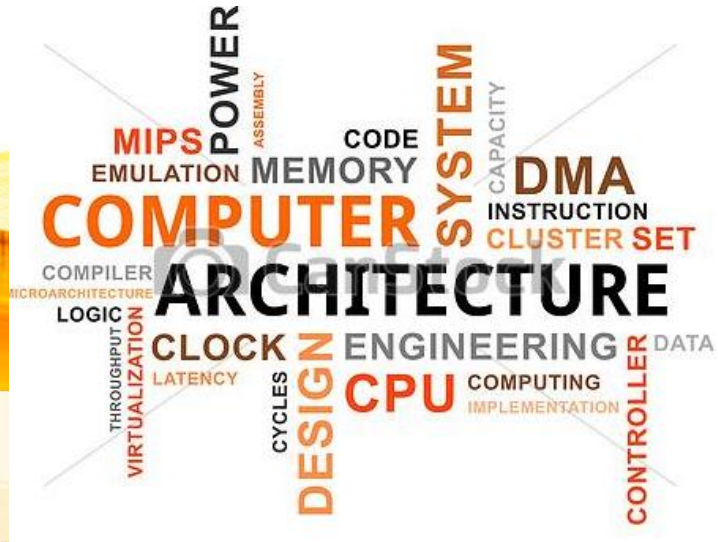
UNIT V
I/O ORGANIZATION AND PARALLELISM

Accessing I/O devices – Interrupts – Direct Memory Access - Buses– Interface circuits - Standard I/O Interfaces (PCI, SCSI, USB)–Instruction Level Parallelism : Concepts and Challenges – Introduction to multicore processor - **Graphics Processing Unit**.



Recap the previous Class

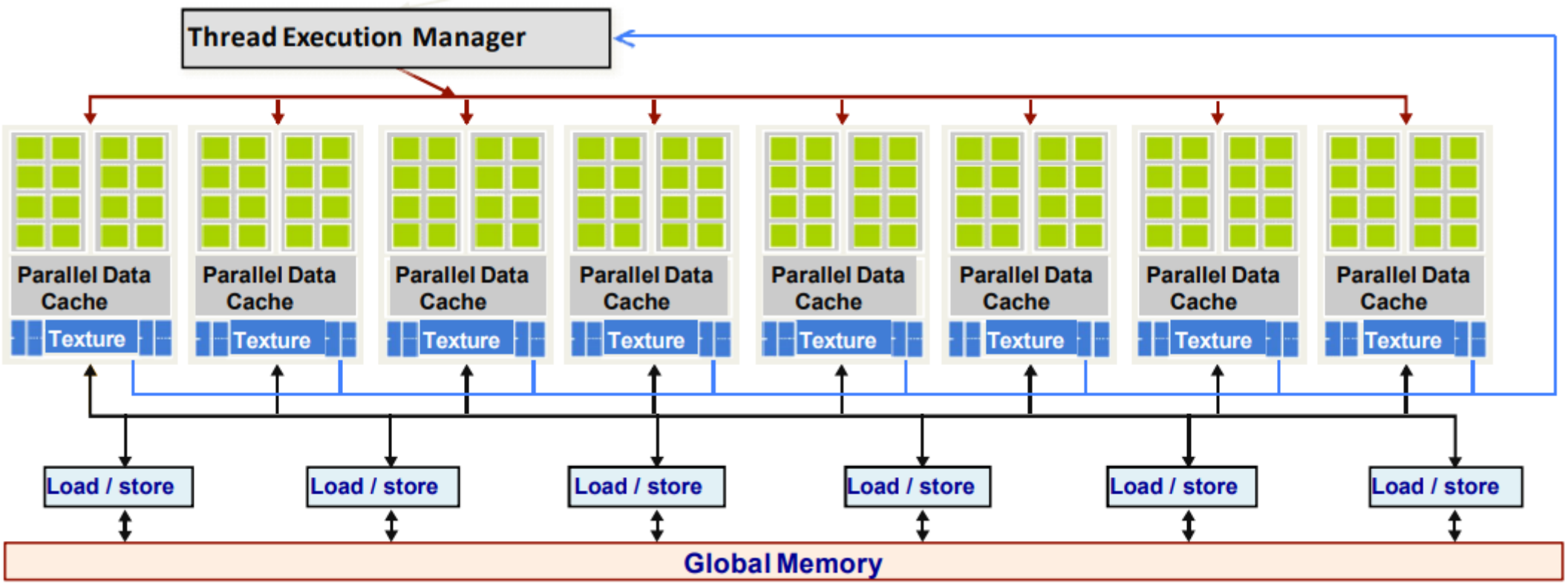
Introduction

- Graphics Processing Unit (GPU) is a processor optimized for **2D/3D graphics computation, video rendering and visual computing**.
- It is a highly parallel, highly multi-threaded multiprocessor optimized for visual computing.
- It provides real-time visual interaction with computed objects via graphics objects like image and video.
- It serves both as a programmable graphics processor and a scalable parallel computing engine.

– Heterogeneous computing systems combine a GPU with a CPU.

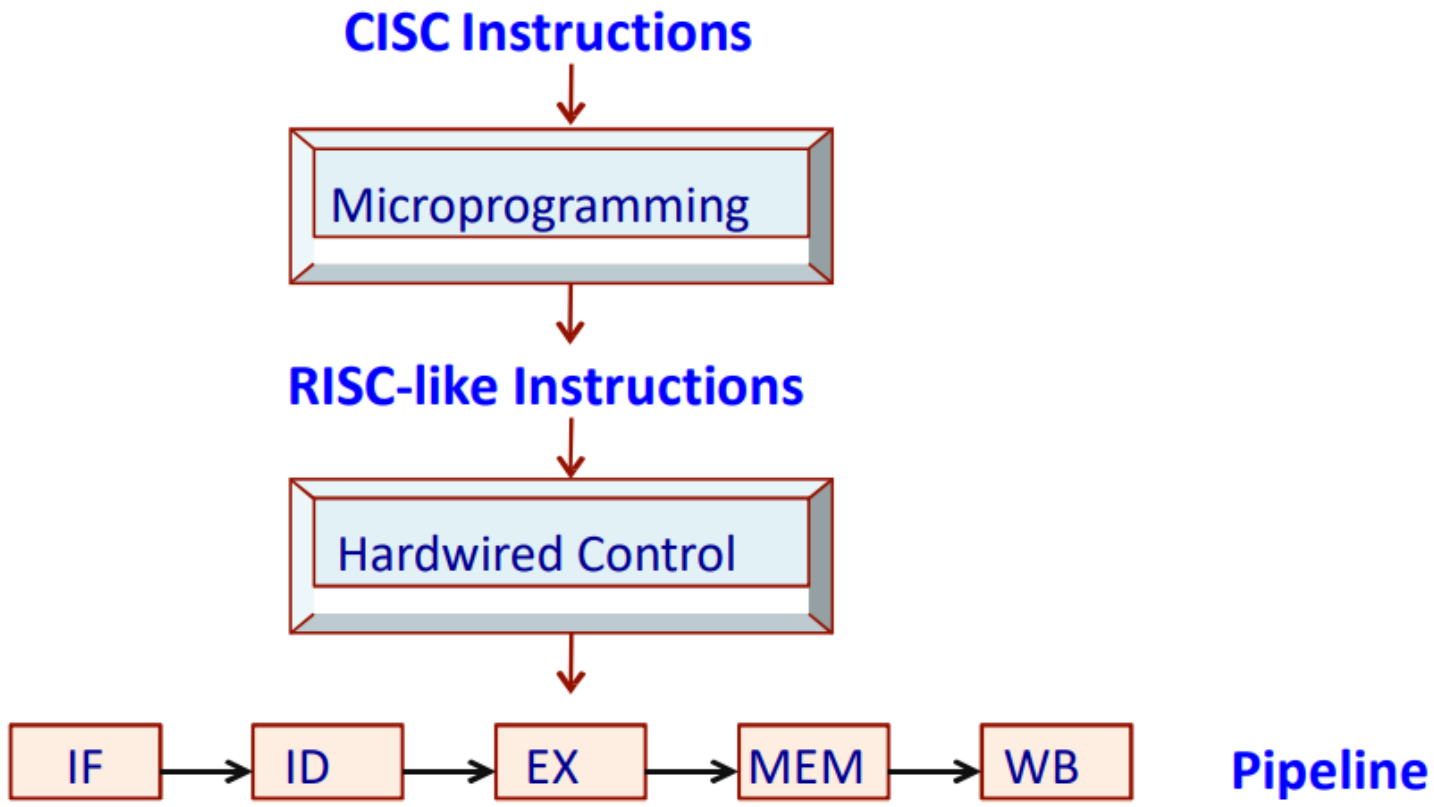
- GPUs contain much **larger number of dedicated ALUs than CPUs.**
  - –Well suited to Single Instruction Stream Multiple Data Stream (SIMD) type of processing.
  - –Can be used to advantage in applications where there are lot of data parallel computations.
- Each processing unit on a GPU contains local memory that **improves data manipulation and reduces data fetch time from memory.**

- Hybrid systems combine CPU and GPU.
  - For programs that have **one or very few threads**, CPUs achieve better performance than GPUs as they have lower operation latencies.
  - For programs having a **large number of threads,** GPUs with higher execution throughput can achieve much higher performance than CPUs.
  - Many applications use both, executing the sequential parts on the CPU, and numerically intensive parts on the GPU.
- Modern GPUs are massively parallel with more than 100 cores, supporting 1000s of threads.


Thread Execution Manager
Parallel Data Cache
Texture
Load / store
Global Memory

- The dilemma:
  - –RISC architecture supposedly execute instructions faster than CISC.
  - –RISC architecture can be efficiently implemented using hardwired control.
  - –CISC architecture may prefer microprogramming because of complex instructions.
- So what is actually done in a CISC processor like x86?
  - –A combination of microprogramming and hardwired control.

- **Intel chips are CISC based**, which use microprogramming to break the complex instructions into simpler sub-operations.
- The sub-operations are very similar to RISC instructions.
- So, at the instruction level the processor can be considered to be using microprogramming.
- At the lower (hardware) level, it may be considered to be using hardwired control to execute the RISC-like instructions in a pipeline.


CISC Instructions
Microprogramming
RISC-like Instructions
Hardwired Control
IF
ID
EX
MEM
WB
Pipeline

TEXT BOOK

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6th Edition 2012.

REFERENCES

1. David A. Patterson and John L. Hennessey, “Computer organization and design”, MorganKauffman ,Elsevier, 5th edition, 2014.
2. William Stallings, “Computer Organization and Architecture designing for Performance”, Pearson Education 8th Edition, 2010
3. John P.Hayes, “Computer Architecture and Organization”, McGraw Hill, 3rd Edition, 2002
4. M. Morris R. Mano “Computer System Architecture” 3rd Edition 2007
5. David A. Patterson “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 5th edition 2011

THANK YOU