PVT Variation Impact on Voltage Island Formation in MPSoC Design

Sohaib Majzoub[1], Resve Saleh[1], and Rabab Ward[2]
[1]SoC Research Lab, University of British Columbia, 2356 Main Mall Vancouver, BC, Canada
[2]Image Processing Lab, University of British Columbia, 2366 Main Mall Vancouver, BC, Canada
Email: {sohaibm, res, rababw}@ece.ubc.ca

Abstract

On-chip process, voltage, and temperature (PVT) variations are projected to be a major bottleneck in deep submicron design. Such variations can change performance characteristics and push power budgets beyond their limits. In a voltage/frequency island (VFI) design, the initial VFI's determined using optimization without considering PVT may not be suitable after fabrication. This can lead to degradation in energy that largely offsets the advantage of using VFI. Thus, it is crucial to include PVT variations in any pre-fabrication energy optimization algorithm to improve the post-fabricaiton design quality. In this paper, we present a methodology that can reduce the differences by including PVT variations in the optimization. We analyze the PVT impact for different PVT characteristics and propose ways to handle the issue with a penalty of only 3%.

Keywords

Process, voltage and temperature variations, multi-processor system on chip, voltage/frequency islands, energy optimization.

1. Introduction

Parameter variations have negative impact on design quality in the deep submicron era. On-chip variations include process variation due to manufacturing limitations in nanoscale technologies, and voltage variation in the power supply and temperature variation due to activity and workload distribution. These three main sources are generally referred to as PVT (process-voltage-temperature) variation. Process variation comes in different forms: wafer-to-wafer (W2W), die-to-die (D2D) and within-die (WID) variation. Within-die variation is manifested as systematic and random effects. Systematic variation exhibits a spatial correlation [2]. As a result, it has major implications on multiprocessor system-on-chip (MPSoC) design as it creates asymmetric frequency distribution among the cores. On the other hand, the random variation creates independent implications at the core-to-core (C2C) level [1][2][3]. In this work, we are concerned with effects that cause C2C variation as part of the WID variation. In designs using voltage/frequency islands (VFI), PVT variations can negate most of the improvements of power optimization due to its impact on VFI size and location. Thus, it is important to understand and predict the impact of PVT variations and to pursue approaches that take such effects into account.

2. Background

There are only handful of papers so far addressing the problem of PVT variations on MPSoC. Humenay et al [1] introduced the impact of the process variation on MPSoC. They provide a model based on a polynomial equation to construct the systematic variation. They showed the process variation as well as thermal throttling implications on the cores' frequency and proposed an on-chip adaptive regulation to the voltage supply.

In Das et al. [3], the authors modeled the random and systematic variation based on the process parameters. For instance, gate length, threshold voltage, line width, interconnect-metal thickness, and inter-layer dielectric thickness were varied and modeled statistically to construct the process variation impact. They attempt to reduce power consumption using voltage/delay dependency in the presence on the process variation.

In Wang et al. [4], the paper presents a variation-aware task scheduling on an MPSoC. They introduce performance yield as a metric to assist the task allocation. The task graph was mapped as a statistical timing task graph.

Coskun et al. [5] provide an exploration of thermal-aware task allocation on an MPSoC in the presence of temperature variation. They presented an OS-level dynamic task allocation as the new technique and provided a comprehensive examination for different policies with negligible performance overhead.

Suresh et al. [5] present a new load-balancing strategy for MPSoC's that includes process variation. First, they explain the impact of process variation on the performance symmetry among the cores in an MPSoC platform. Based on their observation, they optimize task allocation and distribution to the cores.

Li et al. [7] describe a tool for network-on-chip (NoC) architecture space exploration under process and temperature variation. The tool is used in the early stages of design to provide insight for different design components of the NoC platform. Donald and Martonosi [8], provide a technique for an MPSoC to run within a predefined power budget taking into consideration process variation. They introduce PTCMP (Parallel Turandot Chip Multi-processor) as an efficient simulator to verify their technique. To keep the power within the constraint, they shut down the extra power-consuming cores.

To the best of our knowledge, most of the previous research either did not include energy optimization techniques, or they did not incorporate the bus network overhead due to parameter variations in their optimization.

In this paper, we will utilize voltage/frequency scaling and voltage/frequency island partitioning, in the presence of PVT variations for MPSoC/NoC low-power design. The approach is to minimize power by creating voltage islands based on the task graph associated with the application. One of the methods used to modify VFI size and location is by using DVFS (dynamic voltage and frequency scaling). However, DVFS may have very high implications on power,

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.

Netlist EX2124
Samsung v. Netlist
PGR2025-00071

especially if the islands are widespread across the MPSoC platform. Using a recent process variations model [2], we generate different process-variation profiles and study its effect on VFI size and location created on an MPSoC/NoC platform. Our overall goal is to determine and minimize the impact of PVT on the performance and power of such platforms and provide new methods of accommodating such variations.

3. Process, Voltage and Temperature Variations Modeling

3.1. Process Variation Model

To capture the WID process variation for circuit parameters, such as V_t and L_{eff}, we used the model described in [2]. The systematic process variation is created using a multivariate normal distribution with a spherical spatial correlation structure. The chip is divided into small rectangular-shaped fragments. Each fragment is given a normal distribution of V_t and L_{eff}, the main causes of the systematic variations, with 0 mean and a standard deviation given by σ_{sys}. The correlation factor, $\rho(r)$, is a function of the distance r between fragment X and Y [2]:

$$Correlation(X,Y) = \rho(r);\ r = \left|\vec{X} - \vec{Y}\right| \qquad (1)$$

By definition, if $r = 0$ then $\rho(0) = 1$, which means it is totally correlated. On the other hand, if $r = \infty$, then $\rho(\infty) = 0$ and it is totally uncorrelated. The derived correlation function will be considered as shown below:

$$\rho(r) = \begin{cases} 1 - \frac{3r}{2\varphi} + \frac{r^3}{2\varphi^3}, & r \le \phi \\ 0, & otherwise \end{cases} \qquad (2)$$

Figure 1 shows an example of the V_t systematic variation. The mean of V_t is considered to be 400mV, while the mean of L_{eff} is 60nm, and φ is considered to be 0.5 of the chip size [2]. The random variation is represented by σ_{rand}. The random segment is due to uncorrelated random doping effects. To add the random component, the total V_t variation will be:

$$\sigma_{total} = \sqrt{\sigma_{rand}^2 + \sigma_{sys}^2} \qquad (3)$$

As for L_{eff}, the systematic variation σ_{sys}/μ considered to be half that of V_t [2]. Other variations in mobility, channel width, etc. can be incorporated in the model in the same way.

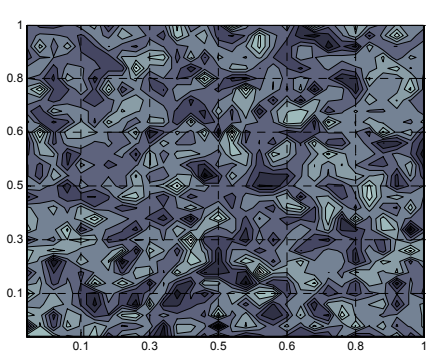


Figure 1: V_t systematic variation for one processor.

Figure 2 shows different normalized frequency distribution with respect to σ/μ ratio of V_t. As the ratio increases, the probability range will also increase. We considered the ratio to be 0.12 in our work.

The model is then extended to a multiprocessor platform by dividing the chip into separate processor regions. The process variation within a region has a direct impact on the speed of the processor. Based on the variation across a processor and its location on the chip, the respective frequency is computed for each processor. Figure 3 shows an example of process variation effects on the frequency of the processors of an 8x8 MPSoC. The processor frequency is determined by computing the delay along the critical path with 10-12 gates. In our case, many different critical path delays were computed for each processor to determine the longest delay. Each color represents a different frequency and produces an overall frequency map.

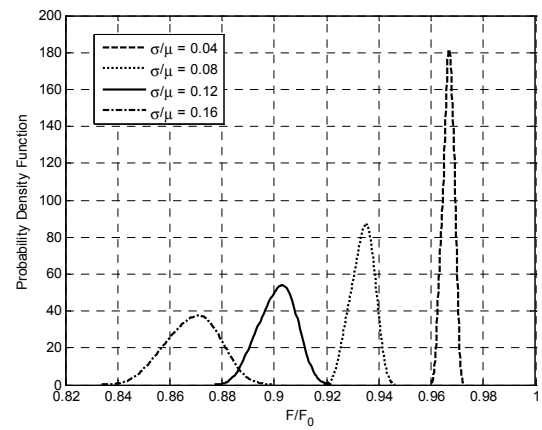


Figure 2: Normalized Frequency distribution with respect to V_t's σ/μ.

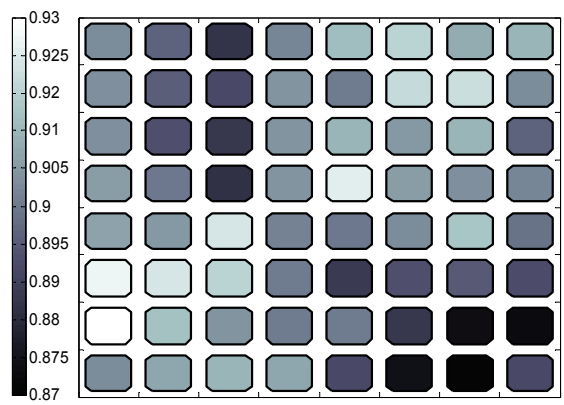


Figure 3: 8x8 MPSoC normalized frequency map with process variation applied

3.2. Estimating Temperature Variation

In order to compute the temperature variation in the system, the HotSpot tool of [10] is used to calculate the temperature of each processor during execution of a segment of code. The leakage power as well as the threshold voltage V_t, which are dependent on temperature, are recalculated based on the new temperature value.

3.3. Estimating Voltage Variation

The power grid model used in [12] was utilized to estimate the voltage variation. The power grid is considered

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.

to be a resistive network, and each processor is replaced by a current source representing its current load. At each cycle, a Modified Nodal Analysis (MNA) matrix is built to represent the current power grid state. The average voltage at every node is updated, and then delay and power numbers are updated accordingly.

4. MPSoC/NoC Architecture

In this paper, we consider MPSoC/NoC architectures with homogenous processing cores. The architecture consists of NxN processing elements with NxN routers/switches, as shown in Figure 4.

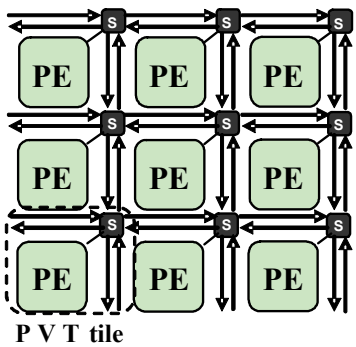


Figure 4: NoC with 3x3 cores

Each PE is connected to a switch, and every switch is connected to 4 adjacent switches via two-way links. We assume that the PVT variations will impact each switch in the same way as its PE. The voltage supply of every switch is also assumed to be the same as the PE associated with it. The dotted region shown in Figure 4 represents the fact that the PE and switch form one PVT tile with one voltage level. A task will be assigned a PE until it finishes and data will be transferred through the network switches. The network has its own energy and speed characteristics.

5. Power/Performance Optimization

The optimization flow graph *without considering PVT effects* is shown in Figure 5, similar to the work in [9]. They used a number of nested genetic algorithms (GA) on a set of heterogeneous processors. Since we are dealing with a homogenous MPSoC here, the outermost genetic algorithm loop, used in the original work for tile mapping, has been removed. The application to be mapped is represented by a computation graph with communication dependences. The directed task graph contains M periodic tasks and E directed edges. Each edge in E has a communication cost C. This cost represents the amount of data to be transferred on the specified edge. The basic steps can be summarized as follows: 1) path allocation is performed for the assigned tasks; 2) voltage scaling of the PE's is carried out; 3) voltage-island partitioning is performed; and 4) the energy reduction is computed.

5.1. Routing Path Allocation

Initial tasks are assigned to processors and the data routing for the tasks in the application is carried out using GA. The genes are first created with n binary bits using 1's and 0's to reflect the shortest path, the Manhattan distance. The crossover operation has an important stipulation to avoid illegal routing. The number of 1's in the parents must be preserved during crossover operation. The mutation operation involves picking a new path at random or altering the current parent path.

5.2. Task Scheduling and Voltage Scaling

The delays of all the tasks are scaled relative to the critical path of the task graph. The algorithm iterates through the non-critical paths in the task graph and slows them down based on their slack times. The voltage scaling starts with the critical path and assigns to it the maximum available voltage level. The remaining critical path voltage levels are assigned based on the available slack. The worst-case communication cost is considered at this point. This cost is determined by assuming that all edges running at the specified time cycle share the same link.

5.3. Voltage Island Partitioning

After the assignment of different voltage levels to each core, the cores are then grouped together to form islands to minimize the number of voltage levels used in the system. This step is effectively voltage island partitioning. In [9], they proposed a new greedy one-step look-ahead algorithm, 1-STEPLA. The algorithm attempts to group adjacent processors using energy as the cost function. It merges two adjacent cores and evaluates the cost of each merge and then selects the best one. The algorithm will reduce the number of voltage levels until it reaches the allowable number.

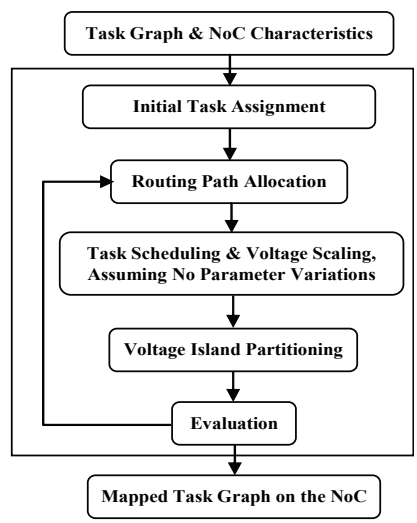


Figure 5: Design Flow graph

5.4. Fitness Function and Energy Model

For every GA iteration, a fitness function is calculated for the energy cost function. The energy model includes the energy consumed in every core, link, and switch in the design. The dynamic power considered is:

$$P_{dynamic} = CV_{dd}^{2} f \qquad (4)$$

where C is the effective switching capacitance, V_{dd} is the operating supply voltage, and f is the frequency at which the core is running. For the static power, the equation used is:

$$P_{static} = I_o e^{-\frac{qV_t}{nkT}} V_{dd} \qquad (5)$$

where I_o is a current coefficient, V_t is the threshold voltage, and T is the temperature. The delay equation used is:

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.

$$D = \frac{C_o V_{dd} / 2}{K.(V_{dd} - V_t)^{\alpha}} \quad (6)$$

where C_o is the switching capacitance of the critical path, K and α are technology dependent constants. The total core energy, E_{core}, will be then:

$$E_{core} = (P_{dynamic} + P_{static}) \times D \quad (7)$$

The communication energy per bit is given by:

$$E_{bit} = n_{hops} \times E_{S_{bit}} + (n_{hops} - 1) \times E_{L_{bit}} \quad (8)$$

where n_{hops} is the number of utilized switches during transmission (or "hops"), while E_S and E_L are the energy levels consumed at the nominal voltage by the switch and the link, respectively.

5.5. Incorporating PVT Variations

The approach described above does not incorporate process variation so it is difficult to predict the actual results if the method is used. Figure 6 shows the new design flow to incorporate process and temperature variations. The frequency map, based on the model described in the previous section, is inserted into the genetic algorithm main body as shown in Figure 6.

In our algorithm, changes in frequency due to PVT variations will be offset by using a new voltage level. The voltage scaling step adjusts for the modified frequency in the fourth step. In most cases, the voltage levels will be increased to compensate for the frequency loss in the processors and switches. As a result of new voltage level assignments (relative to the original algorithm), a different result is produced in the voltage island partitioning step. HotSpot [10] is used to report the temperature for a given MPSoC floorplan. The reported temperature can be used to recalculate the frequency and correct static power number.

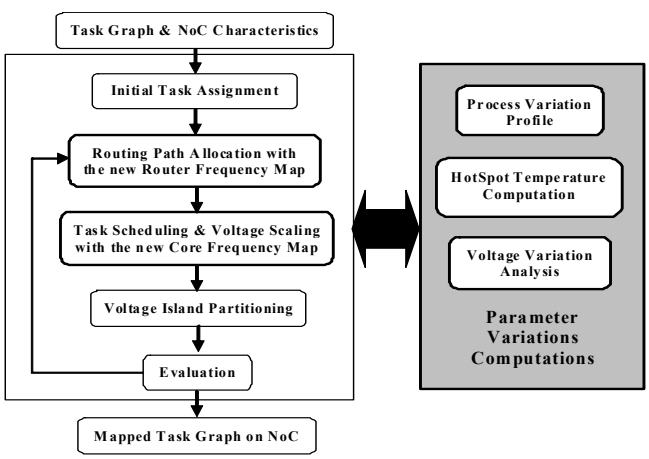


Figure 6: The New Design Flow graph

6. Experimental Results and Analysis

We examined a pool of 220 task graphs mapped onto MPSoCs sized from 3x3 up to 16x16 processors. The task graphs have either 50 or 100 tasks per graph, with a parallelism factor ranging from 4 up to 50 [11]. The number of voltage levels, VL's, allowed in the implementation can be either 3, 4 or 5. The value of the VL's can be any value from 0.6V up to 1.4V in 0.1V increments. Any unused cores are switched off completely. The technology considered in the computation is 65nm with nominal V_t=400mV, and the maximum V_{dd}=1.4V. The maximum speed at which the processor can run is 50MHz with the network running at 100MHz. The dynamic power was based on 0.5mW/MHz. The energy-saving results are grouped and averaged.

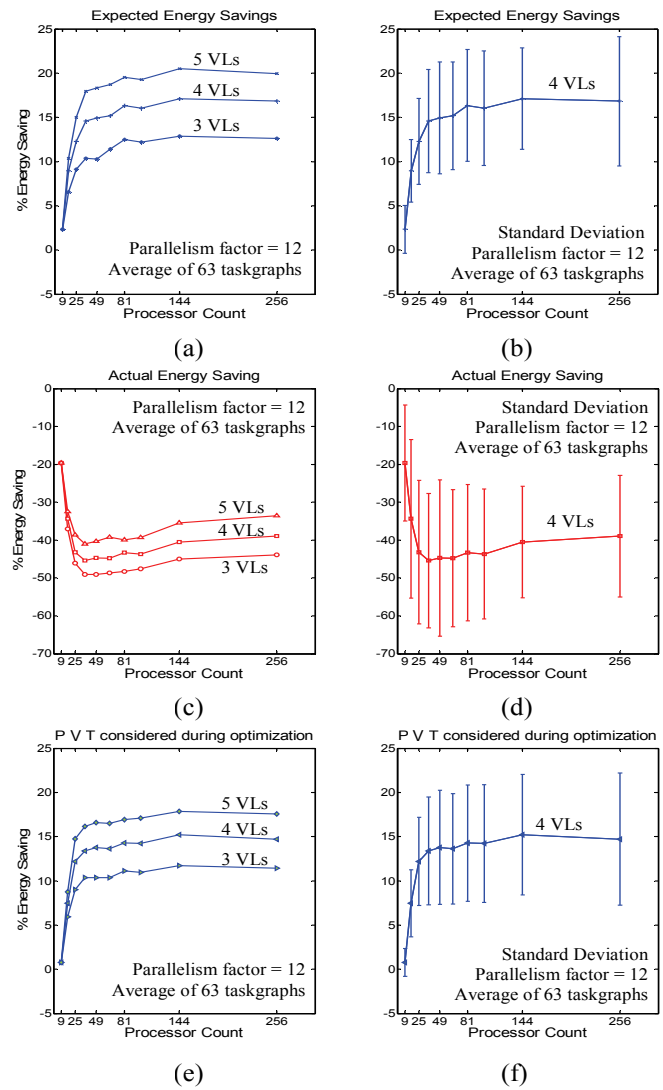


Figure 7: Expected, actual, and PVT considered energy savings for task graphs with 50 tasks.

To show the significance of including PVT effects during energy optimization algorithm, the first part of our analysis presents a comparison of the energy saving between the initially expected energy savings, the actual energy saving after imposing PVT impact, and then the energy saving if PVT considered during optimization. Figure 7 shows the results of the three cases. Plot (a) corresponds to the overall average of the expected energy savings for different VL's. Plot (c) shows the actual energy savings after imposing PVT, and plot (e) is the case where PVT is considered during optimization. Plots (b), (d), and (f) show the standard deviations of the averaged task graphs across 63 applications when the number of allowed voltage islands is 4. Figure 7 (a)-(d) illustrates the significant difference in energy savings between expected and actual when process and temperature variations are not considered during optimization. However,

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.

as shown in plots (e)-(f), the energy savings can be made to approach the initial plots using the new approach. In fact, even the exact PVT profile is not known in advance, the use of the proper models of Section 3 is sufficient to produce much improved results, whereas any optimization that does not consider PVT variations may drop from 20% to -30% (cf. 5 VLs).

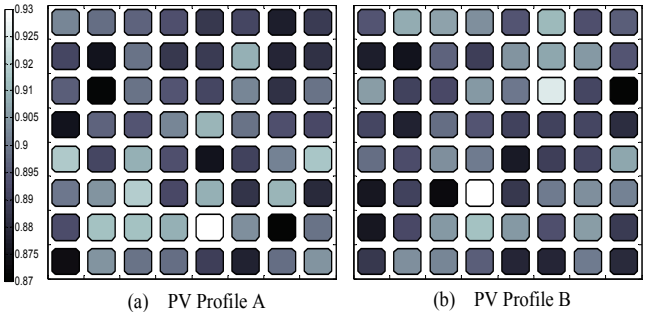
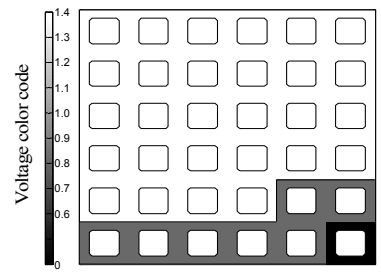
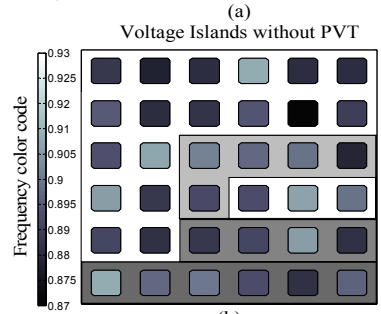


Figure 8: The two Process Variations profiles used in the analysis, (case of 8x8 cores).



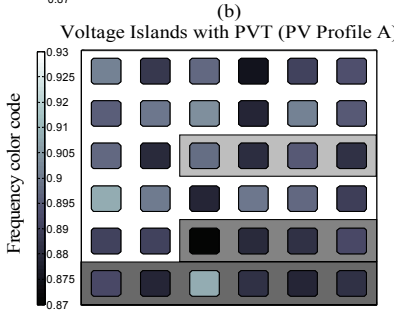
	N# of PE	V_{dd}
VI_1	28	1.4V
VI_2	7	0.8V
VI_3	1	0.6V

(a)
Voltage Islands without PVT



	N# of PE	V_{dd}
VI_1	18	1.4V
VI_2	3	1.4V
VI_3	5	1.1V
VI_4	4	0.9V
VI_5	6	0.8

(b)
Voltage Islands with PVT (PV Profile A)



	N# of PE	V_{dd}
VI_1	34	1.4V
VI_2	4	1.1V
VI_3	4	0.9V
VI_4	6	0.8V

(c)
Voltage Islands with PVT (PV Profile B)

Figure 9: Voltage Islands (a) without PVT, (b) PVT, PV profile A and (c) PVT, PV profile B

In the second part of our analysis, we used two different process variation profiles to further demonstrate this point. We assume that PV profile A is pre-fabrication and PV profile B is post-fabrication. Figure 8 shows the two profiles for an 8x8 core test case. Figure 9 shows an example of how the size and shape of the voltage island will change with respect to its PV profile. Figure 9(a) shows the application running without including PVT. However, Figure 9(b) and 9(c) show same application with the new voltage islands under PVT considering PV profile A and profile B, respectively. It is interesting to see the big difference between case 9(a) and either 9(b) or 9(c). On the other hand, there is minimal difference between 9(b) and 9(c). Noting this, if PVT is included in the pre-fab optimization, the expected difference with post-fab will be relatively small in VFI formation compared to when PVT is not included.

On the other hand, handling such differences between PV profile A and B is necessary as well to avoid any loss in energy. Figure 10 shows the average energy savings of 110 task graphs, with 50 tasks each, for both PV profile A and B in the case of re-optimization for each profile alone. To handle the difference between A and B, we can either enforce the VFI of A on B or we can re-adjust the VFI's in A to meet the new requirements of B. Enforcing VFI of A may lead into extra delay. Figure 11(a) and 10(b) show the energy savings in this case. Only 3% of the energy savings is lost due to the mismatch between case A and B.

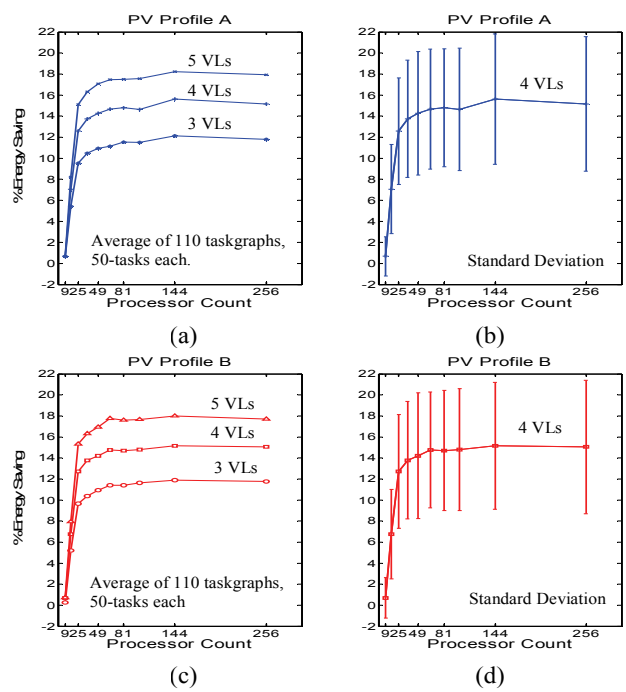


Figure 10: Energy Saving applying PV Profile A, and PV Profile B.

On the contrary, if the voltage islands are adjusted to meet the new requirement of PV profile B, Figure 11 (c)-(f) shows the energy saving outcomes. The figure shows two cases: Figure 11 (c)-(d) shows the case where the VFI size is limited at 2x2 cores, and Figure 11 (e)-(f) shows the case where there is no limit of the VFI size. The energy savings will decrease by about 2% in the first case (smaller VFI size, hence less PVT impact), while it will decrease by 4% in the second (bigger VFI size, hence greater PVT impact).

Furthermore, we carried additional analysis on the orientation of the mapping on the PV profile. The orientation involved rotating and flipping of the PV profile. Eight

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.

different orientations were tried to match PV profile A to B, as shown in . Due to the variability in PVT, the orientation did not appear to have much of an impact considering the sizes of the MPSoC and the task graphs under examination here.

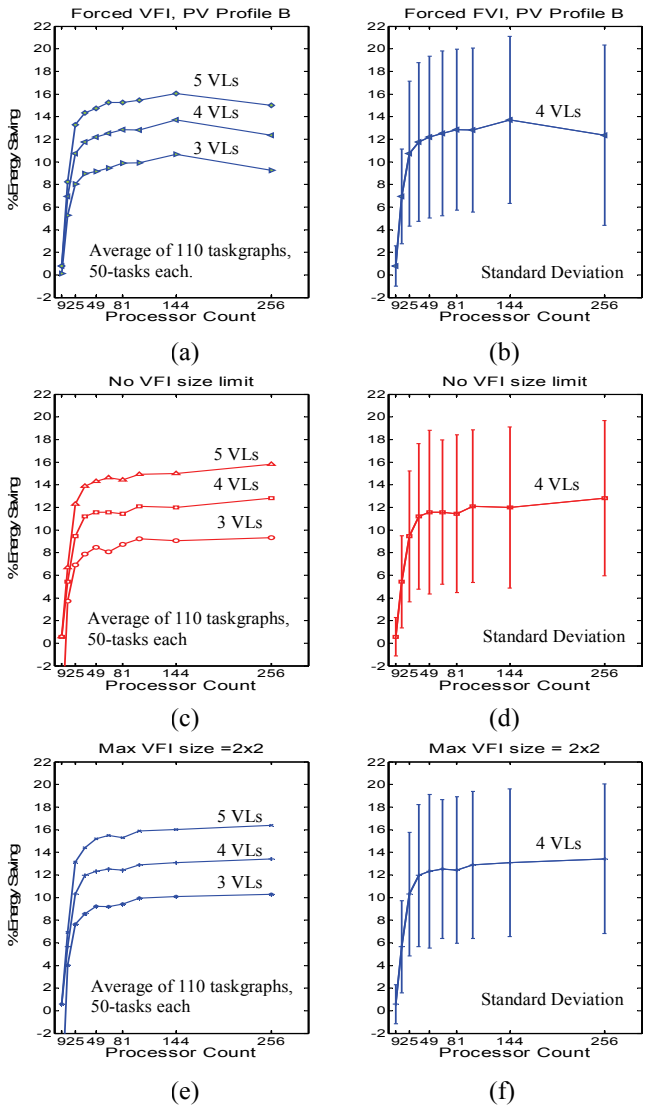


Figure 11: Energy Saving, using forced VFI, modifying VFI with no VFI limit and modifying VFI with VFI size 2x2.

7. Conclusions

In this paper, we showed the impact of PVT on voltage/frequency island formation on MPSoC platforms. We provided suitable analysis to show the implications for PVT on the size and location of VFI in the design. Based on the results, PVT should be part of any pre-fab power/performance optimization algorithm with suitable PVT models. We proposed a methodology to handle PVT with different process variation profiles with minimum impact on energy saving.

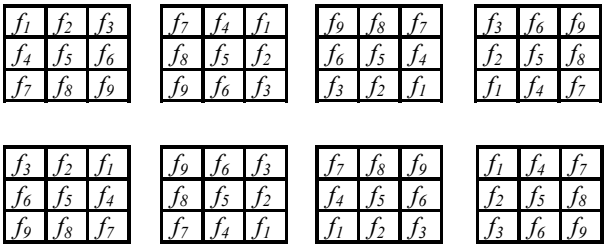


Figure 12: Eight different frequency orientations onto which the application can be mapped.

8. References

[1] E. Humenay, D. Tarjan, K. Skadron, "Impact of process variations on multicore performance symmetry." DATE, Nice, France, Apr. 2007.

[2] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas. "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," IEEE TSM, Feb. 2008.

[3] A. Das, S. Ozdemir, G. Memik and A. Choudhary. "Evaluating Voltage Islands in CMPs under Process Variations." ICCD, Lake Tahoe, CA, USA, Oct. 2007.

[4] F. Wang, C. Nicopoulos, X. Wu, Y. Xie, N. Vijaykrishnan, "Variation-aware task allocation and scheduling for MPSoC." ICCAD, San Jose, CA, USA, Nov. 2007.

[5] A. Coskun, T. Rosing, K. Whisnant, "Temperature aware task scheduling in MPSoCs." DATE, Nice, France, Apr. 2007.

[6] S. Suresh, R. Raghavan, N. Vijaykrishnan "Process Variation Aware Parallelization Strategies for MPSoCs". SOCC, USA, Sep, 2006.

[7] B. Li, L-S. Peh, P. Patra. "Impact of Process and Temperature Variations on Network-on-Chip Design Exploration". IEEE International Symposium on Networks-on-Chip, April, 2008.

[8] J. Donald and M. Martonosi "Power Efficiency for Variation-Tolerant Multicore Processors" ISLPED, USA, 2006.

[9] L.-F. Leung, C.-Y. Tsui, "Energy-Aware Synthesis of Networks-on-Chip Implemented with Voltage Islands." DAC, USA, 2007.

[10] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, M. R. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design." IEEE TVLSI 2006.

[11] T. Tobita, H. Kasahara, "A standard task graph set for fair evaluation of multiprocessor scheduling algorithms." ICS'99 Workshop. (1999).

[12] A. Ramalingam, G. Devarayanadurg, D. Pan, "Accurate Power Grid Analysis with Behavioral Transistor Network Modeling", ISPD, USA, 2007.

Authorized licensed use limited to: Steven Przybylski. Downloaded on June 01,2026 at 00:10:02 UTC from IEEE Xplore. Restrictions apply.