System-level PVT Variation-Aware Power Exploration of On-Chip Communication Architectures

SUDEEP PASRICHA
Colorado State University
and
YOUNG-HWAN PARK, NIKIL DUTT, and FADI J. KURDAHI
University of California, Irvine

With the shift towards deep submicron (DSM) technologies, the increase in leakage power and the adoption of power-aware design methodologies have resulted in potentially significant variations in power consumption under different process, voltage, and temperature (PVT) corners. In this article, we first investigate the impact of PVT corners on power consumption at the system-on-chip (SoC) level, especially for the on-chip communication infrastructure. Given a target technology library, we then show how it is possible to "scale up" and abstract the PVT variability at the system level, allowing characterization of the PVT-aware design space early in the design flow. We conducted several experiments to estimate power for PVT corner cases, at the gate level, as well as at the higher system level. Our preliminary results are very interesting, and indicate that (i) there are significant variations in power consumption across PVT corners; and (ii) the PVT-aware power estimation problem may be amenable to a reasonably simple abstraction at the system level.

Categories and Subject Descriptors: B.71 [**Integrated Circuits**]: Types and Design Styles—*VLSI* (*very large scale integration*); C.1.2 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)—*Interconnection architectures* (*e.g., common bus, multiport memory, crossbar switch*)

General Terms: Design, Experimentation

Additional Key Words and Phrases: PVT variation, on-chip communication architectures, performance exploration, power estimation, digital systems, high-level synthesis

This research was partially supported by grants from SRC (2005-HJ-1330).
Authors' addresses: S. Pasricha, Dept. of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373; email: sudeep@engr.colostate.edu; Young-Hwan Park, N. Dutt, F. J. Kurdahi, Center for Embedded Computer Science, University of California Irvine, Irvine, CA 92697; email:younghwp@uci.edu, dutt@uci.edu,kurdahi@uci.edu.

DOI 10.1145/1497561.1497563 http://doi.acm.org/10.1145/1497561.1497563

ACM Reference Format:
Pasricha, S., Park, Y.-H., Dutt, N., and Kurdahi, F. J. 2009. System-level PVT variation aware power exploration of on-chip communication architectures. ACM Trans. Des. Autom. Electron. Syst. 14, 2, Article 20 (March 2009), 25 pages, DOI = 10.1145/1497561.1497563
http://doi.acm.org/ 10.1145/1497561.1497563

1. INTRODUCTION

With the advent of the deep submicron (DSM) era, more and more system-on-chip (SoC) designs are being fabricated in sub-100nm technologies. Unfortunately, *process, voltage and temperature (PVT) variability* makes it hard to achieve "safe" designs in such nanometer technologies. This is because PVT variability causes fluctuation in timing as well as power for SoC designs [Borkar et al. 2004]. Consequently, timing and power estimates derived early in the design flow are no longer valid, and considerable redesign effort is needed to account for these variability-induced fluctuations. Recently, many research efforts have focused on statistical timing analysis [Le et al. 2004; Khandelwal and Srivastava 2005] to address variability in timing. However, utill now very few efforts [Papanikolaou et al. 2005] have looked at addressing the effect of PVT variability on system-level power estimation. Since reducing power consumption is increasingly becoming the most important goal for SoC designs, especially for portable battery-driven embedded systems [Lahiri et al. 2002], it becomes essential to address the issue of reliable power estimation for these designs, in the face of PVT variability.

In modern IP-based design, the communication architecture backbone has become a significant factor in influencing overall system power, performance, cost, and time-to-market [Ho et al. 2001; Pasricha and Dutt 2008]. In particular, it has been shown that for some SoC designs, on-chip communication architectures (wires and bus logic) can consume anywhere between 20 to 50% of overall system power [Lahiri et al. 2004]. The amount of power consumed in the various bus logic components is also steadily increasing with design complexity, and has been shown to be as high as 80% of the total on-chip communication power [Lahiri et al. 2004; Pasricha et al. 2006]. Furthermore, a significant portion of the on-chip communication architecture power consumption is converted into heat, which has been shown to not only increase interconnect delay (reducing performance), but also increase electro-migration (EM), which significantly increases the device failure rate [Banerjee et al. 2001]. These observations motivate the need for system-level estimation of on-chip communication architecture power consumption early in the design flow, where design decisions have a much greater impact on power consumption than at lower levels.

While in the past, leakage was negligible and dynamic power did not vary much between technology corners, today the increase in leakage power and the adoption of power-aware design methodologies (such as voltage islands and DVS/DFS) has resulted in considerable variations in power consumption under different process, voltage, and temperature (PVT) technology corners. In this article, we first explore the impact of PVT corners on power consumption at the system level, especially for the on-chip communication architecture. We then show how the variability due to different PVT corners can be abstracted up to

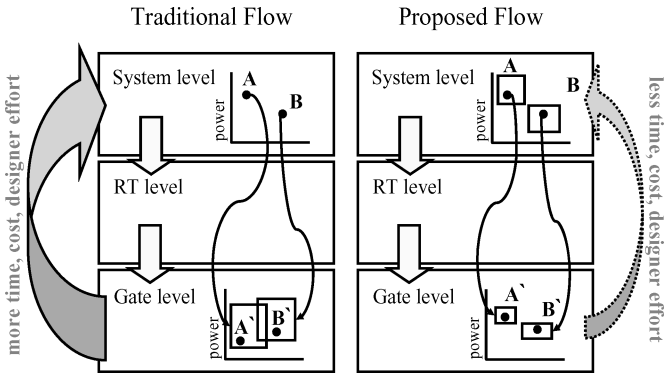


Fig. 1. Traditional approach compared with the proposed PVT variation-aware system-level exploration approach.

the system level for the on-chip communication architecture, where the corners can be explored early in the design flow. To the best of our knowledge, this is the first piece of work to incorporate PVT variations at the system level during power exploration of the on-chip communication architecture.

Figure 1 illustrates the difference between the traditional SoC design approach, and the approach proposed here. In the *traditional approach*, designers explore the power space of the design at the system-level design and select the design configuration with the least power consumption. Note that a SoC design can have different design configurations, all of which implement the requisite functionality of the design, but differ slightly in their implementations. For instance, different design configurations may have different numbers of buses, clock frequencies, buffer sizes, and so on, resulting in differences in power dissipation and performance. In Figure 1, points A and B represent two design configurations of a SoC design, and a designer would select configuration B with the lower power consumption at the system level. Later in the design flow, at the gate level, designers encounter process, voltage, and temperature (PVT) variations that alter the power characteristics and behavior of the synthesized design. Each of the configuration points becomes a large region of uncertainty (representing possible power consumption under different PVT conditions), and it is no longer clear whether configuration A or B is superior in terms of lower power consumption. It is possible that an instance of the design configuration A (shown as A in the figure) is found to be superior to the best instance of design configuration B (shown as B in the figure). As a result, designers may end up spending considerable time and effort in exploring design configurations at the gate level. It is also important to ensure that PVT variations do not violate design constraints for the selected design configuration. Design reiterations might be required if violations are detected (requiring changes in the design at the system level), which can severely influence design cost and time-to-market. In contrast, in our *proposed approach*, we attempt to "scale up" and abstract the PVT variability at the system level to provide a more realistic characterization of the design space early in the design flow. The designer can then select a design configuration with greater confidence, after analyzing its

behavior under PVT variations. This significantly reduces the exploration and redesign efforts later in the design flow. We conducted several experiments to estimate power for PVT corners of DSM technology libraries, at the gate level as well as at the higher system level, especially for on-chip communication architectures. Our preliminary results are very interesting and indicate that (i) there are significant variations in power consumption across PVT corners; and (ii) the PVT-aware power estimation problem may be amenable to a reasonably simple abstraction at the system level. Note that in this article we do not consider intra-die PVT variations, which are beyond the scope of this work.

The rest of this article is organized as follows. Section 2 presents related work in the area of on-chip communication architecture power estimation. Section 3 discusses the role of PVT corners in ultra-deep submicron technologies. Section 4 illustrates the role of PVT variations on power consumption. Section 5 describes scaling relations that can be used to determine power consumption across PVT corners. Section 6 presents an overview of our PVT variation-aware power estimation methodology for on-chip communication architecture power estimation. Section 7 presents experimental results on CMP application that highlight the effectiveness and accuracy of our PVT-aware power estimation methodology at the system level. Finally, Section 8 presents the conclusion to this work.

2. RELATED WORK

System-level power estimation approaches typically create power models for heterogeneous system components (e.g., buses, memories, caches, processors) and integrate them to get overall power estimates [Lajolo et al. 2002; Dhanwada et al. 2005; Lee et al. 2006]. Several approaches have proposed power estimation techniques for bus-based communication architectures [Benini et al. 2000; Sotiriadis et al. 2002; Caldari et al. 2003; Bona et al. 2004; Pasricha et al. 2006]. While early work focused mainly on power estimation for bus wires [Benini et al. 2000; Sotiriadis et al. 2002], more recent work has shown the importance of considering bus logic components as well [Lahiri et al. 2004]. System-level power estimation approaches for communication architectures that consider the contribution of both bus logic and wires have been proposed for the AMBA hierarchical shared bus [Caldari et al. 2003], STBus interconnection network [Bona et al.2004], and the AMBA bus matrix [Pasricha et al. 2006]. None of the abovementioned power estimation approaches have studied the effects of PVT variability on power consumption at the system level. To the best of our knowledge, our work is the first to try and understand how PVT variability affects power consumption, especially for on-chip communication architectures, and then attempts to abstract this variability up to the system level, for early power exploration of the true design space.

3. PVT CORNERS IN ULTRA-DEEP SUBMICRON (UDSM) TECHNOLOGIES

Traditionally, the most important means by which a foundry communicates process, voltage, and temperature variations to designers is through library characterization at design corners, known as PVT corners, relating cell metrics (timing, power) to process, voltage, and temperature variations. Up until

Table I. PVT Corners in UDSM Technologies

Nominal V_{dd}	Corner	Process	Temp	V_{dd}
1.0V	MaxPerf	F-F	0	1.1
	TypPerf	T-T	25	1
	WorstPerf	S-S	125	0.9
	WorstLeakage	F-F	125	1.1
	TypLeakage	T-T	125	1
0.7V	MaxPerfLowV	F-F	0	0.77
	TypPerfLowV	T-T	25	0.7
	WorstPerfLowV	S-S	125	0.7
1.2V	MaxPerfHighV	F-F	0	1.32
	TypPerfHighV	T-T	25	1.2
	WorstPerfHighV	S-S	125	1.08
	WorstLeakageHighV	F-F	125	1.32
	TypLeakageHighV	T-T	125	1.2

the 130nm technology library node, design tools relied on three corners: *Typical*, *Worst*, and *Best* corners. The adjectives associated with these corners relate mainly to timing. The *Worst* corner combines high temperature, low V_{dd} (nominal- 10%), and a Slow-Slow (S-S) process that leads to worst-case timing. The *Best* corner goes the opposite way, combining low temperature, high V_{dd} (nominal+10%), and Fast-Fast (F-F) process to achieve maximum performance. The *Typical* performance corner lies between these two extremes. Synthesis tools currently use the *Worst* and *Best* corners during synthesis, guaranteeing that all the resulting functional chips meet timing (by using the *Worst*-case corner to avoid set-up time violations at register inputs, and using the *Best*-case corner to guarantee that no hold time violations occur). The *Typical* corner is usually used to characterize power consumption under nominal conditions. Since leakage was negligible up until the 130nm technology library node, the only factor affecting power consumption (mostly dynamic) was V_{dd}, and variations of about $\pm$20% were expected among the three corners.

With ultra-deep submicron (UDSM) technologies under 100nm gearing into production, some changes became necessary with corner characterization. This was due to a variety of factors, the most important being the drastic increase in the device leakage power. Other factors include IR drop as well as power management strategies such as DVS/DFS and voltage islands. Today, IPs, especially cell libraries, I/O and memories are thus characterized at many more PVT corners, shown in Table I. These corners become necessary for a variety of reasons: A *TypicalPerf* corner, for example, does not provide a realistic assessment of leakage power under typical conditions because while the application is running, a die would heat up to well above 25^oC. With temperature being an exponential factor in leakage, a more realistic *TypicalLeakage* corner must be considered with a Typical-Typical (T-T) process, nominal V_{dd} and 125^oC. A *WorstLeakage* corner is used to assess the absolute maximum leakage under Fast-Fast (F-F) process (i.e., low V_t) and high V_{dd} (Nominal+10%).

Power management strategies such as voltage islands and discrete voltage scaling (DVS) cannot be validated at the chip level unless IPs are characterized under several low V_{dd} conditions. This requires another set of corners. IPs such as the Metro libraries from Artisan (ARM) [Metro IP 2008] are characterized

for V_{dd} increments of 100mV for a range of possible V_{dd} values. *MaxPerfLowV*, *TypPerfLowV*, and *WorstPerfLowV* are needed for each V_{dd}. For 90nm, the lowest safe V_{dd} is 0.7V. On the other hand, and under certain conditions more performance may be needed. For that case, some cell libraries are characterized for higher than normal operating conditions. In the case of 90nm, IPs can operate up to 1.2V nominal V_{dd}. Thus, another set of corners are needed including *MaxPerfHighV*, *TypPerfHighV*, and *WorstPerfHighV*. Since leakage can be significantly higher under those conditions, additional *TypicalLeakageHighV* and *WorstLeakageHighV* are sometimes available in order to assess typical and worst-case leakage under high V_{dd} conditions. Note that the PVT corners shown in Table I do not constitute a maximal set. Many more corners can be added during library characterization to support more elaborate design methodologies and possible operating environments and conditions. Alternatively, some technology libraries may not support *HighV* corners for reliability reasons. The existence of a multitude of these corners motivates the need to mitigate the complexity and achieve more reliable designs by understanding and incorporating PVT effects early in a design flow, at the system level.

In the case of timing analysis, we may argue that corner characterization is of limited use. However, we note that while timing analysis concentrates on critical path characterization, power characterization introduces significantly more degrees of freedom to the analysis, such as data dependence, power management, and so on, all of which are very hard to incorporate into an amenable statistical analysis, especially at the system level. Thus, employing corner-based analysis with a larger and more realistic corner set helps reduce the complexity of the designers' task in exploring the design space, albeit at the cost of perhaps slightly more pessimistic assumptions.

4. IMPACT OF PVT VARIABILITY ON POWER CONSUMPTION

We will now present some experimental results to show how PVT variability affects power consumption in ultra-deep submicron technology nodes for the on-chip communication architecture (Section 4.1); the entire SoC design (Section 4.2); and a multi-Vt design flow (Section 4.3). Unlike the technology nodes up to the 130nm node, the ultra-deep submicron technology nodes such as the 90nm and 65nm have many more PVT corners that need to be explored during the design phase to ensure that power constraints in a design are satisfied. The experiments in this section explore power consumption characteristics for these different PVT corners, for different ultra-deep submicron technology nodes, designs, and operating frequencies.

4.1 Impact on the On-Chip Communication Architecture

Our first set of experiments was conducted with the aim of understanding the impact of PVT variation on power consumption of the on-chip communication architecture. In the first experiment, we selected four SoC designs of varying complexity. Each of the designs had an AMBA AHB bus matrix [AMBA AHB Interconnection Matrix 2008] communication architecture configuration with a different structure and traffic characteristics: (i) a 2 master, 3 slave bus matrix with 32-bit data bus width (2×3); (ii) a 3 master, 4 slave bus matrix with

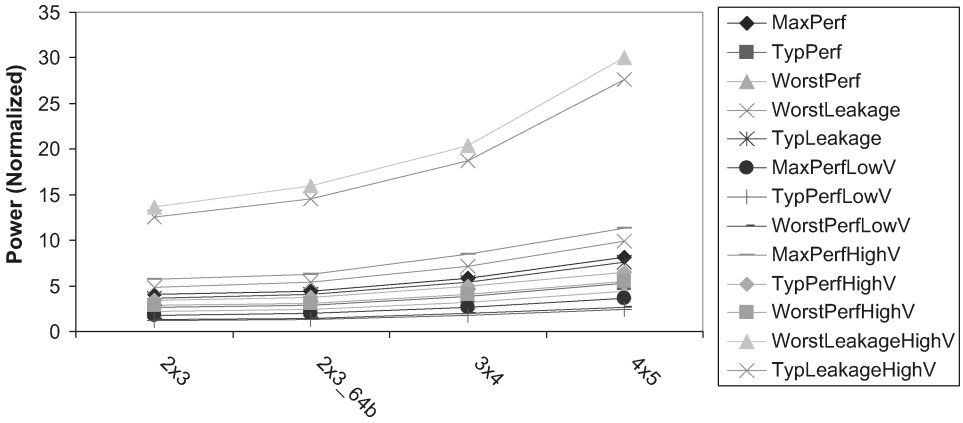


Fig. 2. Normalized power for bus matrix configurations at 90nm.

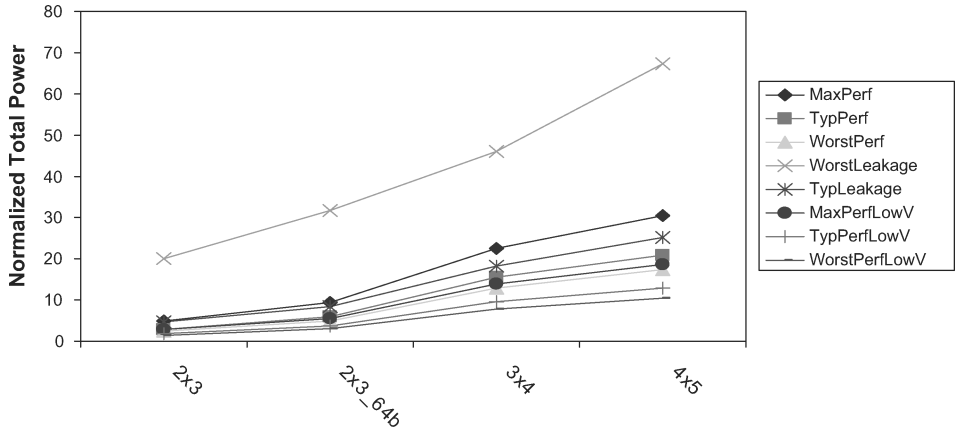


Fig. 3. Normalized power for bus matrix configurations at 65nm.

32-bit data bus width (3×4); (iii) a 4 master, 5 slave bus matrix with 32-bit data bus width (4×5)-and (iv) a 2 master, 3 slave bus matrix with 64-bit data width ($2 \times 3\_64b$). We targeted the 90nm and 65nm general-purpose technology libraries, synthesized these designs for a 100 MHz bus clock frequency, and estimated power for the different PVT corners shown in Table I using Synopsys PrimeTime PX [Synopsys CoreTools 2008] at the gate level.

The normalized power of the bus matrix communication architecture for the different PVT corners is shown for all four bus matrix configurations in Figure 2 (90nm) and Figure 3 (65nm). It can be seen from the figures that there is significant variability in estimated power for 90nm and 65nm libraries, especially between the *WorstLeakageHighV* and *TypPerfLowV* corners (more than a $10\times$ difference). As mentioned earlier, just considering the traditionally used corners (e.g., *TypPerf* vs. *TypLeakage*) is not realistic because there is a large variation in power consumption for sub-100nm libraries due to DSM effects (Section 3), which can only be captured by additional corners. Thus, in order to meet power

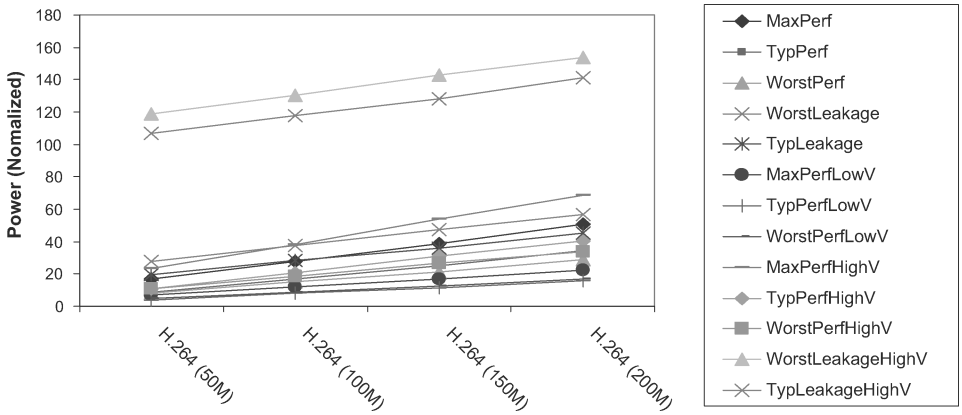


Fig. 4. Normalized power for H.264 SoC subsystem.

goals, designers need to consider multiple PVT corners to understand the power characteristics of a design. *It can be concluded from our experimental results that there is a significant (as much as* 10×*) variation in power consumption across PVT corners, for on-chip communication architectures.*

4.2 Impact on SoC Design

In our next set of experiment, we were interested in investigating the impact of PVT corners on power consumption not just for the communication architecture, but for an entire SoC design. We were also interested in estimating the impact of clock frequency on power consumption for the corners. For the purpose of this experiment, we implemented a complex SoC subsystem for the H.264/AVC codec [JVT ISO/IEC MPEG, ITU-T VCEG 2008] at the RTL level, consisting of the chroma inter-prediction IP, buffers, and several memory blocks interconnected using the AMBA AHB bus matrix [AMBA AHB Interconnection Matrix 2008]. Figure 4 illustrates normalized total power of the synthesized H.264 subsystem, obtained after detailed gate-level power analysis [Synopsys CoreTools 2008] for a 90nm technology library implementation, with clock frequency ranging from 50 MHz to 200 MHz. A near linear increase in power is observed with frequency, which implies that power increases by an approximately constant ratio due to clock scaling. The impact of leakage power can be observed by considering the difference in power (almost 2×) between *TypPerf* and *TypLeakage* corners under the same V_{dd}. *From our experimental result, it can be concluded that there is significant variation (as much as* 60×*) in power consumption across PVT corners, for SoC designs.*

4.3 Impact on Multi-Vt Design Flow

The multi-Vt technique is an effective way to reduce subthreshold leakage current without sacrificing performance. High-Vt libraries can be used to reduce leakage current while low-Vt libraries can be used to get high performance on critical paths. We conducted several experiments for the 90nm and 65nm libraries, where we performed multi-Vt synthesis for different configurations

of the AMBA AHB bus matrix communication architecture running at various bus clock frequencies.

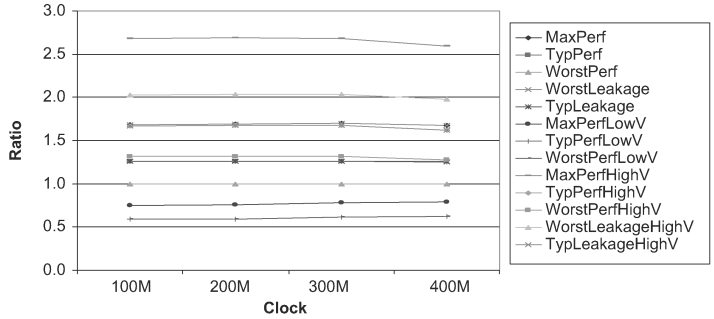
Figure 5 shows the total power ratio for a 2×3 bus matrix across corners for the Single-Vt and Multi-Vt cases. Approximately fixed scaling factors for each corner are again observed. Synthesis of the bus matrix configurations across different frequencies ranging from 100-400 MHz demonstrates some interesting results, as shown in the figure. First, as expected, we observe large leakage power saving through the Multi-Vt technique. Second, there is only slight distortion at higher frequencies for the Single-Vt synthesis cases (shown in (a) and (c) of Figure 5). However, many more distortions in the power ratio are noticeable at high frequencies for multi-Vt cases (shown in (b) and (d) of Figure 5). The reason for these distortions is that additional low-Vt libraries were required to meet the more stringent timing requirements at higher frequencies in multi-Vt synthesis, unlike the single-Vt synthesis case. *Our experimental results indicate significant variations in power consumption across PVT corners (as much as* $27\times$ *for dynamic power and* $80\times$ *for leakage power) for multi-Vt implementations of the on-chip communication architecture.*

5. SCALING RELATION FOR POWER ESTIMATION ACROSS PVT CORNERS

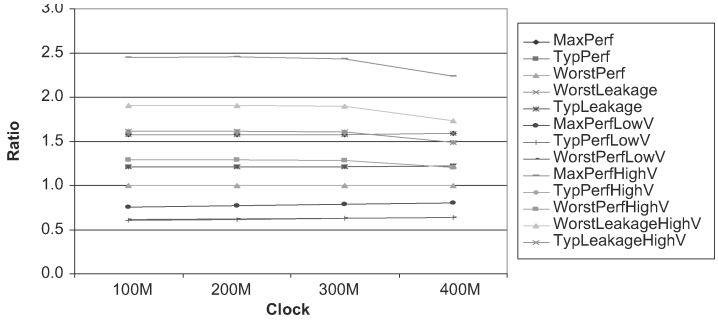
From the results of the previous experiments, it can be seen that power for PVT corners increases at a certain ratio. Figure 6 compares the dynamic and leakage power for the corners for different design configurations: the AMBA bus matrix and the H.264 subsystem, operating at different clock frequencies. The dynamic power for the corners, normalized to the *WorsePerf* corner, is shown in Figure 6 (a). An almost constant ratio is observed for the different design configurations, even as the clock frequency is varied. Figure 6 (b) illustrates a similar, near-constant ratio for leakage power (the results are again normalized to the *WorsePerf* corner).

The analysis of the results above indicates that (i) PVT corner power ratios have a large dynamic range (about 4 orders of magnitude); (ii) corner power ratios for the dynamic component are fairly constant across different designs and design configurations, and these ratios are within about 10% of the V_{dd}^2 ratios of the corners; and (iii) corner power ratios for the leakage component may vary by as much as 22% across designs. Note that variations in the same design at different clock frequencies are due to the logic synthesis tool increasing the number of logic gates, and consequently the component area, in order to meet timing constraints, as shown in Figure 7. This artifact is especially noticeable at higher clock frequencies, where the additional logic required to meet the stricter timing constraints leads to an increase in leakage power.

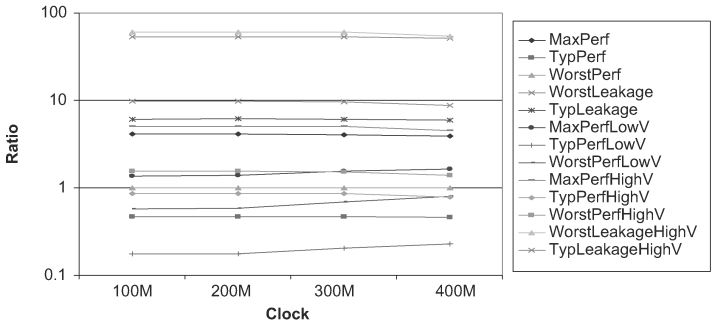
From the results of these experiments, we find that the power consumption for a PVT corner scales almost linearly with frequency. However, a much more interesting and important observation is that the power consumption numbers obtained for the different PVT corners show an almost constant ratio relative to each other. Thus, if the power consumption of the bus matrix on-chip communication architecture for an implementation with PVT corner *C1* is expressed



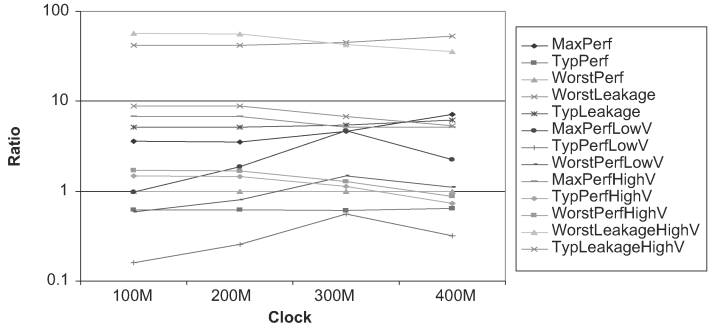
(a) Dynamic power, Single-Vt.



(b) Dynamic power, Dual-Vt.

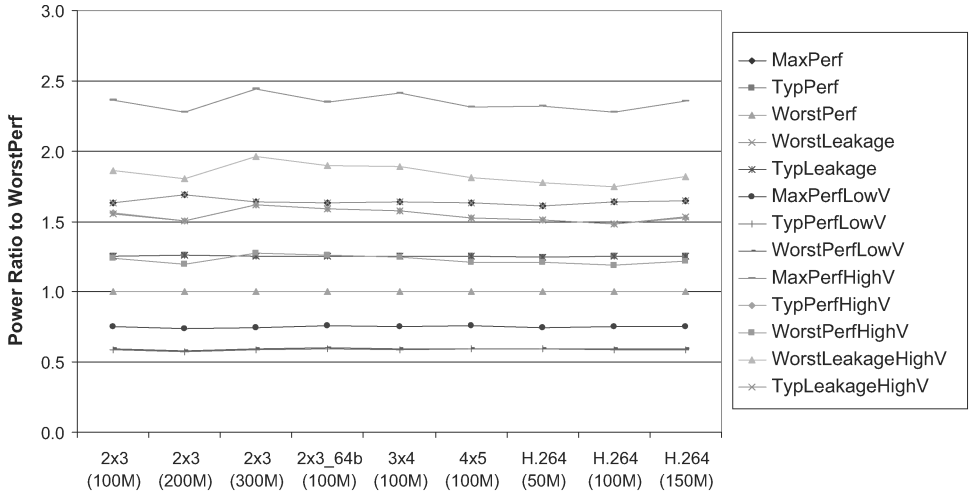


(c) Leakage power, Single-Vt.

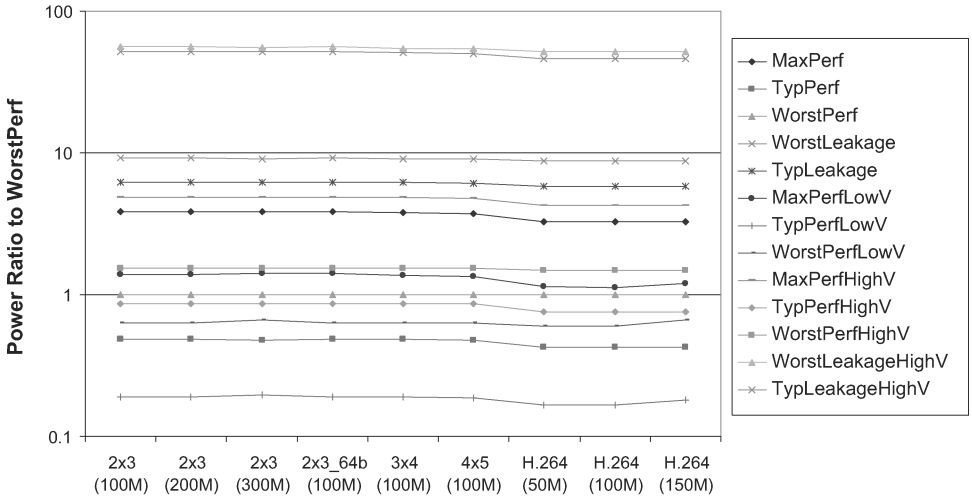


(d) Leakage power, Dual-Vt.

Fig. 5. Power ratio for 2 × 3 bus matrix.



(a) Normalized dynamic power



(b) Normalized leakage power

Fig. 6. Power ratios for the AMBA bus matrix and the H.264 subsystem, operating at different clock frequencies.

as

$$P_{C1} = P_{L1} + P_{D1} \times f \tag{1}$$

where P_{C1} gives the base-level total power for a corner *C1* that has base-level leakage power P_{L1} and base-level dynamic power P_{D1} at frequency f, then the power consumption for an implementation under any other PVT corner $C2$ can be expressed as

$$P_{C2} = \alpha_{1-2} \times P_{L1} + \beta_{1-2} \times P_{D1} \times f \tag{2}$$

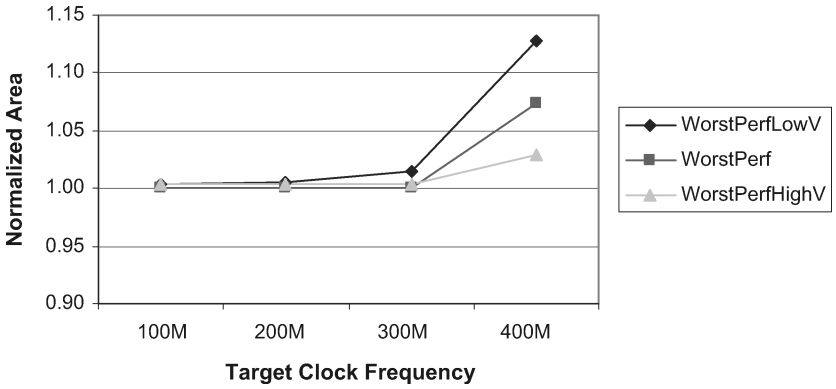
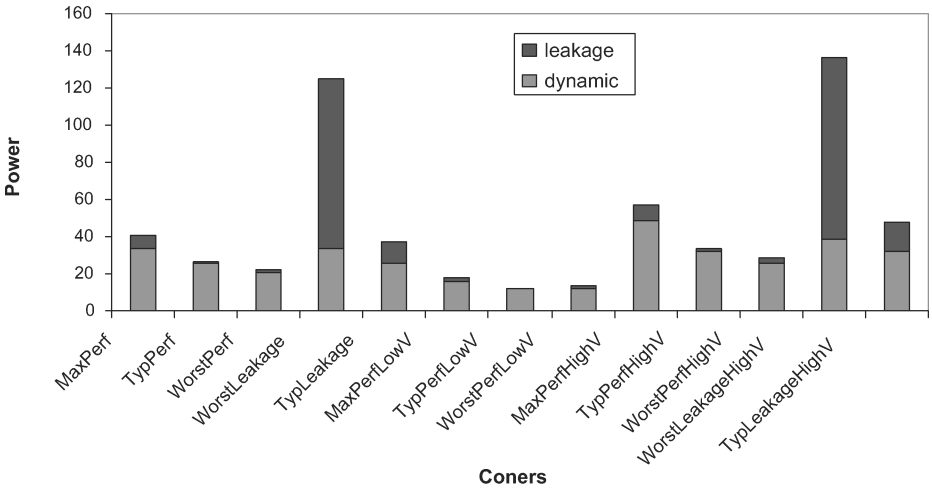


Fig. 7. Normalized area of 2 × 3 bus matrix with varying target clock frequency, for different corners.

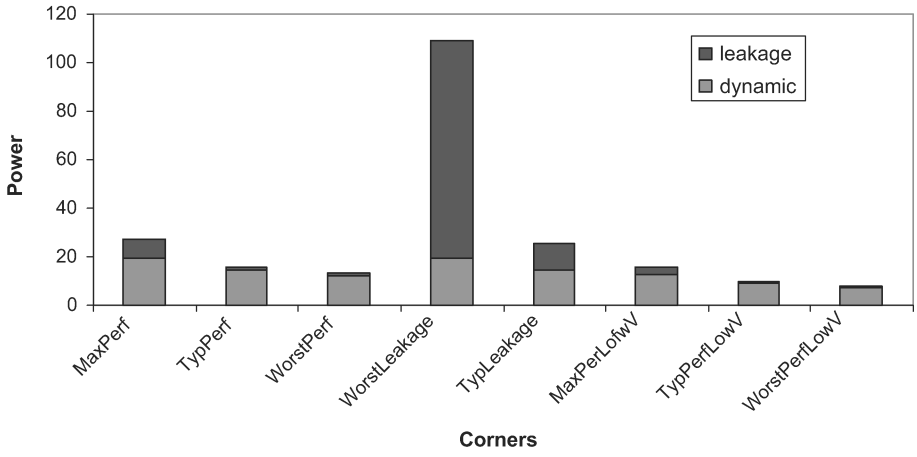
where the total power P_{C2} for another corner *C2* can be linearly scaled from the base-level power relation in Eq. (1) by using scaling factors α_{1-2} and β_{1-2} for leakage and dynamic power, respectively. The scaling factors can be easily obtained by decomposing the total power into dynamic and leakage components (as shown in Figure 8), and averaging the ratio values. *Thus, knowing the leakage and dynamic power for one PVT corner can enable us to obtain the power for other PVT corners using a simple linear model with good accuracy, which considerably speeds up power exploration across corners.*

It is important to differentiate the proposed linear corner-to-corner scaling proposed in (2) from the power (specifically leakage) dependence on P, V, and T. While it is well known that a super-linear relation exists between leakage and each of those P, V, and T parameters [Rabaey et al. 2003], what (2) reflects is an observation that, outside of frequency, which affects dynamic power, the ratio of power (dynamic+leakage) shows little variation *across design instances.* In other words, once a design implementation is characterized for power across the PVT corners (say through detailed gate-level simulation), the corner ratios obtained can be reused to "scale" another design instance without having to do another full characterization run for that instance.

Figure 9(a) and (b) show the maximum estimation error for the leakage and dynamic power for the 90nm and 65nm libraries, respectively, when the scaling factors are used to estimate power consumption for different PVT corners for different design configurations: the AMBA AHB bus matrix communication architecture, the H.264 SoC subsystem and a register file, operating at different clock frequencies. It can be seen from the figures that by using the scaling technique we propose, it is possible to estimate power consumption for different PVT corners with extremely good accuracy for dynamic power ($<$5% in most cases), and fairly good accuracy for leakage power ($<$10% in most cases). This amenability to scaling for PVT corners is an extremely important result, and has been obtained for cell libraries characterized with industrial-strength numbers.



(a) 90nm case

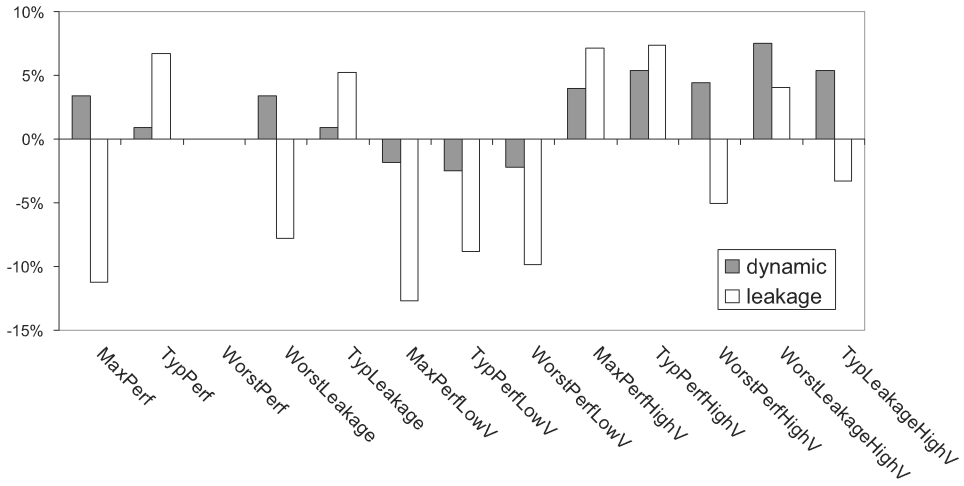


(b) 65nm case

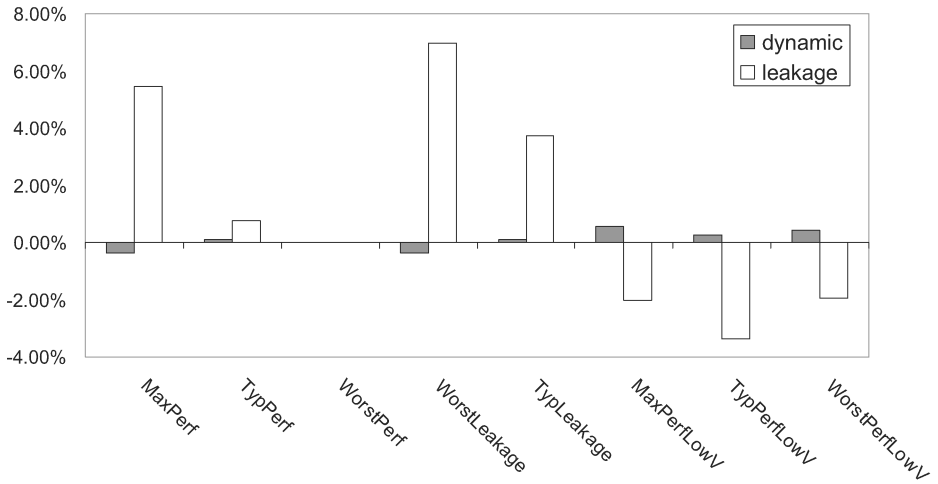
Fig. 8. Decomposed power for 2×3 bus matrix configurations at 100MHz.

6. PVT-AWARE SYSTEM-LEVEL POWER MODELING OF ON-CHIP COMMUNICATION ARCHITECTURES

In this section we present our approach to abstract the power estimation of on-chip communication architectures across PVT corners early in the design flow, up to the system level. We use the AMBA AHB bus matrix on-chip communication architecture as an exemplar, to demonstrate our approach. Section 6.1 presents a very brief overview of energy macro-modeling, and Section 6.2 describes how to create energy macro-models for system-level power estimation for the bus matrix communication architecture. Section 6.3 then describes how we can extend this methodology to enable PVT-aware system-level power



(a) 90nm case.



(b) 65nm case.

Fig. 9. Normalized power estimation error.

estimation for the AMBA AHB bus matrix communication architecture by using the results of our observations from the previous sections.

6.1 Overview of Energy Macro-Models

The energy consumption of a bus matrix can be obtained by identifying events that cause a noticeable change in its energy profile. For this purpose, we create energy macro-models that can encapsulate events or factors having a strong correlation to energy consumption for a given component. A macro model consists

of variables that represent factors influencing energy consumption, and of regression coefficients that capture the correlation of each of the variables with energy consumption. A general energy macro model for a component can be expressed as

$$E_{component} = \alpha_0 + \sum_{i=1}^{n} \alpha_i \cdot \Psi_i \tag{3}$$

where α_0 is the energy of the component that is independent of the model variables, and α_i is the regression coefficient for the model variable Ψ_i . Note that we consider a linear energy model even though quadratic models can theoretically provide higher accuracy. The motivation for choosing a linear model is that if the linear model can provide us with high estimation accuracy, there is no need to consider more complex quadratic models.

For the purpose of our energy macro-models, we considered three types of model variables representing factors influencing energy consumption: *control*, *data*, and *structural*. The *control* factor represents control events, involving a control signal that triggers energy consumption either when it transitions from 1 to 0 or 0 to 1, or when it maintains a value of 0 or 1 for a cycle. Control variables can either have a value of 1 when a control event occurs, or 0 when no event occurs, in the energy macro model relation in Eq. (3). The *data* factor represents data events that trigger energy consumption on data value changes. Data variables take an integer value in Eq. (3) representing the Hamming distance (number of bit-flips) of successive data inputs [Caldari et al. 2003]. Finally, *structural* factors, such as data bus widths and number of components connected to the input also affect the energy consumption of a component; they are represented by their integer values in Eq. (1).

6.2 System-Level Energy Macro-Model Creation Methodology

To estimate power for the bus matrix communication architecture early in the design flow, at the system level, we make use of the power estimation methodology for the bus matrix proposed by us in Pasricha et al. [2006]. A high-level overview of this methodology is shown in Figure 10. We start with a system testbench, consisting of masters and slaves interconnected using the AMBA AHB bus matrix fabric. The testbench generates traffic patterns consisting of single and burst transactions of varying sizes and different modes (e.g., SPLIT/RETRY, locked bus) that exercise the matrix under different operating conditions. Synopsys Coretools [Synopsys CoreTools 2008] is used to configure the bus matrix (specify data bus width, number of masters and slaves, etc.) and generate a synthesizable RTL description of the bus matrix architecture (Step 1). This description is synthesized to the gate level with the Cadence Physically Knowledgeable Synthesis (PKS) tool [Cadence PKS 2008], for the target standard cell library (Step 2). PKS preplaces cells and derives accurate wire-length estimates during logic synthesis. In addition, it generates a clock tree, including clock de-skewing buffers. The gate-level netlist is then used with Synopsys PrimePower [Synopsys CoreTools 2008] to generate power numbers (Step 3).

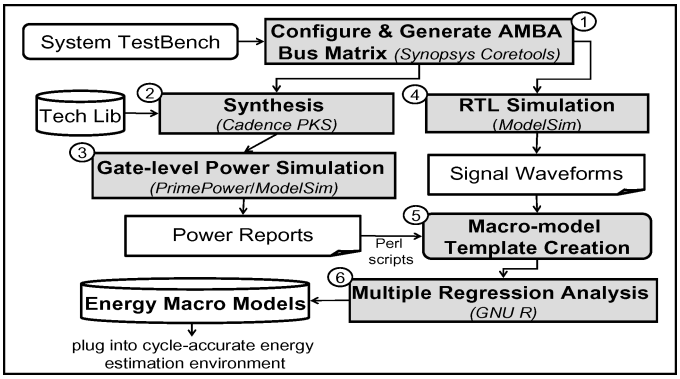


Fig. 10. Methodology for energy macro-model creation.

In parallel with the synthesis flow, we perform RTL simulation to generate signal waveform traces for data and control signals of the bus matrix (Step 4). These signal waveforms are compared with cycle energy numbers, obtained after processing PrimePower-generated power report files with Perl scripts, to determine which data and control signals in the matrix have a noticeable effect on its energy consumption. For our analysis, we consider signals that change in value when an increase in bus matrix energy consumption of at least 0.01% is observed over the base case (i.e., no data traffic). Note that a finer-grained selection criteria can be chosen (e.g., 0.001%), which would result in even more accuracy, but at the cost of more complex macro models that take longer to create and evaluate. The selected data and control events become the variables in a macro-model template that consists of energy and variable values for each cycle of testbench execution (Step 5). This template is used as an input to the GNU R tool [GNU R 2008] that performs multiple linear regression analysis to find coefficient values for the chosen variables (Step 6). Steps 1 to 6 are repeated for testbenches having different structural attributes, such as data bus widths and number of masters and slaves, to identify structural factors (variables) that may influence cycle energy. Statistical coefficients such as *Multiple-R*, *R-square* and *standard deviation for residuals* [Faraway 2004] are used to determine the goodness of fit and the strength of the correlation between cycle energy and model variables.

Once a good fit between cycle energy and macro model variables is found, the energy macro models are generated in the final step. These models can then be plugged into any system-level cycle-accurate or cycle-approximate simulation environment to get energy consumption values for the AMBA AHB bus matrix communication architecture. A high-level simulated annealing floorplanner [Adya et al. 2003] is used for early core placement and Manhattan routing is used to determine wire lengths. The wire lengths are subsequently used to determine wire energy in formulations proposed in Kretzschmar et al. [2004], which we extended to incorporate power for delay-optimally-inserted repeaters. The power estimates for the bus matrix communication architecture were shown to be within 5% accuracy of gate-level power estimates in Pasricha et al. [2006].

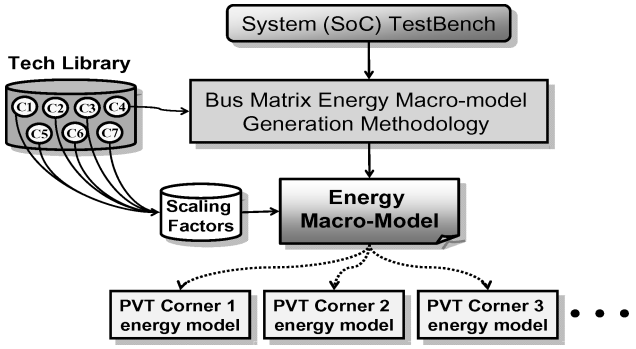


Fig. 11. System-level energy macro-model generation for PVT corners.

6.3 Incorporating PVT Corners in a System-Level Power Estimation Methodology

Our approach from Pasricha et al. [2006] makes use of energy macro-models to determine power consumption for the bus matrix logic components such as the input buffer stages, decoders, arbiters, and output stages. Creating the energy macro-models for the bus matrix, however, requires a one-time effort to identify macro-model variables and coefficients using multiple linear regression analysis [Faraway 2004], which can take several hours. Since a different energy macro-model is required for every PVT corner, the task can take a long time (∼several days to weeks).

Figure 11 shows our proposed methodology to speed up bus matrix energy macro-model creation for different PVT corners of a technology library. This approach extends our previous work on energy macro-model-based power estimation [Pasricha et al. 2006]. Initially, a system testbench consisting of a diverse set of bus matrix-based SoC designs is used to generate an energy macro-model for one of the PVT corners of a technology library, according to Pasricha et al. [2006]. Subsequently, scaling relations presented in Section 5 are used for the other PVT corners to modify the base energy macro-model and create macro-models for each of the other PVT corners. This enables a considerable saving in time because only one macro-model generation iteration needs to be performed in order to obtain the energy models for all the corners in the selected technology library. *Since a single macro-model iteration can take an the order of hours, this approach can save us in the order of days to estimate power for the various PVT corners in ultra-deep submicron technology library nodes.* In the next section, we present experiments to show how accurately these PVT corner macro-models can estimate power at the system level for the bus matrix communication architecture.

7. EXPERIMENTAL RESULTS

To verify whether the scaling factor-based PVT corner power estimation approach can be used to accurately and efficiently explore power consumption across different corners for the bus matrix communication architecture, we performed experimental studies at the system level. We selected four proprietary industrial-strength chip multiprocessor (CMP) networking SoC applications

Table II. Networking CMP Applications

CMP application	# of processors	# of masters	# of slaves
PSWITCH1	4	5	24
DFILTR2	6	8	33
DPROC	9	10	41
MNROUTE	10	12	45

used for data packet processing and forwarding, and modeled the SoC applications in SystemC [SystemC 2008] at the Transaction-based Bus Cycle Accurate (T-BCA) [Pasricha 2002; Pasricha et al. 2004] abstraction. Table II gives an overview of the characteristics of the applications considered, including the number of programmable processors, masters, and slaves (memories, peripherals) in the implementations of each of the applications.

7.1 Estimation Error and Speedup Analysis over Traditional Approach

The goal of our first set of experiments was to estimate power for the PVT corners at the system level using two methods: first, with the traditional approach of creating energy macro-models for each PVT corner separately; and second, with the proposed scaling-based approach. The results of all the power exploration experiments conducted at the system level were verified by detailed gate-level simulation.

In the remainder of this section, we focus on detailed experimental results for the PSWITCH1 application (we summarize the results for the rest of the applications at the end of the section). The PSWITCH1 implementation, shown in Figure 12, consists of 4 ARM processors, a DMA engine, and 24 slaves, which include memories, network interfaces, peripherals, and ASIC components, interconnected using the AMBA AHB bus matrix. We used our communication architecture synthesis framework from Pasricha et al. [2006] to generate a set of bus matrix solutions (each solution having a different number of buses) that satisfies the performance constraints of the PSWITCH1 application. Next, we created bus matrix energy macro-models for each of the different PVT corners of the 90nm general-purpose technology library. We plugged the energy macro-models into a T-BCA simulation model of the networking SoC, and simulated the design for each of the solutions in the solution set to get power consumption for the application. Figures 13 and 14 show the normalized power and energy obtained after simulating the design for all the PVT corners for each of the solutions. As mentioned earlier, these numbers are within 5% accuracy of gate-level power estimates. The X-axis shows different bus matrix solutions, each having a different number of buses (and consequently different number of logic components such as arbiters, decoders, etc.).

From the figures, it can again be seen that there is a significant variation in power and energy consumption across different PVT corners, and the power and energy ratios for different corners are fairly constant. Table III shows the percentage change in performance for the different solutions with fewer buses, compared to the solution with 42 buses. It can be seen that solutions with a lesser number of buses have lower performance. This is because there are more delays due to traffic conflicts when data streams originating from different

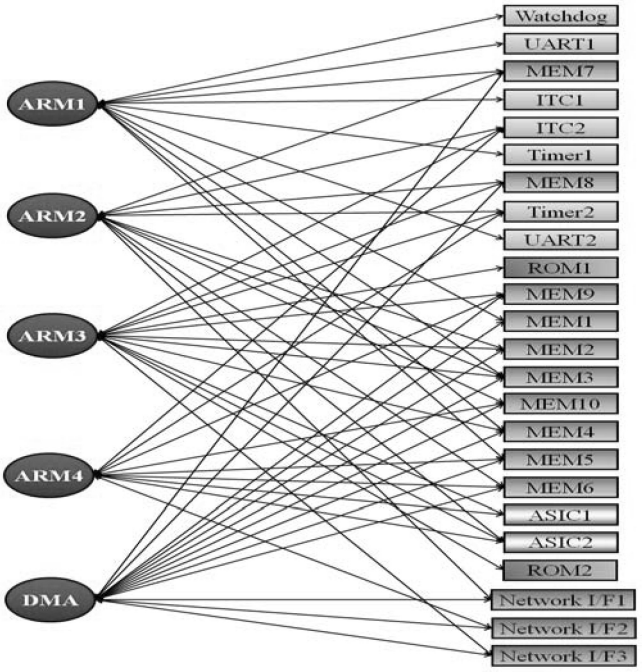


Fig. 12. PSWITCH1 CMP application.

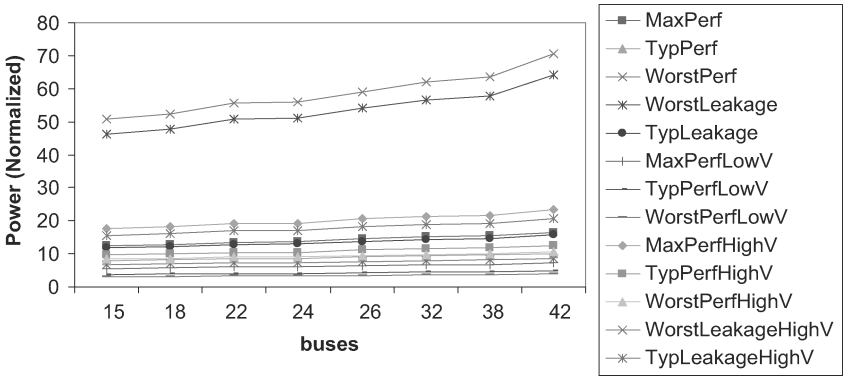


Fig. 13. Normalized power for PSWITCH1.

masters must share fewer buses. *Note that the performance numbers in Table* III *represent a lower bound on performance that can be achieved for all the PVT corners.*

Creating energy macro-models for each PVT corner case turns out to be very time-consuming, requiring a few days in designer effort. There is a need to speed this process up. Clearly, one solution is to use the scaling relations from Section 5 and the methodology in Figure 11. The question is, *can this approach be used at the system level to accurately estimate power for PVT corners?* To answer this, we selected one of the PVT corners (*WorstPerf*) as a base reference and scaled

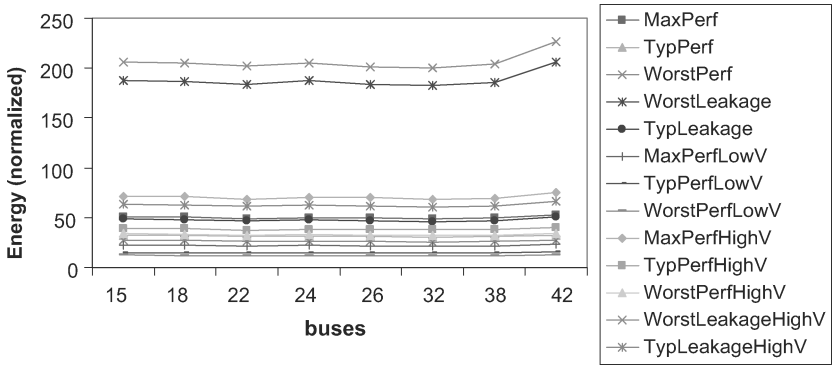


Fig. 14. Normalized energy for PSWITCH1.

Table III. % performance variation for bus matrix configurations for pswitch1

# buses	15	18	22	24	26	32	38	42
% perf. variation	−26.5	−21.9	−12.6	−14	−6.1	−0.7	−0.2	0

its power results to create energ macro-models for the other PVT corners. We then plugged these energy macro-models into our T-BCA simulation model, and simulated the design for each solution in the bus matrix solution set to obtain power numbers.

Figure 15 shows the error in power estimation for the solutions obtained for PSWITCH1, when we used the scaling technique to estimate power at PVT corners, compared to the traditional technique of creating energy macro-models separately for the corners (as in the previous experiment). It can be seen that the maximum absolute cycle error compared to macro-model estimates is less than 9%. *The maximum absolute cycle error compared to gate-level estimates is less than 14%* (since the macro-models are within 5% accuracy of gate-level estimates [Pasricha et al. 2006]), *which is an extremely good accuracy for PVT corner power estimation at the system level.* The results imply that we only need to create an energy macro-model for one of the PVT corners and use scaling factors to quickly obtain power for other PVT corners at the system level. The overall time taken for creating energy macro-models for all the PVT corners from the base reference macro-model in this case is in the order of a few seconds, and *several orders of magnitude less* than the case where macro-models have to be separately created for every PVT corner. More specifically, the speedup over the traditional technique is approximately $N\times$ where N is the number of PVT corners in a technology library for which power must be estimated. For the 90nm technology library with 13 corners, we verified an estimation speed-up of approximately $13\times$ for each of the four CMP applications, using our scaling-based approach.

7.2 Estimation Error and Speedup Analysis Over Gate Level

Designers today typically perform power estimation for PVT corners at the gate level. We compared our system-level scaling-based power estimation technique

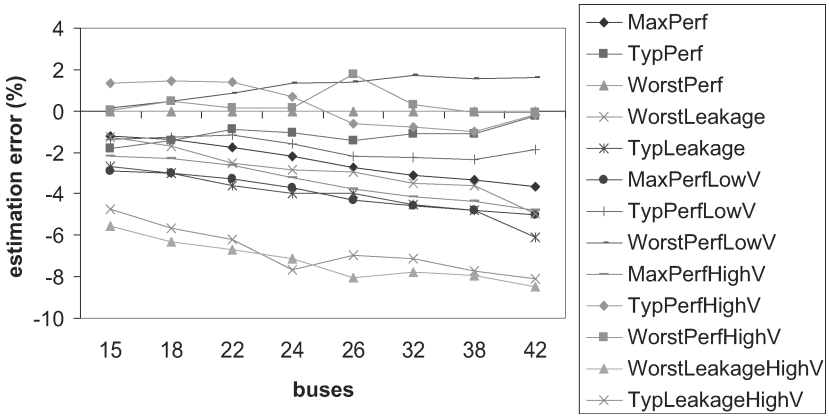


Fig. 15. % power estimation error for PSWITCH1 corner cases when Eq. (2) is used to obtain power at PVT corners.

Table IV. Comparison between System Level and Gate Level Estimation

CMP application	# Max. Absolute Cycle Error	Speedup (X times)
PSWITCH1	13.8	18031
DFILT2	11.4	15652
DPROC	10.7	22672
MNROUTE	13.1	25376

with Synopsys PrimePower-based gate-level power estimation for the AMBA AHB bus matrix communication architecture. The power estimation was performed for all of the PVT corners of the 90nm target technology library. Table IV summarizes the maximum absolute estimation error and speedup using our PVT-variation-aware system-level estimation methodology, compared to gate-level estimation for the four CMP applications. It can be seen that the system-level scaling-based power estimation methodology results in significant power estimation speedup over gate-level power estimation for different PVT corners. This speedup comes at the cost of a slight estimation error, which is quite acceptable for early system-level power estimation.

7.3 Importance of System-level PVT-Variation-Aware Exploration

In the previous sections, we illustrated how our scaling-based system-level power estimation approach can accurately and quickly determine power/energy for the different PVT corners. In this section, we discuss the importance of having such a PVT-variation-aware power estimation methodology available to designers for a more comprehensive early system-level exploration.

Consider again the PSWITCH1 CMP application. Figures 16 and 17 show the percentage change in power and energy for the solutions in the bus matrix solution set, compared to the solution with the most number of buses (42). As explained earlier, the solutions with the fewer number of buses have lower power and energy dissipation at the cost of performance (Table III). There is a large variation in energy and power consumption across the different PVT corners

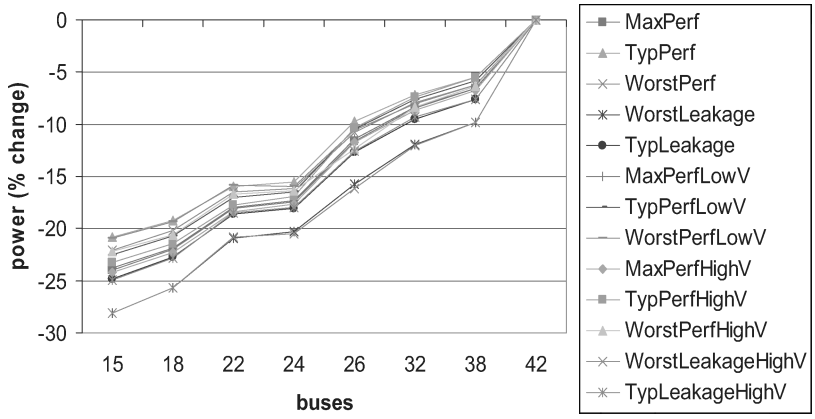


Fig. 16. % change in power for PSWITCH1.

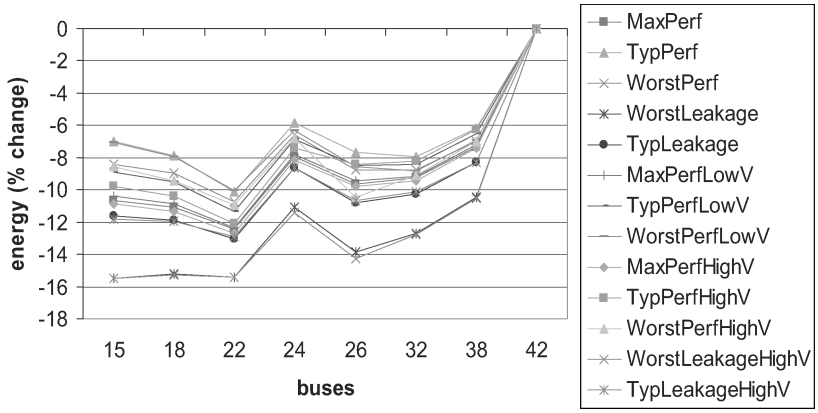


Fig. 17. % change in energy for PSWITCH1.

for the solutions. We present two exploratory scenarios where our framework would be invaluable.

***Exploration Scenario* 1.** Consider the case where the average power dissipation constraint for the design is 80mW. Assume that average power dissipation for the solution with 42 buses is 100 mW. In such a case, from Figure 16, it can be seen that the solution with 15 buses meets the constraint for all corners (i.e., % power reduction compared to 42 bus case > 20% for all corners). The solution with 18 buses meets the constraint for all corners except the *TypPerf* and *WorstPerfLowV* corners. The solutions with 22 and 24 buses meet the constraint only for the *WorstLeakage* and *WorstLeakageHighV* corners. If the designer used traditional techniques, he or she would have only a single power number for a technology library to rely upon during early exploration; now the designer can select a solution based on a more comprehensive characterization of the technology library across various PVT corners. The designer in this case can select the 15 bus solution conservatively, or go for the better performing

solution with 18 buses (refer to Table III), with a reasonable degree of confidence that the solution will not violate constraints under most PVT conditions.

***Exploration Scenario* 2.** Consider another case where a solution with minimum energy dissipation needs to be selected. From Figure 17 it can be seen that under maximum leakage conditions the lowest energy is obtained for the solution with 15 buses. However, if such maximum leakage conditions are not encountered, then the lowest energy dissipation (for the other corners) is obtained for the solution with 22 buses. Such PVT-aware exploration information can aid the designer in selecting the appropriate solution with greater confidence and more accuracy than the traditional approach of considering only a single corner for a technology library during early system-level power exploration.

8. CONCLUSION AND FUTURE WORK

In this article, we investigated the impact of PVT corners on power consumption at the system level, especially for the bus matrix on-chip communication architecture. We first conducted several experiments to show how there are significant variations in power consumption across different corners for a given technology library. Next we showed how it is possible to "scale up" and abstract the PVT variability to characterize the true design space early on in the design flow, at the system level. We used scaling relations to quickly create power models for the different PVT corners, to estimate the power consumption of the bus matrix at the system level. The scaled power models took several orders of magnitude less time to create than the traditional macro-modeling and gate-level estimation techniques at the cost of a slight estimation error, which is quite acceptable for early system-level power estimation. Finally, we experimentally established the importance of considering PVT corners during system-level power exploration. While we currently do not consider intra-die PVT variations, we believe that the simplicity of the abstractions described in this article will make it feasible to incorporate such variability into future work. Additionally, our PVT-aware power estimation methodology can be used in conjunction with recent works on process variation-aware bus encoding [Raghunandan et al. 2008; Tuuna et al. 2008] that attempt to optimize communication in the face of process variations. In particular, we believe our methodology can help accurate characterization of techniques for robust process-variation-aware power-optimal encoding schemes.

REFERENCES

ADYA, S. N. AND MARKOV, I. L. 2003. Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Trans. VLSI Syst. 11*, 6, 1120–1135.

AMBA AHB Interconnection Matrix. 2008. http://www.synopsys.com/products/designware/amba_solutions.html.

BANERJEE, K. AND MEHROTRA, A. 2001. Coupled analysis of electromigration reliability and performance in ULSI signal nets. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 158–164.

BENINI, L., MACII, A., PONCINO, M., AND SCARSI, R. 2000. Architectures and synthesis algorithms for power-efficient bus interfaces. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst. 19*, 9, 969–980.

BONA, A., ZACCARIA, V., AND ZAFALON, R. 2004. System level power modeling and simulation of high-end industrial network-on-chip. In *Proceedings of the Conference on Design, Automation and Test in Europe.*

BORKAR, S., KARNIK, T., AND DE, V. 2004. Design and reliability challenges in nanometer technologies. In *Proceedings of the* 41st *Design Automation Conference*. 75.

CADENCE PKS. 2008. www.cadence.com/datasheets/pks_ds.pdf.

CALDARI, M., CONTI, M., COPPOLA, M., CRIPPA, P., ORCIONI, S., PIERALISI, L., AND TURCHETTI, C. 2003. System-level power analysis methodology applied to the AMBA AHB bus. In *Proceedings of the Conference on Design, Automation and Test in Europe: Designers' Forum*.

DHANWADA, N., LIN, I., AND NARAYANAN, V. 2005. A power estimation methodology for system C transaction level models. In *Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*.

FARAWAY, J. J. 2004. *Linear Models with R*. CRC Press.

GNU R. 2008. http://www.gnu.org/software/r/R.html.

HO, R., MAI, K. W., AND HOROWITZ, M. A. 2001. The future of wires. *Proc. IEEE 89*, 4, 490–504.

JVT ISO/IEC MPEG, ITU-T VCEG. 2003. Final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC). JVT-G050.

KHANDELWAL, V. AND SRIVASTAVA, A. 2005. A general framework for accurate statistical timing analysis considering correlations. In *Proceedings of the* 42*nd Design Automation Conference*. 89–94.

KRETZSCHMAR, C., NIEUWLAND, A. K., AND MÜLLER, D. 2004. Why transition coding for power minimization of on-chip buses does not work. In *Proceedings of the Conference on Design, Automation and Test in Europe.*

LAHIRI, K. AND RAGHUNATHAN, A. 2004. Power analysis of system-level on-chip communication architectures. In *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis (CODES + ISSS)*. 236–241.

LAHIRI, K., RAGHUNATHAN, A., DEY, S., AND PANIGRAHI, D. 2002. Battery-driven system design: A new frontier in low power design. In *Proceedings of ASP-DAC 7th Asia and South Pacific Design Automation Conference and Proceedings of the 15*th *International Conference on VLSI Design.* 261–267.

LAJOLO, M., RAGHUNATHAN, A., DEY, S., AND LAVAGNO, L. 2002. Cosimulation-based power estimation for system-on-chip design. *IEEE Trans. VLSI Syst. 10*, 3, 253–266.

LE, J., LI, X., AND PILEGGI, L. T. 2004. STAC: statistical timing analysis with correlation. In *Proceedings of the* 41st *Design Automation Conference*. 343–348.

LEE, I., KIM, H., YANG, P., YOO, S., CHUNG, E., CHOI, K., KONG, J., AND EO, S. 2006. PowerViP: SoC power estimation framework at transaction level. In *Proceedings of the Asia and South Pacific Conference on Design Automation*. 24–27.

METRO IP. http://www.arm.com/products/physicalip/metro.html.

PASRICHA, S. 2002. Transaction level modeling of SoC with System C 2.0. In *Proceedings of the SNUG Conference.*

PASRICHA, S. AND DUTT, N. 2008. *On-Chip Communication Architectures*. Morgan Kaufmann/Elsevier.

PASRICHA, S., DUTT, N., AND BEN-ROMDHANE, M. 2004. Extending the transaction level modeling approach for fast communication architecture exploration. In *Proceedings of the 41st Annual Conference on Design Automation*. 113–118.

PASRICHA, S., PARK, Y., KURDAHI, F. J., AND DUTT, N. 2006. System-level power-performance tradeoffs in bus matrix communication architecture synthesis. In *Proceedings of the* 4^{th} *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. ACM, 300–305.

RABAEY, J., CHANDRAKASAN, A., AND NIKOLIC, B. 2003. *Digital Integrated Circuits: A Design Perspective* 2nd Ed. Prentice-Hall, Englewood Cliffs, NJ.

RAGHUNANDAN, C., SAINARAYANAN, K. S., AND SRINIVAS, M. B. 2008. Process variation aware bus-coding scheme for delay minimization in VLSI interconnects. In *Proceedings of the 9th International Symposium on Quality Electronic Design (ISQED)*. 43–46.

SOTIRIADIS, P. P. AND CHANDRAKASAN, A. P. 2002. A bus energy model for deep submicron technology. *IEEE Trans. Very VLSI Systems 10*, 3, 341–350.

SYNOPSYS CORETOOLS, PRIMETIME PX. 2008. http://www.synopsys.com.

SYSTEMC INITIATIVE. 2008. http://www.systemc.org.

TUUNA, S., ISOAHO, J., AND TENHUNEN, H. 2008. Analysis of delay variation in encoded on-chip bus signaling under process variation. In *Proceedings of the 21st International Conference on VLSI Design*. 228–234.

WANG, H., MIRANDA, M., LOBMAIER, F., CATTHOOR, F., AND PAPANIKOLAOU, A. 2005. A system-level methodology for fully compensating process variability impact of memory organizations in periodic applications. In *Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. 117–122.

Received July 2008; revised August 2008; accepted September 2008