2018 24th IEEE International Symposium on Asynchronous Circuits and Systems

Case Study of Process Variation-based Domain Partitioning of GPGPUs

Shomit Das Michael LeBeane Bradford Beckmann Greg Sadowski
Advanced Micro Devices, Inc.

Abstract— We can unlock higher performance in general purpose graphics processing unit (GPGPU) chips by adhering to local environment conditions while setting a timing (clock) reference. We propose partitioning the GPU chip into smaller independent domains to enable finer control of the chip voltage frequency (V/F) settings. Post silicon (Si) analysis is used to classify the domains as fast, 'typical', or slow. This information is then provided to the system management unit (SMU) to enable the appropriate settings for power optimization. In this paper, we present an initial analysis of power/performance associated with fine-grain domains in GPUs.

I. Introduction

Due to process variation (PV), there can be a significant difference in transistor characteristics, especially across a large die. This variance can only be accurately determined on silicon (Si) postfabrication. In case of a single, global frequency control, the clock frequency is determined, in part, by the worst case critical path in the entire chip. Voltage/Frequency (V/F) settings for power and performance optimization are also currently limited by full-chip considerations. This worst-case limitation introduces higher margins for minimum voltage (Vmin) and maximum frequency (Fmax) parameters. In a throughput oriented, highly parallel architecture, such as GPUs, this constraint can lead to performance loss. The effect can be even more pronounced for newer technology nodes on large dies that have a high variance in transistor characteristics. To combat this, we propose a multi-frequency configuration of a GPU with distributed V/F control. The GPU is divided into domains and each domain is characterized based on local PV conditions. The multi-frequency operation in GPGPUs leads to some interesting trade-offs, as described in this paper.

II. Background

Process variation is based on manufacturing characteristics of the circuit die and remains relatively static throughout the lifetime of the die. There are several methods to model PV, and certain statistical, circuit-based, or system-based methods to mitigate its effect [1]. Variation aware dynamic voltage frequency scaling (DVFS) has been studied in the context of multicore systems [2]. Similarly, the effect of fine-grained clocking on GPU power supply noise has also been presented [3]. This paper studies the effect of distributed V/F control on benchmark application power and performance on a GPGPU. To enable V/F islands on GPU, several circuit solutions are required. Fast low dropout oscillators and clock stretching techniques are described by Singh *et al.* [4]. Fine-grained adaptive clocks have also been presented by Cortadella *et al.* [5]. Also, there have been recent papers describing circuits for low overhead clock domain crossing [6], [7].

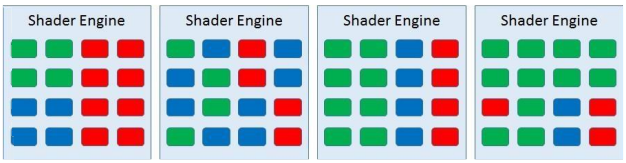

Fig. 1: Fine-grained CU Frequency Mapping

III. Tuning for Process Variation

In this work, we consider the availability of three independent clocks. The compute units (CUs) are distributed in three domains, each corresponding to one clock. The clock frequencies can operate in of three modes – slow (s), typical (t), and fast (f). Each domain has independent control over its V/F. Post-Si measurements using ring oscillators (ROs) are performed to classify the domains into 's/t/f' buckets. The settings can be selected to optimize power/performance of the GPU system. Depending on the objective, a domain can be operated at a lower voltage setting to balance its performance with the other domains; or it can be run at a high voltage setting to maximize its throughput.

Once the post-Si characterization has been completed, the information is stored in the SMU. Certain branches in the clock and power distribution trees can be activated based on the hardware settings applied to the chip. The power states in different partitions are modified, so that each state may mean somewhat different voltage and frequency. The power management runs on the top of the partitions settings. Fig. 1 shows a toy example of CUs being mapped to slow (blue), typical (green), or fast (red) clocks. The actual mapping may be coarser grained depending on the overhead of clock domain crossings and power distribution.

The timing imbalance introduced by the non-uniform frequencies can be exploited. The work dispatching module in the GPU is responsible for managing workloads across compute units. Different kernels can be mapped to different 'speed' buckets to balance parallel tasks and boost critical ones. Similarly, high priority tasks can be scheduled to better performing CUs.

IV. Simulation Framework

To evaluate our work, we employ a simulation framework based on the open-source gem5 simulator [8] including the AMD public GPU compute model. We have configured our hardware to resemble an APU system based on a forward looking exascale design [9]. All memory accesses are coherent through a directory-based coherence protocol and occur in a unified virtual address space.

978-1-5386-5883-3/18/$31.00 ©2018 IEEE
DOI 10.1109/ASYNC.2018.00037

TABLE I: Simulation configuration.

GPU Configuration	
Wavefronts	40 (each 64 lanes)
D-Cache	16kB, 64B line, 16-way
I-Cache	32kB, 64B line, 8-way
L2-Cache	2MB, 64B line, 16-way
DRAM	DDR3, 4 Channels, 800MHz

TABLE II: Evaluated Workloads.

CoMD	DOE Molecular-dynamics algorithms
LULESH	Hydrodynamic simulation
SNAPC	Discrete ordinates neutral particle transport app
HPGMG	Ranks HPC systems

The baseline GPU models a single clock domain for all components. We simulate an APU-style system with coherent, shared physical memory between the GPU and the CPU. Each CU contains 4 SIMD-16 engines, each with 10 active wavefronts. We simulate a system with 24 CUs and small input sizes to decrease simulation times. Table I illustrates the specific configuration for the major components of our infrastructure.

Power analysis is done using a trace-based simulator [10] that gathers data from performance monitors while running the application on commodity hardware. Using an offline machine learning algorithm, the simulator provides guidance on the expected power consumption.

V. Results and Conclusions

Table II describes the workloads used in this analysis. Except for CoMD, the workloads are memory-intensive applications. The relationship between the 'slow', 'typical', and 'fast' frequencies can be described as 1X, 1.25X, and 1.5X respectively. For ease of simulation, we divide the CUs into three groups that are arbitrarily mapped to one of the three clock sources. This is not a requirement, and there are several possible combinations of CU domain and clock source. Voltage projections are made in accordance with the method in Zhang *et al.* [10].

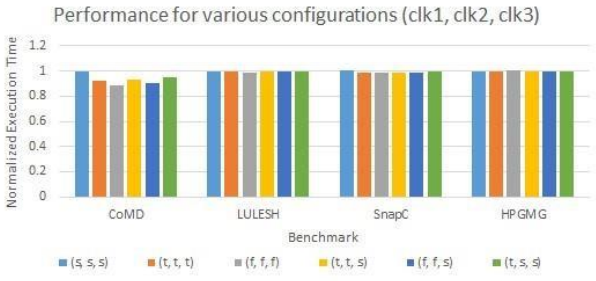

Fig. 2: Performance for different mappings

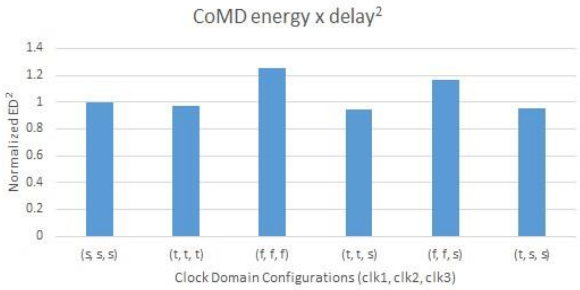

Fig. 3: Energy-Delay squared product for CoMD

It can be seen that this method produces noticeable results only for compute-intensive applications. This intuitively makes sense, since boosting CU frequency will have little effect on memory-stalled kernels. Fig. 2 shows the performance effect of fine-grained frequency control. It shows that the cases (t, t, t), (t, t, s), and (f, f, s) have similar performance characteristics. Therefore, purely from a performance standpoint, any of these configurations may be used. However, the energy consumed in these settings may differ widely. As can be seen from Fig. 3, the (t, t, s) configuration provides the best Energy * Delay^2 metric for the CoMD benchmark.

In the absence of hybrid clock frequency distribution, the only options available are (s, s, s), (t, t, t), or (f, f, f). It is clear that the Energy * Delay^2 product can be lowered by using a distributed frequency control.

VI. Future Work

There are several future directions for research. Intelligent task scheduling based on CU performance is possible. Also, we have only simulated a static CU-frequency mapping. DVFS based techniques are possible with distributed V/F control. Further, several schemes for dynamic V/F adjustment can be explored. Finally, a more rigorous treatment of the latency and power cost of clock domain crossing is needed for more accurate results.

References

[1] S. S. Sapatnekar, "Overcoming variations in nanometer-scale technologies," IEEE JETCAS, 2011.

[2] S. Herbert and D. Marculescu, "Variation-aware dynamic voltage/frequency scaling," in IEEE HPCA, 2009.

[3] D. A. Kamakshi, *et al*., "Modeling and analysis of power supply noise tolerance with fine-grained gals adaptive clocks," in IEEE ASYNC, 2016.

[4] T. Singh, *et al..,* "Zen: An energy-efficient high-performance times 86 core," IEEE JSSC, 2018.

[5] J. Cortadella, *et al*., "Ring oscillator clocks and margins," in IEEE ASYNC, 2016.

[6] B. Keller, *et al.*, "A pausible bisynchronous fifo for gals systems," in IEEE ASYNC, 2015.

[7] A. M. S. Abdelhadi, *et al.,* "Interleaved architectures for highthroughput synthesizable synchronization fifos," in IEEE ASYNC, 2017.

[8] N. Binkert, *et.al*., "The gem5 simulator," SIGARCH Comput. Archit. News, 2011.

[9] M. J. Schulte, *et.al*., "Achieving exascale capabilities through heterogeneous computing," IEEE Micro, 2015.

[10] D. Zhang, *et.al.,* "Top-pim: Throughput-oriented programmable processing in memory," in ACM HPDC, 2014.