Impact of Variability on Power and Performance

Daisy Lee
July 29, 2005
Advisor: Prof. D. Marculescu
Carnegie Mellon University

Table of Contents

Introduction 3

Experimental Setup 4

Experimental Results 9

Conclusion 11

References 12

Table of Figures

Figure 1: Probability Distribution for $T_{cp,max}$ 5

Figure 2: 000.art 6

Figure 3: 177.mesa 6

Figure 4: 000.ijpeg 6

Figure 5: 175.vpr 6

Figure 6: 000.gcc 6

Figure 7: 256.bzip2 6

Figure 8: Power Consumption 8

Figure 9: Performance 9

Figure 10: Quality 10

Introduction

Increased variability in high-performance processors poses a challenge to processor design and analyzation. Because of aggressive techniques for increasing processor performance, such as the use of sub-wavelength lithography and deep-pipelining, and for lowering power consumption, such as the use of voltage scaling, variability is introduced to process and system parameters that appear across a single die (WID) or across multiple dies (D2D) [9]. Variability in itself can be detrimental to product robustness; ie. the product may not adhere to a set of performance specifications [8]. Furthermore, variability decreases the reliability of logic and memory available on chip [9]. Thus it is crucial that processors are designed taking into account the negative effects of variability.

One possible approach that lessens the negative effects of and even exploits variability is the Globally Asynchronous, Locally Synchronous (GALS) implementation. Such a design involves splitting the processor into five regions, each respectively dealing with fetch and decode, rename and dispatch, integer, floating point, and memory. Each region retains its own clock speed suited to the processes involved. Communication between the regions is enabled through high-speed FIFOs based on arbiters or synchronizers [10]. Through this implementation, clock speed in the rename/dispatch region, where instructions are accounted for, is noticeably reduced due to fewer critical paths and lower temperature, thus improving performance and reducing power leakage. If voltage scaling is applied to each of the regions, power consumption and leakage are reduced further. GALS is a relatively new design that is currently in the research and development stage; most of today's processors are built using the synchronous implementation, which involves only one clock speed across the entire processor.

As of now, there is no unanimously agreed upon metric for the assessment of processor design quality that takes into account the effects of variability. The metric used in this paper was proposed by Professor Diana Marculescu of Carnegie Mellon University:

Q = T_cp,max * CPI * Power (1)

where T_cp,max is the maximum critical path delay, CPI is the number of cycles per instruction, and Power is the sum of dynamic and leakage power [9]. A smaller Q value signifies better quality design. The goal of my research is to compare Q values among different cases, namely synchronous, GALS, GALS with temperature effects, and GALS with voltage scaling, and lastly, gain insight on the nature of variability and its impact on power consumption and performance.

Recently, there has been growing interest in process and system parameter variability and its effects on power and performance. Eisele et al. [6] have shown how WID variations affect performance on low power designs. Bowman et al. [3] developed statistical models for WID and D2D variations and their impact on critical paths and logic depth. Borkar et al. [2] have shown that adaptive body biasing reduces the effect of variations in speed and leakage current. Finally, there have also been a number of papers addressing the benefits and tradeoffs of the GALS design, specifically papers written by Semeraro et al. [11], Iyer et al. [7], and Talpes et al. [12].

Experimental Setup

WID Variation

In order to find $T_{cp,max}$, or the maximum critical path delay, for each of the five asynchronous regions, I considered the following equation:

$$T_{cp,max} = T_{cp,nom} + \Delta T_{D2D} + \Delta T_{WID} \quad (2)$$

where $T_{cp,nom}$ is the nominal path delay which is assumed to be 10, ΔT_{D2D} is the die-to-die variation which is normally distributed with zero mean and standard deviation approximately 10-12% of $T_{cp,nom}$, and ΔT_{D2D} is the within-die variation with a distribution defined by

$$f_{\Delta T\_D2D}(t) = N_{CP} * f_{WID\_Tcp,nom}(t - T_{cp,nom}) * (F_{WID\_Tcp,nom}(t - T_{cp,nom}))^{Ncp - 1}$$

where $f_{WID\_Tcp,nom}$ is a Gaussian centered at $T_{cp,nom}$ with standard deviation that is 5% of $T_{cp,nom}$ and a cumulative distribution of $F_{WID\_Tcp,nom}$ [9]. The mean values of ΔT_{D2D} and ΔT_{WID} are added to $T_{cp,nom}$ to obtain a mean value for $T_{cp,max}$ which will be plugged into cycle-accurate processor simulators, similar to the one described in [11][7], that provide performance statistics based on power models using the Wattch framework [4]. One simulator models GALS while the other models synchronous.

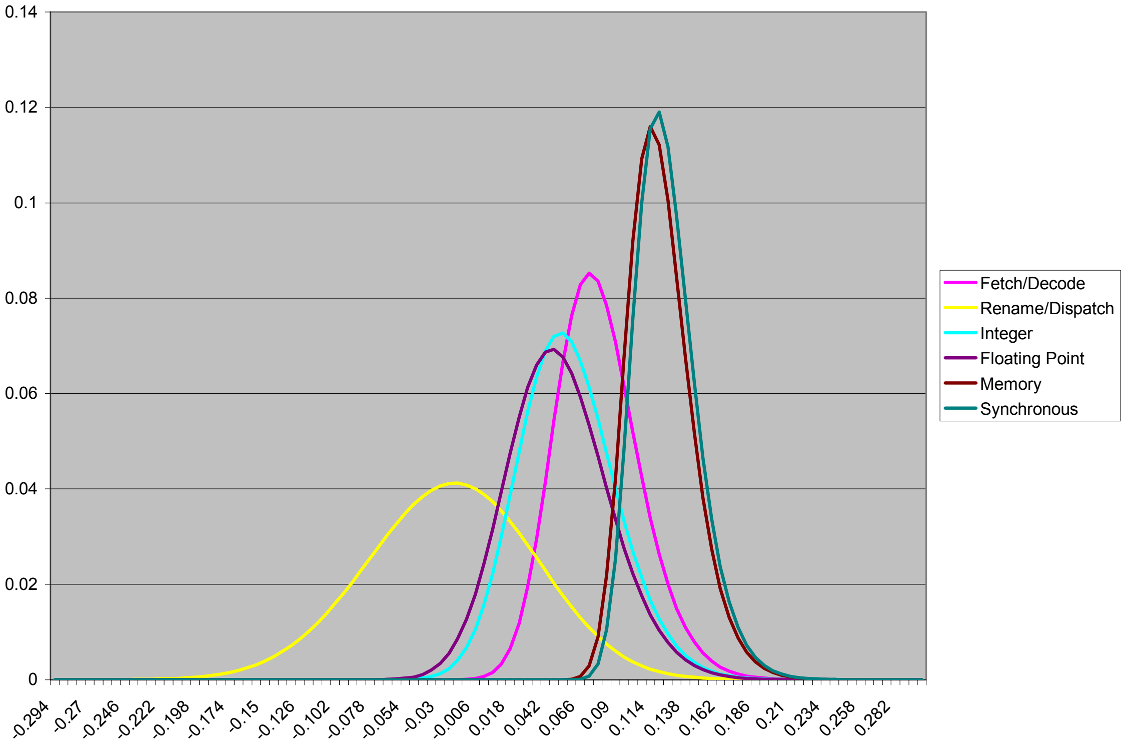
To find the mean of ΔT_{WID} for each region, I first found N_{CP}, the number of critical paths, for each region using the following relation:

$$N_{CP}(\text{region})/N_{cp,Total} = N_{devices}(\text{region})/N_{devices,Total} \quad (3)$$

where $N_{devices}$(region) is the number of devices in a region, $N_{devices,Total}$ is the total number of devices, and $N_{cp,Total}$ is the total number of critical paths which is assumed to be 100 [9]. In other words, the number of critical paths per region is proportional to the number of devices in that region. Note that the values for $T_{cp,nom}$ and $N_{cp,Total}$ are chosen arbitrarily; other values can be chosen just as long as there is a consistency when making comparisons between GALS and synchronous.

Second, I wrote a program that models $f_{\Delta T\_D2D}(t)$, plugged in N_{CP} (region), and found sample values for $f_{\Delta T\_D2D}(t)$ accordingly. Since the mean is located at the point where $F_{WID\_Tcp,nom}$ $F_{\Delta T\_D2D}(t)$, the cumulative distribution of $f_{\Delta T\_D2D}(t)$, equals 0.5, I enabled my program to find sample values of $F_{\Delta T\_D2D}$ as well. Finally, I obtained the mean by finding the t value that corresponds to a cumulative probability of 0.5. Figure 1 models $f_{\Delta T\_D2D}(t)$ for each region and their mean values.

Figure 1: Probability Distribution for $T_{cp,max}$



Temperature

To obtain temperatures for the five asynchronous regions, I used HotSpot-2.0, a software that calculates temperatures of various regions given sample power profiles. Thus I ran multiple simulations, using my newly obtained $T_{cp,max}$ values in the GALS case, on various testbenches that model different applications. For each testbench, I obtained three sample profiles, each modeled after 50 million instructions representing different segments of the application simulation. Finally, I plugged in the power profiles to HotSpot and obtained the temperatures for various regions of the processor, identified the regions associated with each of the five asynchronous regions, and found the associated temperatures accordingly. Figures 2 through 7 show the temperatures per region for various testbenches in both GALS and synchronous cases.

Figure 2: 000.art

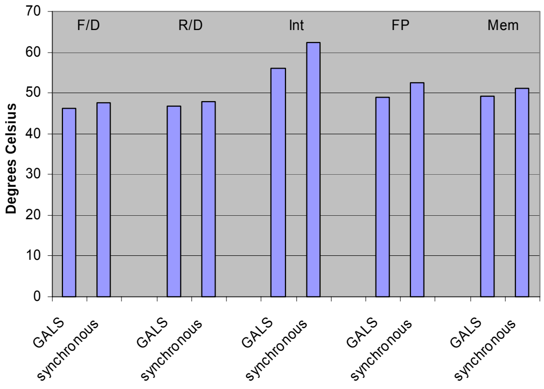


Figure 3: 177.mesa

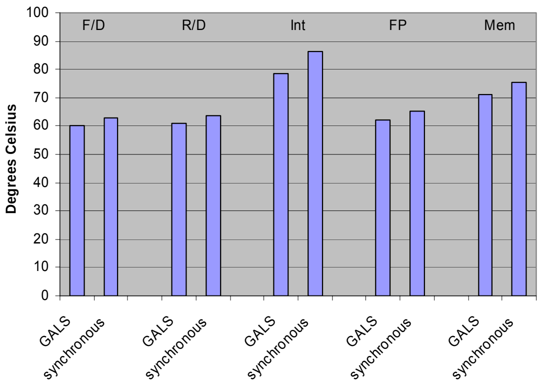


Figure 4: 000.ijpeg

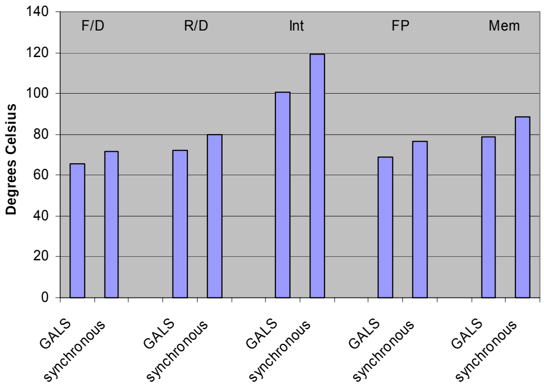


Figure 5: 175.vpr

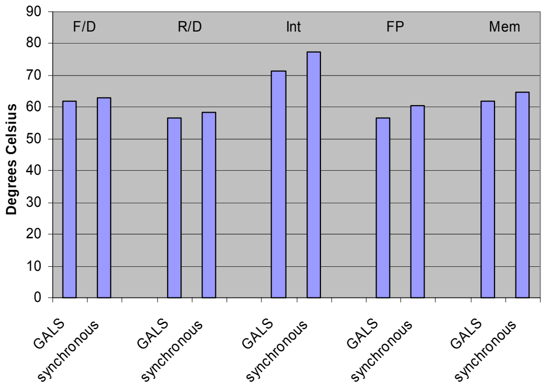


Figure 6: 000.gcc

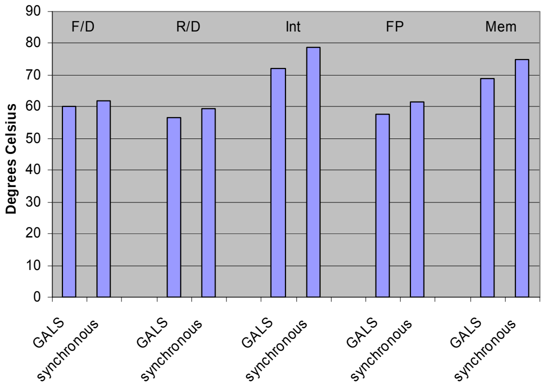
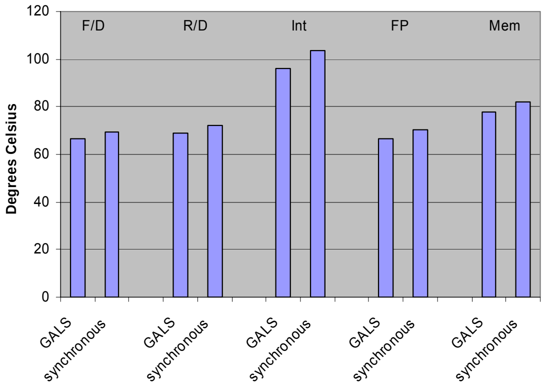


Figure 7: 256.bzip2



A $T_{cp,max}$ that takes into account temperature is found using the following relation:

$$T_{cp,max,\ adjusted} = T_{cp,max} * \text{Temp(region)/Temp(hottest region)} \quad (4)$$

where Temp(region) is the temperature of the region and Temp(hottest region) is the temperature of the hottest region [1]. Hence, when taking into account temperature, the maximum critical path delay is reduced slightly for each region that is not the hottest. Simulations not considering temperature effects produce power statistics based on the assumption that all regions are as hot as the hottest region when in actuality, that is not the case.

Leakage Variation

Because the simulators compute the total power leakage per testbench assuming all regions are the same temperature as that of the hottest region in the synchronous design, the following relation must be used for a more accurate assessment of leakage power per module:

$$\text{Leakage Power} = ke^{-Vt/Temp} \quad (5)$$

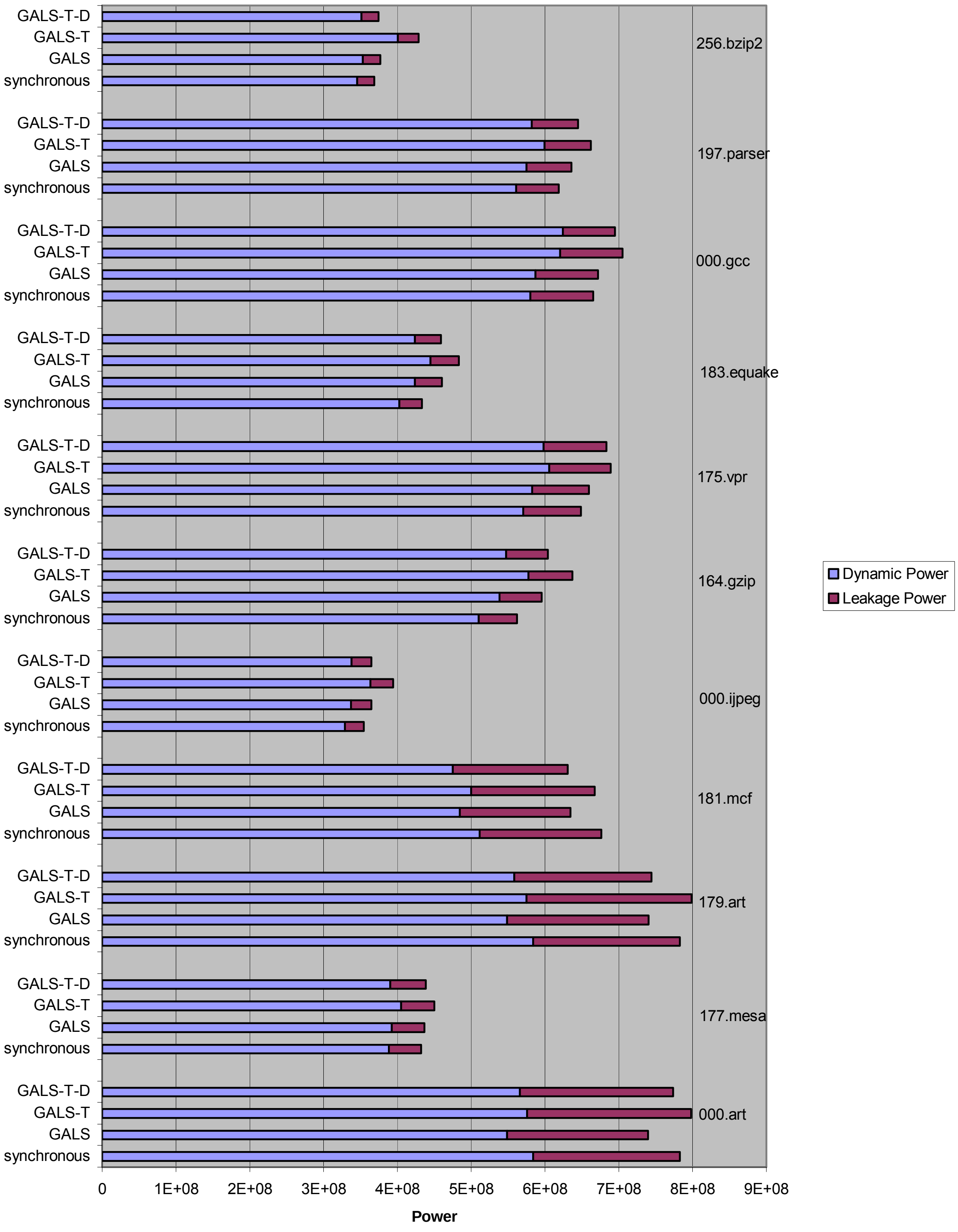
where k is a constant, Temp is the temperature of the region, and V_t is the threshold voltage having normal distribution with $V_{t,0}$ mean, which is 0.2 V in this case, and standard deviation determined by WID and D2D variations [5].

Since the correct power leakage for synchronous in the hottest domain, Leakage1, is given, the correct power leakage for each of the other domains, Leakage2, can be obtained by solving for k as follows:

$$\begin{aligned} \text{Leakage1} &= ke^{-Vt/Temp1} \\ k &= \text{Leakage1}*e^{Vt/Temp1} \\ \text{Leakage2} &= ke^{-Vt/Temp2} \\ \text{Leakage2} &= \text{Leakage1}*e^{Vt/Temp1 - Vt/Temp2} \end{aligned}$$

where Temp1 is the temperature of the hottest region in synchronous and Temp2 is the temperature of the examined region. Finally, total power consumption to be plugged into (1) is obtained by adding the adjusted leakage power of all five domains and the dynamic power consumption per testbench. Figure 8 shows the adjusted power consumption, dynamic power, and total power consumption for each testbench.

Figure 8: Power Consumption



Experimental Results

Figure 9 shows performance for each of the testbenches on synchronous, GALS, GALS with temperature effects, and GALS with voltage scaling and temperature effects.

Figure 9: Performance

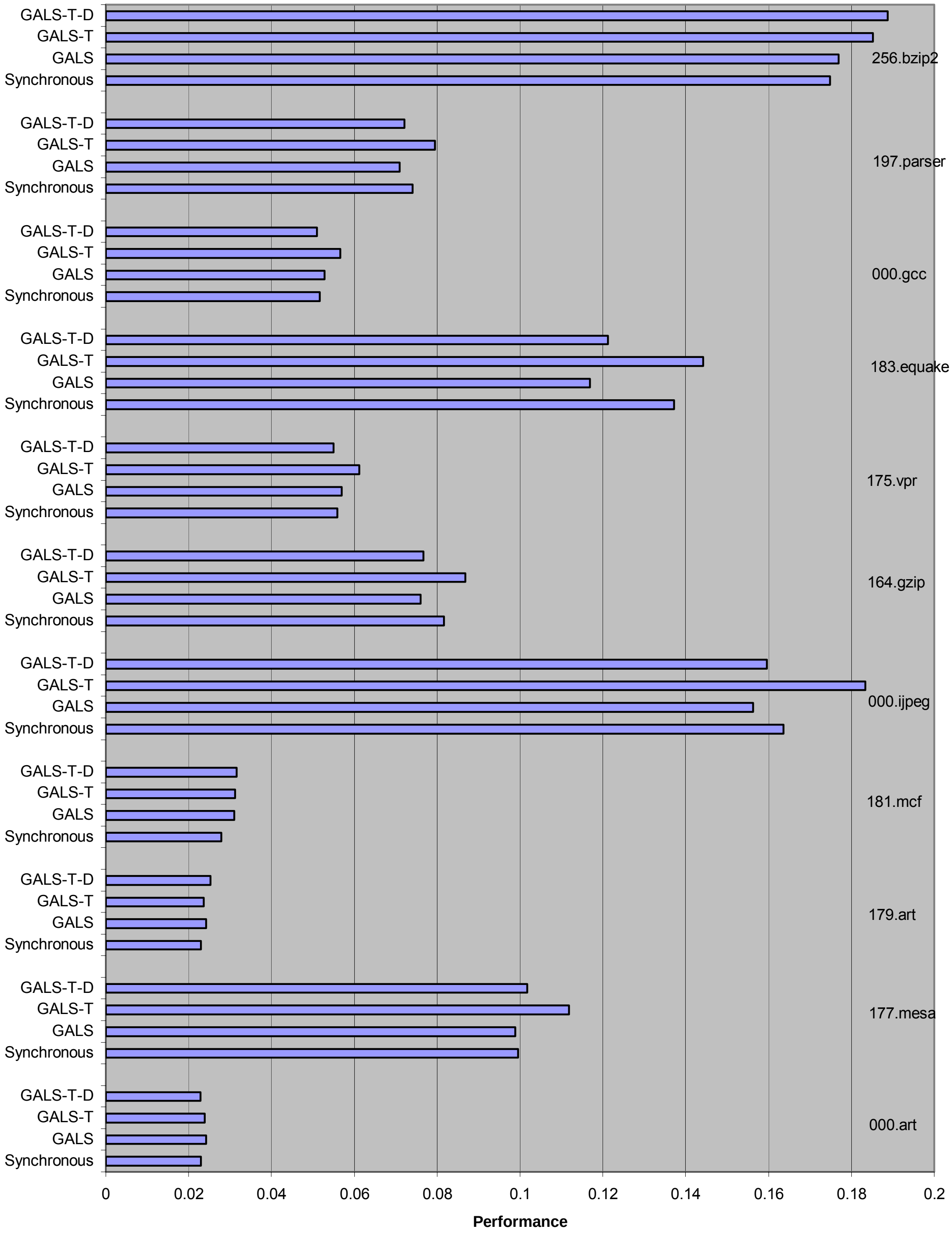
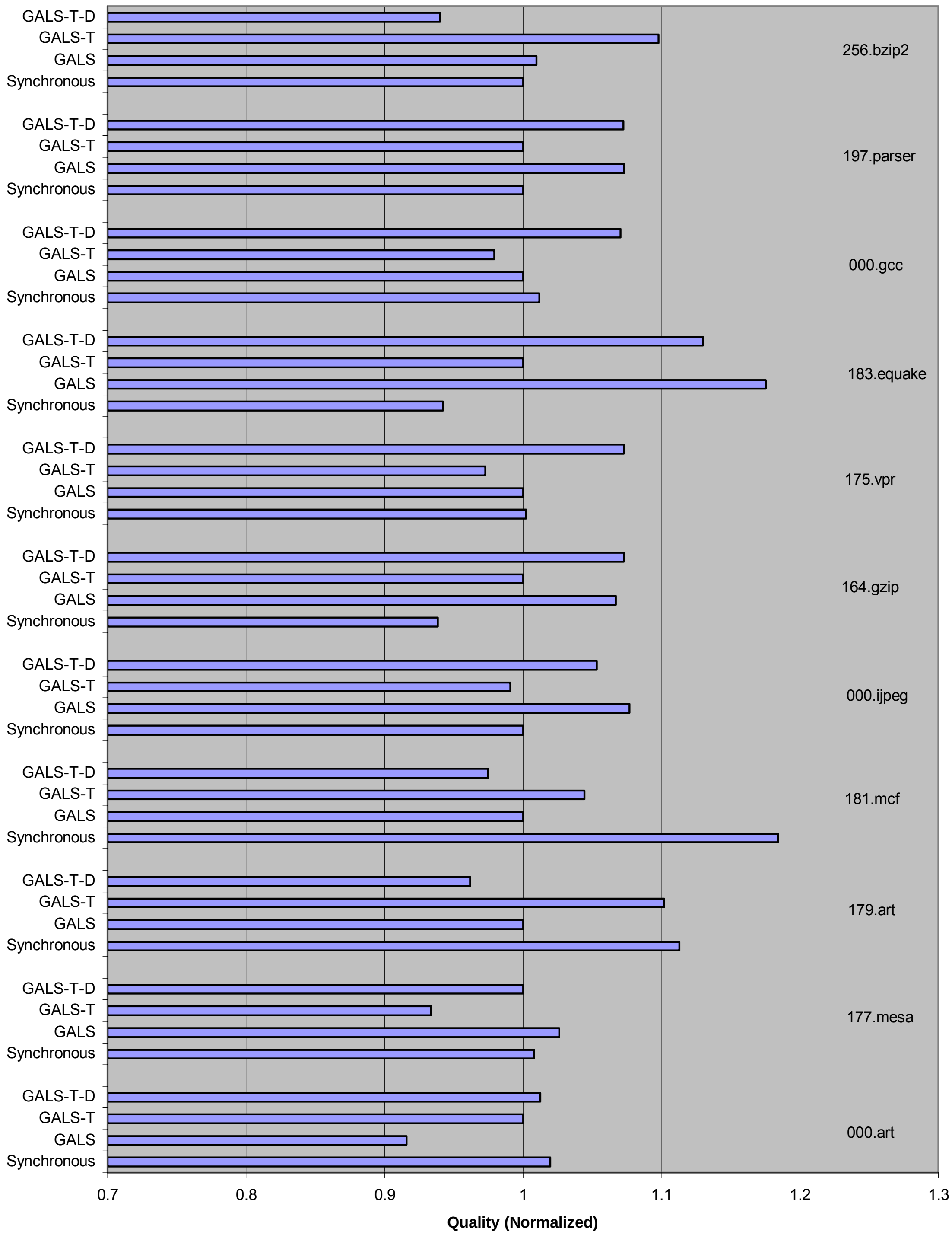


Figure 10 shows normalized quality (1) for each of the testbenches on synchronous, GALS, GALS with temperature effects, and GALS with voltage scaling and temperature effects. Recall that a smaller value for Q signifies better quality design.

Figure 10: Quality



Conclusion

The results show that GALS with thermal considerations with or without voltage scaling do not necessarily result in better quality in terms of energy, performance, and variability. Although four of the testbenches, 181.mcf, 179.art, 177.mesa, and 000.art, show marked improvement when using GALS, the remaining seven testbenches indicate that synchronous is a sufficient, if not optimal, choice.

The initial supposition was that GALS would have a better quality value than synchronous based on the fact that performance increases when different clock speeds are applied to different regions of the architecture and that voltage scaling should decrease power consumption. Although voltage scaling reduces performance in some cases, the initial hope was that the reduction would be small enough to nevertheless ensure a better quality value. Moreover, higher overall temperatures in synchronous suggested greater leakage power which theoretically made GALS appear to have an advantage, but actual results show that leakage power has a minimal effect on the quality metric. Thus, although GALS may be a good design for processors that specialize in .art, .mesa, and .mcf files, the results of this experiment indicate that synchronous still possesses the competitive advantage. A few wins for GALS is not likely to be sufficient for drastically changing the status quo.

Possible future research work includes addressing the impact of wire delay variability as well as the underlying causes for the GALS design’s mediocre quality value.

References

[1] A. Basu, S. Lin, V. Wason, A. Mehrotra, and K. Banerjee, “Simultaneous Optimization of Suypply and Threshold Voltages for Low-Power and High-Performance Circuits in the Leakage Dominant Era,” in Proc. ACM/IEEE Design Automation Conference, June 2004.

[2] S. Borkar, T. Karnik, V. de, “Design and Reliability Challenges in Nanometer Technologies,” in Proc. ACM/IEEE Design Automation Conf., June 2004.

[3] K. A. Bowman, S. G. Duvall, J. M. Meindl, “Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration,” in IEEE Journal of Solid-State Circuits, vol. 37, no. 2, Feb. 2002.

[4] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A Framework for Architectural-Level Power Analysis and Optimizations,” in Proc. ACM Intl. Symp. On Computer Architecture, June 2000.

[5] J. A. Butts, and G. S. Sohi, “A Static Power Model for Architects,” in Proc. Intl. Symp. On Microarchitecture, pp. 191-201, Dec. 2000.

[6] M. Eisele, J. Berthold, D. Schmidt-Landsiedel, R. Mahnkopf, “The Impact of Intra-Die Device Parameter Variations on Path Delays and on the Design for Yield of Low Voltage Digital Circuits,” in Proc. ACM/IEEE Intl. Symp. On Low Power Electronics and Design, Aug. 1996.

[7] A. Iyer and D. Marculescu, “Power efficiency of Multiple Clock, Multiple Voltage Cores,” in Proc. IEEE/ACM Intl. Conf. On Computer-Aided Design, San Jose, CA, Nov. 2002.

[8] Kenneth Crow DRM Associates, 2002. Variability Reduction. http://www.npd-solutions.com/vr.html. Accessed 2005 July 25.

[9] D. Marculescu, E. Talpes, "Variability and Energy Awareness: A Microarchitecture-Level Perspective," in Proc. ACM/IEEE Design Automation Conference, (DAC), Anaheim, CA, June 2005.

[10] K. Niyogi, D. Marculescu, "Speed and Voltage Selection for GALS Systems Based on Voltage/Frequency Islands," in Proc. ACM/IEEE Asian-South Pacific Design Automation Conference (ASPDAC), Shanghai, China, Jan.2005.

[11] G. Semeraro, D. H. Albonesi, S. G. Dropsho, G. Magklis, S. Dwarkadas, and M. L. Scott, “Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture,” in Proc. ACM Intl Symp. On Microarchitecture, Nov. 2002

[12] E. Talpes and D. Marculescu, “A Critical Analysis of Application-Adaptive Multiple Clock Processors,” in Proc. ACM/IEEE Intl. Symp. On Low Power Electronics and Design, Aug. 2003.