中山大學
SUN YAT-SEN UNIVERSITY

国家超级计算广州中心
NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

# Computer Architecture

# 计 算 机 体 系 结 构

## 第8讲：DLP & GPU（2）

张献伟

xianweiz.github.io

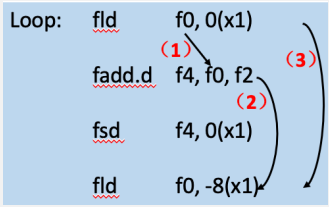
DCS3013, 10/31/2022

# Quiz Questions


**Plz email to zhangxw79@mail.sysu.edu.cn (no later than 14:35).**

- Q1: which is anti-dependence? (1)/(2)/(3)/None

  (2): read f0, then write → WAR, the reverse of RAW




- Q2: side effect of loop unrolling?

  Code size, compiler complexity.

- Q3: differences between Scoreboard and Tomasulo?

  Tomasulo relies on hardware to enforce and resolve deps.

- Q4: what is VLIW?

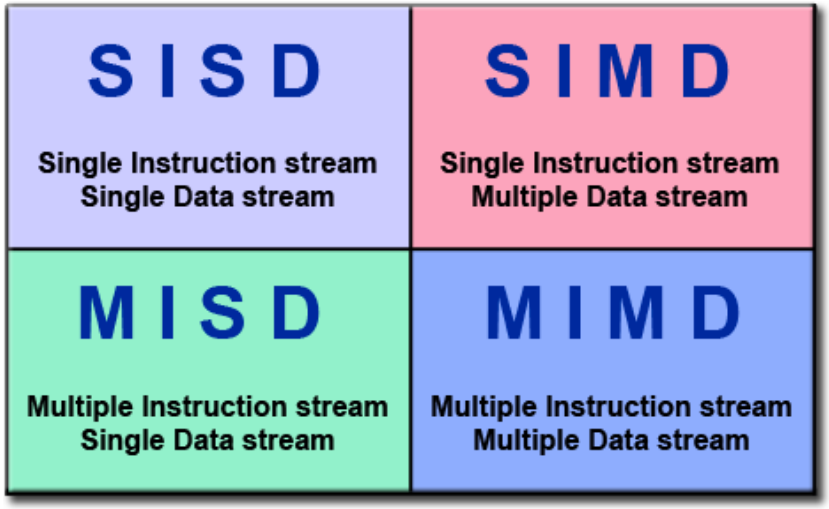
  Very Long Instruction Word. Packing multi-insts into one.

- Q5: for a 5-way superscalar machine, what's the ideal CPI?

  0.2. Issue 5 instructions per cycle, and finish in one cycle.

# Taxonomy[分类]

- **Flynn's Taxonomy** (1966) is widely used to classify parallel computers
  - Distinguishes multi-processor computer architectures according to how they can be classified along the two independent dimensions of ***Instruction Stream*** and ***Data Stream***
  - Each of these dimensions can have only one of two possible states: ***Single*** or ***Multiple***
- 4 possible classifications according to Flynn




https://hpc.llnl.gov/training/tutorials/introduction-parallel-computing-tutorial

# Execution Model[执行模型]

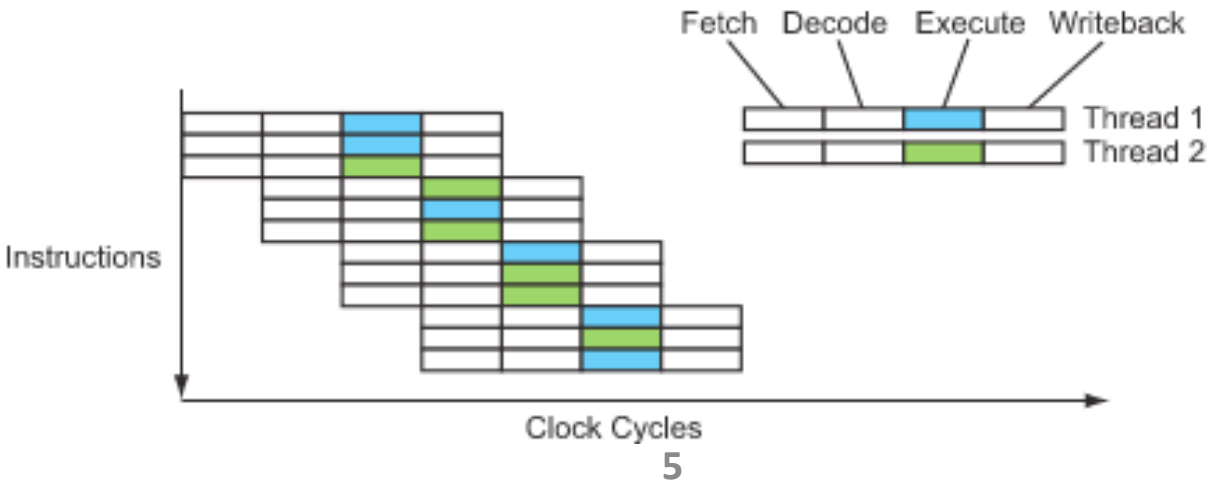
| MIMD | SIMD | SIMT |
| --- | --- | --- |
|  |  |  |
| Multiple independent threads | One thread with wide execution datapath | Multiple lockstep threads |
| Multicore CPUs | x86 SSE/AVX | GPUs |

- SI(MD/MT)
  - Broadcasting the same instruction to multiple execution units
  - Replicate the execution units, but they all share the same fetch/decode hardware

**SIMD and SIMT are used interchangeably**

# SMT[多线程]

- SMT: simultaneous multithreading
  - Instructions from multiple threads issued on the same cycle
    - Use register renaming and dynamic scheduling facility of multi-issue architecture
  - Needs more hardware support
    - Register files, PC's for each thread
    - Support to sort out which threads to get results from which instructions
    - Thread scheduling, context switching
  - Maximize utilization of execution units

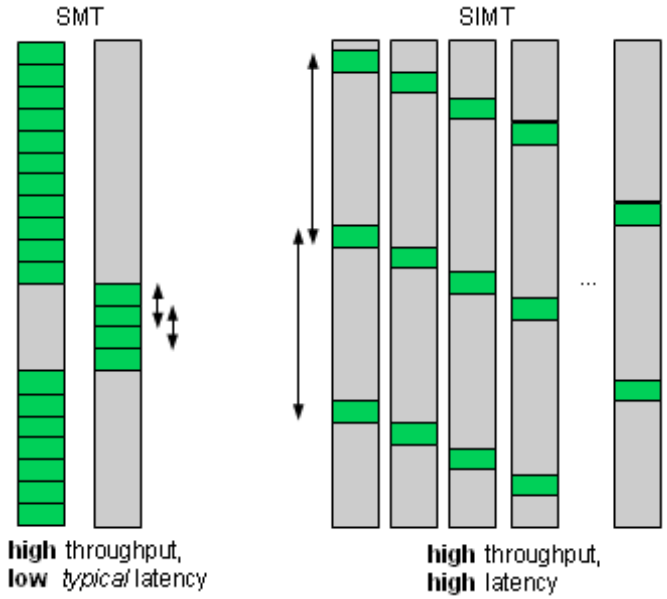





http://yosefk.com/blog/simd-simt-smt-parallelism-in-nvidia-gpus.html

# SMT vs. SIMT[比较]

- SMT: maximize the chances of an instruction to be issued without having to switch to another thread
  - superscalar execution
  - out-of-order execution
  - register renaming
  - branch prediction
  - speculative execution
  - cache hierarchy
  - speculative prefetching



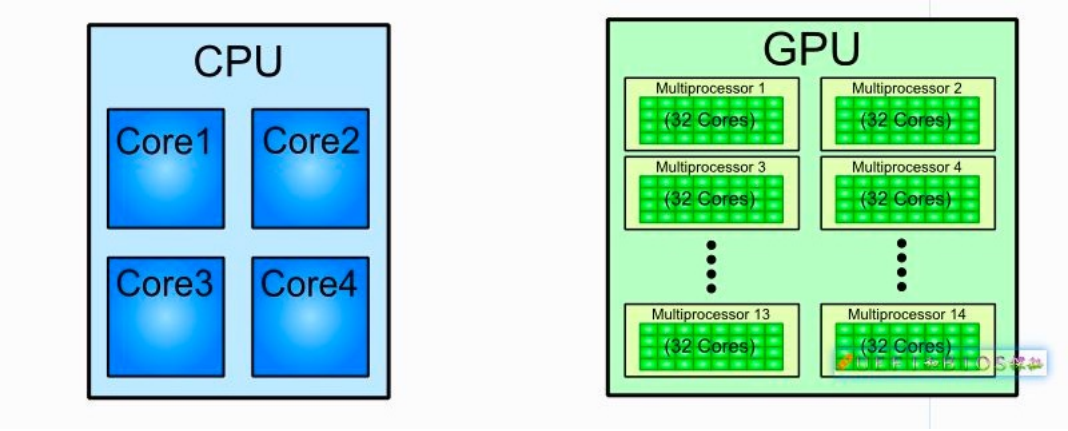

- SIMT: keep massive threads to achieve high throughput
  - Hardware becomes simpler and cheaper
  - No OoO, no prefetching, …



http://yosefk.com/blog/simd-simt-smt-parallelism-in-nvidia-gpus.html

# CPU vs. GPU[比较]

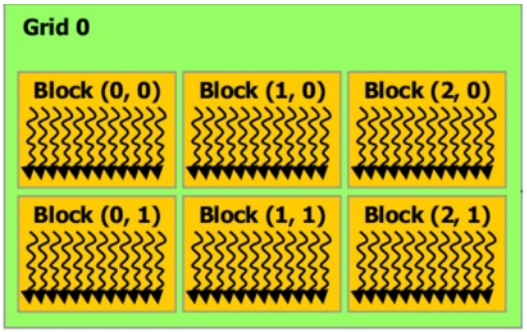
- CPU
  - Low compute density
  - Complex control logic
  - Fewer cores optimized for serial operations
    - Fewer execution units (ALUs)
    - Higher clock speeds
  - Low latency tolerance
- GPU
  - High compute density
  - Simple control logic
  - 1000s cores optimized for parallel operations
    - Many parallel execution units (ALUs)
    - Lower clock speeds
  - High latency tolerance



# GPU Overview

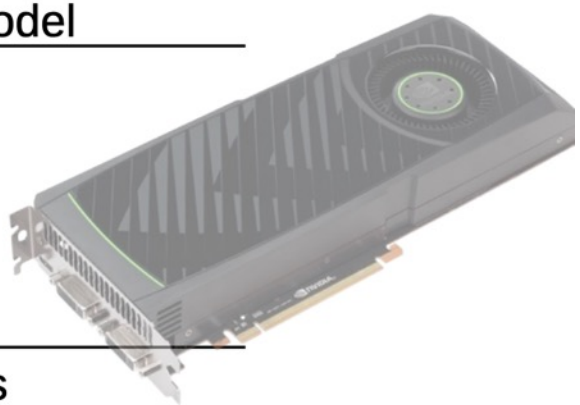
Software

```
__global__ void scale(float a, float * X)
{
    unsigned int tid;
    tid = blockIdx.x * blockDim.x
          + threadIdx.x;
    X[tid] = a * X[tid];
}
```

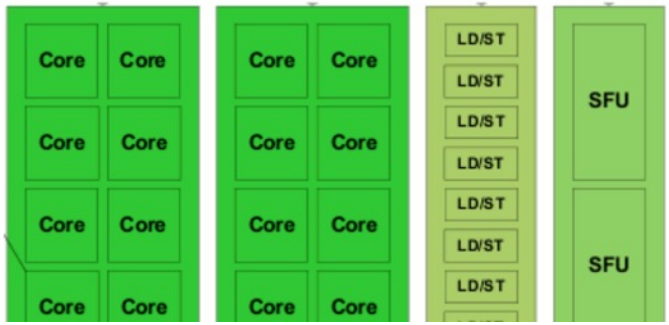



Architecture: **multi-thread** programming model

**SIMT microarchitecture**



Hardware datapaths: **SIMD** execution units




Hardware

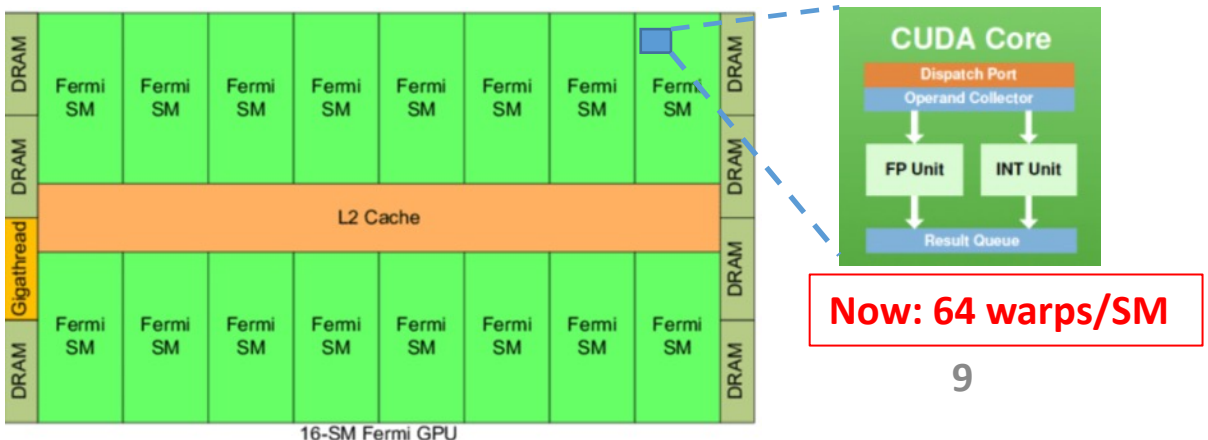
3

# GPU Overview(cont.)

- A GPU contains several largely independent processors called "**Streaming Multiprocessors**" (SMs)
  - Each SM hosts multiple "cores", and each "core" runs a thread
  - For instance, Fermi(2010) has up to 16 SMs w/ 32 cores per SM
    - So up to 512 threads can run in parallel

**A100: 128 SMs w/ 64 cores per SM**
**H100: 144 SMs w/ 128 cores per SM**

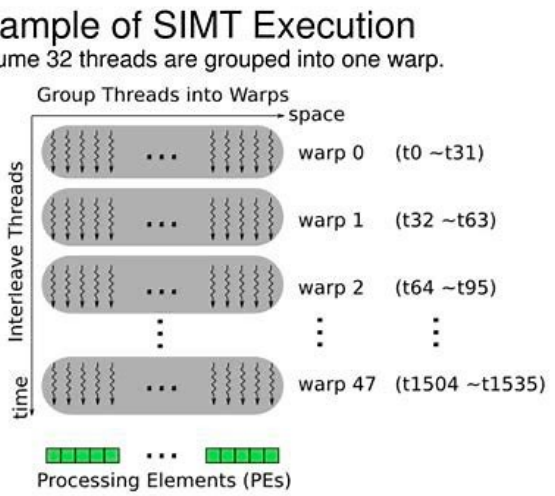
- Some SIMT threads are grouped to execute in lockstep
  - One warp contains 32 threads
- Multiple 'groups' can be executed simultaneously
  - For Fermi, up to 48 warps per SM



16-SM Fermi GPU

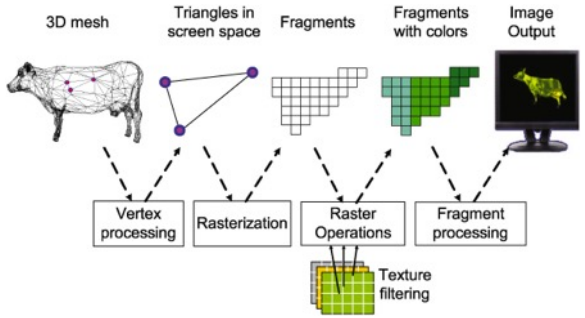



**Now: 64 warps/SM**

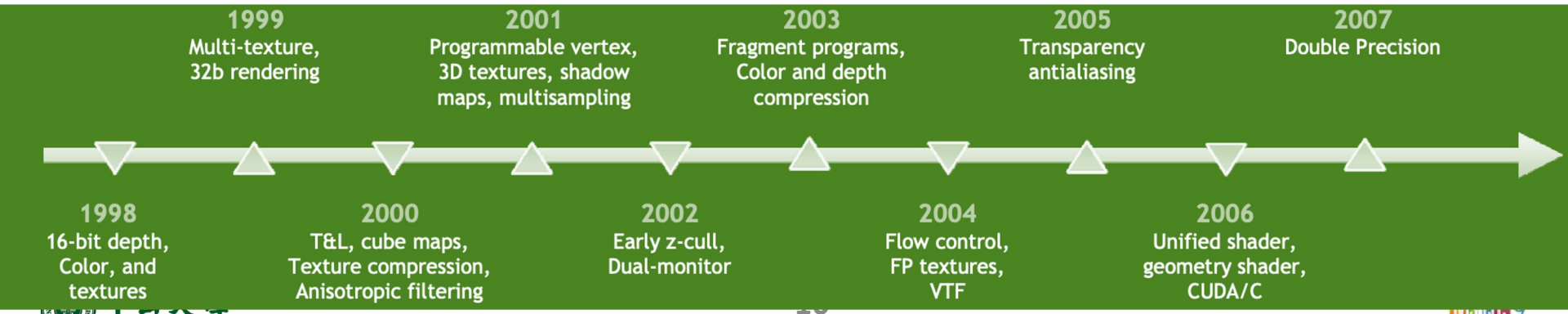


# GPU Evolution[演进]

- Arcade boards and display adapters (1951 - 1995)
  - ATI: founded in 1985
  - Nvidia: founded in 1993
- 3D revolution (1995 - 2006)
  - Term “graphics processing unit”: 1999
    - Nvidia GeForce 256
  - Rivalry between ATI and Nvidia




- General purpose GPU (2006 - present)
  - AI, data analytics, scientific computing, graphics rendering, etc.





https://www.nvidia.com/content/nvision2008/tech_presentations/Technology_Keynotes/NVISION08-Tech_Keynote-GPU.pdf

# GPGPU History[简史]

| Year | AMD | Nvidia | Note |
|---|---|---|---|
| 2006 | AMD acquired ATI | Tesla (CUDA Launch) | Unified shader model |
| 2007 | TeraScale | | Unified shader uarch |
| 2009 | TeraScale 2 | | |
| 2010 | TeraScale 3 | Fermi / GTX580 | First compute GPU |
| 2011 | GCN 1.0 / gfx6 | | VLIW → SIMD |
| 2012 | | Kepler / GTX680 | CUDA cores: 512 → 1536 |
| 2013 | GCN 2.0 / gfx7 | | |
| 2014 | GCN 3.0 / gfx8 | Maxwell / GTX980 | Energy efficiency |
| 2016 | GCN 4.0 / gfx8 | Pascal / GTX1080 | |
| 2017 | GCN 5.0 / gfx9 | Volta / GV100 | First chip with Tensor cores |
| 2018 | GCN 5.1 / gfx9 | Turing / RTX2080 | |
| 2019 | RDNA 1.0 / gfx10 | | |
| 2020 | RDNA 2.0 / gfx10<br>CDNA 1.0 / gfx9 | Ampere / RTX3090 | First chip with Matrix cores |

# TFLOPS[衡量算力]

- A100 Tensor Core GPU
  - 108 SMs
    - GA100 Full GPU with 128 SMs
  - Base clock: 1065 MHz
  - Boost clock: 1410 MHz
  - Performance
    - FP64: 9.7 TFLOPS
    - FP32: 19.5 TFLOPS


- Calculate TFLOPS
  - FP64: 1410 MHz x (32 x 2) ops/clock x 108 SMs



https://images.nvidia.cn/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf