CS 320
Ch 19 General Purpose Graphical Processing Units

The architectural components of a typical GPU include a **few hundred to a thousands of parallel processor cores on a single IC. Some small embedded micros may have fewer than 10 processors cores acting in parallel such as on a smart phone.**

GPUs are used for: **3D graphics rendering and video processing. Workstations, tablets, smartphones, and laptops all have GPUs of one sort or another.**

GPUs are not specialized to a few applications using graphics, **In the past decade people have begun using GPUs anywhere there are massively parallel programming environments. The author lists medical imaging, DSP, computational finance, oil and gas exploration (analyzing sonar signals), etc.**

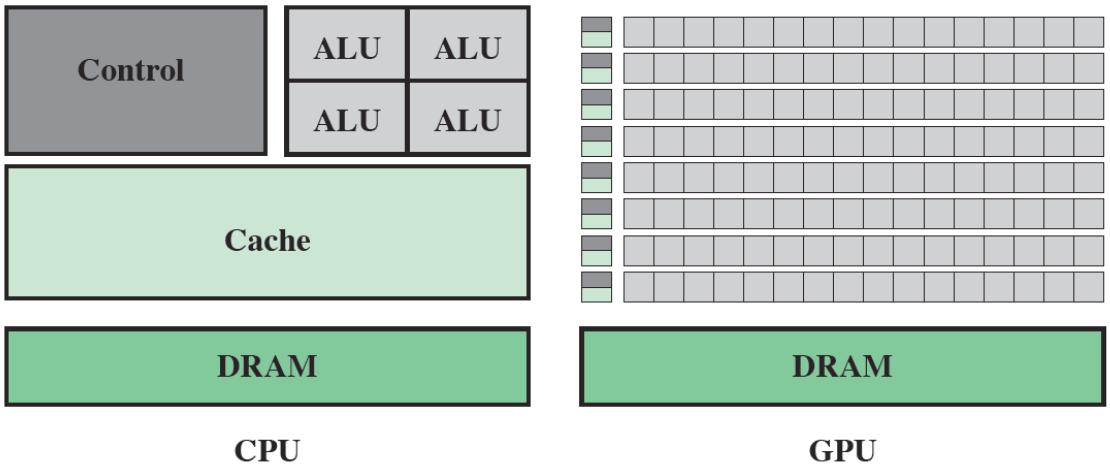
This is likely out of date at this point but the author says you can by NVIDIA's GeForce GTZ 660 with 960 parallel cores for about $200.

General purpose GPUs (GPGPU) have two main languages for support: **NVIDIA's CUDA and Khronos Open CL. Both of these are relatively programmer friendly and allow the user to write code for a GPGPU.**

CUDA (**Compute Unified Device Architecture) was created by NVIDIA. CUDA is a programming language that runs on NVIDIA's GPUs. CUDA is a C/C++ based language. It has three sections: Code to run on the host CPU, code to be run on the general purpose sections, and code related to the transfer of data between the host and the device.**

To execute the code generated by CUDA, e**ach unit in the GPU is assigned a block consisting of multiple threads to be executed. It's up to the user to see to it that the number of blocks does not exceed the number of units in the GPU.**

The figure illustrates the difference between a CPU and a GPU.

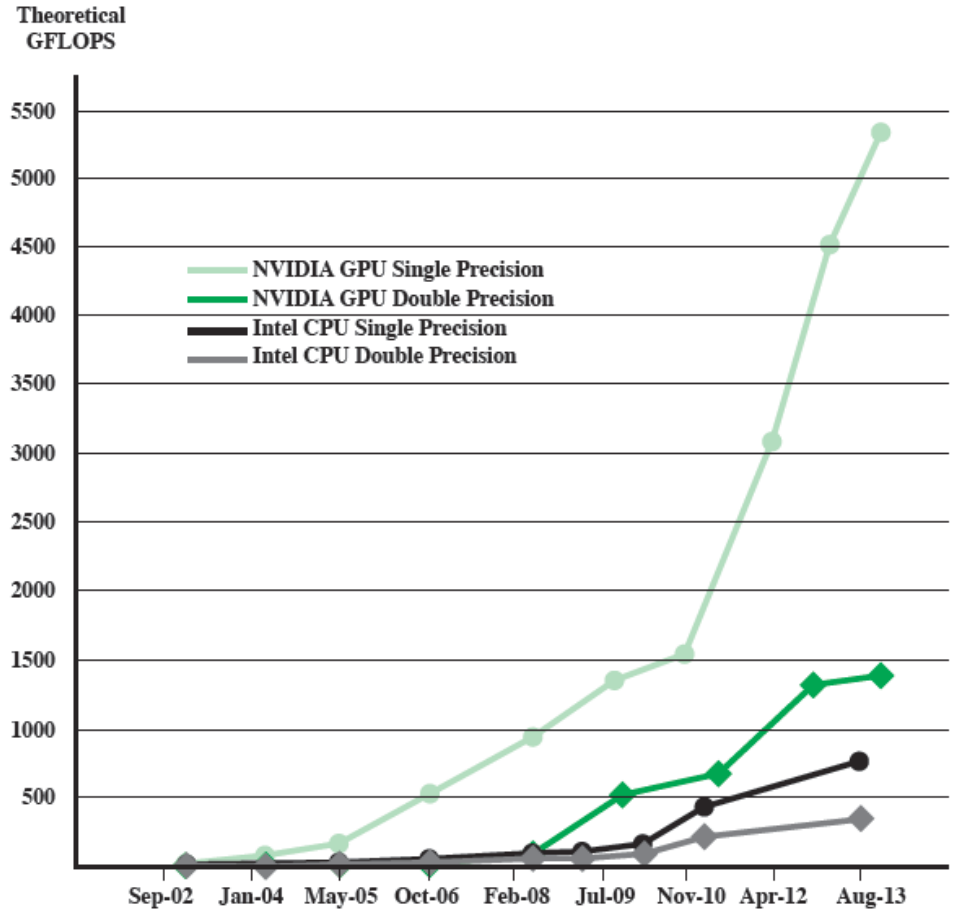


CPU left and GPU right. Shows the relative area for transistor dedication.

Note that the GPU has relatively small space for control since it is the same for all units. Likewise there is not much need for a large cache and the ALUs are not pipelined and neither are they superscalar.

The main market for GPUs **is the gaming market.**

GPUs tend to optimize the number of FLOPs possible as well as the number of FLOPS/watt. They do this by decreasing the system clock rate and increasing the number of transistors used.



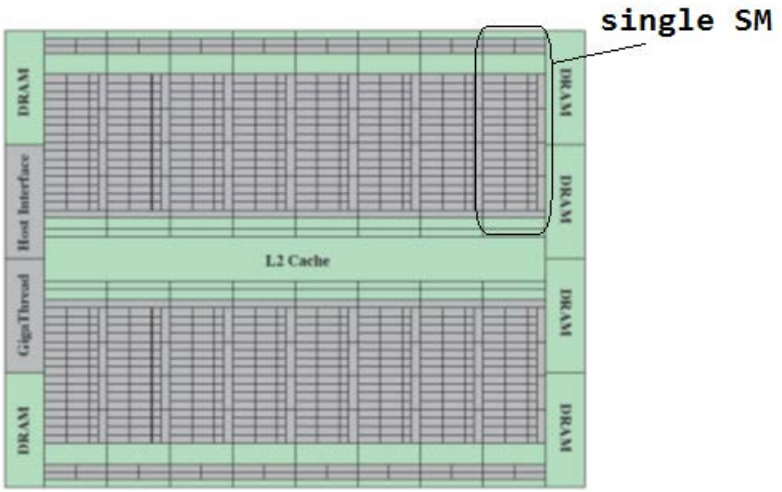
GPUs have evolved from the 1980s when the early machines started adding special hardware to handle the screen graphics. There are three major phases of GPU architectural development?

1) 80s-90s: fixed nonprogrammable specialized processor stages such as raster, shader, etc;
2) 90s to mid-00s: iterative modification of phase 1 GPU from hardware pipeline to programmable GPUs;
3) 05 to present: GPGPU as a highly parallel SIMD machine used for things other than graphics.

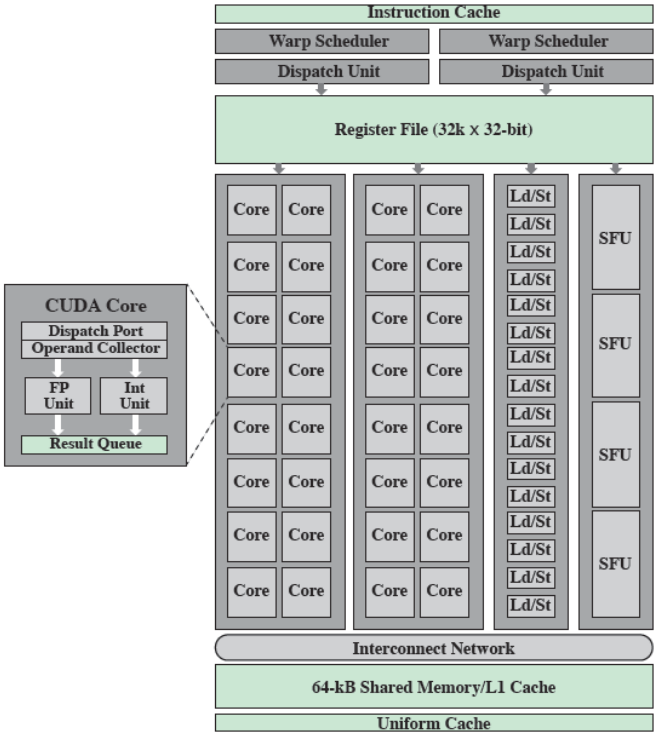
At this time the major companies making GPGPUs? **NVIDIA, Intel, AMD, and ATI (Array Technology Inc now owned by AMD).**

Since GPGPUs have move to phones and tablets almost all major software companies now provide some support. This includes MATLAB, Microsoft, Google, and Apple. This chapter looks mostly at NVIDIA.

The Characteristics of the NVIDIA Fermi Architecture are shown below:



NVIDIA Fermi Architecture



NVIDIA Single SM (streaming multiprocessor) architecture.

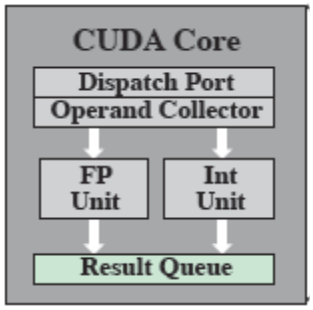
• Single L2 Cache shared by 16 streaming multiprocessors.

• Each SM has 32 Cores plus 16 load and store units plus 4 special function units.

• Each SM has a 64 KB shared L1 cache for the 32 cores.
• Each GPGPU has six 64-bit DRAM interfaces.
• Supports up to 6 GBytes of external DRAM.
• There is a 32K x 32 bit register file on each SM

There is a GigaThread Scheduler on the GPGPU chip (not shown) which distributes thread blocks to the SMs. The dual warp scheduler on each SM breaks the thread blocks into *warps* which are a bundle of 32 threads that have consecutive thread IDs and start at the same address. Each thread goes to a core and has its own program counter and register set.

The Cores are called CUDA Cores or just CUDA (Compute Unified Device Architecture). Each core has two pipelines: integer and floating point. Only one of these units can be used in a single clock period. The integer unit can do 32/ or 64 bit integer operations including arithmetic and bitwise ops. The floating point unit can do only a single precision floating point operation. A double precision operation requires two cores.



Cuda Core

The Special Function Units **do trigonometric operations.**

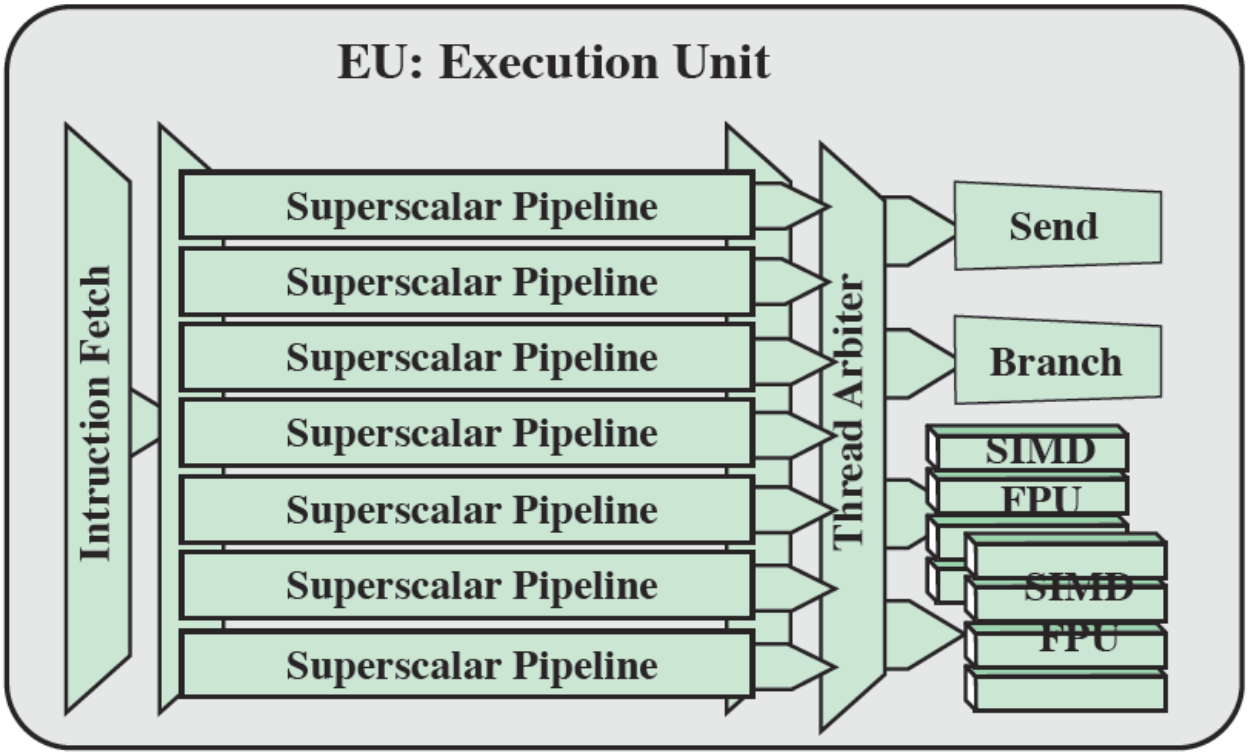
Each load/store unit calculates the source and destination addresses for a single thread per clock cycle. The addresses can be in cache or DRAM.

Programming a NVIDIA GPGPU has some peculiarities. **The programmer must be careful to set the thread block size to be greater than the total number cores in an SM and less than the maximum number of allowable threads per block. The programmer must also be careful about the size of data types, their access times, and their accessibility limitations. Also threads cannot share date even though they are in the same SM except by shared memory (not shared registers). So the programmer must assign certain threads to read and write to memory.**

GPGPU provide significant advantage to the following program types:
• Programs which have highly parallelizable portions of code which can be replicated into thousands of lightweight threads.
• Serial code which has a large for loop which does calculations on equations which have little or no dependencies are ideal for GPGPU operations.

• At this point there is no parallelizing compiler to convert serial code to a form that is optimized for a GPGPU core.

Intel's GEN8 GPU
Intel's GEN8 GPU was introduced about 2014 which consists of multiple execution units (EUs) as shown in the figure below.



Intel's GEN8 Execution Unit.

Each EU has seven superscalar pipelines and can handle seven threads for Simultaneous Multi-Threading. Each pipeline has 128 GP registers.

But each register is *32-Bytes* long to give each Superscalar Pipeline 2^{12} bytes – 4Kbytes for a general purpose register file (GFR).

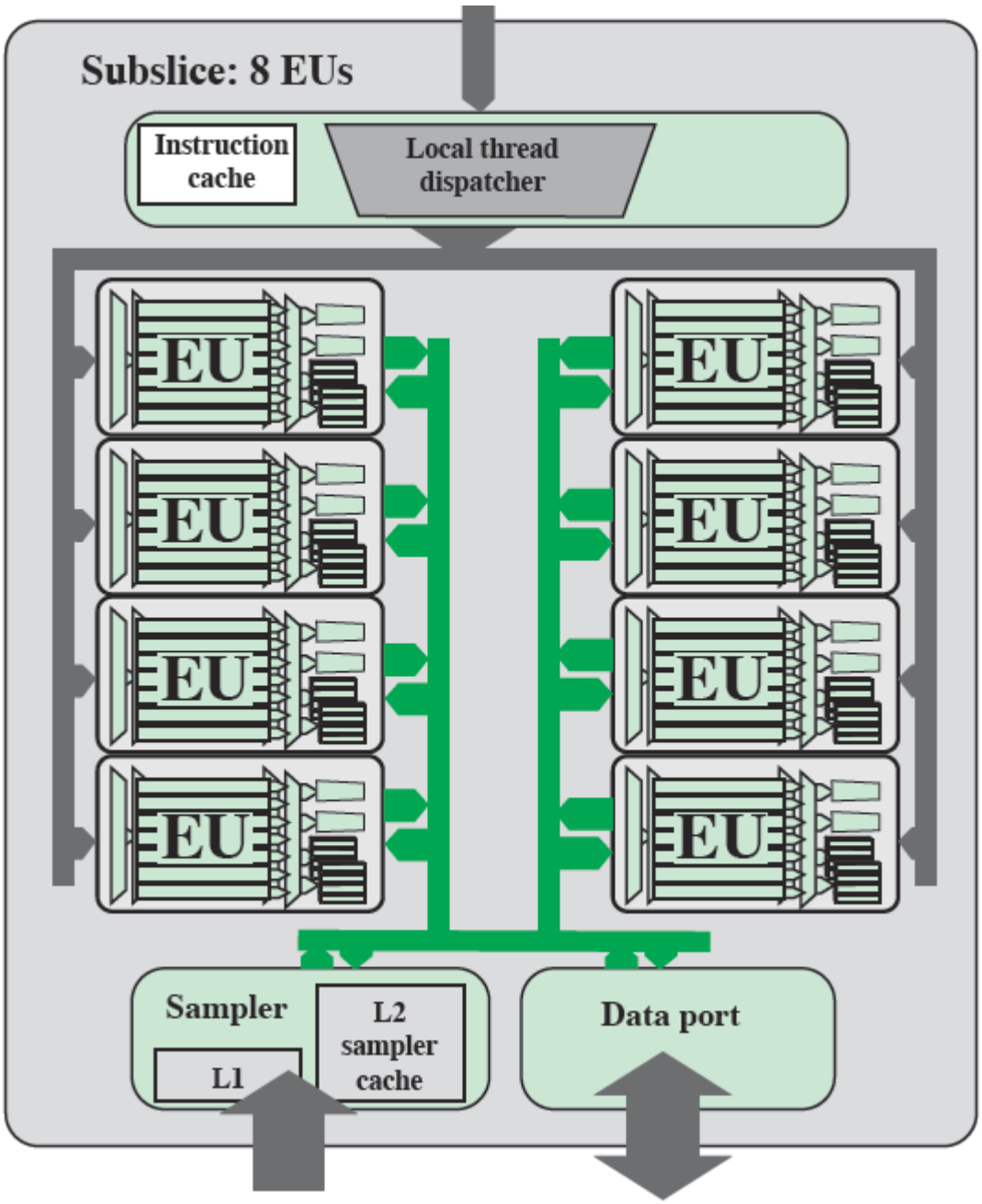
The Superscalar pipelines can do some basic computation (shift, add, logic, etc) but the primary computation unit at the two SIMD FPUs.

The FPUs can complete a floating point add and a floating point multiply each cycle. The seven Superscalar pipelines share the FPUs via the Thread Arbiter.

A Branch unit handles branch instructions and a Send unit handles memory operations.

Each Execution Unit can issue up to four different instructions simultaneously (Branch, Send, FPU1, FPU2) with the Thread Arbiter deciding where the instructions go.

The Execution Units are put together in a *Subslice* as shown in the figure below.



GEN8 Subslice.

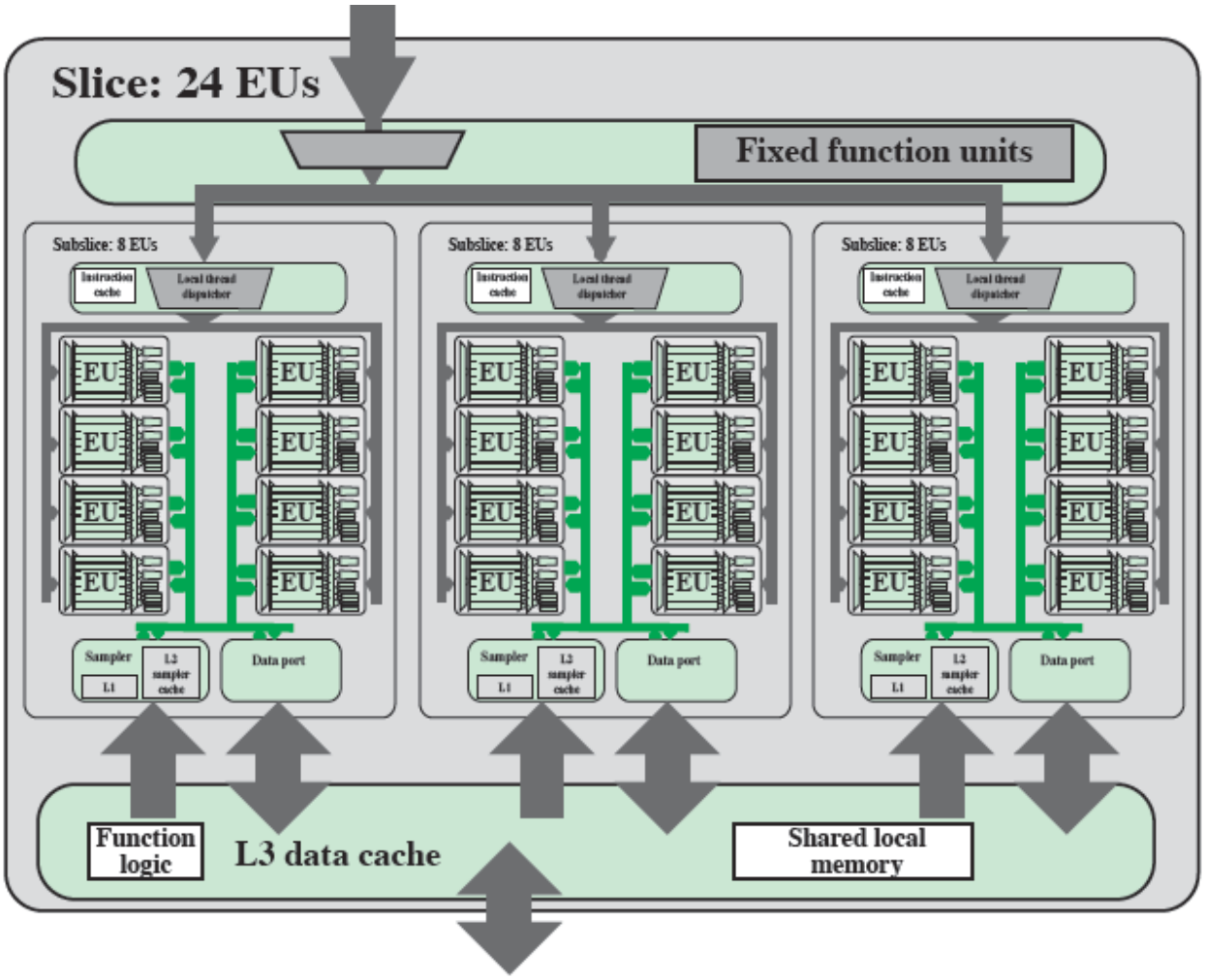
Each Subslice contains eight EUs.

Each Subslice also has its own thread dispatcher and its own L1 Instruction cache.

A Subslice can handle 56 simultaneous threads

The sampler unit is used for image processing. It has its own fixed-function logic to support compression/decompression and other image texture operations.

Subslices can be clustered into groups called *Slices*. A slice can have up to three subslices or 24 execution units. A Slice is shown in the figure below.



GEN8 Slice

The subslices are interconnected by way of a multi-bank L3 Data cache. The L3 cache is organized in banks which can be accessed in parallel. If two subslices attempt to get data from the same location in the L3 cache the logic serializes the access.

Hewlett-Packard makes a micro-server that uses the GEN8 graphics card among others.