Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2023, 10(9):225-228**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Low Power Methods for GPU Design Optimizations

Apoorva Reddy Proddutoori

San Diego, California
Email: apoorvaproddutoori@gmail.com

Abstract The paper comprises of modern-day low power solutions to enhance the design of graphics processing unit (GPU). The change in the design to low voltage distribution network will allow the GPU to sweep bi-directionally, increasing the scope of the parallelization of the load. This indeed can be achieved by optimally determining and controlling the data load.

Further, irrelevant data load can lead to the lagged processor and lead to potential increase in the memory bandwidth. Higher bandwidth of memory directly leads to more power consumption rising the energy consumed. Therefore, reducing the data traffic results in better and faster design. This paper highlights the procedures to reduce the data traffic for newly innovated applications, focusing the improvement and efficiency of the architecture.

Keywords LVDN, Power, CMA, Graphics Processor Unit

1. Introduction

The System on Chip (SOC) designs primarily focus on cost and area reduction, pushing power optimizations to secondary. But the rise in the demand for lesser battery power consumption of the devices bought power optimization techniques into spotlight, making it one of the main goals of the design advancements.

A variegate of techniques have been developed as the necessity is rising to reduce the power consumption. Multi Voltage domains and power gating are generally combined with clock gating to provide an optimal solution for creating low power islands, used to minimize the usage of power.

Creating a multi voltage distribution network will improve the speed and advance the development of the graphics processor unit (GPU) designs. Enabling the network would lead to understanding the data traffic better and brining generators capable of optimizing active power. Programming the processors for rendering more efficient and complex operations with flexibility.

Further, the hardware has be potentially utilized to improve parallelization, thereby reducing, or offloading the data traffic developing multi core stream processor with unified kernel to lower the power consumed.

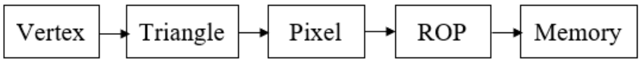

Figure 1: Graphics Pipeline (Processor per Function)

The stream processor core is integrated into the SoC to enable the mobile to become the host for interfacing processor applications and memory. The master and slave construction are used to perform the interfacing to help transfer the code to instruction memory from off chip memory. Not only this, but it can also transfer data from graphics processor to stream input and output registers.

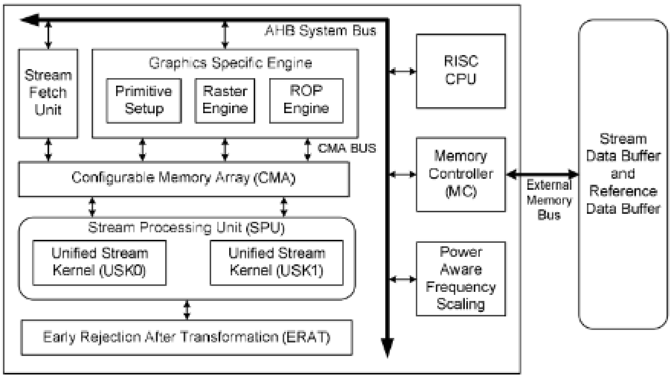
AHB System Bus
Stream Fetch Unit
Graphics Specific Engine
Primitive Setup
Raster Engine
ROP Engine
RISC CPU
CMA BUS
Configurable Memory Array (CMA)
Memory Controller (MC)
External Memory Bus
Stream Data Buffer and Reference Data Buffer
Stream Processing Unit (SPU)
Unified Stream Kernel (USK0)
Unified Stream Kernel (USK1)
Power Aware Frequency Scaling
Early Rejection After Transformation (ERAT)

Figure 2: Graphics Processor Unit Architecture

The Configurable Memory Array (CMA) comprises of stream input registers, constant registers, and stream output registers. The name itself indicated it can be modified flexibly based upon the memory needs of the processor applications, hence SRAM is added as an on-chip memory in accordance with the CMA. Another main purpose of the CMA was to mainline the stream of input and output data to reduce the memory bandwidth. Configuring the memory bandwidth has been of prime importance to innovate ways to efficiently strategize the on-chip data requirements. In this paper, we will mostly focus on highlighting the reduction of the unnecessary data traffic created due to ordering memory constraints. The introduction of possible ways of reduction is depicted in the following figure.

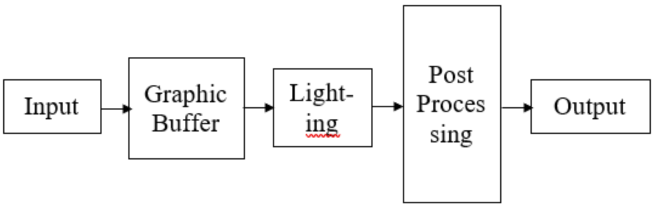

Figure 3: Graphics Processor Memory Load

The workload can be broken down into multiple blocks as shown in the pipeline diagram, bringing in the scope for parallelization too. The synchronization operation is enabled to ensure the working of all blocks or stages at once without any constraints while ensuring data dependencies.

2. Data Traffic in GPUs

In recent times, most of the computational problems are eradicated by increasing the potential of parallelization on CPUs and GPUs. The number of cores incorporated would technically increase, considering the cost would be of primary concern here though. In the Low Voltage Domain Network (LVDN) theory, parallel computation is mainly used to improve the computational times, reducing the forward and backward data load.

Global optimization would involve online procurement of the LVDN operation, involving backward forward sweep of the data. Differential Evolution (DE) technique is used to determine reference values for distributed generators.

As modern-day applications thrive to excel, the demand for memory bandwidth and raw compute power is forever rising. From figure 3, we understand that the output of each stage if fed as input to the later stage. This can create an unnecessary utilization of cache as each stage of this pipeline delivers data used by the subsequent stages leading to production of irrelevant load and creates displayable frames for the computational gaming load.

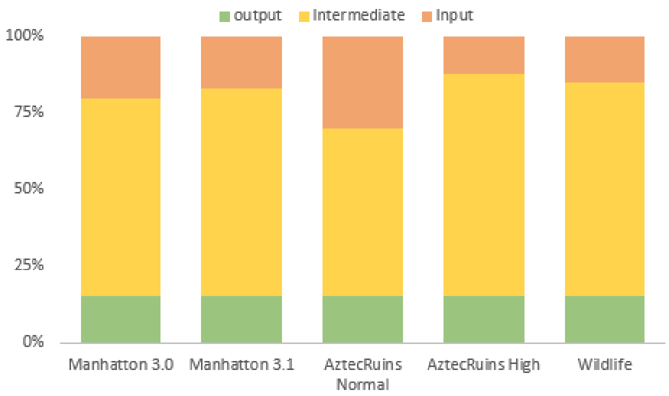

Figure 4: Graphics Benchmark Workloads

For example, Aztec Ruins Vulkan Normal algorithm uses the pipeline including graphic buffer, lighting, and post processing. Each of the frames produced by this algorithm is not immediately produced after the input stage but in the following subsequent stages all together. Each stage produces subsequent sub frames which are further used by the next stage. Finally, the output stage produces a frame using all these sub frames or passes. The resulting frame is then displayed on the screen, with a usual resolution of Full HD (1920 x 1080p). On an average the compression rate of the data is expected to be around 40% and the size of each pixel is 4 Bytes, leading to 4MB output data.

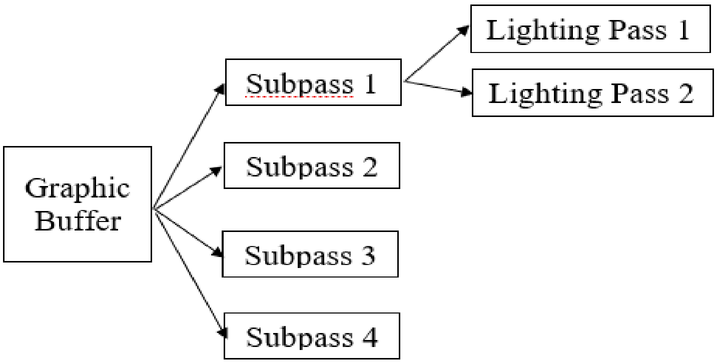

Figure 5: Memory Sub-Frame Topology

The compressed output produced of size 4MB is initially generated though 4 sub frames in the graphics buffer each of 16MB memory load as shown in figure 5. This sub frame is latter fed as input to two lighting passes each of 8MB along with additional data relevant to depth map and bandwidth compression ratio. Finally, using this bandwidth compression ratio, the lighting pass would result an output of 4MB, the original worth of data along with some 32MB of intermediate memory load. This intermediate memory is indeed the minimum necessary footprint built based on the original input, the intermediate data traffic generated could often be more than expected.

For example, from figure 4 it indicates that the intermediate data traffic generated is way too high than the actual input or output itself. If we consider Aztec Ruins Vulkan High usecase, the intermediate data traffic depicts to be comparably higher than the input. Hence based upon the benchmark workload, the data traffic generated varies as various benchmarks data is presented in figure 4.

The intermediate data becomes irrelevant once the output is produced, or in most of the cases when the subpass has consumed the required data to produce the output. The benchmark can significantly reduce the workload as soon as this intermediate data becomes irrelevant and save the memory traffic for the entire benchmark.

3. Hardware implementation

A tactical block needs to be implemented in the hardware design block, where it signals whenever the data traffic becomes irrelevant. Once this signal is passed onto the blockchain, the memory it cleared up for the actual data to be stored.

This block performs a functional operation, which is also capable of curtailing the active power. Moreover, it is predicted that the load in the voltage domain network can be controlled using a switch. The measurement system inputs the voltage network control system with the actual data, based on this data the LVDN structure analyzes the references for reactive power generation for individual DGs. This process is repeated constantly with respect to time to determine the updated reference values creating as minimum as possible intermediate data traffic.

This configuration is specifically targeted to minimize the traffic generated in DRAM due to the flow of GPU from last level cache (LLC). Identifying the relevancy of the data in the cache line while writing back to DRAM helps in evicting the unnecessary data traffic load. Hence in this architecture such cache lines are discarded from writing back to the main DRAM memory.

The address of such cache lines can be determined in the following ways:

1. Unified Stern Kernel - Adapts a dual core architecture to increase the performance and utilization, by dispatching the irrelevant data.
2. Multi-Threading – The workload is divided into multiple threads to implement adaptive scheduling of accessing the cache lines from DRAM and avoid any bottle neck issues.
3. Tracking Address – If the address belongs to known irrelevant data cache, it is dropped immediately.

4. Future development

From the theoretical analysis, the parallelization on GPU can outperform the multi core CPU. Considering the increase in the number of generations. Furthermore, smart grids can be implemented for better performance and local LVDN control for cost savings. With advanced adjustments to the objective functionality, the demand vs response can be incorporated within this method. This advancement can be expanded into smart cars and batteries with just basic network flow modifications.

5. Conclusion

New age applications lead to production of significant amounts of irrelevant intermediate data, not of any value once the output is generated. The stream processor architecture proposed in this paper can help eliminate large portion of the DRAM memory traffic. The mechanism used to write back the data by discarding the irrelevant cache lines can be further extended to reduce the entire on chip traffic.

References

[1]. Chia-Ming Chang, Shao-Yi Chien, You-Ming Tsao, Chih-Hao Sun, Ka-Hang Lok, Yu-Jung Cheng, “Energy-Saving Techniques for Low Power Graphics Processing Unit”, IEEE, November 2008.

[2]. Shen-Fu Hsiao, Shang-Yu Li, Kai-Hsiang Tsao, “Low Power and High Performance of OpenGL ES 2.0 Graphics Processing Unit for Mobile Applications”, IEEE, January 2015.

[3]. Anshujit Sharma, Sushant Kondguli, Michael Huang, “Irrelevant Data Traffic in Modern Low Power GPU Architectures”, IEEE Explore, June 2022.

[4]. Ernest Belic, Niko Lukac, Klemen Dezelak, Borut Zalik, Gorazd Stumberger, “GPU Based Online Optimization of Low Voltage Distribution Network Operation”, IEEE, April 2017.

[5]. Wenjie Wang, Qinali Chen, “Research on Low Power Schemes based on Large GPU Chip”, 3rd International Conference on Frontiers of Electronics, Information and Computation Technologies, IEEE, June 2023.