PowerRed: A Flexible Modeling Framework for Power Efficiency Exploration in GPUs

Karthik Ramani[‡], Ali Ibrahim [†], Dan Shimizu [†]
[‡] School of Computing, University of Utah
[†] AMD Inc.

Abstract

The tremendous increase in the complexity (memory and computations) of graphics models have been supported by a similar increase in the computational resources available in graphics processing units (GPUs). The inherently high parallelism of such systems has led to a significant increase in power dissipation, thereby necessitating expensive cooling solutions. In addition, general purpose processing on such specialized architectures poses new problems yet opens avenues for power optimizations at the architectural level. In this paper, we present a modular architectural power estimation framework that will help GPU designers with early power efficiency exploration. The framework employs a combination of analytical and empirical basic blocks for power estimation. In addition, we associate an area model to the different units within a GPU to estimate power across the interconnects. We explore two cases of optimizations to demonstrate the utility of the framework. The results show that simple bus encoding reduces the power of a medium/high activity texture cache global bus interconnect by 15-30% while exploring the trade-offs between power savings, data activity, and interconnect length. We also showcase the power benefits (around 13%) of optimally sizing repeaters and spacing between them on the same global bus.

Keywords: *GPU Power Modeling, Architectural Exploration, Power optimization, Encoding.*

1 Introduction

In recent years, there has been a tremendous increase in the memory and computational requirements of graphics models (e.g. polygon rendered models, etc). Advances in process technologies and the availability of abundant transistor budgets have facilitated a similar growth in the number and functionality of resources available on a Graphics Processing Unit (GPU). With the emergence of commercial avenues such as stream computing and GPGPU, the evolution in graphics hardware is likely to keep up with the pace of the application requirements.

The increase in the sheer complexity and performance of GPUs across generations also incurs a commensurate increase in the power dissipation of such systems and necessitates expensive cooling solutions. In addition, the relative lack of understanding of the power impact of executing new applications exacerbates the problem. This provides new opportunities for researchers to employ power modeling and evaluate optimizations at different levels of design. In this study, we explore an early stage power estimation framework that employs a modular design for architectural and circuit based power optimizations and complements the low level design tools.

Early stage power estimation for CPUs has been a popular research area in the academic community. Wattch [6], a power simulator employs parameterizable analytical models of units like memory structures, clock tree network, and execution units, etc. to estimate dynamic power dissipation in a CPU. Other models (SimplePower [12], TEM2P2EST [7]) employ empirical models based on known circuit implementations for better accuracy. These models tradeoff ease and speed of simulation for estimation accuracy and/or scalability across process technologies. In addition, these models do not accurately model power dissipation for wires and interconnects. Given the unique design issues in GPUs, the market need for extremely tight design schedules, and the lack of accurate but flexible power models, our ongoing study attempts to address the above issues.

In the first part of the paper (Section 2), we present a unified architectural power estimation framework that employs the combination of two different models. We employ analytical models for regular and predictable structures like memory, FIFOs, etc. Power dissipation for complicated structures like execution units, control logic, etc., depend greatly on the implementation and hence, we employ empirical models based on low level RTL based power models. Interconnect power dissipation contributes to a significant fraction of total chip power [10] and hence, we employ an area cost to build models for interconnects based on the methodology described in [5, 13].

In the second part of the paper, we demonstrate the utility of the framework by exploring two techniques for interconnect power optimizations. First , we show how inversion encoding (Section 3) on global interconnects saves power at minimal latency overheads. Second, we explore the power impact of changing repeater sizes and spacing between them for latency tolerant global buses. Finally, we explore the architectural impact (Section 4.2) of such optimizations on an interconnect fabric with-in the texture cache. In summary, this ongoing study provides the following contributions:

- We discuss a flexible architectural power modeling framework that employs analytical, empirical, and area based models to provide a foundation for circuit and architectural optimizations in a GPU.
- We demonstrate the utility of the framework by employing the optimizations of bus encoding and repeater sizing/spacing and study their impact on the power dissipation of a bus interconnect fabric in a texture cache.

2 Power Modeling Framework

The goal of the power modeling framework is to estimate power at different levels of detail. Each level trades off estimation accuracy for speed of simulation and hence, facilitates design space pruning to explore a variety of architectures. At the top most level, we employ coarse-grained macro-architectural power estimation at the granularity of large structures (memories, queues, arithmetic units, etc.) to estimate power for different designs at modest accuracy. Once we prune the design space to a few candidate choices, we employ more detail for improved accuracy, thereby providing flexibility. Figure 1 shows the high level overview of the power modeling framework and its interface against an internal cycle accurate performance model or a benchmarking suite for stand-alone power model testing (shown by a multiplexer-like structure). The performance model provides the capability to simulate a modern GPU and employs counters and queues to track information (data activity, control values, etc.) necessary for power estimation. This information is delivered to the power model through a trace interface. On the other hand, statistical benchmarks can also deliver similar information to the power model. Process related information (transistor diffusion, voltage, frequency, etc.) is delivered to the power model through the technology file.

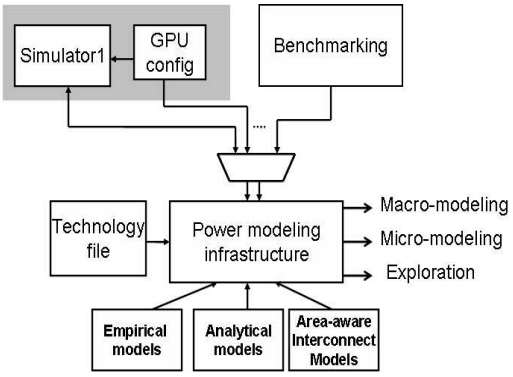

Figure 1. Power Modeling Infrastructure, Interfaces, and Usage Modes

In CMOS circuits, the dynamic power dissipation (P_d) is defined as

$$P_d = aCV^2f,$$

where f and V are the frequency and voltage of operation of the circuit respectively, C is the load capacitance, and a is the switching activity factor. The

Structure	Analytical Model	Empirical Model
Cache	dynamic, leakage	dynamic, leakage
FIFO	dynamic, leakage	dynamic, leakage
Register file	dynamic, leakage	dynamic, leakage
Bus	dynamic, leakage	-
Crossbar	dynamic, leakage	dynamic, leakage
Arbiter	dynamic, leakage	dynamic, leakage
Bus encoder	-	dynamic, leakage
Bus decoder	-	dynamic, leakage
Arithmetic	-	dynamic, leakage
control	-	dynamic, leakage
data path	-	dynamic, leakage

Table 1. Types of models available for the different structures within a GPU

performance model estimates the activity factor for structures such as FIFOs, buses, caches, etc. For internal circuits, where the modeling is not accurate enough to calculate activity factor, we assume values as recommended by [6, 5]. Leakage power, which also contributes to a significant fraction of total power dissipated in CMOS circuits today, is determined using analytical models for memory structures (similar to HotLeakage [14]) and empirical table lookup models for all other circuits.

Analytical Models These types of models are employed for parameterizable regular structures and we employ a methodology similar to Wattch [6] to build the models for various structures. Each of the structures are broken into different constituent stages and equivalent RC models are built for each of them. Finally, we sum the capacitances for each of the stages and then calculate the dynamic power for the structure. This type of modeling is relatively well understood and models power for wires internal to a circuit as well. However, they do not model global interconnects and is our motivation for employing analytical models for global interconnects. Table 1 shows the type of modeling available for each type of structure within a GPU. For structures represented by both the models, the choice of model is dependent on the required level of accuracy, speed of simulation, and the required level of detail.

RTL based Empirical Models for Dynamic and Leakage power This type of modeling is employed for all structures where the underlying implementation varies across different units and in structures where it is difficult to build parameterizable analytical models such as control circuits, custom datapath, arithmetic units, etc. Power dissipation for such structures is determined by the activity factor of data and the control signal that determines the type of operation performed in the structure. For example, in an FIFO circuit, the control signal (push, pop) determines the operation type and the activity factor of input data determines the switching activity in the circuit. Hence, we estimate the power dissipation for various activity and control values using commercial low level power simulators (PowerTheater [1] for RTL code and PrimePower [2] for net-list) and form a table for the circuit. The table contains both the dynamic and the leakage power for the circuit. For power estimation, we perform a table lookup based on the control signals and activity. Table 2 shows an example for a FIFO that is modeled empirically.

Interconnect Power Models Power dissipation on interconnects contributes to a major fraction of the total chip power [10]. Hence, we employ an analytical model of a bus to estimate power dissipation for global and local buses that contribute a significant fraction to the total power of a macro-block. For global buses, we assume a methodology similar to [4], with appropriately sized buffers and repeaters, inter-buffer distances, etc., depending on the delay and power requirements. In addition, we compute the length of the bus using the chip floor-plan assum-

ing that the bus runs from the center of one block to that of the other. This simple technique helps us in exploring various layouts for a given architecture and can be further improved. For other interconnects, we employ analytical models similar to [13] for matrix-based crossbars, arbiters, and empirical models for a multiplexer based crossbar.

Communication Across Models and Exploration Implementing a flexible but robust framework that models power at different levels of detail requires the establishment of clean interfaces across the performance model, power model, and the configuration boundaries. While the goal of the project is to integrate the power model with the performance model in a seamless manner so as to perform on-line tracking, currently, the model is built as a stand alone tool in C++ and uses a trace format to read the activity data from the performance model. The processor configuration and the technology configuration are provided to the power model at the start of simulation for initial setup. The user can also choose the level of detail for power modeling or for exploration. An example of exploration would be to investigate different designs for a global bus depending on the performance requirements. The user mentions the latency requirement for the bus either through the performance model or on the command line and the power model will explore different options using techniques discussed above and quantify the performance and power impacts of different choices.

3 Exploration

The modeling framework discussed in the last section is employed to showcase different case studies of power optimizations at the architectural level. we first discuss some offline optimization techniques followed by on-line optimization techniques. It should be noted though that we only evaluate the power impact of offline optimizations due to the current status of the existing infrastructure. The first two case studies form part of a wide interconnection interface that delivers data from the texture cache to the GPU back-end units. We investigate how encoding reduces the switching activity of global buses to reduce the total power and then explore the optimal repeater sizing and spacing for such interconnects.

Optimization I - Bus Invert Encoding The observation that medium to high activity global buses dissipate a significant fraction of their total power due to data switching necessitates techniques that can reduce such activity. At the same time, the technique should have minimal overhead in terms of latency and area. Hence, we investigate simple bus inversion encoding on global buses in the GPU and quantify their power impact on wide data buses. Given an n bit data bus, and we add one additional encoding bit to get an $n + 1$ bus. In the first step, the encoder computes the hamming distance between present cycle bit values and previous cycle bit values. If the computed distance is greater than n/2, we invert the all bits to reduce the activity of switching. Thus, power savings are obtained as a result of a reduction in switching activity. There are two potential problems with this technique: it can be observed that the savings diminish as the bus size increases due to the increase in the tails of the hamming distance, and the encoder, decoder, and the encoding bits add area, latency and power overhead to the bus. The former can be solved by partitioning a large bus into smaller encoded sub-buses. Recent studies have shown that a sub-bus size of 8 bits [9, 3, 11] represents a good solution in terms of minimal latency and area overheads. In our studies, such a configuration doesn't affect the clock frequency though the power overhead is around 10-15% for a 256 bit global bus depending on the bus length.

To model the global bus, we employ empirical models for the decoder and encoder units and use the analytical power model for the global bus with delay optimal repeater sizing and spacing.

Optimization II - Power Optimal Repeater Size and Spacing Industrial designs for global buses employ buffers separated by a fixed distance for signal restoration and repeaters after a fixed number of buffers. The repeaters acts as pipeline stages and usually incur more power than the buffers. The sizing of repeaters and buffers, and the spacing be-

Activity Factor	Control (push, pop)	Dynamic Power	Leakage Power
0.2	00	0.105	0.118
1.0	00	0.105	0.118
0.5	01	1.610	0.122
.	.	.	.
.	.	.	.
0.7	11	1.610	0.122

Table 2. Empirical Table for a FIFO

tween buffers are optimized for delay and the devices are conservatively over-sized, leaving a sufficient delay margin for power optimizations. We employ the methodology proposed by Banerjee *et al.* [5, 4] and apply the technique in GPUs with latency tolerant interconnect fabrics. For example, the interconnection interface between the texture cache and the GPU back-end is a deeply pipelined fabric that is normally over-provisioned. We exploit the latency margin of such global buses to reduce the size of buffers and repeaters and increase the spacing between the buffers to create power optimal designs that have minimal latency overheads. In addition to employing the repeater sizing technique discussed in [5], we also vary the size of the buffers that drive the signal between two repeater stations.

Proposal I: Exploring Global Bus Interfaces for Texture Caches In a modern GPU, data delivery from the texture cache to the back-end consists of a wide global bus interface that delivers data in a SIMD fashion. The interface is a deeply pipelined interface and is a good example of a latency tolerant interconnect where the impact of the above proposed optimizations can be studied. It should also be noted that global bus interfaces are employed frequently at different parts of the chip (shaders, texture cache hierarchy, interconnection between blocks, etc.) and contribute a significant fraction (around 40%) to the total interconnect power dissipated in the chip. Hence, we employ our simulation framework to explore the power impact of bus encoding and optimal repeater insertion.

Proposal II: Exploring Crossbars in Shaders/Texture Cache Crossbar implementations are commonly used in GPUs for providing functionality for data alignment, data width selection, and for swizzle operations. One avenue of exploration is to compare various implementations of crossbars for latency and power dissipation characteristics using offline power estimation. Another proposal exploits the fact that current designs employ fully connected circuits in places where they are not required and hence, designs are conservative. In blocks where the design need not be fully connected, the performance model can study the data patterns within the application suite and can reduce the number and functionality of interconnect paths to service only the requirements, thereby saving area and power.

4 Results

4.1 Experimental Methodology

Our power models are employed to simulate a GPU running at a frequency of 750 MHz (1.1V) on a 55nm process. For simulating power across the interconnection networks and buses, we employ the analytical model for repeaters, buffers, and wires, while the decoder and encoder circuitry employs empirical power estimates. Validation of the models at the block level and the chip level is an important task and we are currently investigating such an approach.

For simulating the traffic on the interconnects, we employ statistical benchmarking including normal and uniform random distributions for ten million cy-

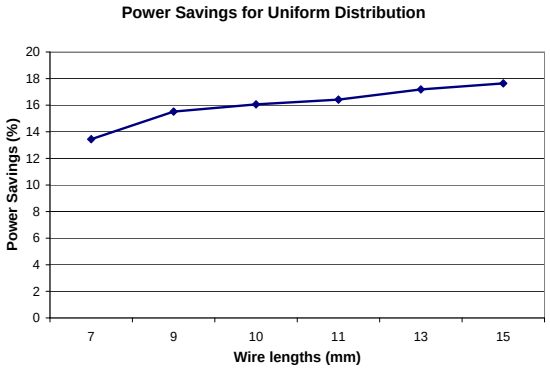

Figure 2. Power Savings for Uniform Distribution for a 7 mm Global bus

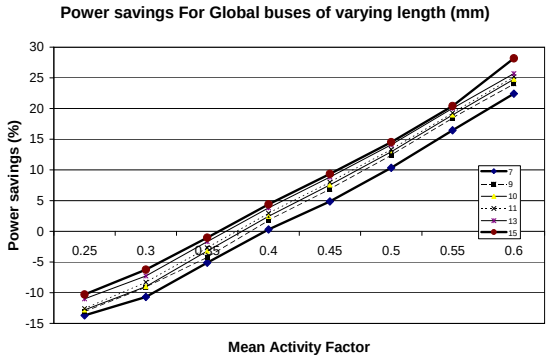

Figure 3. Power Savings for Normal Distribution

cles. The results are averaged over fifteen runs to ensure consistency and repeatability. We would like to eventually execute real world graphics programs, but for simulating behavior across interconnects, random distributions approximate the bursty nature of data traffic reasonably well.

4.2 Evaluation

In this section, we evaluate the impact of optimizations I and II on the global bus interface in the texture cache (proposal I). As mentioned previously, evaluating proposal II requires on-line performance and power estimation for application analysis, and hence, is left as part of future work.

Impact of Inversion Encoding on Texture Cache Interface We now evaluate the power impact of inversion encoding (Case Study I) under uniform random and normally distributed traffic. Figure 2 shows the power savings for a 256 bit global bus for different lengths. It should be noted that for any given length, the power savings are computed with the un-encoded bus of the same length as the baseline. Hence, the encoded bus models incur the power overhead of the encoder, decoder and the extra bits employed for encoding. It can be observed from the plot that inversion encoding provides between 15-20% power savings as compared to un-encoded buses. This is mainly attributed to the reduced switching activity of the encoded bus during periods of high activity. As the length of the bus increases, the power savings increase due to two reasons: the overhead of the decoder/encoder circuitry is amortized over the length of the bus and a greater reduction in the switching of a longer bus.

Figure 3 shows the power savings for normally distributed data traffic for various mean activity factors. The general observation is that the power savings increases as the activity factor and the length increases. Global buses with low activity (around 0.2) are affected adversely by bus encoding. The power (mainly leakage) incurred in the encoding and the decoding circuitry adds to the power dissipation while not affecting the switching activity of the bus. The power overhead of encoding is as high as 13% for a 7mm bus and 10% for a 15mm bus. Global buses with medium to high activity ($\geq$0.35) benefit significantly from encoding. It can be observed that at a mean activity of 0.5, we save power by as much as 10% for a 15 mm global bus. For higher activity buses, the power savings obtained increases rapidly to as much as 22-28% for various lengths. Encoding thus is inefficient for low activity buses and beneficial for wide global buses with medium to high activity. It can also be observed that the break even point (when power saving equals the power overhead) moves to the left (less activity) as the length of the bus increases. This is due to the amortization of the encoding overhead as the length increases.

Impact of Optimal Repeater Sizing and Spacing for Power on Texture Cache Interface Table 3 shows the power savings obtained in a 7 mm global bus when the repeater size and spacing are varied in a 55nm process. We employ three different sizes of buffers(D16,D12, and D8) and two different sizes for

repeater stations (RP12, RP8). In our studies, we only show results for combinations that minimally affect timing and circuit level issues. It can be observed that we can achieve 11-13.7% power savings for latency overheads of less than 5% without affecting issues such as signal integrity, crosstalk, etc. Given the over-constrained designs in today's GPUs, the power savings are likely to be achieved without any effect on the clock frequency. Though the power savings are significant given the minimal or zero latency and area overhead, our numbers are not as high as reported in [5] due to the following reasons: previous studies assume a greater contribution of leakage to total power dissipation($\geq$ 50%) and do not include the power overheads of buffers that drive the signals in addition to the repeater stations.

5 Related Work

Power Models Brooks *et al.* [6] developed Wattch, the most popular power simulator for CPUs. This academic simulator employs analytical models for various regular structures based on implementations in the Alpha 21264 and our methodology for building analytical models is similar. The difference is that we employ internal models used for implementation. Also, for custom regular structures such as high density register files and certain memories, we employ empirical models for greater accuracy. SimplePower [12] is another academic simulator that employs empirical models. TEM2P2EST [7], an industry simulator employs a combination of analytical and empirical models for power and temperature estimation in CPUs. To our knowledge, Qsilver [8] is the only simulation framework that allows for architectural optimizations in GPUs. The difference between these simulators and our framework is twofold: first, Simplepower, Wattch, and TEM2P2EST are geared towards CPUs while our framework is for GPUs. Secondly,the above simulators do not include power models for interconnects, and with interconnects incurring a major fraction of power dissipation in future processors, our framework provides a single unified framework for full chip power simulation at a greater level of accuracy.

Interconnects This work builds on [4] that explores repeater sizing and spacing for long interconnects. To our knowledge, this is the first work that explores the power impact of encoding on long interconnects in GPUs. In addition, we also explore different techniques of power optimization across various interfaces in the design of a texture cache.

6 Conclusions and Future Work

As processing requirements for graphics models increase rapidly, GPU resources will also scale up in number and functionality. A linear increase in resources and the addition of new functionality exacerbates the problem of power dissipation and further heat removal, necessitating expensive cooling solutions. Simulation is a very important tool and early stage power estimation is essential to study the impact of resource and functionality addition. Towards this goal, we have presented our on-going work for a flexible architectural power modeling framework that trade-offs accuracy and speed of simulation at different levels. A combination of macro and micro-architectural modeling employs analytical and empirical models estimates power at sufficient accuracy to complement low level models. We demonstrate the effectiveness of the framework in exploring how encoding on long buses can save as much as 15-30% power at medium to high activity levels. Our study also demonstrates the power impact of varying the size of repeaters and buffers for long interconnects. Overall, this study demonstrates the effectiveness of the modeling framework in exploring the architectural impact of various optimizations.

This ongoing study requires further enhancements to improve it robustness. On-line power estimation is currently underway. Once on-line power estimation is ready, workload characterization is required to identify bottlenecks for power and to investigate architectural solutions. The interconnect based area models will be further enhanced to build a temperature model for predicting transient chip temperatures. Validation is required at the chip level to guarantee the accuracy of power estimation.

Buffer-Repeater size/Inter-buffer distance(mm)	0.5	0.6	0.7	0.8
BUFFD16-RP12	-	4.46	5.53	8.33
BUFFD12-RP8	4.69	8.03	9.14	11.14
BUFFD8-RP8	8.7	11.20	12.03	13.70

Table 3. Power savings for power optimal repeater insertion

7 Acknowledgments

We thank the anonymous reviewers for their constructive and helpful feedback. We also thank Steve Presant, AMD for his help in setting up the fabrication process related information. Finally, we thank Professors Al Davis and Rajeev Balasubramonian for reviewing our drafts.

References

[1] Powertheater. http://www.sequencedesign.com.

[2] Primepower. http://www.synopsys.com.

[3] B. A.R., J. Zhang, and Q. Qiu. Low power bus encoding with an adaptive hybrid algorithm. In *43rd ACM/IEEE Design Automation Conference*, pages 987–990, 2006.

[4] R. Balasubramonian, N. Muralimanohar, K. Ramani, and V. Venkatachalapathy. Microarchitectural Wire Management for Performance and Power in Partitioned Architectures. In *Proceedings of HPCA-11*, February 2005.

[5] K. Banerjee and A. Mehrotra. A Power-optimal Repeater Insertion Methodology for Global Interconnects in Nanometer Designs. *IEEE Transactions on Electron Devices*, 49(11):2001–2007, November 2002.

[6] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA*, pages 83–94, 2000.

[7] A. Dhodapkar, C. H. Lim, and G. Cai. TEM2P2EST: A Thermal Enabled Multi-Model Power/Performance Estimator. In *Proc. Workshop on Power-Aware Computer Systems (PACS'00), Cambridge, MA*, Nov. 2000.

[8] J.W.Sheaffer, K. Skadron, and D. Luebke. Studying thermal management for graphics-processor architectures. In *International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2005.

[9] C.-G. Lyuh and T. Kim. Low power bus encoding with crosstalk delay elimination. In *15th IEEE International ASIC/SOC Conference*, pages 389–393, 2002.

[10] N. Magen, A. Kolodny, U. Weiser, and N. Shamir. Interconnect Power Dissipation in a Microprocessor. In *Proceedings of System Level Interconnect Prediction*, February 2004.

[11] U. Narayanan, K.-S. Chung, and T. Kim. Enchanced bus invert encoding for low-power. In *International Symposium on Circuits and Systems*, 2002.

[12] N. Vijaykrishnan, M. T. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using simplepower. In *ISCA*, pages 95–106, 2000.

[13] H.-S. Wang, L.-S. Peh, and S. Malik. Power-Driven Design of Router Microarchitectures in On-Chip Networks. In *Proceedings of MICRO-36*, December 2003.

[14] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects. Technical Report CS-2003-05, Univ. of Virginia, March 2003.