Measuring and Modeling Variability using Low-Cost FPGAs

Michael Brown Cyrus Bazeghi Matthew R. Guthaus Jose Renau
Dept. of Computer Engineering
University of California Santa Cruz

ABSTRACT

The focus of this paper is to measure and qualify high-level process variation models by measuring variability on FPGAs. Measurements are done with high spatial resolution and demonstrate how the high-resolution data matches two industry test cases. The benefit of such an approach is that several inexpensive FPGAs, which are normally on the leading edge of technologies compared to ASICs, obviate the need of fabricating many custom test chips. Specifically, our evaluation shows how measurements of an Altera Cyclone II FPGA can be used to derive variability models for several 90*nm* commercial designs such as the Sun Niagara and Intel Pentium D. Even though the FPGAs and commercial processors are produced by different fabs (TSMC, TI, and Intel, respectively), we find the FPGAs to be very useful for predicting variation in the commercial processors.

1. INTRODUCTION

Test chips are routinely made by industry to determine, with great accuracy, the values of process parameters across die, wafer, and lots in a given process technology. Such methods are costly, but knowing these statistics can provide invaluable insight into the validity of new design techniques. Manufacturing tolerances and product yields are staunchly guarded industry secrets, however. Therefore, researchers are prevented from validating ideas using production model processor data while exploring architectural solutions.

In this paper we propose a new application for FPGAs, specifically we use the 90*nm* Cyclone II from Altera, to measure process variability. This paper shows that FPGAs can be used to calibrate high-level variability models instead of expensive test chips. To this end, we describe how to use FPGAs to measure variability and then confirm that these measurements agree with variability of modern processors.

The main driving force behind adopting a new technology is transistor density. Either the same design can be made smaller (and hence cheaper) or more functionality can be integrated. Because of these reasons, FPGAs are often early adopters of new technologies. In addition, FPGAs are very regular structures and are easy to test. Because of this, FPGAs tend to be the first chips that the foundries mass produce. Together with their low-cost, they are ideal test vehicles to measure process variability. We use a 90*nm* technology because production designs for FPGAs and several off-the-shelf processors are readily available in this technology.

The model we investigate was originally proposed by Bowman et al. [2]. The Bowman FMAX model is widely cited and its predictions have been compared with small test chips. However, an often cited shortcoming of the Bowman model is its lack of spatial correlation. We propose to extend the FMAX model to account for spatial correlations and then determine whether such an extension is necessary. Many other works in the CAD community, such as the infamous Pelgrom model [12], have preached the importance of correlation but the importance for the Bowman model has not been investigated. This is the first time that the FMAX and extended FMAX models have been calibrated with FPGAs and applied to several modern commercial processors.

Similar to our work, several researchers have measured variability on FPGAs. While we use FPGA variability to extrapolate CPU variability, previous works have focused solely on making FPGA designs more robust. Li et al [8] used FPGAs as a polysilicon critical dimension (CD) process monitor. Parametric yield in FPGAs due to within-die delay variations were explored in [14]. Matsumoto et al. [11] proposed a method for improving timing yield considering random within-die variation by selecting an appropriate configuration from a set of functionally equivalent configurations. Katsuki et al. [6] fabricated a LUT array to confirm that FPGAs have clear within-die and die-to-die delay variations.

To validate the FMAX and the extended FMAX model we analyze several existing commercial processors; the Sun Niagara and the Intel Pentium D, which both are manufactured in similar 90*nm* technologies. As the evaluation will show, the measured FPGA variability exhibits significant die-to-die (D2D) variation and high spatial correlation. The Niagara chip exhibits the same correlation as the FPGA. The Pentium D only has two cores and is not suitable for measuring spatial correlation at a high-level. More importantly, the evaluation shows that the FPGA variability measurements can be used to predict the variability of the commercial processors. It is important to emphasize that the processors are only used to validate the results and not to fit the model.

The rest of the paper is organized as follows: Section 2 describes the proposed infrastructure; Section 3 evaluates the proposed infrastructure and the accuracy of the models; and Section 4 presents conclusions and future work.

This work was supported in part by the National Science Foundation under grants 0546819, 720913, and 0751222; Special Research Grant from the University of California, Santa Cruz; Sun OpenSPARC Center of Excellence at UCSC; gifts from SUN, nVIDIA, Altera, Xilinx, and ChipEDA. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the NSF.

2. MEASURING PROCESS VARIABILITY

Process variation is typically classified as independent, correlated, or deterministic. Systematic effects can introduce either correlated behavior or completely deterministic variations. Independent variation has no discernible cause and effect relationship. Many systematic variations, however, may be predictable but are too complex to model accurately during the design process.

The sources of these variations can be further assigned to two types: within-die (WID) or die-to-die (D2D) variation. WID variation, also called intra-die variation, causes on-chip variations in a single design. A transistor or wire on one portion of the die may behave differently than an identical transistor on another portion of the die. D2D variation, also called inter-die variation, causes wafer or lot variation. It affects entire dies in the same way. D2D variation cannot, however, be considered only as an offset due to the differing path and clock signal sensitivities to the D2D variation sources. In this paper, we consider both D2D and WID variation.

Several models [2, 3, 4, 5, 9, 15] have been developed that capture the WID and D2D variation. Sometimes the assumptions used in the models contradict each other. For example, [1] assumes that WID variation is bigger than D2D variation, while [13] assumes the opposite. Most models assume variation parameters from the ITRS roadmap, however, these are industry goals and the actual parameters may be significantly different. Some models determine their process variation parameters by measuring small custom chips, but as mentioned previously, this is prohibitively expensive for large chips which are required for measuring any significant WID variation.

In Section 2.1, we describe an infrastructure to measure the variability using low-cost FPGAs. Subsequently, we present the data in Section 2.2. Finally, Section 2.3 explains how to adapt a recent variability model [2] so that the data gathered from the FPGA can be used to predict CPU variability.

2.1 FPGA Measurement Infrastructure

Field Programmable Gate Arrays (FPGA) are very dense, regular circuit structures. The logic structure primitives are referred to as Logic Elements (LEs) and a single chip can contain a few thousand to many hundreds of thousands of LEs. In addition to the LEs, an FPGA may contain other regular structures such as embedded memories (SRAMs), DSP blocks (hardware multipliers) and even embedded processors.

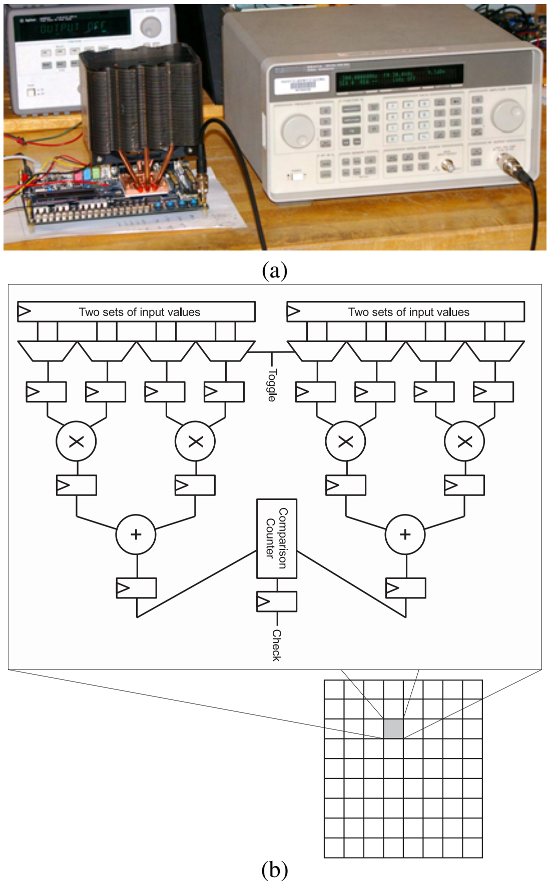
FPGAs are ideal environments to measure the variability in modern technologies for three major reasons: FPGAs are early adopters of new technologies, they are low cost, and they have large die areas. Due to their high regularity and large volume, FPGAs tend to be the first devices to use new technologies. Although it is possible to use custom or ASIC designs to analyze variability, the cost and overhead is significantly larger than using FPGAs. A large FPGA costs under $1K, but building a custom chip with equivalent area can easily exceed $500K. In addition, FPGA die area is equivalent to state of the art processors; a low cost FPGA like the Altera Cyclone II is over $60mm^2$.

FPGAs also have the ability to be reprogrammed after they are manufactured. This trait is ideal for our variability research since configuring the FPGA to perform combinational and sequential logic operations at specific areas of the die is essential for WID variation analysis. The fine-grain mapping of circuit elements allows us to precisely map logic across a die.

The FPGA development system also enables us to easily control the test environment by changing clock frequencies and controlling the scan chains and reset control. To cover the majority of a chip, we created 80 "builds", each covering roughly 1% of the total chip area available for programmable logic. Each build is a synthesized logic design comprised of a test block and clock control circuitry. The general methodology to measure variability on FPGAs is to replicate a self-checking circuit all over the FPGA area, and slowly increase the clock frequency until a failure is detected. From the failure frequency and block list, a map of WID variability is constructed.

Three major issues need to be addressed to obtain accurate variability maps: supply variation control, temperature control and self-checking circuit regularity.

In order to reduce the impact of IR drop in supply circuits, each build with the self-checking circuit was tested separately. This removes the impact that one self-checking circuit may have on an adjacent block.


(a)

(b)

Figure 1: (a) Development board with heat sink and clock generator (b) Self-test circuit and relationship to FPGA.

In addition, temperature has a significant impact on performance of both interconnect and devices. In order to reduce the temperature impact on our process variability measurements, our setup uses a larger-than-required heat sink to minimize intra-die temperature variations. The cooling consists of two parts: a peltier cooler which sits on top of the FPGA (cold side down) and a CPU heat sink and fan which sits on top of the peltier cooler. The heat sink is capable of dissipating over 100W and keeps the FPGA temperature under 10C. This larger than required heat sink ensures that thermal effects both within the FPGA as well as between FPGAs are kept to a minimum and that temperature is stable regardless of the frequency of the external clock.

It is also important to guarantee that all the self-checking blocks have the same hardware mapping to the LEs. This is more challenging than it seems at first. Designs can be placed and routed, exported, and then imported to any FPGA location with the correct resources. Small adjustments in the routing of these "hard" blocks during the synthesis of each build are required due to IO connections. However, since the circuits are self-checking, these IO routes have little impact on the overall performance. Any remaining differences are accounted for by normalizing each block location among the same block on all of the measured dies.

Figure 1-(a) shows the major components of the measurement infrastructure: the FPGA board, the heat sink, and the external clock generator.

The Cyclone II FPGA is manufactured in TSMC's 90*nm* technology and operates at 1.2v. The Development and Education Board (DE2) developed by Altera (Figure 1-(a)) was used for analysis. Each DE2 board contains a Cyclone II EP2C35 (672 pin package) FPGA. This device contains approximately 33K LEs, 483K memory bits, 35 embedded multipliers, 4 PLLs, and 475 user I/O pins. Our methodology used the Altera Quartus II version 6.1 design software. This tool is used for design entry with Verilog HDL, synthesis, mapping, fitting, and timing analysis.

We divide the FPGA's available programmable space into 80 blocks (Figure 1-(b)). Not all the available block space is used, but we get 75% coverage of physical area over many project builds. Each build is an individual test block. A test block uses 95% of the combinational cells and 58% of the register cells available in its physical region. An example of one of these builds is shown in Figure 1-(b) as a simplified block diagram. The basic test block is synthesized for a clock speed of 200MHz and constrained to an area of 4x5 LABs (a LAB consists of 16 Logic Elements). It is then exported as a hard macro to retain its placement and routing. Then, this basic block is imported into 80 projects, each at a different location on the FPGA.

At the top level of each project, a clock generation unit selects one of two possible clock sources: the external high frequency clock generated by an HP 8647A Signal Generator or an external scan clock. The development board provides debounced buttons which we configure for use as a *reset*, *enable*, and *scan clock* signals. We also use a switch to select which clock source to use. Since switching a clock source while a circuit is running can be problematic, the asynchronous *enable* button is synchronized and then disables the registers before changing clock sources. The registers inside the FPGA have enable ports which allow us to use this approach. Thus we were able to reset the logic, program a clock frequency, wait a short period, disable the clock, switch clock sources, scan the result out, and note if the block fails for a given frequency.

2.2 Measured FPGA Data

This section presents the variability measurements obtained after analyzing the six FPGA development boards. A histogram of the delay is shown in Figure 2. The data for each of the analyzed FPGA chips is shown in a box plot in Figure 3. Here, we see that there is significant variation among the means of each FPGA die (D2D

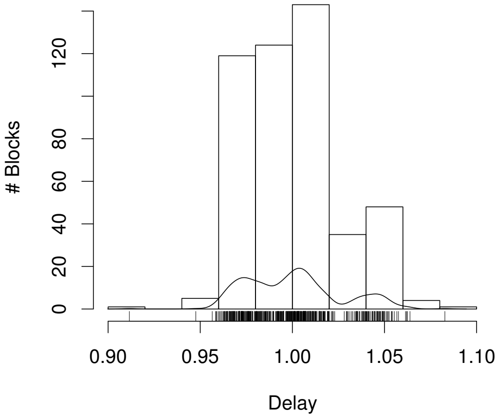


Figure 2: Delay block histogram for the FPGA chips analyzed.

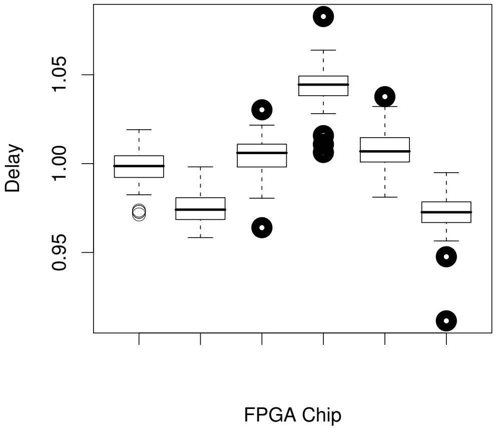


Figure 3: Delay box plot for the FPGA chips analyzed.

variation). We also see the first and third quartiles for each FPGA chip in relation to each other. There are a couple of outlier points on each die, but it was confirmed that these values are repeatable over a period of time. These plots visually show the different types of populations without any assumptions of the statistical distribution.

Figure 4 shows the delay maps of each analyzed FPGA. The level plot shows a 10 x 8 grid annotated with the normalized failure delay (failure delay $= \frac{1}{\text{failure frequency}}$). From this data, we compute the die-to-die (D2D) standard deviation to be 2.56%, the within-die (WID) standard deviation to be 1.04%, and the random standard deviation to be an additional 1.04%. It is observed that there is significant D2D variation and the combination of the WID and random variation is comparable to the D2D variation.

Further examination of Figure 4 shows significant clustering of the failure delay. That is, if an area has a high failure delay, it is likely that adjacent areas have a similar delay. This is due to the spatial correlation of the WID variation due to proximity effects during design and manufacturing. Figure 5 shows three different plots analyzing the correlation coefficient compared to distances in the xy plane, x-dimension, and y-dimension, respectively. It is interesting to note that the directional distance in the xy plane is quite noisy, but has a general trend of decreasing correlation with distance. This corresponds to previous research results. An interesting result, however, is that the x- and y-dimensions do not show equal spatial correlation. The x-dimension has a correlation of roughly 0.9 independent of the distance through about 7mm. The y-dimension, however, has decreasing correlation down to about 0.8 at 7mm. Since the FPGA die size is 8mm x 8mm, we are unable to verify correlations beyond this distance at this time.

2.3 Processor Variability Model

We approach modeling variability by using the FMAX model presented in [2] which is fundamentally stated in (1). The FMAX model says that the frequency that a design will run is the nominal frequency for the design plus the effects of die-to-die variability and the effects within-die variability.

The $f_{T_{cp},max}$ is a probability distribution function (PDF) of the longest delay of a critical path in a design. The component PDFs of this distribution are as follows: the nominal delay ($f_{T_{cp,nom}}$) is an offset, die-to-die variation ($f_{\triangle D2D}$) is a normal distribution, and within-die variation ($f_{\triangle WID}$) is a composite term. To find the PDF for $T_{cp,max}$ we use (2), which convolves the nominal critical-path delay as an impulse with the PDFs of the respective elements of variation. [1]

$$\begin{aligned} f_{T_{cp},max} &= f_{T_{cp},nom} + f_{\triangle D2D} + f_{\triangle WID} \qquad (1) \\ &= \delta(t - T_{cp,nom}) * f_{\triangle D2D}(t) * f_{\triangle WID}(t) \qquad (2) \end{aligned}$$

The value of this impulse can be approximated as the FO4 logic depth of the design, N_{FO4}, multiplied by the delay time of a NAND gate for the modeled technology $T_{cp,nom} = N_{FO4} \cdot T_{nand}$. Returning to the variation components; the D2D variation, $f_{\triangle D2D} = N(0, \sigma^2_{D2D})$, is a normal distribution centered at 0 with a variance determined by measurements.

$$f_{\triangle WID} = N_{cp} \cdot f_{\triangle wid}(t - T_{cp,nom}) \cdot (F_{\triangle wid}(t - T_{cp,nom}))^{N_{cp}-1} \quad (3)$$

The within-die variation is shown in (3), where Ncp is the number of critical paths in the design of a core, approximated by the work

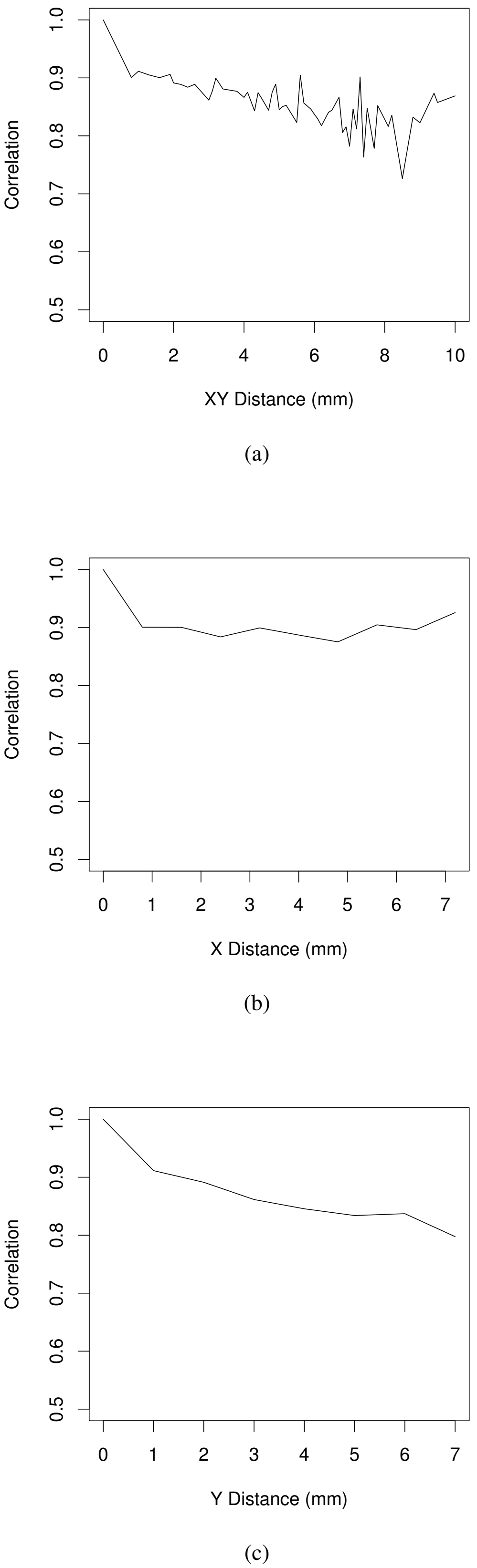


Figure 5: **Spatial correlation on distance (a), for the x-axes (b), and the y-axes (c) for the FPGA chips analyzed.**

[1] Summing gaussian normals is done by convolving them

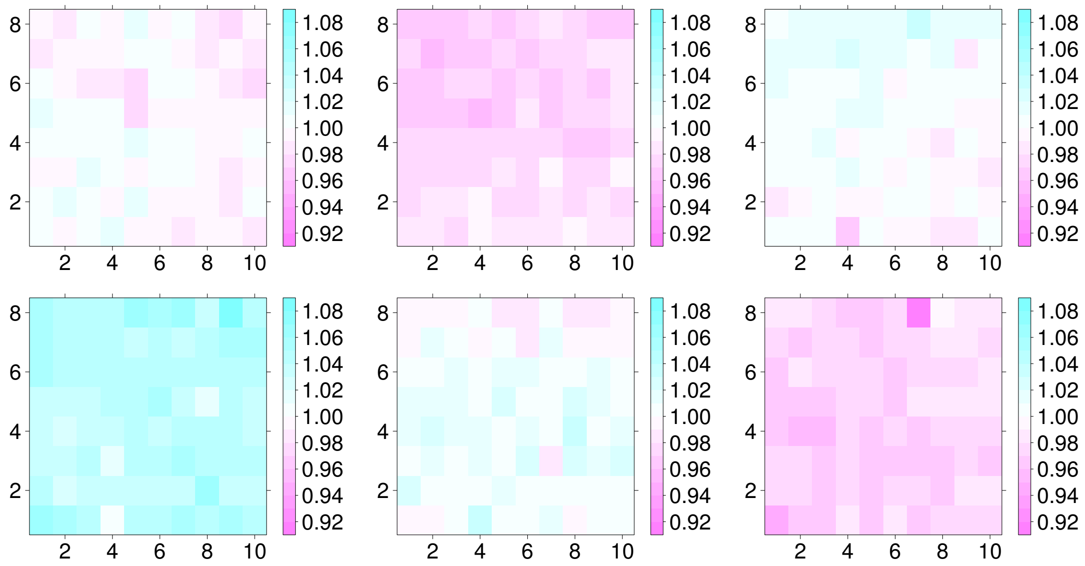


Figure 4: Delay level plots for the FPGA Chips analyzed.

of [10]. In this equation, $f_{\triangle wid}$ is a normal distribution, like $f_{\triangle D2D}$, centered at 0 with an empirically determined standard deviation (4).

$$f_{\triangle wid}(t - T_{cp,nom}) = N(0, \sigma^2_{WID}) \tag{4}$$

The component $F_{\triangle wid}$ is the cumulative distribution function (CDF) shown in (5) by integrating over $f_{\triangle wid}$.

$$F_{\triangle wid}(t - T_{cp,nom}) = \int_0^{t-T_{cp,nom}} f_{\triangle wid}(x)\mathrm{d}x \tag{5}$$

Equation 6 shows the expanded terms using the number of critical paths, the length of the longest critical path, and two empirically determined deviations as inputs. The output is a PDF representing the spread of cycle time delays that the design can be reasonably expected to attain.

$$\begin{aligned} f_{delay}(t) = {} & (\delta(t - (N_{FO4} \cdot T_{nand})) \\ & * N(0, \sigma^2_{D2D}) * (N_{cp} \cdot N(0, \sigma^2_{WID}) \\ & \cdot (\int_0^{t-(N_{FO4} \cdot T_{nand})} N(0, \sigma^2_{WID})\mathrm{d}t)^{N_{cp}-1}) \end{aligned} \tag{6}$$

The FMAX model does not address spatial correlation though. For this, we add a spatial correlation coefficient which takes into account the size and floorplan of the die. By measuring the size and position of cores on a die, we produce matrices which relate the respective X and Y distances among all cores. Using the distance vs. correlation plots obtained from FPGA measurements, we create a correlation matrix which describes the relation in variation between all cores on a die. By extending the FMAX model in (1) and (6), we are able to predict the cycle time spread of spatially correlated multicore dies. To do this, we add a term to account for variation from core-to-core (C2C) to get:

$$f_{T_{cp},max} = T_{cp,nom} + f_{\triangle D2D} + f_{\triangle WID} + f_{\triangle C2C}. \tag{7}$$

Our $T_{cp,nom}$ and $f_{\triangle WID}$ are represented as the maximum of partially correlated Gaussians across all critical paths in a core in (8). In this equation, $N_i(T_{cp,nom}, \sigma^2_{WID})$ is the delay distribution of a single critical path with $N_i = 1...N_{Ncp}$ and ρ_{WID} is the correlation of the WID variation in a single core.

$$\begin{aligned} f_{T_{cp,nom}+\triangle WID} &= N(\mu_{path}, \sigma^2_{path}) \\ &= MAX(\rho_{WID}, (N_1(T_{cp,nom}, \sigma^2_{WID}), \\ & \quad N_2(T_{cp,nom}, \sigma^2_{WID}), ..., N_{Ncp}(T_{cp,nom}, \sigma^2_{WID}))) \end{aligned} \tag{8}$$

To find the distribution of cores on a multicore die, (8) is reformed to take the maximum of several partially correlated cores in the following:

$$\begin{aligned} f_{T_{cp,nom}+\triangle WID+\triangle C2C} &= N(\mu_{cores}, \sigma^2_{cores}) \\ &= MAX(\rho_{[Ncores][Ncores]}, \\ & \quad (N_1(\mu_{path}, \sigma^2_{path}), N_2(\mu_{path}, \sigma^2_{path}), \\ & \quad ..., N_{Ncores}(\mu_{path}, \sigma^2_{path})). \end{aligned} \tag{9}$$

This calculates the delay distribution for a multicore processor. Because the inputs to the MAX function are not values, but distributions, the computation of the MAX function for the core delays depends on the particular layout of the processor. It can be simple in the case of only two cores or require integration of the joint probability distribution in the case of non-uniform distances in a system with many cores.

After calculating the distribution of a single die with multiples cores, we then combine the independent D2D deviations of $f_{\triangle D2D}$ and

$f_{T_{cp,nom}+\triangle WID+\triangle C2C}$ with convolution. This computes the delay distribution of a die with multiple cores.

3. EVALUATION

The evaluation starts by presenting the challenges in measuring commercial processor delay distributions in Sections 3.1 and 3.2. Then, the raw data measured from the processors is presented in Section 3.3. Next, we use the FPGA variability data, including spatial correlation, to generate predicted Pentium and Niagara processor variability in Section 3.4. The confidence of these predictions is then formally computed in Section 3.5.

3.1 Processor Measurement Infrastructure

The best method of measurement available to test processors is built-in-self-test (BIST) as the processors are overclocked at multiple frequencies under a controlled temperature. This usually involves a service processor which can setup the parameters of the test and allows for the most accurate and precise testing since the BIST can give information about specific kinds of failure local to each processing core.

The maximum performance of a core is measured as the frequency where a core no longer works properly. Ideally, a test is performed as the frequency is turned up until the cores start to fail the test. Many machines will let you do this in the BIOS before booting up the machine or by replacing the external clock generator.

Of course, neither the BIST nor the frequency change are available on many systems. Equally important, temperature variations can significantly affect the failing frequency and the manufacturers often bin their systems according to performance. To have accurate processor variability measurements, we need to have BIST, frequency control alternatives, and to compensate for temperature and manufacturer binning.

BIST Alternatives: If a BIST is not available, then a self-checking test program which utilizes most functions of each core can be written. A good self-checking program can be a CRC check, matrix multiplier, or some other CPU intensive program. Since we want to focus our testing on the processor core only, the test program should not generate any off-core traffic such as L1 cache misses and/or I/O operations.

Frequency Increase Alternatives: The maximum operational frequency is dependent on both temperature and voltage. We found that controlling the temperature was more challenging than controlling the supply voltage. Therefore, if frequency overclocking is not available, the alternative is to lower the core voltage. This achieves the same effect as raising the frequency. Several manufactures publish "shmoo" plots which map core voltages to functionally equivalent frequencies [7].

Temperature Impact: Failures due to overclocking inevitably occur at different running temperatures. This requires a correction since temperature affects the processor circuit timing and in turn the frequency at which it will fail. To compensate for this, we force different on-die temperatures and measure for their corresponding different failing frequencies. For the range of temperatures/frequency analyzed, we found the relationship between temperature and frequency to be linear in our setup. This means that if both the failing frequency (F) and the operating temperature (T) that cause the frequency are known, then (10) gives what the frequency would be if all failures occurred at the same temperature. The T_{factor} is obtained for each measured processor.

$$F = F_{measured} + (T_{center} - T_{measured}) \cdot T_{factor} \qquad (10)$$

Manufacture Binning: Manufacturers bin their processors according to frequency. This means that measuring the variability on a binned processor may lead to lower process variability than what really exists. In order for our study to accurately represent the un-binned lot of chips as they are manufactured, we obtain sales records to approximate the size of each bin. In our case, we track the sales ranking on Amazon for the product lifetime to use as a representative bin size. We then weigh the variability data according to the size of its bin.

3.2 Pentium D and Niagara Measurement Setup

While the previous section explained a general infrastructure to measure process variability in off-the-shelf processors, this section provides further details required for the Intel Pentium D and Sun Niagara processors.

Intel Pentium D: The dual core Pentium D 820/840s measured have 200mm^2 die area using Intel 90nm technology operating at 1.3v. Pentium D 820s operate at 2.8GHz and the Pentium D 840s reach 3.2GHz. We were able to overclock the frequency of both processors. Raising the clock frequency of the processor is accomplished by increasing the external system clock which is input to the processor. The chip then multiplies this clock to get its own frequency. By using a processor which is relatively slow when compared to the other system components we ensure that failures are only the result of the chip and not memory or other components. To guarantee that the motherboard never fails we use 1000MHz DDR2 when the processor only requires 600MHz DDR2. Intel does not publish information about the on-chip BIST, therefore, we use the BIST alternatives previously explained.

Sun Niagara: The eight core T1 chip has a 400mm^2 die area using TI 90nm technology operating at 1.3v. To measure performance we lower the Niagara's core voltage rather than raising the clock rate. We do this because the Niagara's CPU frequency can only be changed by raising the system clock. Raising the system clock, however, means that any failures could be from overclocking any of the system components. To isolate the eight cores for failure we instead lowered the core voltage and recorded the point of failure with the BIST activated during system boot. When combined with the shmoo plot (Figure 6) published by Sun [7], we are able to obtain data which is equivalent to overclocking.

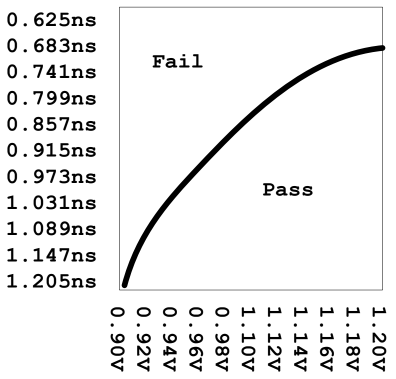


Figure 6: Shmoo plot for the Sun Niagara.

3.3 Processor-Measured Data

We measured the performance for eleven off-the-shelf Pentium D 800 series chips (22 cores total) and three Sun Niagara T1 chips (24 cores total).

Intel Pentium D: The Intel Pentium D 800 series have two cores per chip. For these Pentiums, we measured the temperature at which each chip failed. Since the cores sometimes failed at temperatures as much as 45 °C apart, we measured the range of failure for several chips over the widest possible frequency and temperature spread. The plots showed a near linear relationship between the temperature of failure and the frequency of failure. Using this, we applied an T_{factor} in (10) of 4.4 MHz/°C to account for temperature differences.

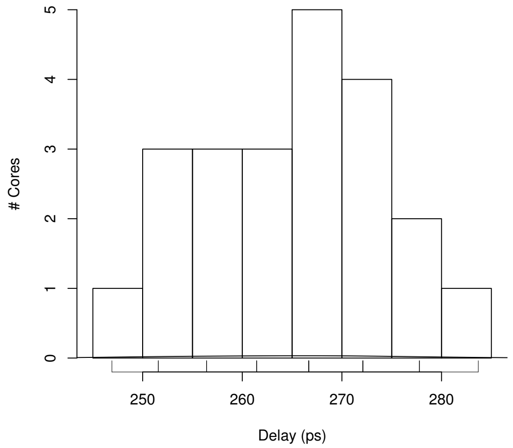


Figure 7: **Un-binned performance distribution of the Pentium D 800 series.**

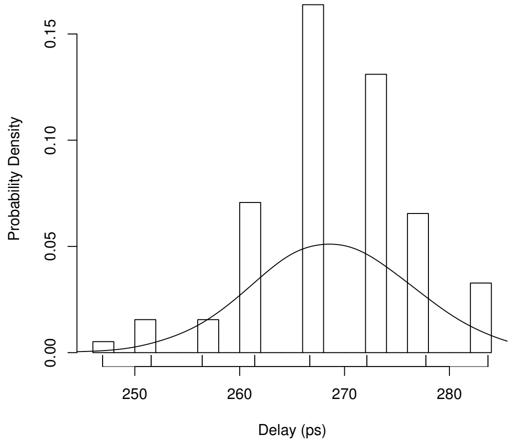


Figure 8: **Binned or adjusted performance distribution of the Pentium D 800 series.**

Of the Pentiums tested, seven were Pentium D 820s and four were Pentium D 840s so that we would have a more complete sample of the chips produced. By retrieving sales records for the Pentium D 800 series for a large scale vendor (Amazon), we extrapolate the approximate production quantities for each bin. We found that roughly 60% of the chips produced were Pentium D 820, 23% Pentium D 830, and 17% were Pentium D 840. While Figure 7 shows the frequency distribution of the original 22 Pentium cores, Figure 8 shows the bin-compensated Pentium frequency distribution using the estimated manufacturing bins. Note the slight slump on the left side of the distribution around 255ps. This is because we could not attain any 830 chips to measure. The missing data from the 830 chips seems very likely to fill the dip between data acquired from the 820s and the 840s. The measured D2D variability is 3.9%, the WID variability is 0.8%, and the Random variability is 3.0%.

Sun Niagara: Figure 9 shows the performance level counter plots measured for each Niagara chip. A visual inspection of the Niagara plots shows a spatial correlation among cores as seen with the FPGAs. The right-most plot had only limited measurements because the master core was the first core to fail. A failing master core prevents the BIST on the other cores since data cannot be retrieved from the chip at any higher performance level than that of the master core. The Niagaras had a 14.2% D2D, 2.8% WID, and 1.0% Random variation. The D2D variation should not be considered because we only sampled three different chips.

3.4 Model Parameter Extraction

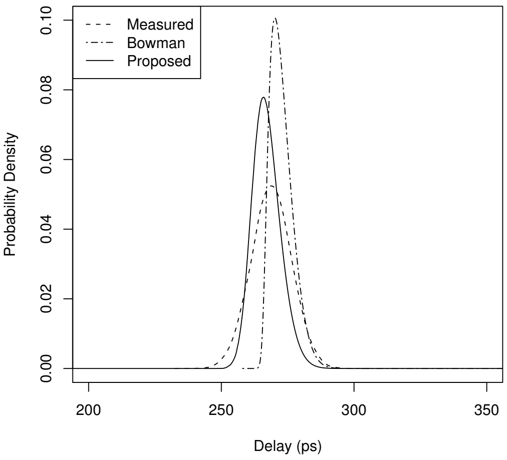


Figure 10: **Pentium D 800 series plots with the measured processor distribution (*Measured*), Bowman model without spatial correlation (*Bowman*), and Bowman model extended with spatial correlation (*Proposed*).**

This section applies the model from Section 2.3 with the FPGA data to estimate variability measured from the Intel Pentium D 800 series and the Sun Niagara T1s.

Figure 10 shows the overall variability distribution estimated for the Pentium D 800 series using the FPGA data without spatial correlation model (*Bowman*) and with spatial correlation (*Proposed*). As we see from the figure, the spatial correlation (between 0.8 and 0.9) has a low impact on the distribution. Adding spatial correlation to the Bowman model tends to increase the standard deviation (σ) and lower the average delay (μ). Despite being non-Gaussian, we still make comparisons using the first two moments of the distributions (μ and σ) since the distributions are not drastically different

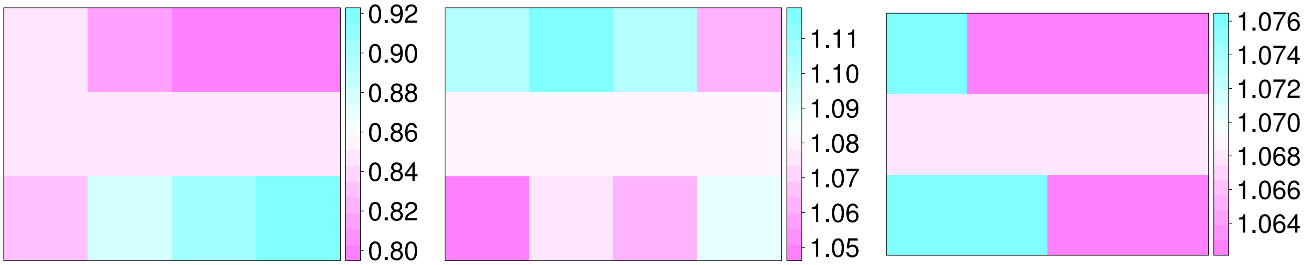


Figure 9: Performance level contour plots for the Sun Niagara analyzed.

from a Gaussian.

The binned Pentium D 800 series have a mean of $269ps$ with a standard deviation of $7.5ps$. Bowman predicts a $\mu = 272ps$ and a $\sigma = 4ps$. Once we add spatial correlation used on the proposed model, we obtain $\mu = 267ps$ and $\sigma = 5ps$. Both Bowman and the Proposed method are very close to the measured data.

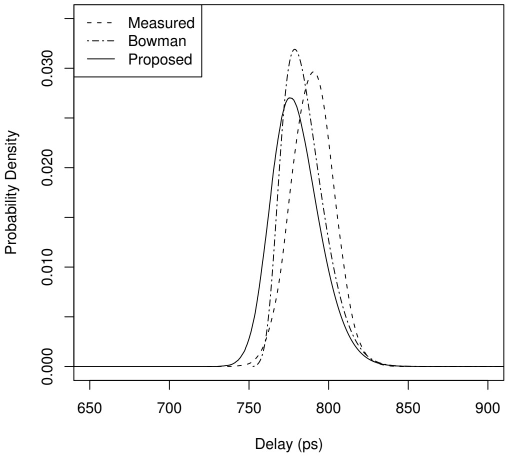


Figure 11: Niagara plots with the measured processor distribution (*Measured*), Bowman model without spatial correlation (*Bowman*), and Bowman model extended with spatial correlation (*Proposed*).

For the Sun Niagara, we triple the number of FO4 (N_{FO4}) and decrease by one fold the number of critical paths (N_{cp}). The Sun Niagaras have a mean delay of $734ps$ with a standard deviation of $82ps$. When we use our model, we obtain a $\mu = 779ps$ with a $\sigma = 15ps$. The μ is very close to the predicted by the model, but the σ is 5.4 times bigger than what was measured over all of our Niagaras. We feel the reason for this is that we only measured 3 Niagara chips and one of them was significantly faster, perhaps as much as two standard deviations away from the mean of all Niagaras produced. This meant that our measured mean was not in our confidence interval calculated from the model. If we remove the fast Niagara chip (Figure 11), we have a $\mu = 789ps$ and a $\sigma = 13ps$. The σ predicted by the model (Proposed) fits between both of them. Bowman predicts a $\mu = 784ps$ and a $\sigma = 13ps$.

We conclude that the FPGA variability is a good source to select parameters for process variability because comparing the measured results to the predicted with the Bowman model and the Proposed model has a low error.

Another conclusion from this section is that, for the data analyzed, there is only a small difference between the probability distributions when we use a plain Bowman model and a Bowman model extended with spatial correlation. This means that modeling spatial correlation does not have a significant affect on measurements.

3.5 Confidence Intervals

Because we were only able to attain a limited number of processors to test, we include a calculation of the statistical confidence of our measurements. Determining confidence in a statistical set requires calculating a confidence interval, which starts with the sample set and an allowable error, α. The sample set is $x_{1...n}$. The known mean of this sample set is $\bar{X}$ and the unknowable mean of the infinite set is μ. The variance of the sample is

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X}).$$

We define Z as $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$. After which we can say that the probability P of the next drawn sample Z being between $-z$ and z is

$$\begin{aligned} P(-z \leq Z \leq z) &= 1-\alpha \\ z &= \phi^{-1}(\phi(z)) \\ \phi(z) &= P(Z \leq z) = 1 - \frac{\alpha}{2}. \end{aligned}$$

By calculating z from the probit function, ϕ^{-1}, we find the bounds of the confidence interval. With the semantics of confidence intervals being something of debate in the statistics community, it may be worth noting that we choose this to mean that with probability $1-\alpha$, the μ will be in the interval $\bar{X} \pm z\frac{\sigma}{\sqrt{n}}$.

Our FPGA setup and extended model estimates a 95% confidence interval for the mean cycle time of Pentium Ds to be between $265ps$ and $269ps$, which the measured of mean of $269ps$ is. Likewise, our 95% confidence interval for the mean cycle time of Niagaras is

772*ps* to 786*ps*. In this case, our measured mean was near the edge of the confidence interval at 789*ps*.

4. CONCLUSIONS AND FUTURE WORK

This paper has several novel contributions: it introduces a measuring infrastructure for FPGAs and processors, it shows that FPGA variation information can be applied to processor models, and it presents new insights on variation and spatial correlation in multicore systems.

The proposed measuring infrastructure targets processors and FPGAs. The processor setup captures operating frequency of modern processors and proposes several compensations and methods to obtain accurate measurements. The FPGA setup also captures variability, but with a higher spatial resolution.

The key contribution of the paper is that high resolution FPGA variability data can be applied to commercial processors. Utilizing FPGAs enables the opportunity to perform many measurements with a reduced cost. Both the absolute variability and the distribution predicted using FPGA measurements is very close to the measured variability on the Sun Niagara and Intel Pentium D 800 series. We feel that such measurements are important for developing new models and to gain further insights.

Both the FPGA and the processors analyzed show a larger D2D variation than WID variation. These preliminary results seem to contradict previous publications [1, 2] that assume a bigger WID than D2D variation. Though other sources like IBM [16] purport that D2D variation is a full three times greater than WID variation which supports our conclusions. These findings have several implications to designers as differences inside a die are less significant than difference between dies. Ideally, future work should validate our insights by analyzing multiple types of FPGAs and technologies.

Another interesting insight shown in the FPGA evaluation is the fairly constant spatial correlation. Our variability model assumes that the spatial correlation changes with distance. A nearly constant spatial correlation [2] could simplify existing variability models. Also, we found that despite having been produced by different fabrication plants, the FPGA still provided good input for our models to predict processor variation. Future work should measure more (and larger) FPGAs and processors to further quantify these observations.

Finally, we feel that the measuring capabilities shown in the paper create the opportunity for further insights. For example, we found that WID variability is lower than D2D variability. The D2D variability could be divided into wafer to wafer (W2W) variability and within-wafer variability (WIW). If the wafer information is known, additional measurements are possible. Building test circuits into FPGAs would enable an even more detailed insights. Also interesting would be to analyze 65*nm* FPGAs. In conclusion, this work has made several interesting contributions and establishes new opportunities for many additional research projects.

Acknowledgments

We like to thank the reviewers for their feedback on the paper. Thanks also to Altera Corporation for their aid in obtaining DE2 boards and fielding questions about their products.

[2] Figure 5-(a) has between 0.8 and 0.9 correlation for all the distances.

5. REFERENCES

[1] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *ICCAD*, page 900, Washington, DC, USA, 2003. IEEE Computer Society.

[2] K. Bowman, S. Duvall, and J. Meindl. Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration. *IEEE Journal of Solid-State Circuits*, 37(2):183–190, Feb 2002.

[3] Y. Cao and L. T. Clark. Mapping statistical process variations toward circuit performance variability: an analytical modeling approach. In *DAC*, pages 658–663, 2005.

[4] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *ISQED*, pages 516–521, 2005.

[5] E. Humenay, D. Tarjan, and K. Skadron. The impact of systematic process variations on symmetrical performance in chip multi-processors. In *DATE*, pages 1653–1658, April 2007.

[6] K. Katsuki, M. Kotani, K. Kobayashi, and H. Onodera. A 90 nm LUT Array for Speed and Yield Enhancement by Utilizing Within-Die Delay Variations. *IEICE Transactions on Electronics*, E90(4):699–707, April 2007.

[7] A.S. Leon, B.Langley, and L.S. Jinuk. The UltraSPARC T1 Processor: CMT Reliability. In *IEEE Custom Integrated Circuits, 2006*, pages 555–562. IEEE Computer Society, Sept 2006.

[8] X.-Y. Li, F. Wang, T. La, and Z.-M. Ling. FPGA as Process Monitor-An Effective Method to Characterize Poly Gate CD Variation and Its Impact on Product Performance and Yield. *IEEE Transactions on Semiconductor Manufacturing*, 17(3):267–272, Aug 2004.

[9] X. Liang and D. Brooks. Microarchitecture parameter selection to optimize system performance under process variation. In *ICCAD*, pages 429–436, 2006.

[10] D. Marculescu and E. Talpes. Variability and energy awareness: A microarchitecture-level perspective. In *DAC*, June 2005.

[11] Y. Matsumoto, M. Hioki, T. Kawanami, and et al. Performance and Yield Enhancement of FPGAs with Within-die Variation using Multiple Configurations. In *FPGA*, pages 169–177. ACM Press, 2007.

[12] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers. Matching properties of mos transistors. In *JSSC*, pages 1433–1439. IEEE, 1989.

[13] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester. Statistical estimation of leakage current considering inter- and intra-die process variation. In *ISLPED*, pages 84–89, New York, NY, USA, 2003. ACM Press.

[14] P. Sedcole and P.Y.K. Cheung. Parametric yield in fpgas due to within-die delay variations: A quantitative analysis. In *FPGA*, pages 178–187. ACM Press, 2007.

[15] A. Srivastava, S. Shah, K. Agarwal, D. Sylvester, D. Blaauw, and S. Director. Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance. In *DAC*, pages 535–540, 2005.

[16] C. Visweswariah. Within die variations in timing: From derating to cppr to statistical methods. In *ICCAD Tutorial*. IBM, Inc., 2007.