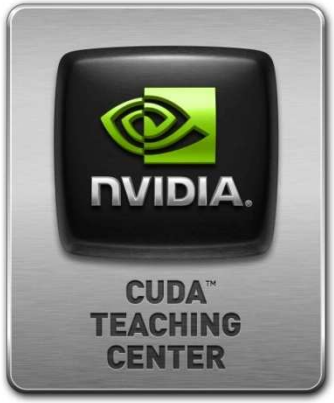
Una década de GPGPU Computing

Martín Pedemonte



Contenido

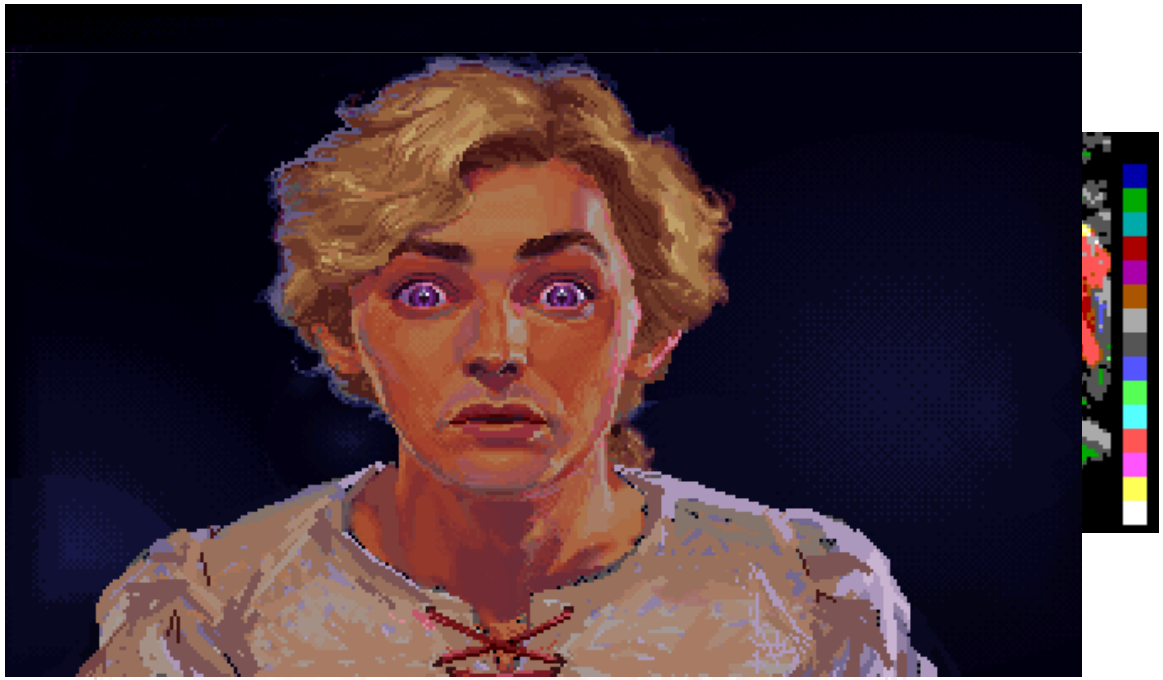
- La prehistoria
- Tarjetas programables
  - Los pioneros de GPGPU Computing
  - Lenguajes de Programación
- CUDA
  - El nacimiento de las arquitecturas unificadas
  - CPU vs GPU
  - Arquitecturas CUDA G80, GT200 y Fermi.
- Fing – CTC

La prehistoria

La prehistoria

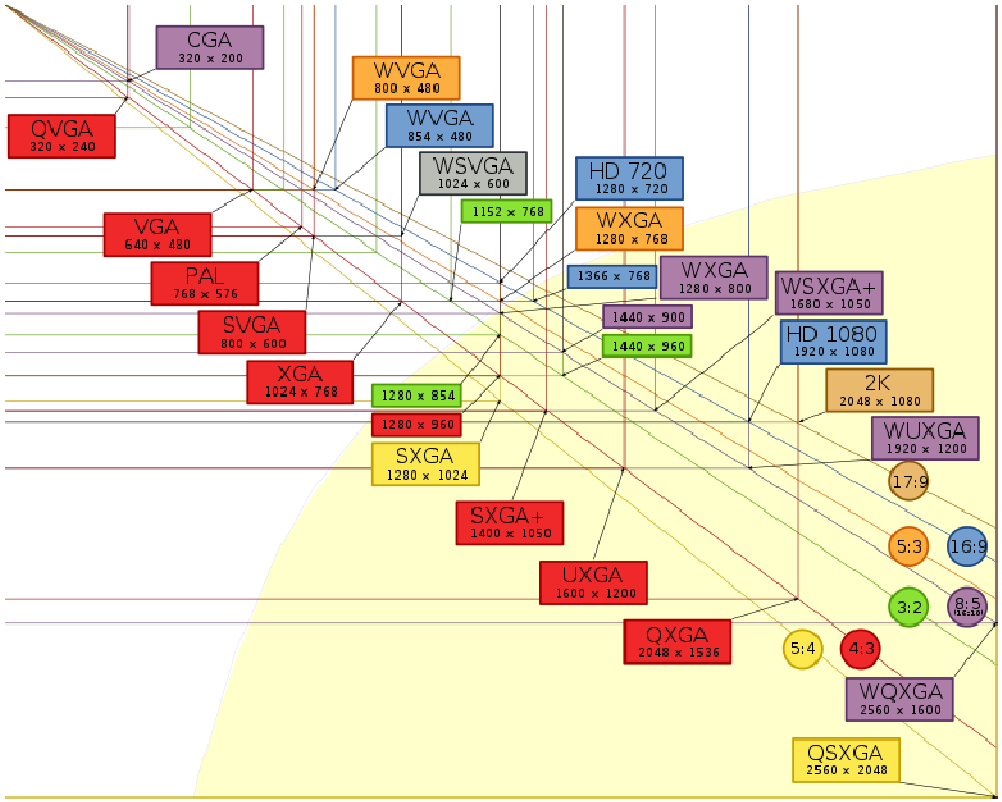
- **Durante la década del ’80**
  - **Las tarjetas de video consumer-level incorporan el color y mejoran las resoluciones**

1984



La prehistoria

- **En la década del ‘90 se continúa mejorando las resoluciones:**
  - **Super VGA: 800 x 600.**
  - **XGA: 1024 x 768 (256 colores, 8 bits por pixel) y 800 x 600 (65536 colores, 16 bits por pixel).**

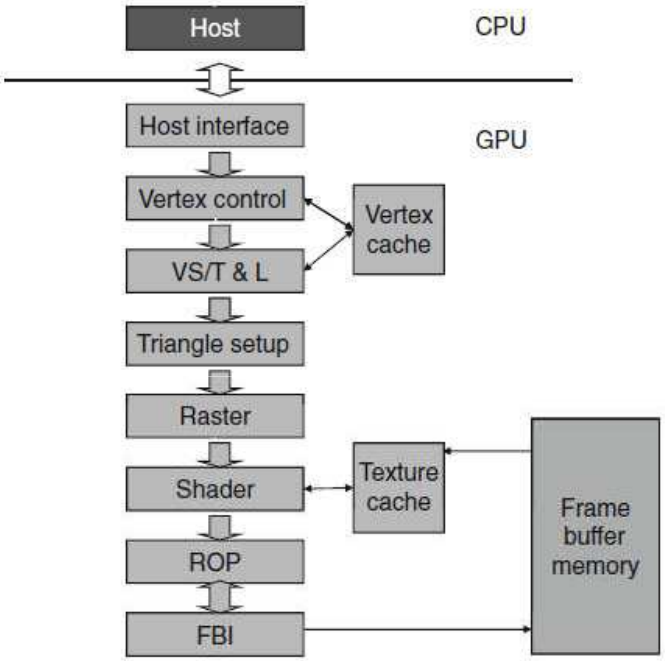


La prehistoria

- **Típicamente se deben desplegar imágenes que son proyecciones de escenas tridimensionales en una pantalla bidimensional.**
- **Las operaciones para visualizar imágenes en pantalla se agrupan en un proceso generalmente conocido como pipeline gráfico.**
- **Durante la década del 90 se diseñan tarjetas gráficas que van incorporando etapas del pipeline gráfico, con el fin de aliviar a la CPU.**

La prehistoria

- Pasos fundamentales del pipeline gráfico:
  - transformación e iluminación en los vértices (vertex shaders)
  - determinar el color final de los píxeles (pixel shaders).
- Originalmente, las GPUs proveían operaciones propias (fijas) para dichos pasos.
- Pero eso estaba por cambiar…



La prehistoria

- **Durante la década del ‘90 se presentan las primeras tarjetas con capacidad de reproducir efectos en 2D/3D:**
  - **3dfx Interactive (desde 1996):**
    - **Línea Voodoo**
  - **ATI (desde 1995):**
    - **Línea Rage**
  - **Nvidia (desde 1998):**
    - **Líneas TNT y GeForce**

La prehistoria

- **En 1999, Nvidia lanza al mercado la GeForce 256 que es considerada la primera GPU (Graphics Processing Unit).**
- **La GeForce 256 es la primera tarjeta de video consumer-level que implementa el pipeline gráfico completo.**

Tarjetas programables

Tarjetas programables

- **Originalmente, las tarjetas de video tenían un pipeline gráfico fijo.**
- **Las operaciones ejecutadas y el orden en que se aplicaban sobre los datos estaba preconfigurado.**
- **Las operaciones sobre los vértices y los píxeles eran provistas por el fabricante de la tarjeta de video.**

Tarjetas programables

- **Desde 1999 a 2006, se produce una mejora significativa en las capacidades de programación de las GPUs.**
- **Algunos hitos son:**
  - **GeForce 3 (2001): primera GPU que ejecutaba vertex shaders programados en DirectX 8.**
  - **ATI Radeon 9700 (2002): introdujo la aritmética de punto flotante de 24 bits en los pixel shaders (DirectX 9 y OpenGL).**
  - **GeForce FX (2002-2003): introdujo el trabajo con aritmética de punto flotante de 32 bits.**

Los pioneros de GPGPU Computing

- **Primeras aplicaciones de uso de tarjetas para la resolución de problemas de propósito general al comienzo de la década del 2000.**
- **Trabajos:**
  - **Cargados de ingeniosidad**
  - **Limitados por las particularidades de la arquitectura de las GPUs y su capacidad de programación**

Los pioneros de GPGPU Computing

- [2001, Larsen y McAllister]
  - Multiplicación de matrices aplicada a la búsqueda de componentes conexas en matrices de adyacencia.
  - Trabajan con números en la precisión disponible en las GPUs de la época (8 bits).
- [2001, Rumpf y Strzodka]
  - Resolución de sistemas lineales con métodos iterativos (Jacobi).
  - Aplicados a la resolución de ecuaciones diferenciales con el método de elementos finitos.
  - También trabajan con 8 bits.

Los pioneros de GPGPU Computing

- [2003, Bolz y otros]
  - Resolución de sistemas lineales con métodos iterativos
  - Estudian distintos métodos: multigrid y gradiente conjugado
  - Trabajan con matrices dispersas y completas.
- [2003, Kruger y Westermann]
  - Primer esfuerzo tendiente a la implementación de BLAS en GPU.
  - Luego CUDA brindará CUBLAS.

Lenguajes de programación

- En un principio, el avance en el hardware no fue acompañado por un avance en el software de manejo de las GPUs.
- Se comenzó desarrollando los shaders en el lenguaje ensamblador específico de cada modelo.
- Esto implicaba la existencia de varios lenguajes y baja portabilidad de los programas.
- Otra alternativa era utilizar las APIs gráficas como OpenGL y DirectX.
- Evidentemente esto representaba una limitación para el desarrollo de aplicaciones.

Lenguajes de programación

- Para solucionarlo se desarrollaron diferentes lenguajes de programación de más alto nivel que funcionaran sobre los modelos de GPU existentes, como: High-Level Shading Language (HLSL) y Cg.
- Posteriormente, otros lenguajes de alto nivel surgieron basados en considerar la GPU como un stream processor como: Brook, Sh, PyGPU, Accelerator Language, Close to the Metal (CTM) y ATI Stream.
- Sin embargo, cada herramienta seguía siendo muy dependiente de la arquitectura de la GPU, el modelo, etc.
- Estas limitaciones fueron eliminadas con la aparición de CUDA.

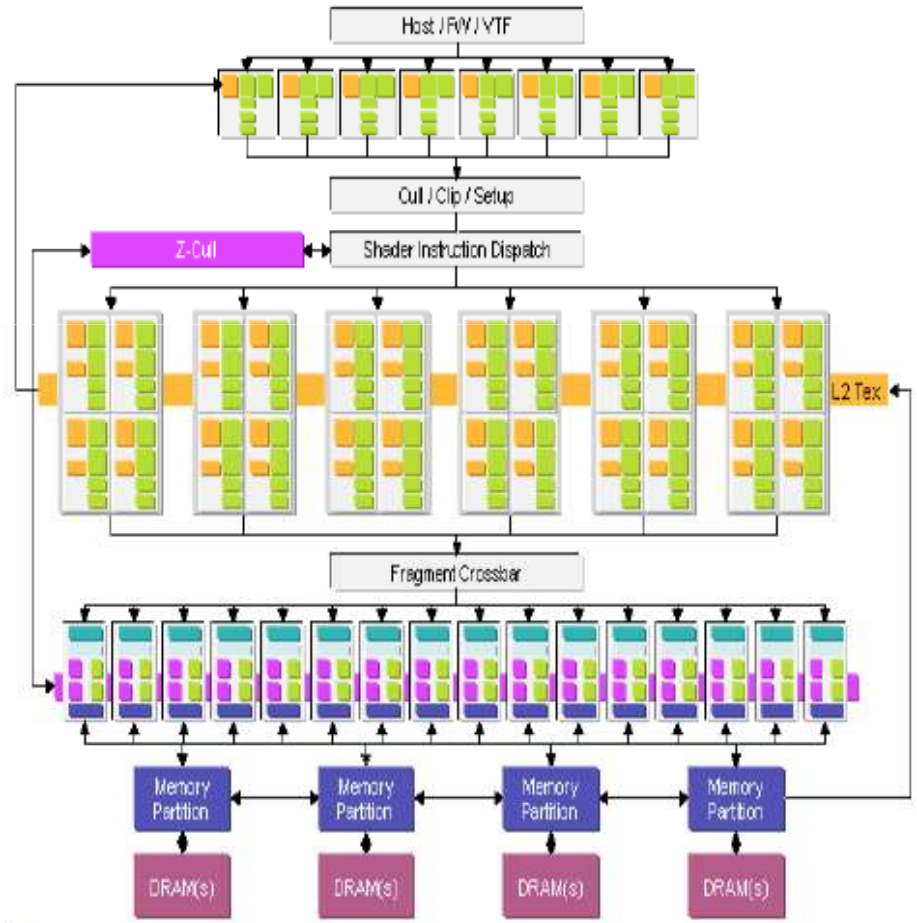
CUDA

El nacimiento de las arquitecturas unificadas

- **Antecedentes:**
  - **Xbox 360 (2005): primera arquitectura unificada de procesadores para el pipeline gráfico. Una sola clase de procesadores computa las distintas secciones del pipeline.**

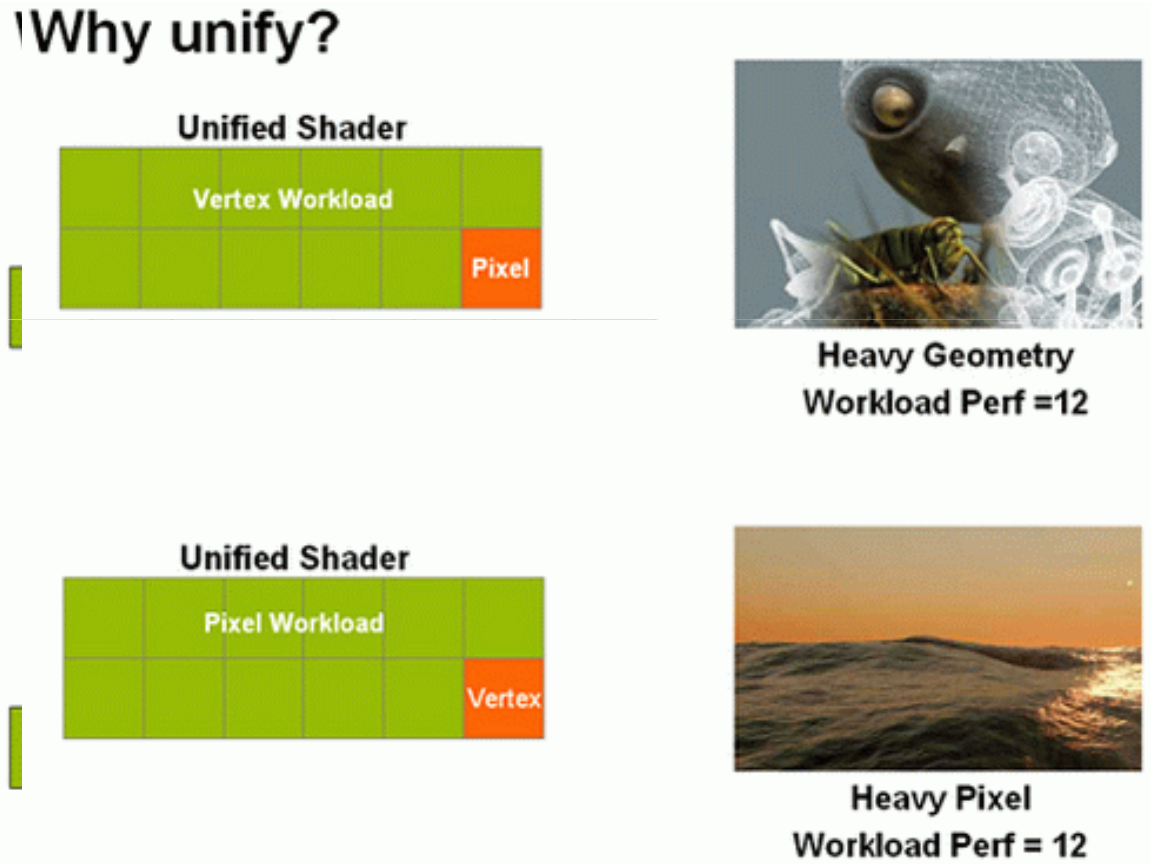
El nacimiento de las arquitecturas unificadas

La arquitectura de la GeForce 7800



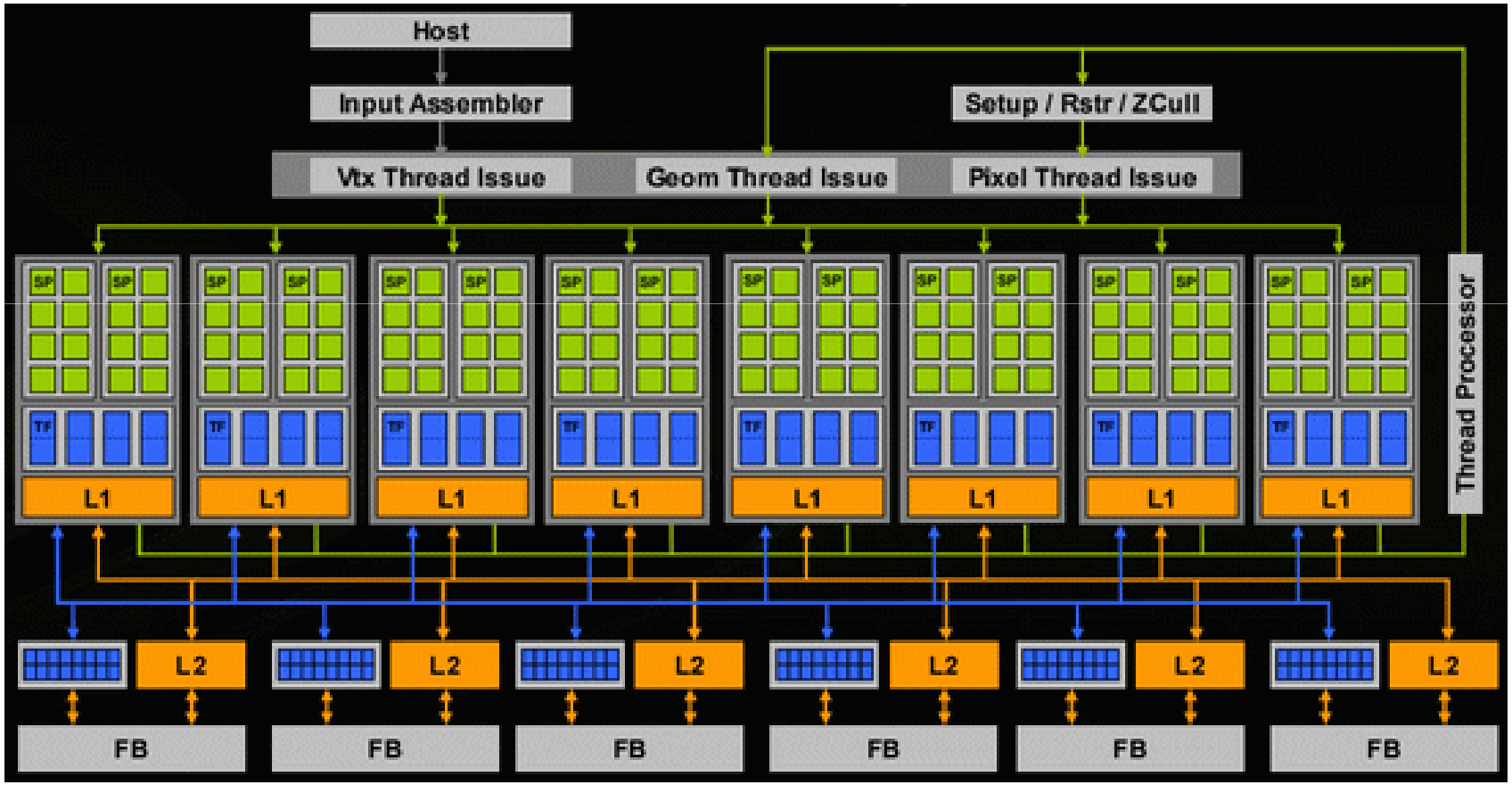
El nacimiento de las arquitecturas unificadas

¿Por qué una arquitectura unificada?



El nacimiento de las arquitecturas unificadas

La arquitectura del chip G80 (GeForce 8800)



El nacimiento de las arquitecturas unificadas

- En el año 2007, Nvidia presenta CUDA (Compute Unified Device Architecture).
- Cambio radical en la arquitectura de las GPUs de Nvidia:
  - arquitectura unificada sin distinción entre procesadores de píxeles y vértices.
- Produjo un cambio radical en el software para desarrollo de aplicaciones en las GPUs de Nvidia.
- Es el mojón más importante desde el nacimiento de la programación de propósito general en GPUs (GPGPU).
- Masificó la GPGPU, ya que dejó de ser un juego para eruditos y se transformó en una alternativa a la alcance de cualquier desarrollador.

CUDA

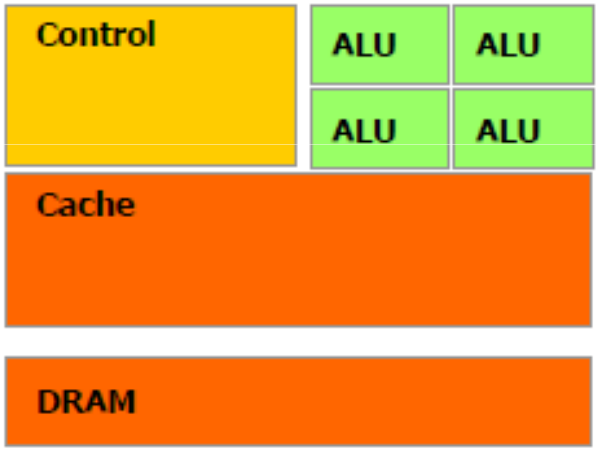
- Es una arquitectura de computación paralela para el cómputo de problemas de propósito general diseñada por Nvidia.
- Está enfocada al cálculo masivamente paralelo y las capacidades de procesamiento que brinda las tarjetas gráficas de Nvidia.
- Permite programar el dispositivo a través de extensiones de lenguajes de programación estándar (C y Fortran)
- Está disponible para las tarjetas gráficas GeForce de la serie 8 en adelante.

CUDA

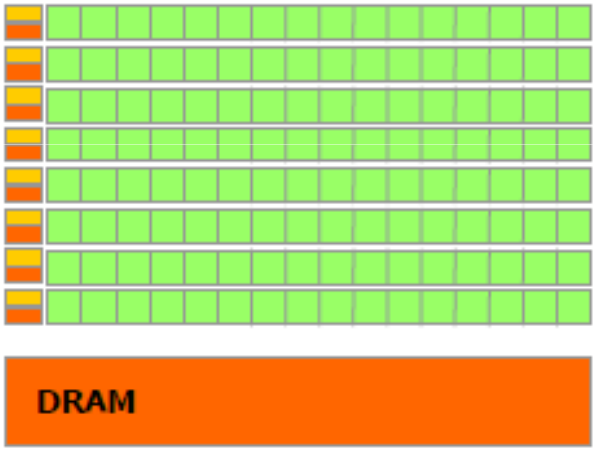
- En los últimos años existe un auge de la utilización de GPUs para el cómputo de problemas de propósito general.
- Este crecimiento se basa fundamentalmente en:
  - La arquitectura es intrínsecamente paralela
  - La industria de los videojuegos presiona a los fabricantes de tarjetas gráficas a aumentar las capacidades de procesamiento gráfico para que los juegos sean más realistas y más rápidos.
  - El surgimiento de lenguajes de programación de propósito general para GPUs como CUDA.

CPU vs GPU

- **La arquitectura de las GPUs es radicalmente distinta a la de una CPU.**



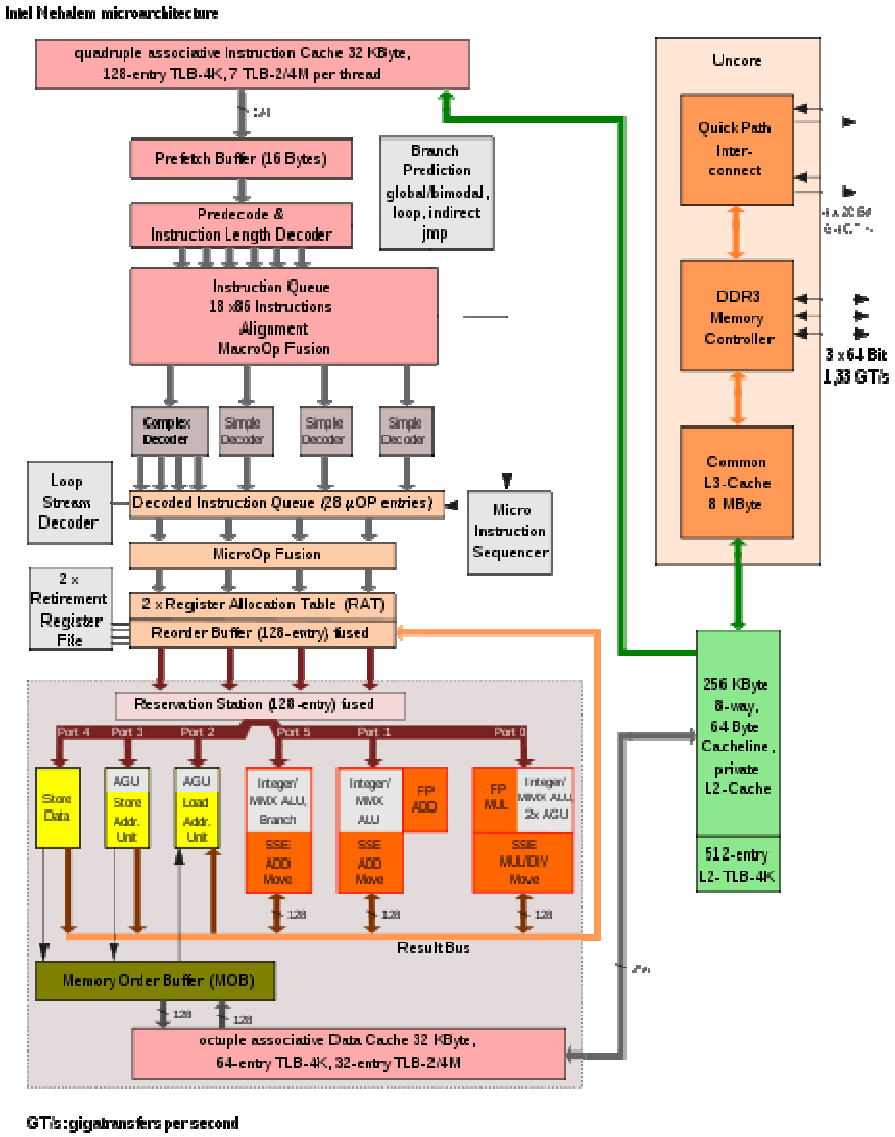
CPU



GPU

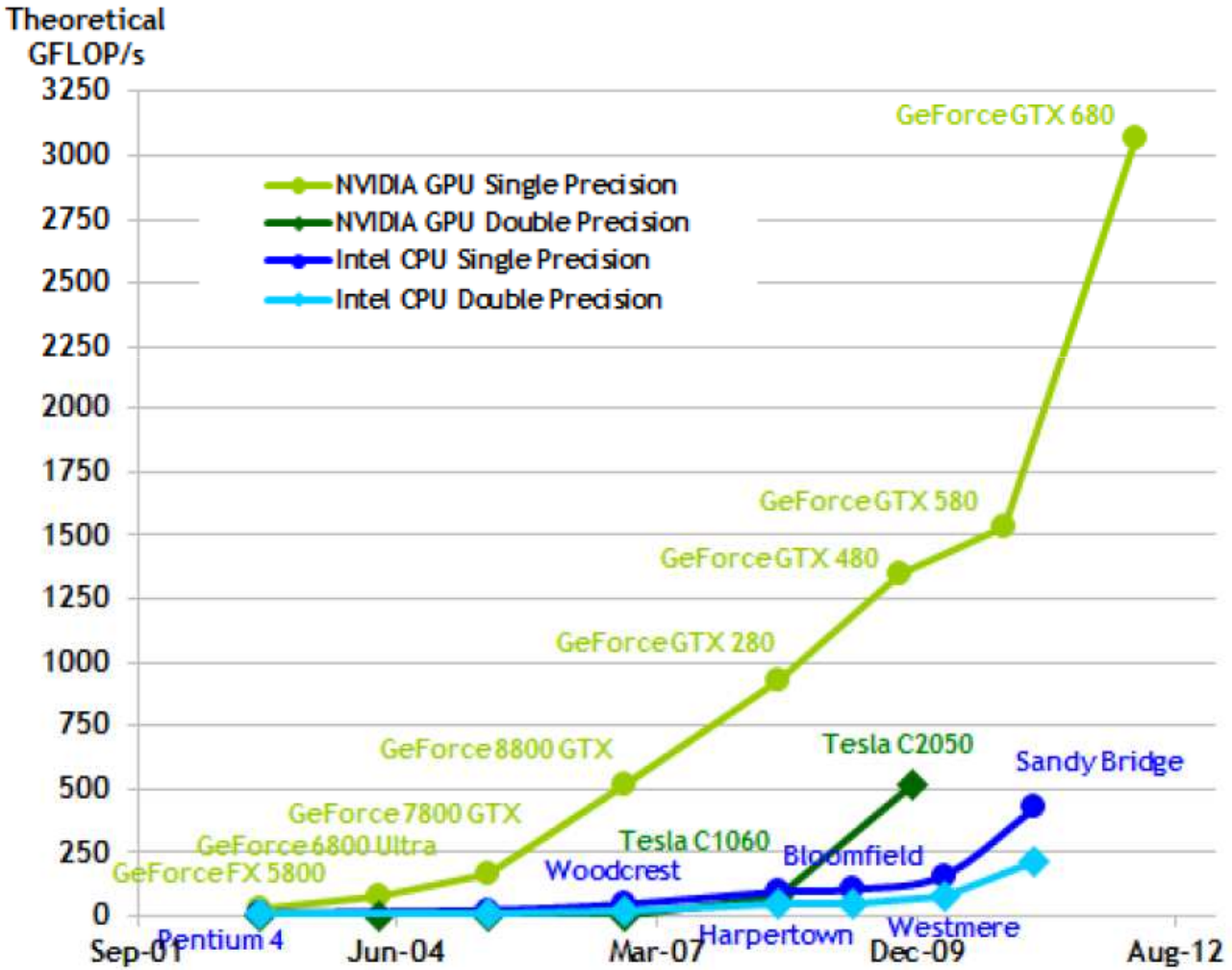
CPU vs GPU

- En una CPU tradicional gran parte de los transistores están dedicados a realizar otro tipo de tareas:
  - Predicción de branches.
  - Prefetch de memoria.
  - Ejecución fuera de orden.
  - Caché de datos.
- En las GPUs hay más transistores dedicados al cálculo.



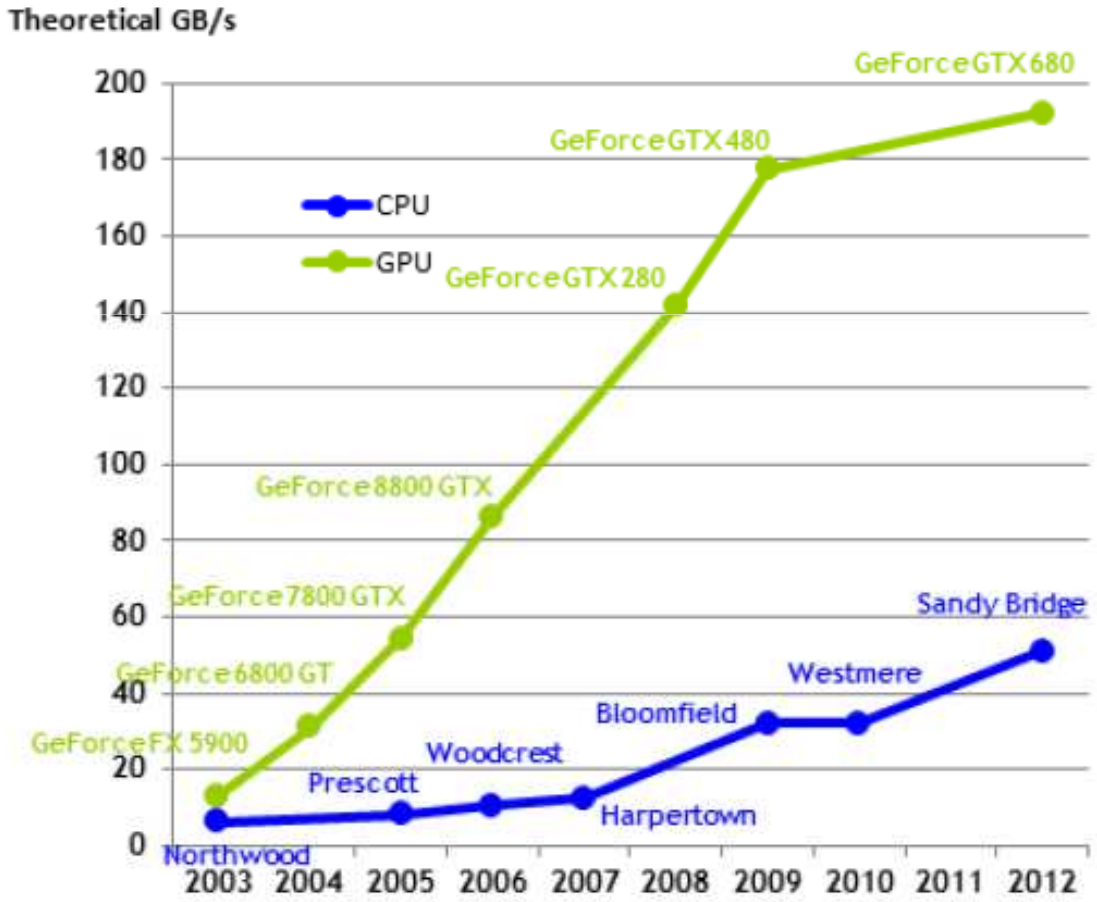
CPU vs GPU

Pico teórico de performance en GFLOP/s



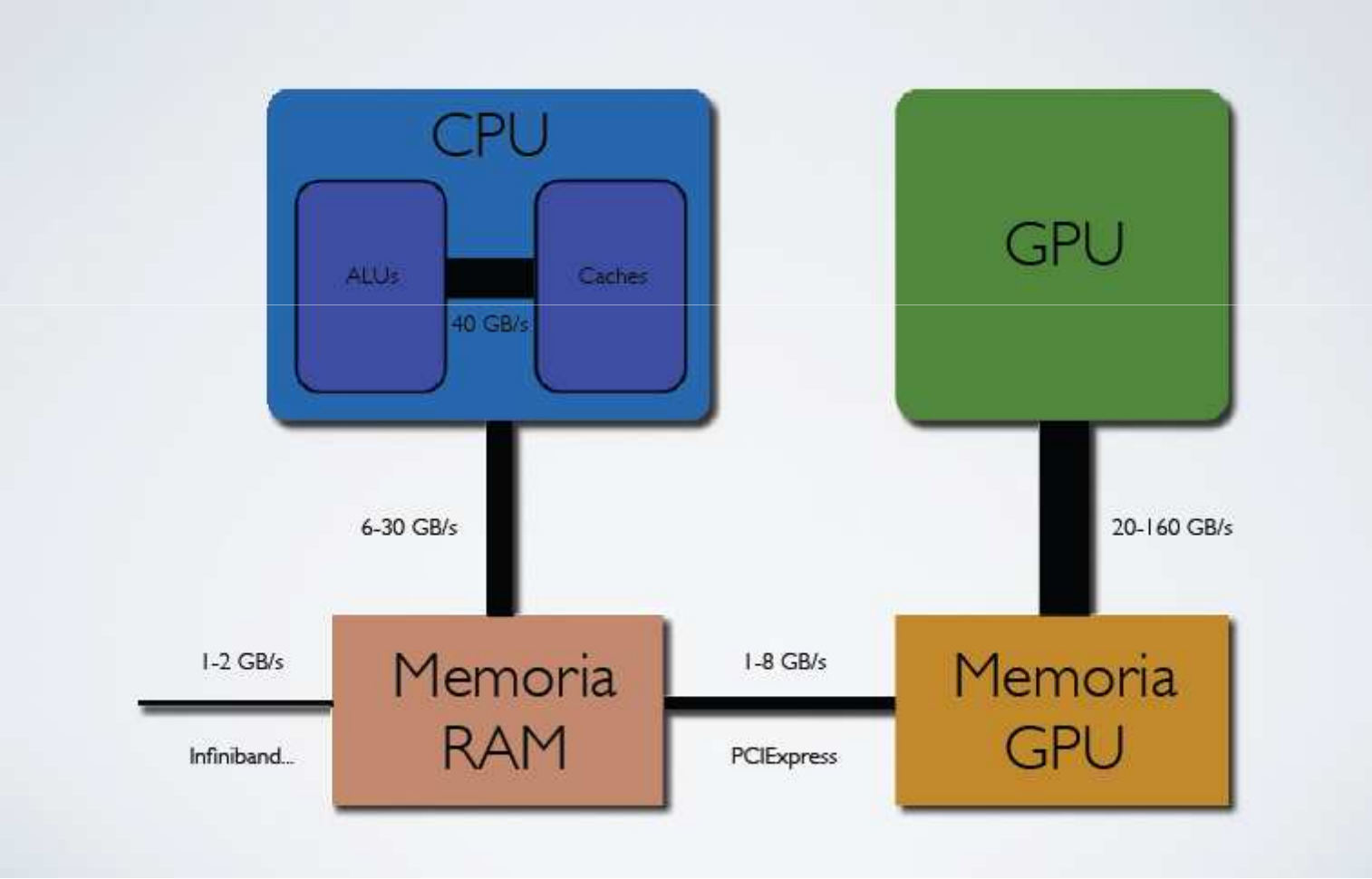
CPU vs GPU

Pico teórico de tasa de transferencia de memoria en GB/s



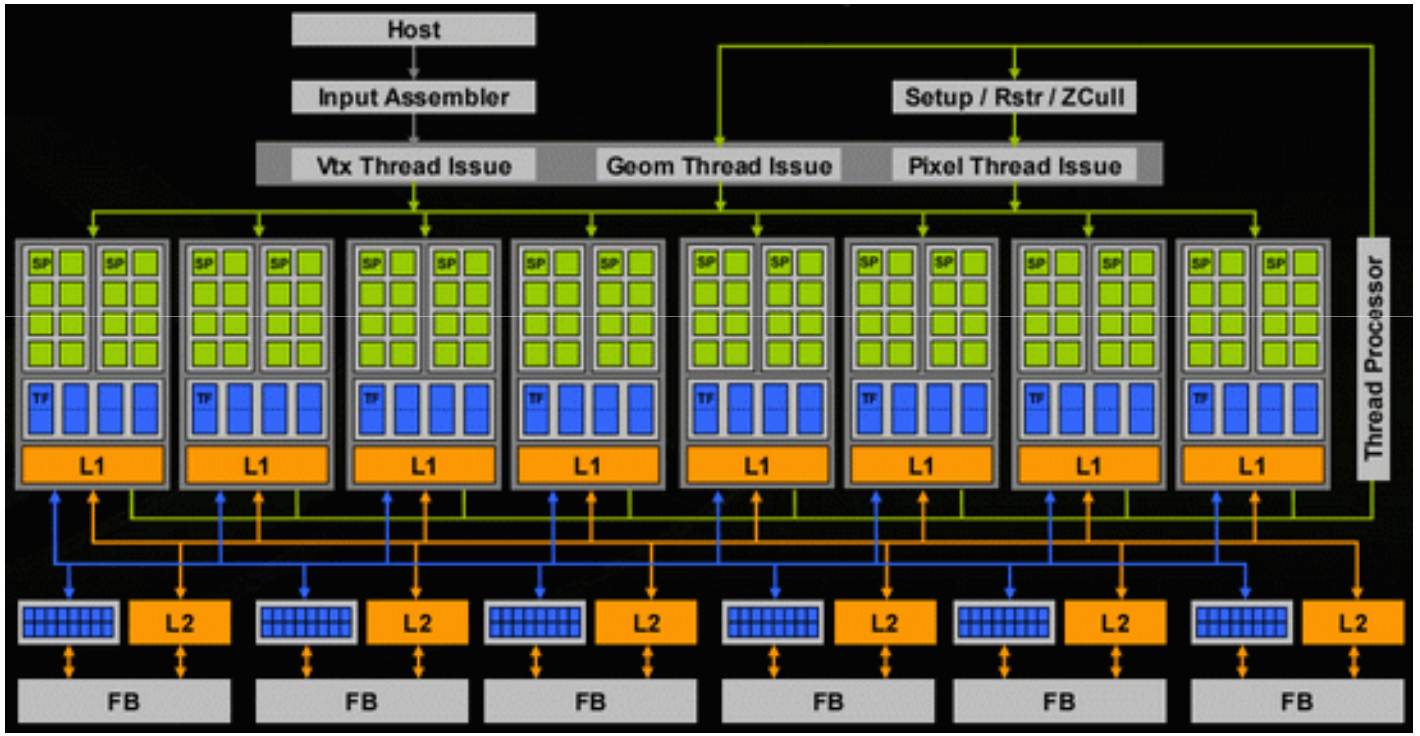
CPU vs GPU

Ancho de banda



Arquitectura CUDA – G80

La Arquitectura G80



Arquitectura CUDA – G80

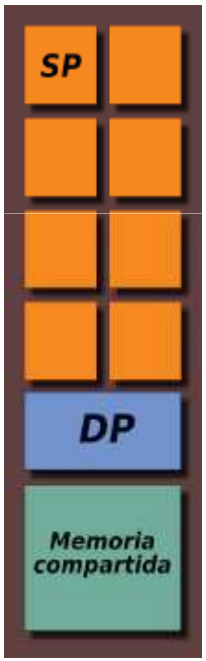
- **Un multiprocesador (SM) de la arquitectura G80 consiste en:**
  - **ocho procesadores escalares (Streaming Processors, SP). También conocidos como CUDA cores.**
  - **unidad de instrucciones multihilo**
  - **chip de memoria compartida**
  - **dos unidades especiales (Special Function Units, SFU) para computar operaciones trascendentales como sin, cos, log, y sqrt.**

Arquitectura CUDA – G80

- Cada CUDA core tiene una unidad que permite realizar una operación multiply-add y una unidad que permite hacer una multiplicación.
- Es decir que los CUDA cores básicamente son una ALU.
- Los CUDA cores no tienen registros propios o caches.
- Los registros se manejan a nivel de multiprocesador.
- Los CUDA cores soportan la aritmética de punto flotante de simple precisión (32 bits) definida en el estándar IEEE 754-1985.

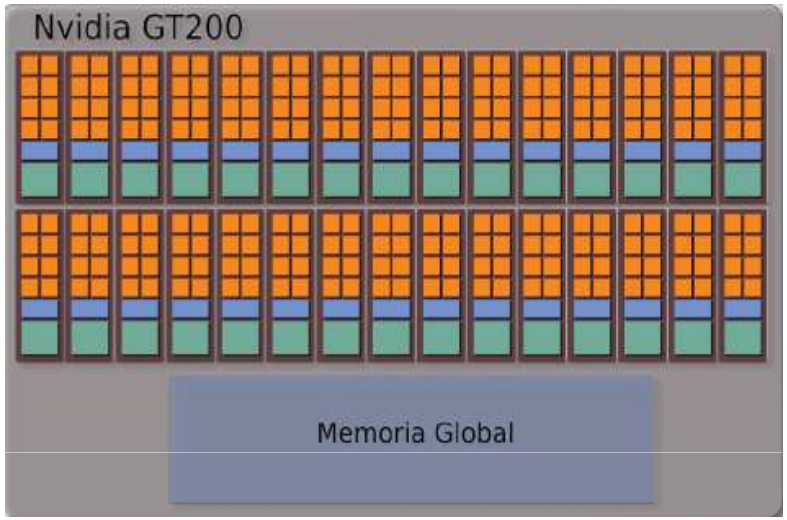
Arquitectura CUDA – GT200

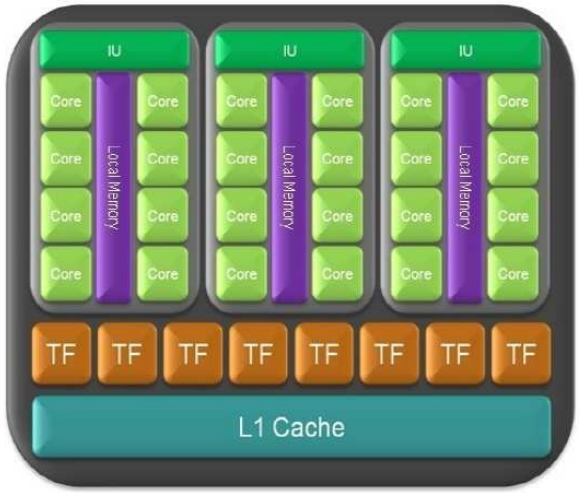
- En 2008 se lanzó la segunda generación de CUDA.
- Es una revisión menor (?) de la primera generación.
- Algunos de los cambios más importantes son:
  - Soporte a aritmética de doble precisión (64 bits) del estándar IEEE 754-1985. Para ello cada SM incorpora una unidad de doble precisión.
  - Soporte a operaciones atómicas read-modify-write en memoria compartida y memoria global.
  - Mejora en la performance de las SFUs.



Arquitectura CUDA – GT200

- **Algunos de los cambios más importantes son:**
  - **Los TPCs agrupan 3 SMs.**
  - **Las GPUs tienen 10 TPCs, resultando en 30 SMs y 240 CUDA cores.**



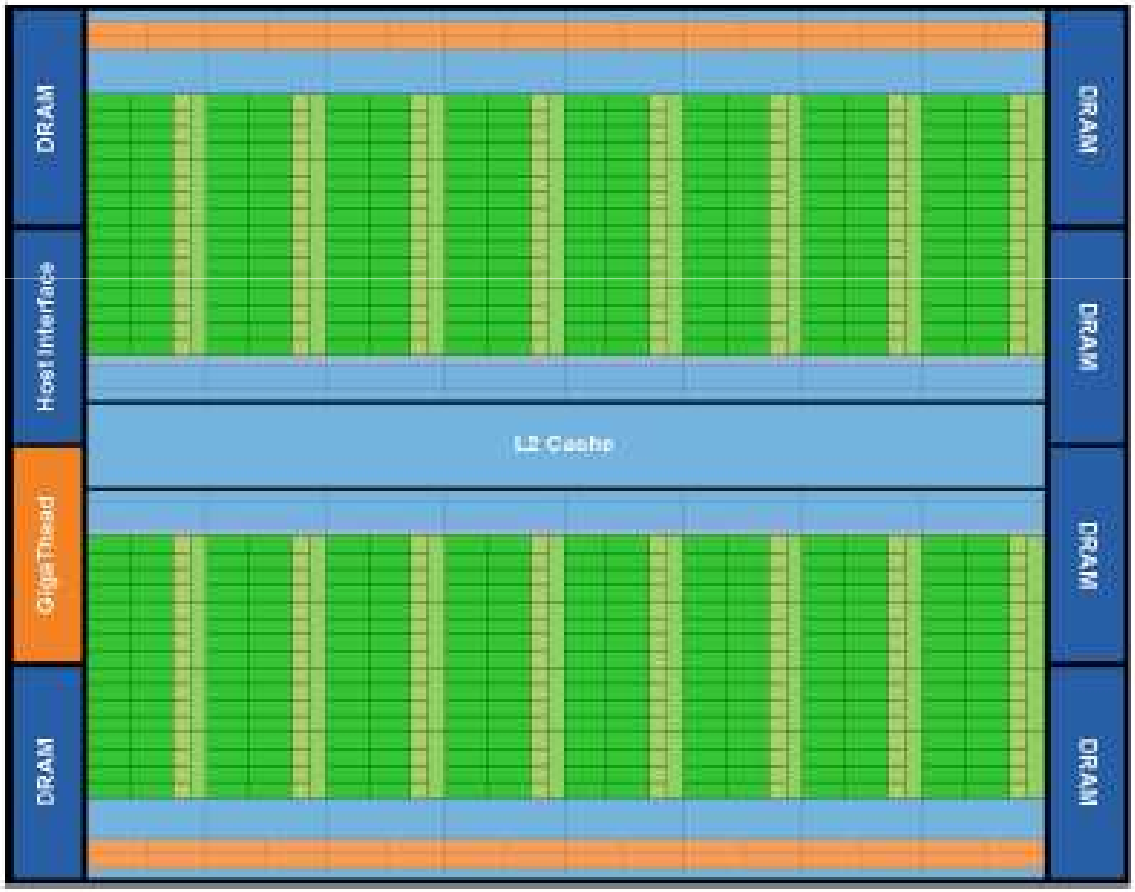


Arquitectura CUDA – Fermi

- En 2010 se lanza la tercera generación de CUDA.
- Esta nueva arquitectura representa una mejora sustancial sobre las arquitecturas previas.
- Algunos de los cambios más importantes son:
  - Los CUDA cores soportan completamente el estándar 754-2008 de la IEEE para simple y doble precisión (se agregó el soporte para números desnormalizados).
  - Se unifica el espacio de direcciones de memoria (global, shared y local), lo que permite dar soporte completo a C++.
  - Mejora en la performance de las operaciones atómicas.

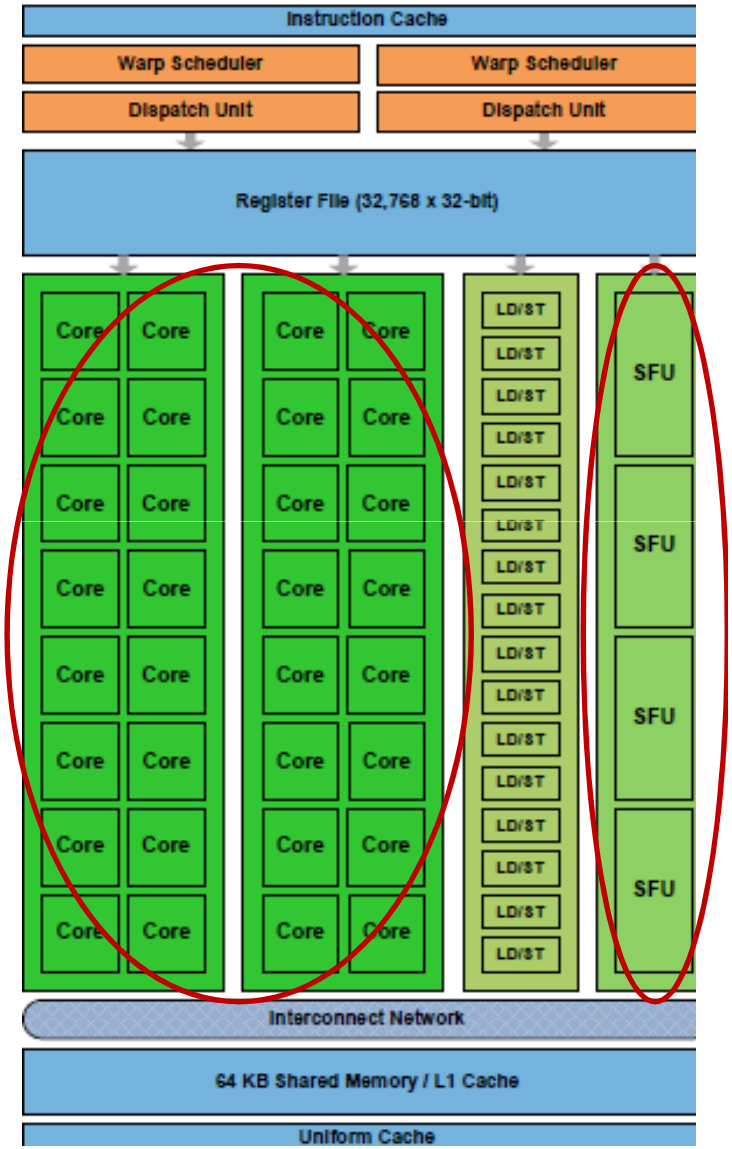
Arquitectura CUDA – Fermi

- **Algunos de los cambios más importantes son:**
  - **Tiene 16 SMs con 32 CUDA cores cada uno (512 cores en total).**



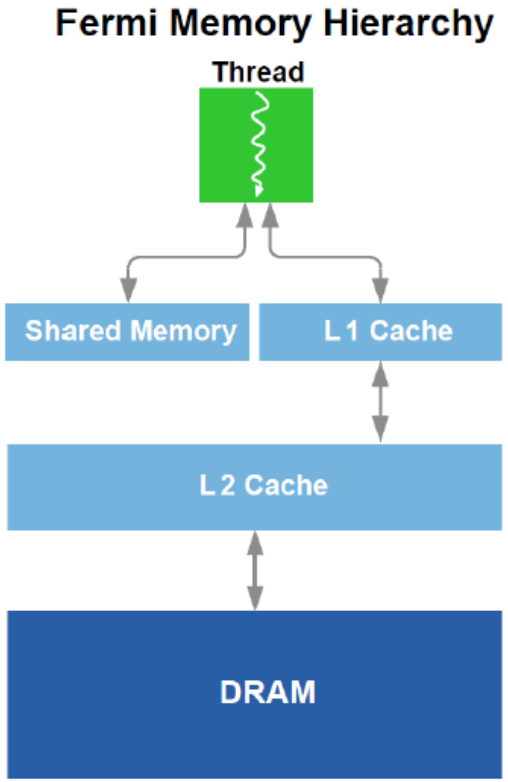
Arquitectura CUDA – Fermi

- **Algunos de los cambios más importantes son:**
  - **Los SMs están organizados en dos bloques de 16 cores cada uno.**
  - **Tienen 4 SFUs lo que permite que en ocho ciclos de reloj ejecute un warp.**
  - **El pipeline de SFUs está desacoplado de la dispatch unit por lo que se pueden despachar instrucciones a otras unidades mientras las SFUs están ocupadas.**
  - **Aumento significativo en la performance de punto flotante de doble precisión (relación 2x con simple precisión).**



Arquitectura CUDA – Fermi

- **Algunos de los cambios más importantes son:**
  - **Se incorporan dos niveles de caché L1 (hasta 48K) y L2 (768 K)**



  - **Se incorporan mecanismos para la detección y corrección de errores en el acceso a datos.**

Fing - CTC

Fing - CTC

- **Las GPUs son:**
  - **Una plataforma ampliamente disponible**
  - **Una tecnología barata**
- **Son ideales para la realidad latinoamericana y uruguaya:**
  - **Es posible disponer del hardware más nuevo a nivel mundial con poco atraso y a un costo razonable (U$S 500)**

Fing - CTC

- **Existe una apuesta fuerte de Nvidia por la adopción de GPUs para la computación de alta performance.**
- **Cientos de instituciones educativas enseñando GPGPU.**
- **Desde el año 2007:**
  - **se viene trabajando en investigación usando tecnologías CUDA**
  - **se ha incorporado CUDA a diversos cursos**
- **Desde 2011:**
  - **Fing se suma al programa CUDA Teaching Center de Nvidia, siendo la segunda institución en Latinoamérica y la primera en Sudamérica en hacerlo.**
  - **Actualmente se dictan los cursos "Computación de propósito general en unidades de procesamiento gráfico" y "Taller de GPGPU"**

Muchas gracias