<http://charlab.github.io/blog/blog.html>

GPU Terminology

This week, I tried to create a GPGPU configuration file for a more modern GPU. However, not long after I started, I came to realize that there were a lot of flags that were ambiguously named or the information could not be found on the internet. For example, it seems likely that the only way to find the latencies of particular instructions is by deducing the values by running a program on the actual GPU. One thing that I was surprised that I never found was the structure of a the GPU caches. It's easy to find the size of the caches, but I can't seem to find the number of blocks or lines.

I came across a lot of unfamiliar terms so I made a list of abbreviations and terminology.

* Block -- A block is a group of warp. The SM will break up the block into warps with one thread for each core in the SM. These threads are then each executed in parallel.
* CUDA core -- In NVIDIA GPUs, this is a single processor for doing computation. A typical modern GPU has several thousand CUDA cores.
* FPU - Floating-point unit - Part of a processor that is specialized for floating point number operations such as adding, subtracting, bitshifting, etc. A typical GPU will have a single FPU.
* GDDR5 -- Graphics Double Data Rate, version 5 - GDDR5 is high performance SDRAM used for graphics cards.
* Grid -- A grid contains many blocks. The blocks in a grid can be processed simultaneously. The blocks can each be sent to a SM. If there are more blocks in a grid than SMs, the gird must be execuded sequentially.
* Kernel -- A kernel controls the flow of a program. Often times a single kernel is large enough to use the entire GPU by itself, but some GPUs support multiple kernels running in parallel.
* Prefetch buffer -- A data buffer in DRAM that allows access to multiple in a single row of memory.
* Read-only cache -- As it's name suggests, a read-only cache stores data that has been declared constant, and therefore will never change during the lifetime of the kernel. Typically in a GPU, each SM has one read only cache that is about 48 KB.
* Register spills -- A register spill occurs when a variable can not be placed in a register. In this case, the variable must be placed in memory, typically in the L1 cache in a GPU. This variable takes much longer to use due to the slower access time.
* ROP -- Render Output Unit or Raster Operations Pipeline -- ROPs work at the end of a pixel pipeline and do anti-aliasing and color compression and write the output to the frame buffer. A typical modern GPU has around 48 ROPs.
* SFU -- Special Function Unit -- Executes functions like sine, cosine, reciprocal and square root.
* Shader -- A kernel function to modify an image. There are three types of shaders: pixel shaders shade a single pixel, vertex shaders calculate 3d locations and map them to a 2d screen, and geometry shaders can create tesselations and other high level effects.
* SIMD -- Single Instruction, Multiple Data -
* SM (or SMX) -- streaming multiprocessor -- holds a group of processors and caches. A typical modern GPU has up to 16 SMs.
* SP -- streaming processor -- A processor for a single thread of a program. For NVIDIA GPUs, this is synonymous with a CUDA core.
* Thread -- A thread is a basic element of data to be processed.
* TMU -- Texture Mapping Units - A separate processor used for rotating and resizing 3D objects. A typical modern GPU has around 128 TMUs.
* Warp -- A warp is a group of many threads. Typically this is up to 32 threads, one thread for each core in a streaming processor. A warp cannot have more threads than processor cores, so it is always executed entirely in parallel.