# A look at the GPU architecture

Pre-G80: Separate vertex and fragment processors.

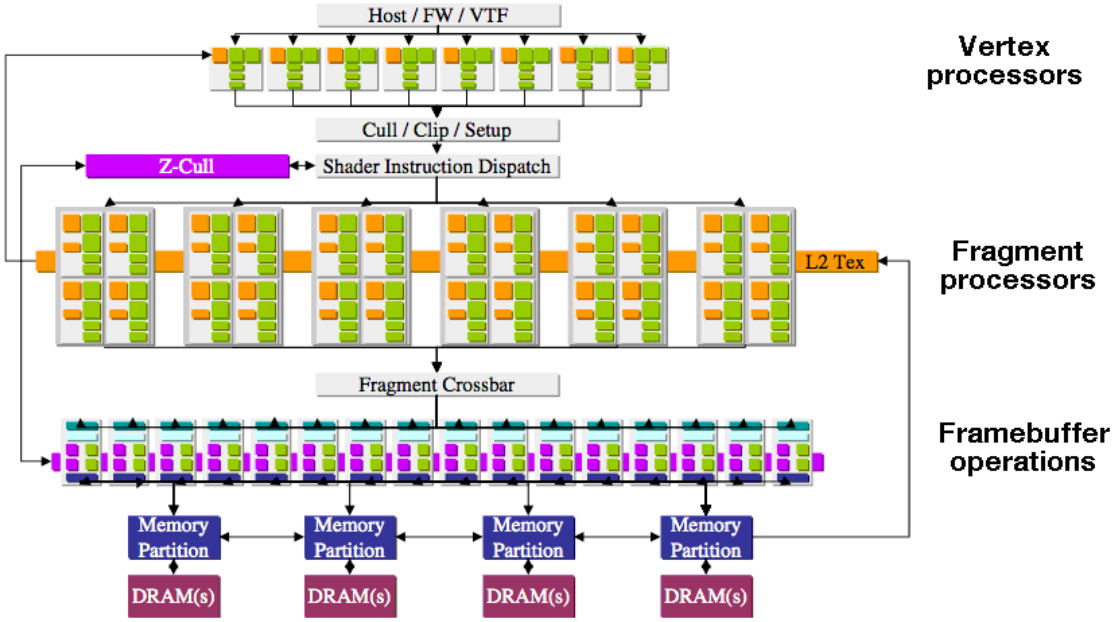
Hard-wired for graphics. Load balance problems.

G80: Unified architecture. More suited for GPGPU. Higher performance due to better load balancing.

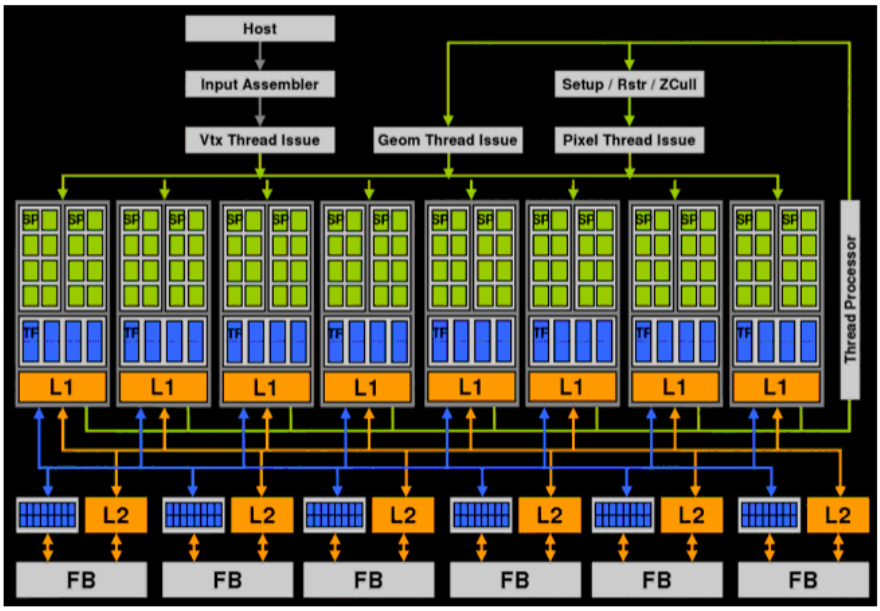
G92: Similar to G80, more cores, more cores per group.

Information Coding / Computer Graphics, ISY, LiTH

# 7800: High-end GPU before G80



# G80



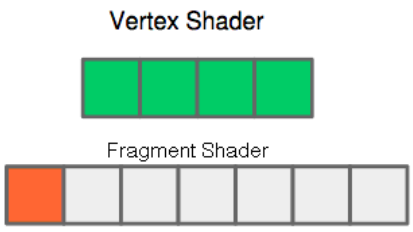
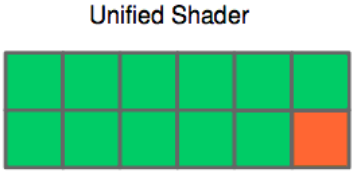
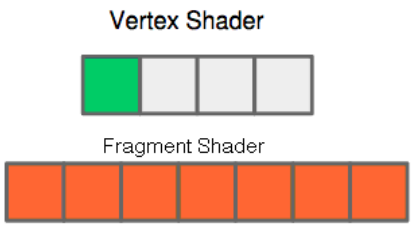

Hardware formerly between vertex and fragment processors

Unified processors

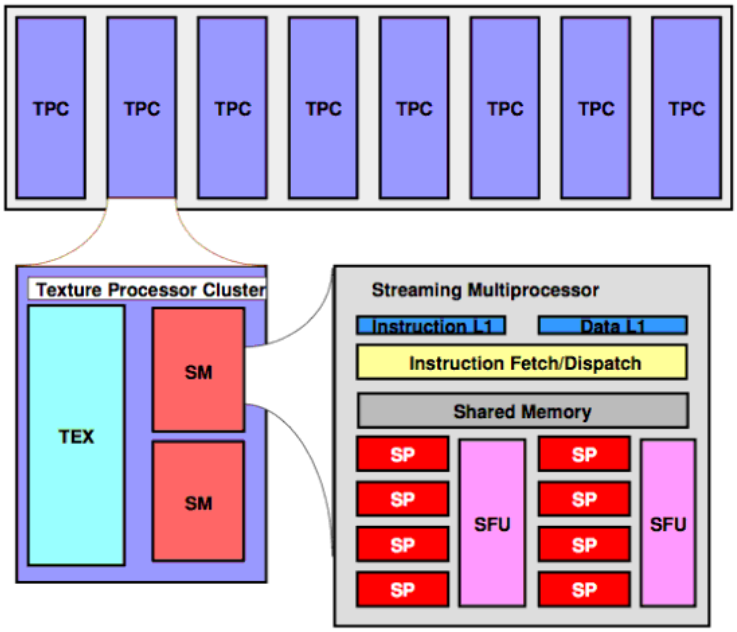
Framebuffer operations

Information Coding / Computer Graphics, ISY, LiTH

# G80: A question of *load balance*!

| | Separate vertex and fragment processors | Unified processors |
|---|---|---|
| Vertex problem (e.g. complex geometry) |   |   |
| Fragment problem (e.g. advanced rendering) |   |   |

# G80 processor hierarchy

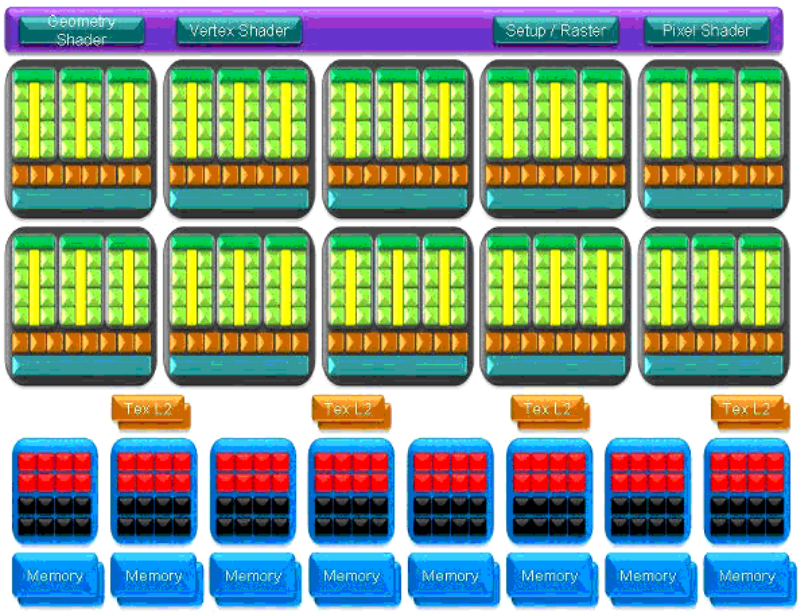



8 top-level groups of TPCs

SM is a group of 8 SIMD cores

Information Coding / Computer Graphics, ISY, LiTH

# GT200

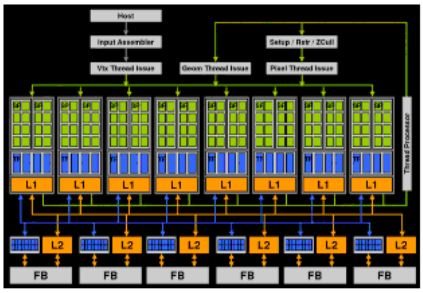


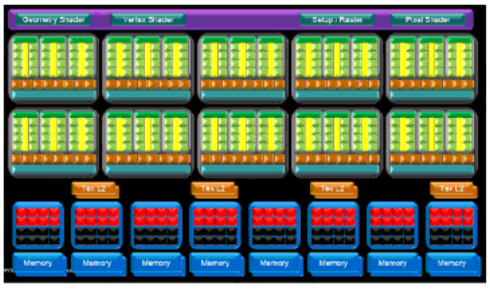

Similar but with a bit more of everything

# G80 vs GT200 in numbers:

8 cores per SM
2 SMs per cluster
8 clusters

10 cores per SM
3 SMs per cluster
10 clusters

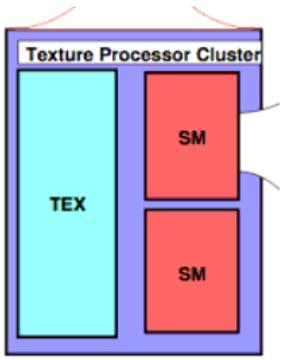



**8 was *not* a magic number - more cores per SM**

Information Coding / Computer Graphics, ISY, LiTH

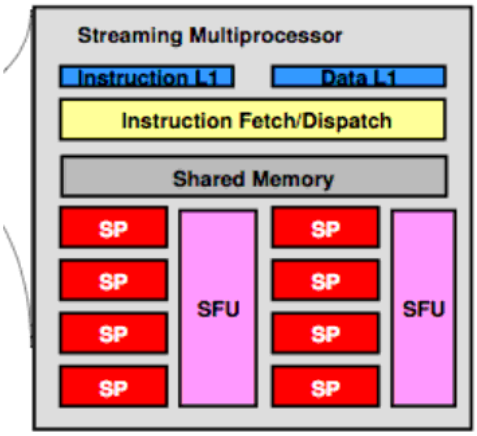
# Vital components




**Texture processor cluster: 2 or 3 SMs and a *texturing unit***

**A texturing unit will provide texturing access with automatic interpolation - vital component for graphics**

# Vital components




**SM: 8 cores**

**but also**

**SFU: Special functions unit**

**Shared memory**

**Register memory in each core**

**Instruction handling/thread management**

Information Coding / Computer Graphics, ISY, LiTH

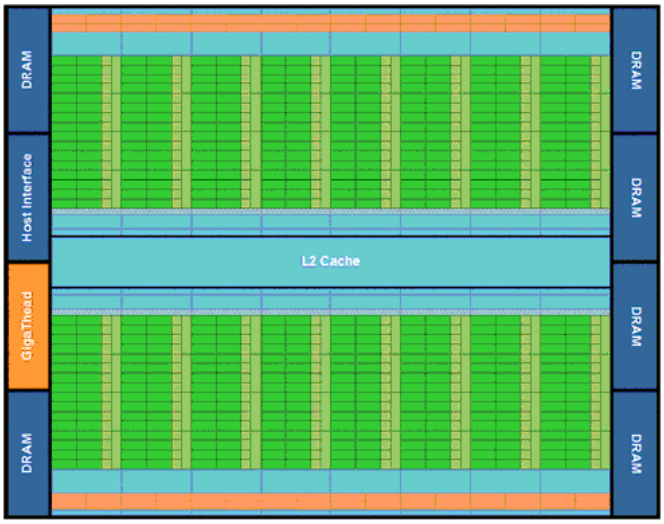
# How much architecture details do we need to know?

Shaders: The architecture is mostly invisible

Cuda/OpenCL: Less so, but number of cores more or less ignored - as long as we provide more parallelism in our algorithm than the architecture has!

Memory usage is specified by the programming languages. More about that later.

# 2010: Fermi (GT100)




Looks like:

16 SMs

32 cores per SM

Support for 24576 threads!

Much area for L2 cache!

Information Coding / Computer Graphics, ISY, LiTH

# 2010: Fermi (GT100)


Four clusters

Four SMs in each

32 cores per SM!

# 2010: Fermi (GT100)

NVidia's new architecture! Major changes in favor of general computing.

512 cores!
Caching closer to the processors!
Concurrent kernels.
64-bit wide
ECC


Information Coding / Computer Graphics, ISY, LiTH

# More on Fermi

4x performance for double (64-bit FP)

More silicon space for cache! More like a CPU.

16 SMs, 512 cores (32 cores per SM)

CGPU = Computing Graphics Processing Unit

=> NVidia aims for GPGPU with Fermi!

# Related parallelization efforts

IBM Cell (next generation canceled!)

Intel Larabee (”put on ice” - dead)

GPUs are the clear winners so far!