7	13.7	14.2

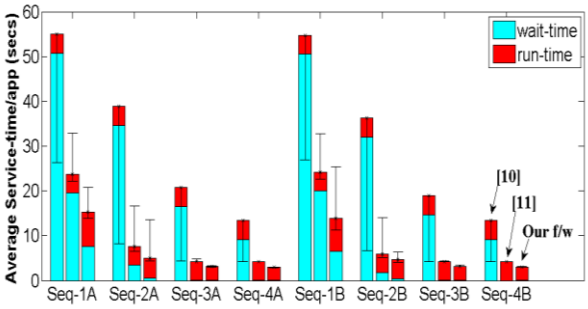
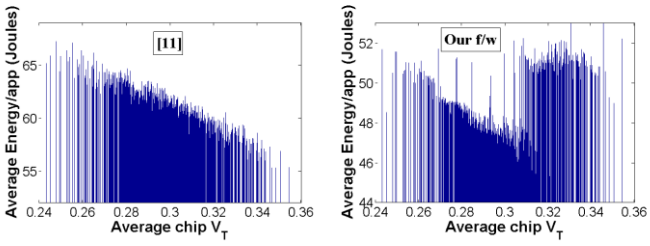


Figure 3: Average service-time per application-instance (wait-time + run-time); the bars represent mean values of service-times across 1000 test-chips, while variation in service-times is shown by confidence intervals.

Our experiments indicate that for the frameworks from [10] and [11], where a fixed high value of V_{dd} (1.2V) is used, slower chips (with high V_T and lower leakage) could prove to be advantageous as they would be left with greater power-slack to accommodate more applications at any given time (less %dark-silicon), thus resulting in better energy values (as shown in figure 4(a) for the framework in [11]). Conversely, in our framework, given a DS-Pc, faster chips (with low V_T and higher leakage power) can usually utilize lower V_{dd}-levels to minimize energy, and slower chips (with high V_T and lower leakage) can utilize higher V_{dd}-levels to improve performance. We observed (as shown in figure 4(b)) that test-chips with moderate leakage and performance profiles produce desirable energy results, whereas chips that are too leaky or too slow yield worse results for our framework.



(a) **(b)**

Figure 4: Average energy/app for all 1000 test-chips using Seq-1B.

For applications with relatively more stringent reliability-constraints, it may not be possible to support nominal-DoP even at the highest V_{dd}-level (1.2V). Therefore, when using reliability-unaware frameworks with no DoP-adaptivity, reliability-constraints (Rc) of such applications may be violated. The Rc-violations (average per sequence) obtained across 1000 test-chips for schedules produced by [10] and [11] are **8.75** and **7** respectively. Our reliability-aware framework, results in no Rc-violations because of its ability to dynamically reduce app-DoPs as well as hike V_{dd}-levels in accordance with application reliability-requirements.

V. Conclusion

This paper represents one of the first efforts to integrate reliability and variation-awareness in a run-time variable degree-of-parallelism application-scheduling methodology to enhance performance of multi-core SoC systems in the new dark-silicon era. The proposed framework produces savings of **35%-80%** in application service-times, **13%-15%** in energy, and avoids reliability violations unlike state of the art prior works that suffer from reliability violations in up to **11%** of application instances arriving at run-time.

References

[1] A. Kaouache et al., "Analytical method to evaluate soft error rate due to alpha contamination," IEEE Trans. Nucl. Sci., 60(6), Dec. 2013.

[2] N. Gaspard et al., "Effect of threshold voltage implants on single-event error rates of D flip-flops in 28-nm bulk CMOS," IEEE IRPS, 2013.

[3] N. Kapadia, S. Pasricha, "VISION: a framework for voltage island aware synthesis of interconnection networks-on-chip", GLSVLSI, 2011.

[4] D. Zhu, R. Melhem, D. Mosse, "The effects of energy management on reliability in real-time embedded systems," Proc. ICCAD, Nov. 2004.

[5] ARM, http://www.arm.com/products/processors/selector.php

[6] S. Sarangi et al., "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," IEEE TSM, (21)1, 2008.

[7] A. Kahng, et al., "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," DATE, 2009.

[8] S. Borkar "Design perspectives on 22nm CMOS and beyond" DAC '09.

[9] J. Allred, S. Roy, K. Chakraborty, "Designing for dark silicon: a methodical perspective on energy efficient systems," ISLPED, 2012.

[10] B. Raghunathan et al., "Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors," Proc. DATE, 2013.

[11] M. Fattah et al., "Smart hill climbing for agile dynamic mapping in many-core systems," Proc. DAC, June 2013.

[12] X. Wang et al., "Design and analysis of a delay sensor applicable to process/environmental variations and aging measurements," IEEE TVLSI, 20(8), pp: 1405-1418, Aug. 2012.