UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

GPUs
Computación Heterogénea

Profesor: Dr. Joel Fuentes - jfuentes@ubiobio.cl

Ayudantes:
- Daniel López - daniel.lopez1701@alumnos.ubiobio.cl
- Sebastián González - sebastian.gonzalez1801@alumnos.ubiobio.cl

Página web del curso: http://www.face.ubiobio.cl/~jfuentes/classes/ch

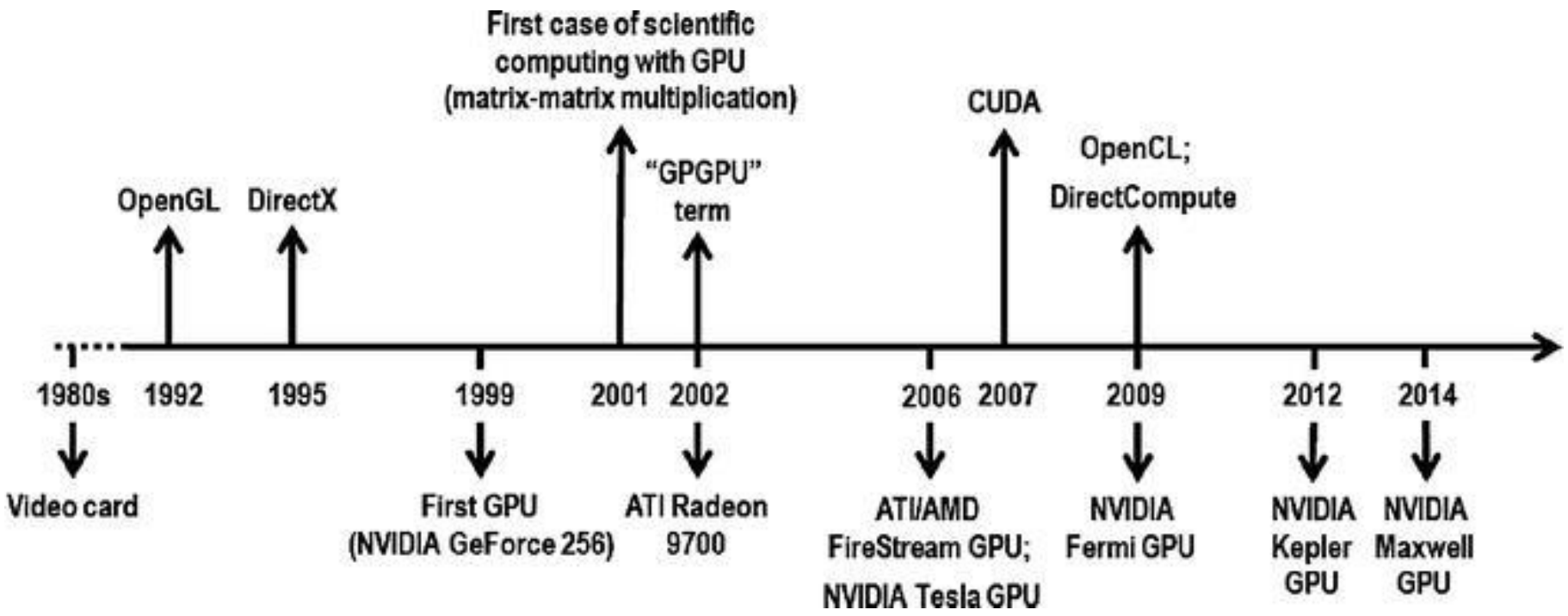
Contenidos

- Historia de las GPUs
- Aplicación de ejemplo: Renderización 3D
- Arquitecturas Nvidia, AMD, Intel
- Programación de GPUs

Historia de las GPUs

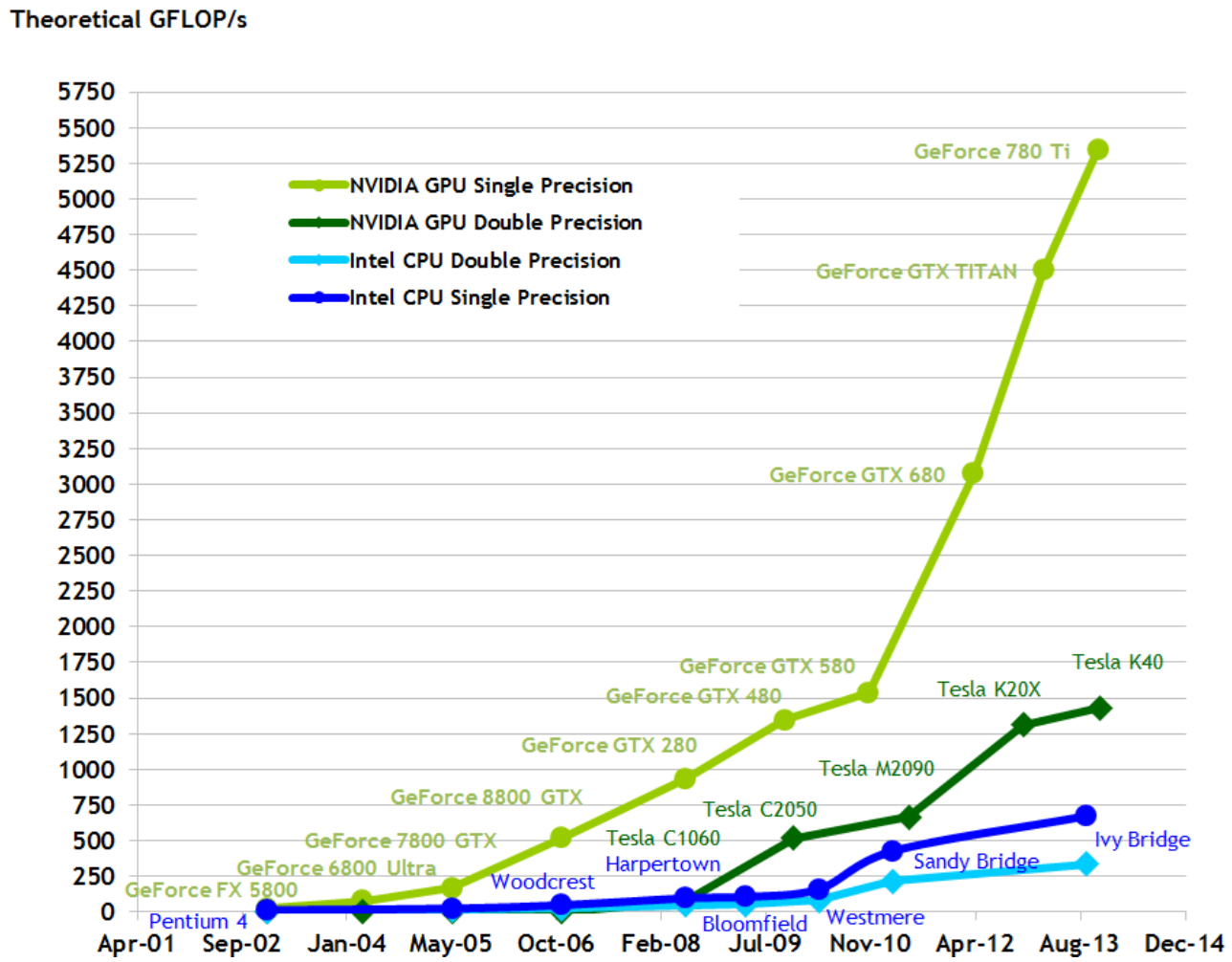
- GPUs (Graphics Processing Units) fueron inicialmente creadas para aceleración gráfica y video juevos 3D.
- Inicialmente diseñadas con funciones fijas sin posibilidad de programación flexible.
- Desde el 2001 se comenzó a incluir la posibilidad de programar GPUs
- Hoy en día sus usos han evolucionado y abarcan:
  - Visión por computador
  - Deep learning
  - Computación científica
  - Cryptomining

Historia de las GPUs



https://link.springer.com/referenceworkentry/10.1007%2F978-3-319-17885-1_1606

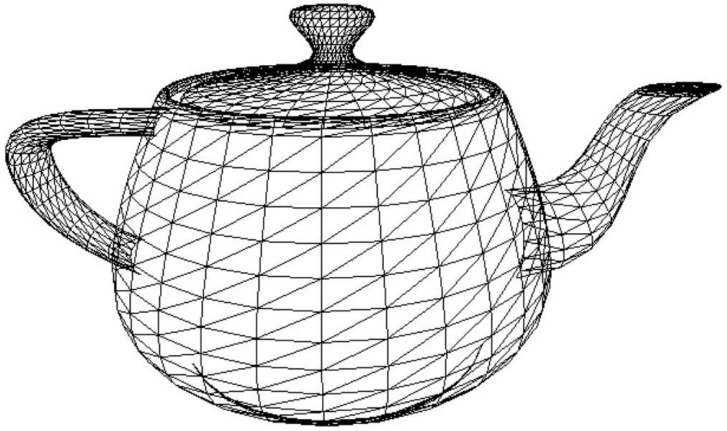
Performance en GFLOP/s vs CPU



http://docs.nvidia.com/cuda/cuda-c-programming-guide/#axzz3X9Fwos00

Aplicación de ejemplo: Renderización 3D

- Tarea consiste en calcular cómo cada triángulo en la malla 3D contribuye en la apariencia de cada pixel de la imagen renderizada.



Descripción de la escena basada en: malla de triángulos, luces, cámara, etc.

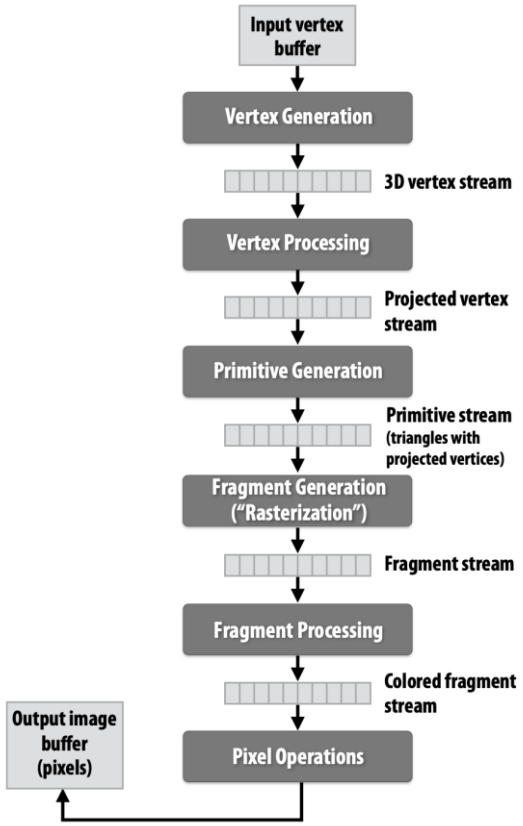
Imagen de la escena

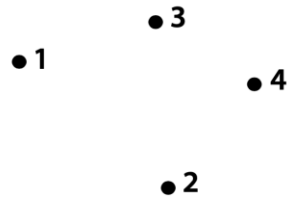
Renderización 3D en tiempo real

Flight Simulator (2021)

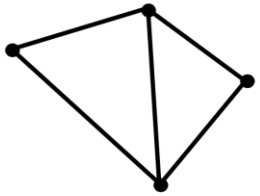
Aplicación de ejemplo: Renderización 3D

- Pasos en la renderización dada una malla de triángulos 3D

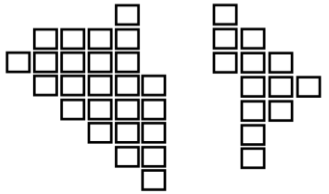




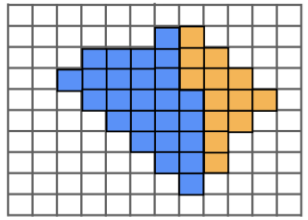
Vertices
(points in space)



Primitives
(e.g., triangles, points, lines)



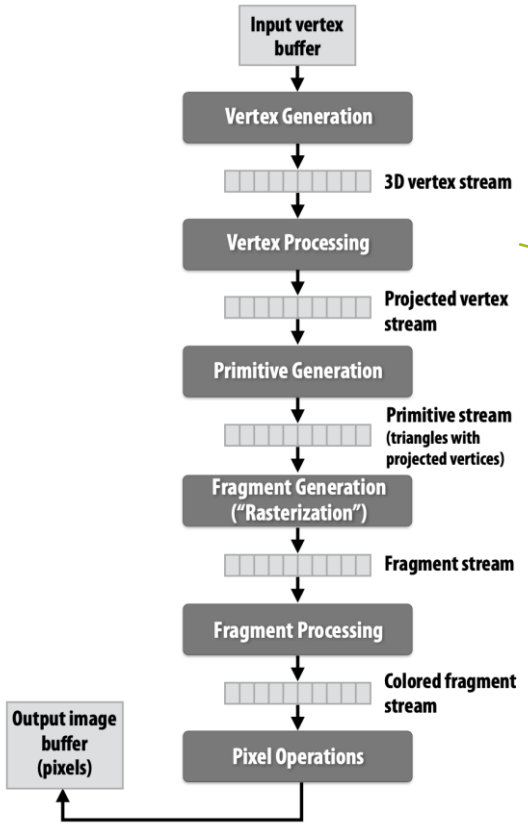
Fragments



Pixels (in an image)

Aplicación de ejemplo: Renderización 3D

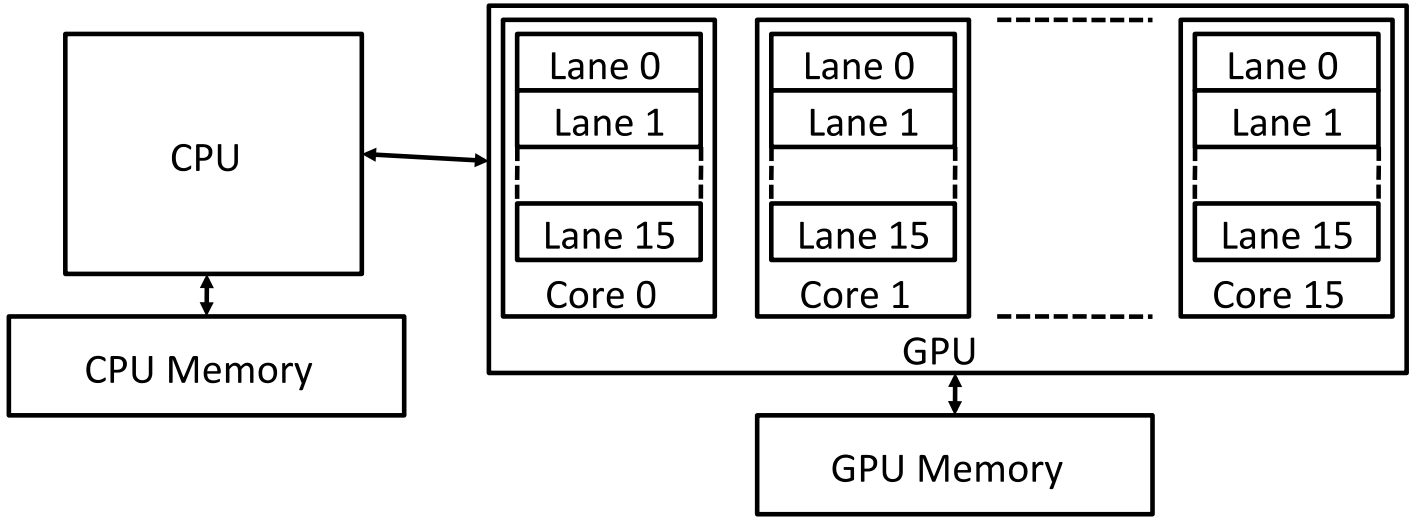
- Pasos en la renderización dada una malla de triángulos 3D



Programadores escriben mini-programas llamados "shaders" que describen la lógica de estos pasos

Arquitecturas GPU

- Modelo de ejecución



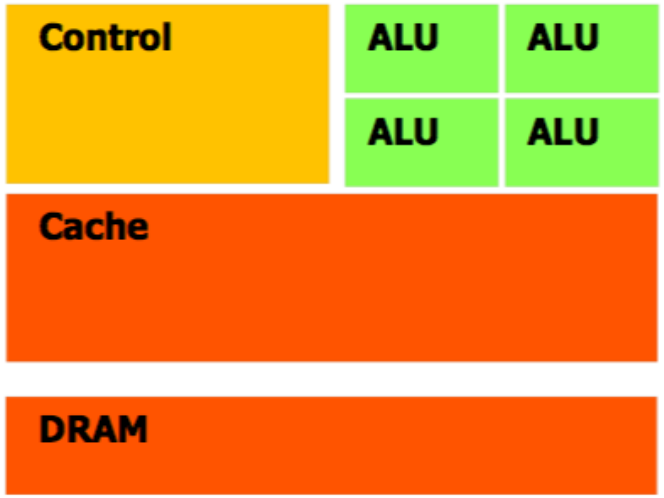
- GPU está construida basado en múltiples cores paralelos simples. Cada core es capáz de ejecutar instrucciones SIMD (Single Instruction Multiple Data).
- La CPU envía tareas de procesamiento (mallas, buffers, etc.) a la GPU, quien distribuye el trabajo en sus cores.

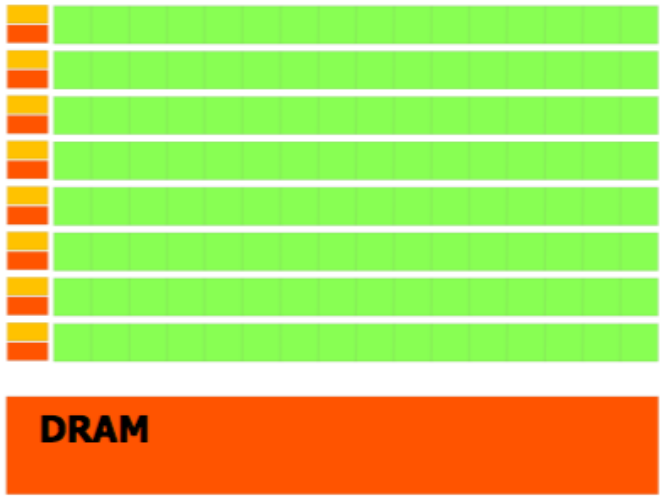
Arquitecturas GPU

- Los cores en la GPU trabajan bajo el modelo de hilos en memoria compartida.
- GPUs pueden correr cientos o miles de hilos en paralelo.
- GPUs usualmente tiene su propia DRAM
- GPUs son buenas para:
  - Procesamiento data-parallel: la misma operación ejecutada en muchos elementos de datos en paralelo.
  - Procesamiento con alta intensidad aritmética.

Arquitecturas GPU

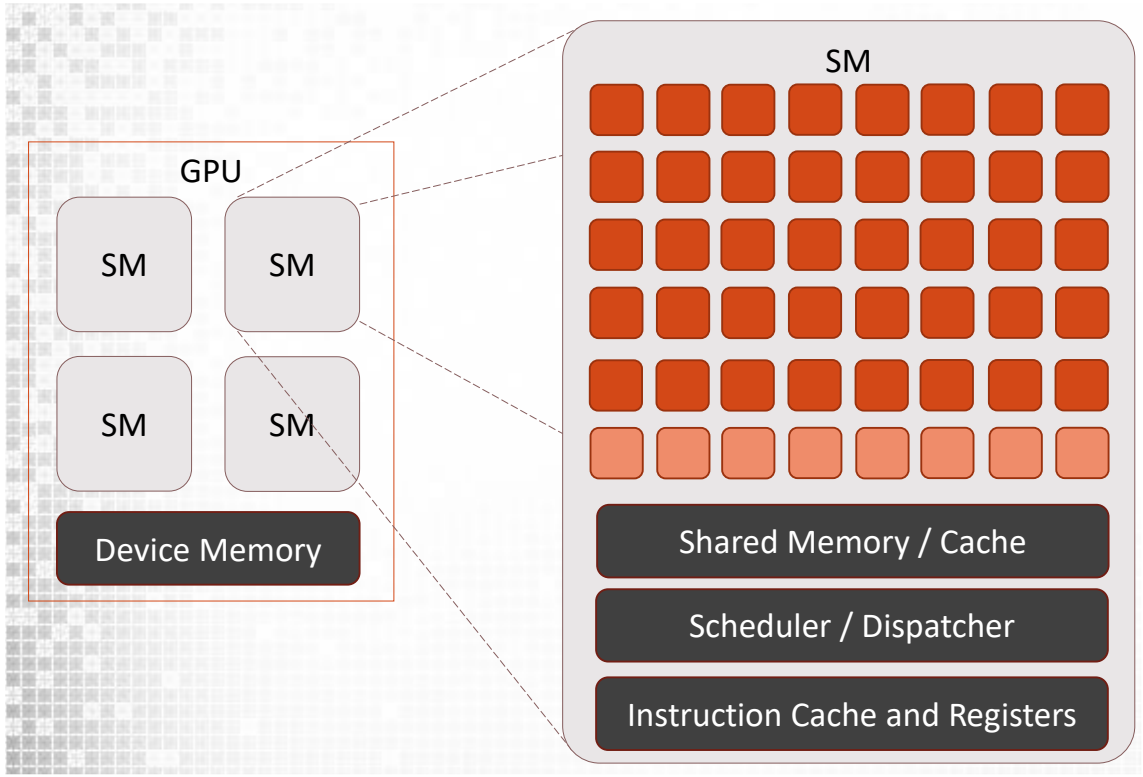
- Comparación con CPU:
  - GPUs ocupan más transistores en procesamiento de datos
  - GPUs poseen cachés más pequeñas
  - La ALU de GPU es más simple que la de CPU





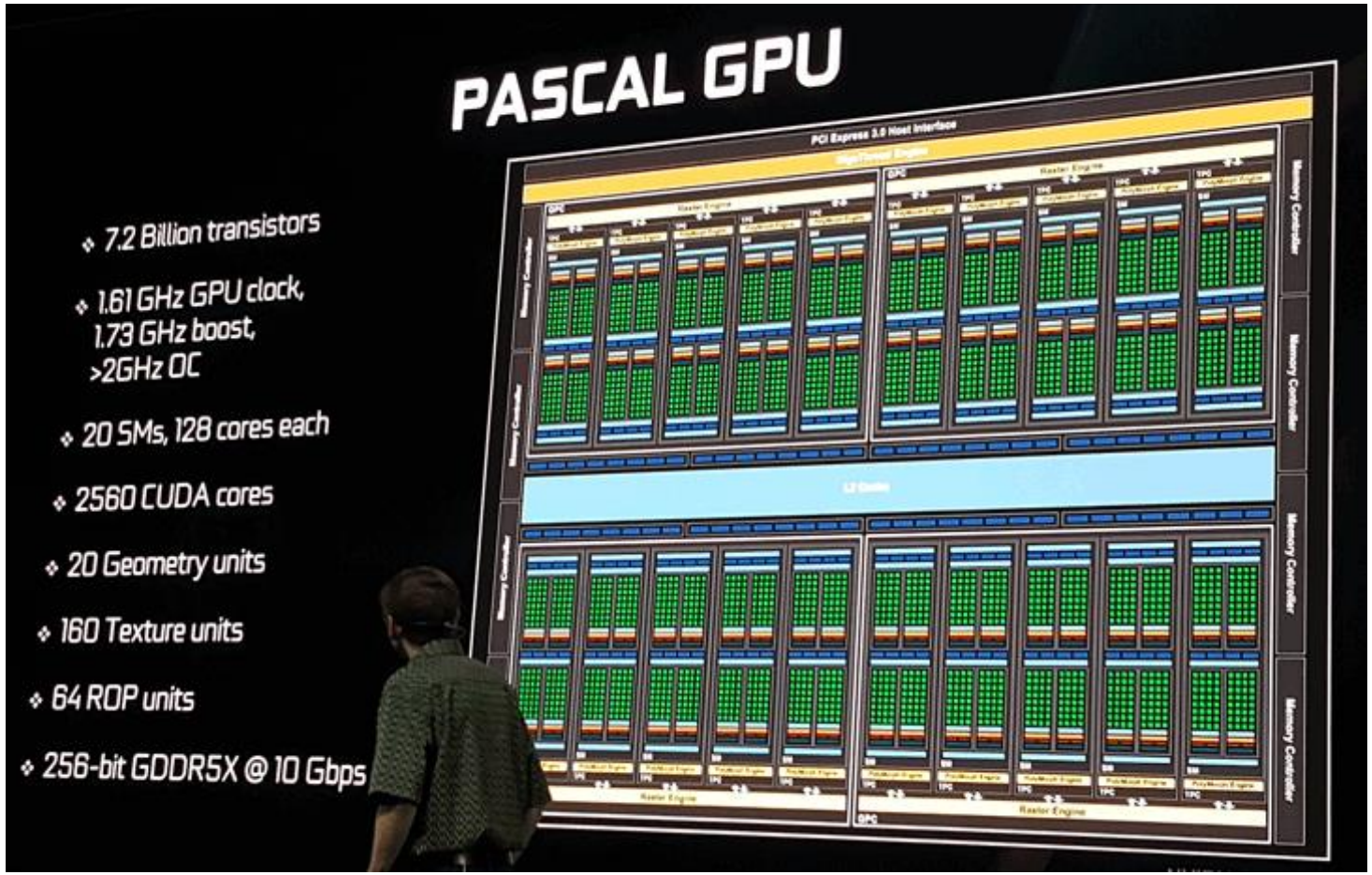
GPU Nvidia

- Tienen una jerarquía de 2 niveles
- Cada procesados de flujo (SM) tiene múltiples CUDA cores
- El número de SMs varía dependiendo del tipo de GPU



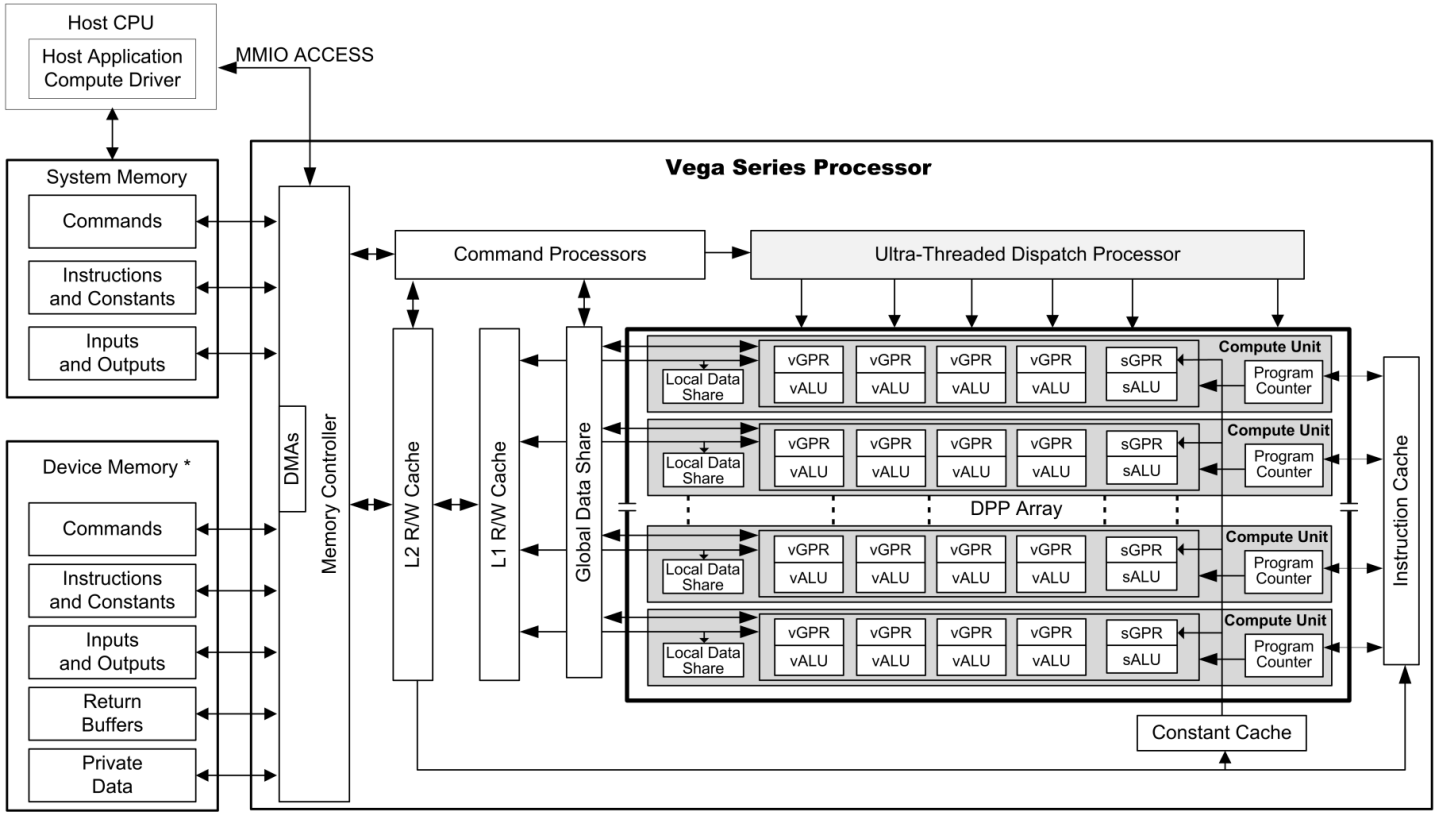
GPU Nvidia

- Imagen arquitectura Pascal (1080 Ti)



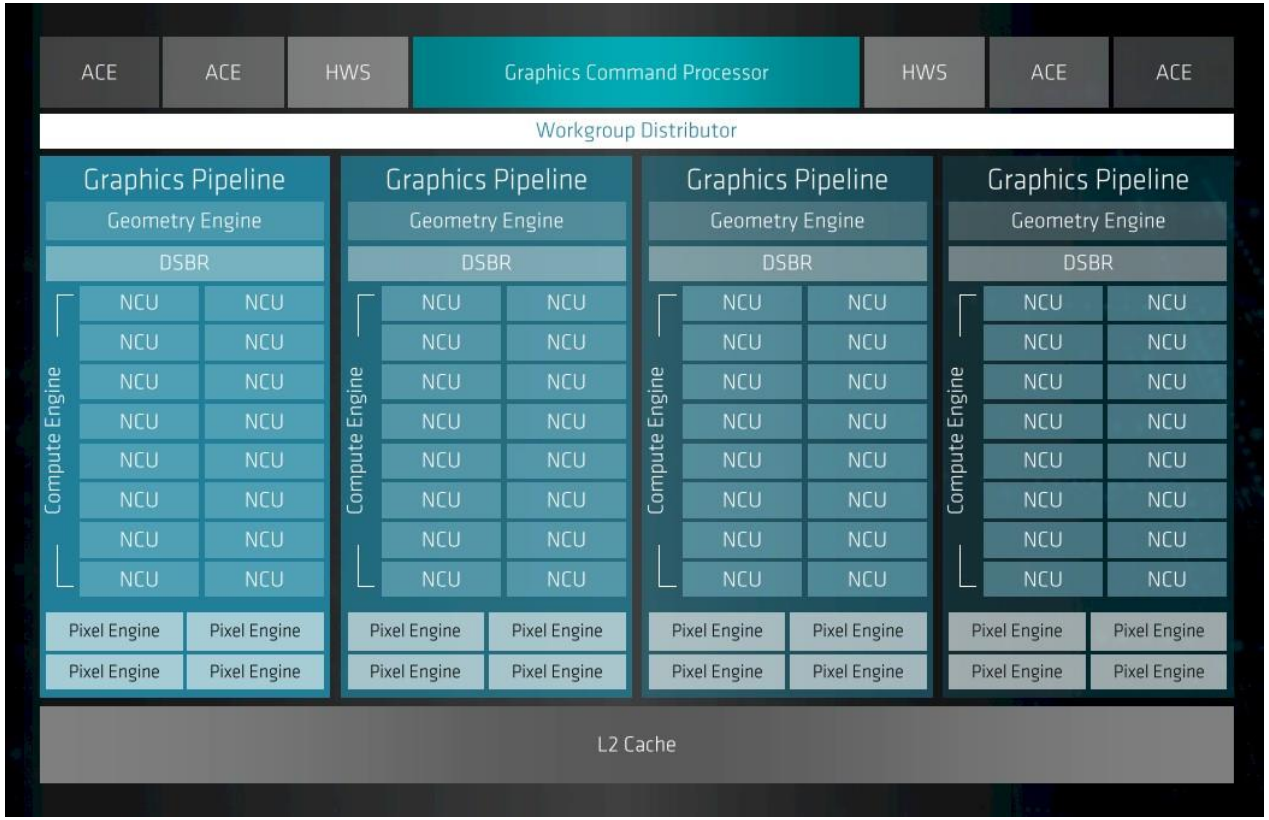
GPU AMD

- GPR: registros de propósito general
- vALU: vector ALU para procesamiento SIMD



GPU AMD

- Imagen GPU Vega 20
- NCU: Next compute unit



GPU Intel

- Unidad de cómputo o core es llamado Execution Unit (EU)
- Múltiples EUs agrupadas en Subslices
- Múltiples Subslices agrupadas en Slices
- Jerarquía de memoria hasta L3

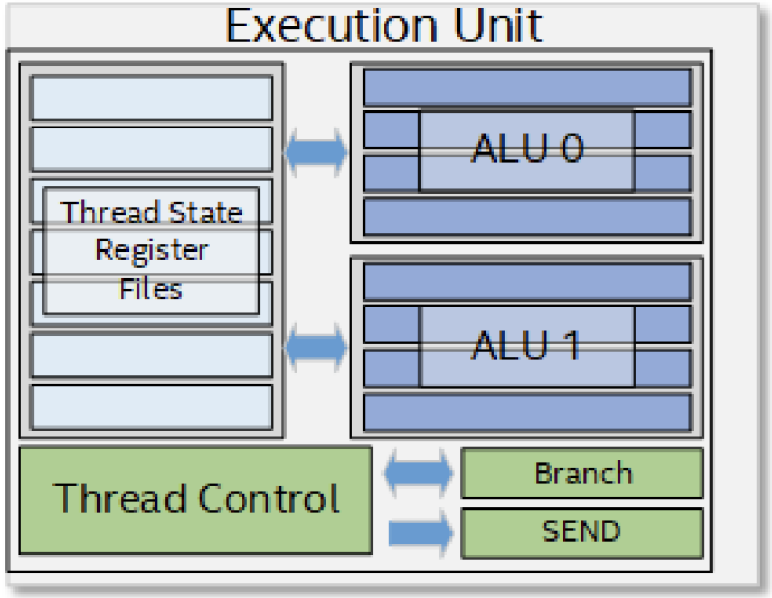
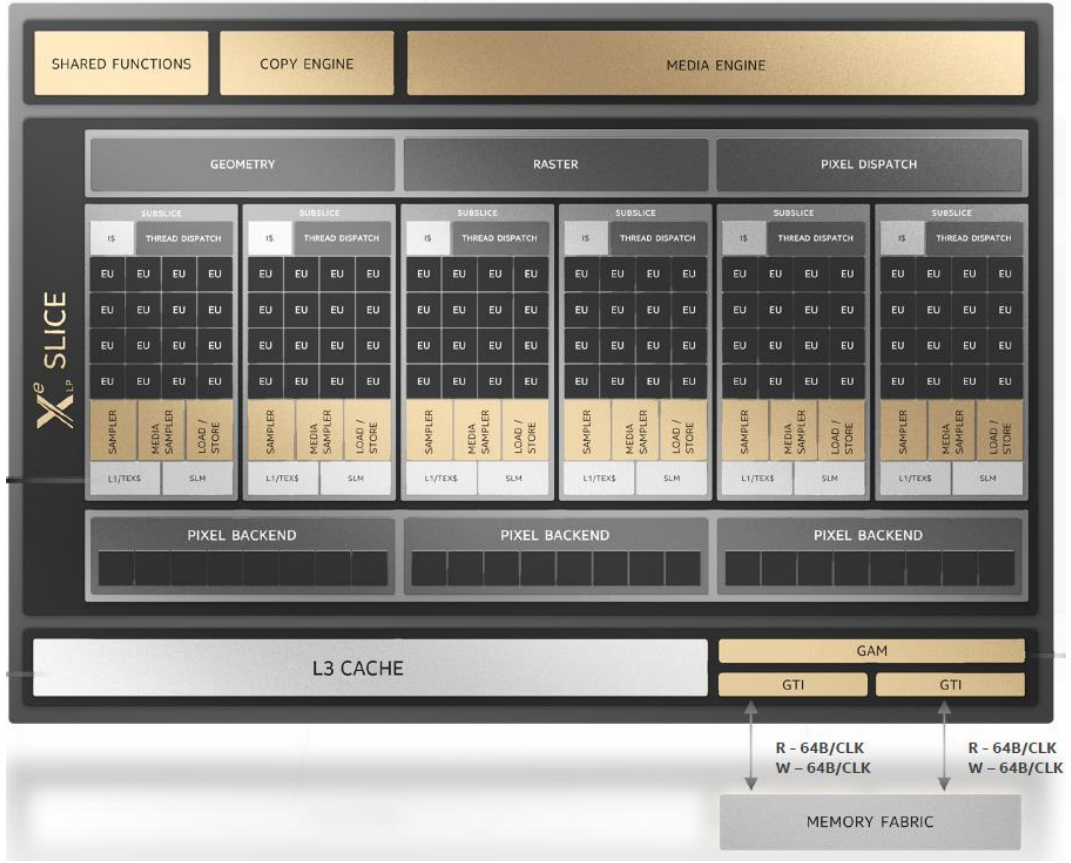


Figure 4: Gen11 detailed block diagram.

GPU Intel

- Diagrama de bloques arquitectura Xe (GPU integrada)
- Pronto Intel lanzará GPUs discretas



Programación de GPUs

- En qué programamos GPUs?
  - CUDA (Nvidia)
  - OpenCL (estándar abierto)
  - OpenACC
  - SYCL (estándar abierto)
  - DPC++

- En la próxima unidad veremos diferentes modelos de programación

Referencias

- Parallel Computing, CS 149 (Fall 2019), Stanford University
- Paul Richmond. GPU Arquitectures http://paulrichmond.shef.ac.uk/teaching/COM4521/
- Qin CZ. (2017) Cuda/GPU. In: Shekhar S., Xiong H., Zhou X. (eds) Encyclopedia of GIS. Springer, Cham. https://doi.org/10.1007/978-3-319-17885-1_1606