FIT9134
Computer architecture and operating systems

MONASH University
Information Technology

Week 11

Hardware II:
CPU, Memory and System Performance

System Performance

- Performance of computer systems has improved continually. In general, the hardware has been evolving faster than the software.

- Since computer system performance is not determined solely by the CPU (*Central Processing Unit*). One needs to consider:
  - what components are involved in determining the overall system performance (eg. CPU, disks, memory, etc)?
  - how can the performance of a computer system be measured (eg. Benchmarks, clock speeds, etc)?

Performance Enhancements

- Computer performance has been greatly enhanced by the following advancements:
  - faster processing within CPU (faster CPU speed)
  - multi-threading (single CPU, multiple-threads)
  - multiple CPU's (single computer, multiple CPUs)
  - distributed systems (multiple co-operating computers)
  - wider, faster data and instruction paths (buses)
  - faster external disk access
  - more (and faster) memory, including advanced cache technology. Better memory managements, such as Virtual Memory.
  - faster, better quality display

CPU Development

- CPU speed is determined by factors such as:

  - clock speed
    - directly affects overall processing speed
    - indirectly depends on other components to keep in sync
  - instruction set architecture (or CPU architecture)
    - RISC versus CISC (see later slides)
  - processing technologies (see later slides)
    - instruction pipelining
    - scalar/superscalar processor

CPU architectures

- *CISC (complex instruction set computers)*
  - eg. IBM mainframes, Intel x86
  - ***characteristics*** : small number of registers, large number of specialised instructions, variable instruction word sizes.

- *RISC (reduced instruction set computers)*
  - eg. PowerPC, Sun SPARC
  - ***characteristics*** : larger number of registers, smaller number of specialised instructions, fixed instruction word size.

CISC vs RISC CPU architectures

- With advancement in computer architecture, CISC and RISC seem to have adopted each other's strategies.
- Some features in RISC are better than CISC and vice-versa, depending what sort of programs the computer runs.
- Researchers try to select best features of each architecture and improve both hardware and OS in order to build faster computer
  - eg. more recent CISC processors use a combination of other technologies, such as pipelining, superscalar and RISC technologies to increase performance.

RISC Principles

Why RISC?

- studies show:
  - only a small number of instructions are very frequently executed
  - programmers & compiler writers generally avoid using complex instructions
  - procedure calls is a bottleneck (eg.arguments need to be passed between the calls); these can be stored in fast registers to improve performance.
- hence we can eliminate rarely used instructions. The resulting reduced instruction set can be more efficiently implemented, and can execute faster (when optimised).
- emphasises more efficient register use (faster) rather than data memory (slower) accesses.

Optimising Most Frequently Executed Instruction Set…

	Instruction	% of executions
BC	Branch condition	20.2
L	Load	15.5
TM	Test under mask	6.1
ST	Store	5.9
LR	Load register	4.7
LA	Load address	5.0
LTR	Test register	3.8
BCR	Branch register	2.9
VC	Move characters	2.1
LH	Load halfword	1.8

Hopkins 1987: IBM/370 instructions ratios : **~5% (ie. 10 out of ~200)** of the instructions accounted for ~70% of total instruction usage!

Procedure Calls

- generally very time consuming
  - depends on number of parameters passed
  - depends on level of nesting
- most programs do not do a lot of calls followed by lots of returns

So the aim is to have as few of these as possible, optimise them, and execute them in the fastest way possible (eg. using fast-speed registers)

RISC Summary

- simpler instruction set
- emphasise on register-based (faster) instructions instead of memory-based (slower) instructions
- fixed-length and-fixed format instructions, so the control unit design is much simpler
- instruction fetch and decode can be performed in parallel
- simplified addressing modes
- large register bank

RISC is powerful. Historically, this is a problem for a company like Intel, which uses a CISC architecture for their main line of CPUs, and needs backward compatibility to maintain market share.

Which architecture is better? Answer is no longer clear-cut with the current CPU designs, so often a mixture of both architectures (RISC & CISC) is used.

Examples of RISC processors

- Currently, almost anything except the Intel x86 uses the RISC architecture

- Sun SPARC line (desktop and server)
- ARM processor (embedded/small devices)
- Alpha (high performance servers until 2003)
- MIPS (embedded/small devices)
- PowerPC (some Apple/IBM/Motorola)
- many Gaming consoles make use of these processors
- …

Improving CPU Performance

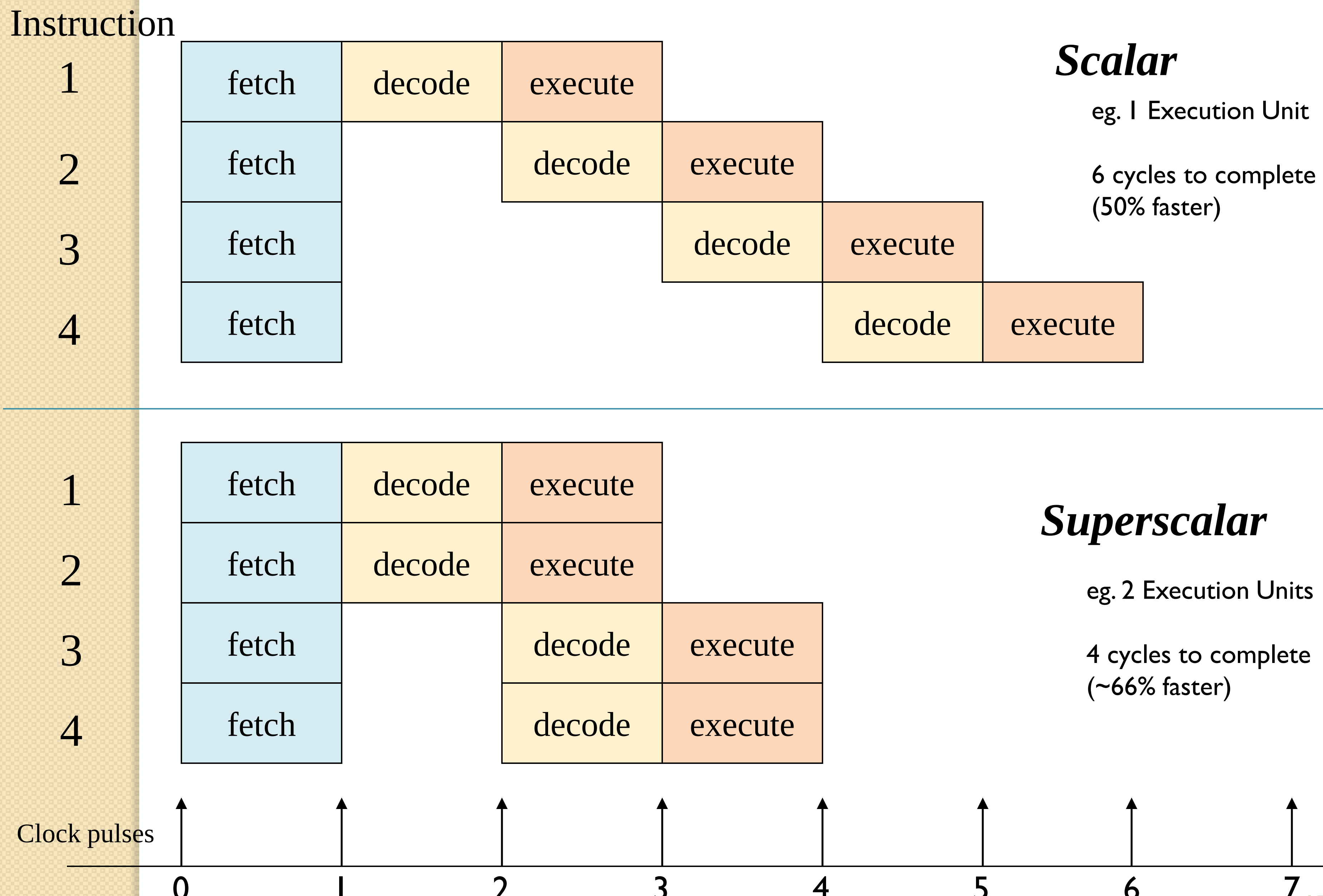
- Techniques to improve CPU performance:
  - Use of separate Fetch and Execute instruction units, allowing ***concurrent*** fetch-execute operations
  - Use of ***pipelining*** to overlap the fetch-execute cycles, allowing one instruction to start before another is finished (analogy : food preparation, factory processing, etc)
  - Use of multiple Execute units with a specialised Fetch unit (which can retrieve multiple instructions simultaneously), allowing ***parallel executions***

Improving CPU Performance

- Pipelining overlaps the stages of the fetch-execute cycle, but completes execution of one instruction at a time (due to only one “execution unit”). This is *scalar* processing.

- Most modern processors also implement *superscalar* processing
  - this uses multiple “execution units” (analogy : adding more lanes on roads to allow more traffic) to complete the execution of several instructions at the same time (“in parallel”).

Scalar vs. Superscalar Processing, with pipelining



Using Cache memory

- A '*small' amount of fast memory between the CPU and primary* memory.

- *All memory* references are checked against the cache, and the cache location used if the memory reference occurs in one of the cache blocks.

- For a 64Kbyte cache, *hit ratios of 90% or more can be achieved, with* execution speed improvements of 50% or more.

Using Cache memory

- Reading is simple – just retrieve from cache

- Writing is more difficult, because writes must be made to primary memory to preserve the integrity of the contents of primary memory.
  - generally two strategies:
    1) *Write Through – write all changes directly to primary memory*
    2) *Write Back – only write changed data when it is to be evicted from cache*

Using other fast memory

- ***Static RAM***** - memory that does not require periodic refreshing to maintain its contents (unlike the ***Dynamic RAM*** commonly used in main memory)
  - faster, consumes less power, and more expensive than Dynamic RAM – so tends to be used in cache/registers

- ***SDRAM*** – synchronous dynamic RAM (designed for better implementation of pipelining, leading to faster overall performance)
  - eg. DDR2/3/4 SDRAM's commonly found in modern PCs

** *RAM ➔ Random Access Memory*

Using Flash Memory

- Flash Memory
  - non-volatile (does not need power to retain contents)
  - limited lifespan (degrades after a certain number of write cycles –typically 3000-5000)
  - can be used as a cheap (but slow) cache,

    eg. ***Windows Readyboost*** technology
  - can be used as portable boot devices

Other components affecting overall system performances

- Disks
  - increased speeds, use of caches, efficient scheduling, etc
- Bus
  - increased width and speed of paths for data and instructions, faster speeds, etc
- Display
  - faster graphics cards, more onboard memory, better panel technologies, etc

Performance Measurement

- There are three common measures used to describe computer system performance:
  - *CPU Clock Speed* – inaccurate, due to pipelining and superscalar processing (5 mhz to 3.5 Ghz over 20+years), and more recently multiple CPUs.
  - *MIPS* (million instructions per second) - more accurate, but accuracy affected by RISC versus CISC.
  - *Benchmarking* programs (CPU, Disk and Graphics) - most accurate for a particular task, but are task dependent.

See www.spec.org for some published benchmarks

Next Week

- **Lecture** :
  - a brief review/summary of all topics covered will be presented.
  - information on ***Final Exam format & tips*** will be provided.