Computer Systems

Kasun Gunawardana

E-mail: kgg

University of Colombo School of Computing

The CPU and Its Organization

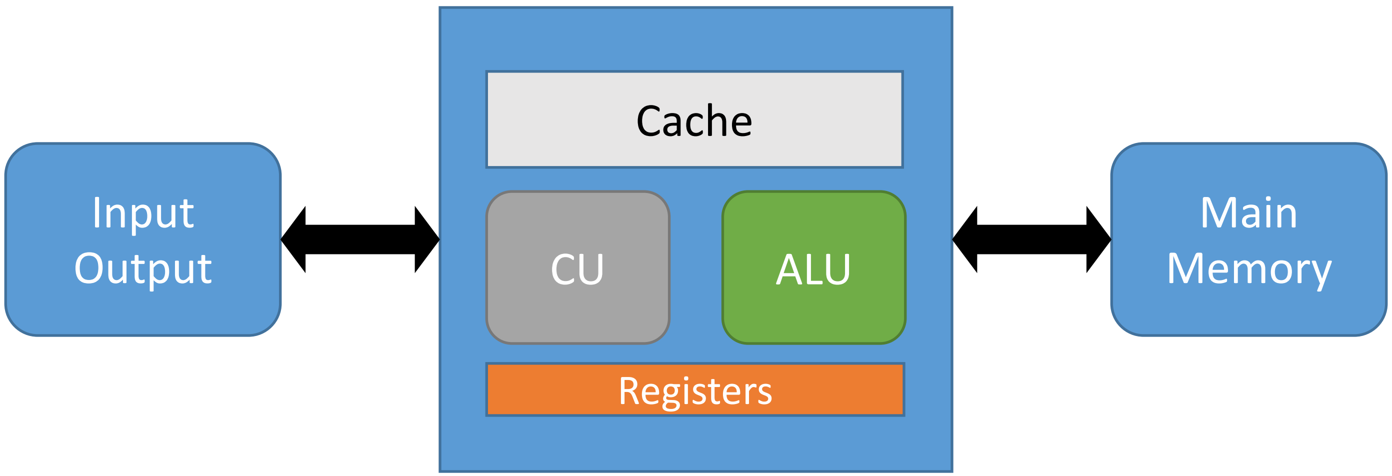
Central Processing Unit

- CPU is the brain of a Computer
  - Execute instructions and perform arithmetic and logic operations
- Key Components
  - Arithmetic Logic Unit (ALU)
  - Control Unit (CU)
  - Registers
  - Cache
  - Input/Output (I/O) interfaces

The organization of a CPU

- The organization of a CPU can be divided into two main conceptual parts:
  - Control Unit (CU and registers)
    - Manages the flow of data and instructions
  - Execution Unit (ALU and cache)
    - Performs operations on data

Hardware components of a CPU



Program Execution - Typical Procedure

- The CPU fetches an instruction from main memory
- The CPU decodes the instruction
- The CPU executes the instruction
- The CPU stores the result of the instruction in a register or main memory.
- The CPU then fetches the next instruction from main memory and repeats the process of decoding, executing, and storing the result.

Control Unit

- Fetch instructions from main memory,
- Decode instructions,
- Determining the appropriate operation to be performed by the ALU.
- Controls the flow of data within the CPU

Arithmetic and Logic Unit

- Perform mathematical and logical operations
- Actual calculations and logical operations specified in the instructions
- Arithmetic and logic operations are performed on binary numbers

Main Memory

- A series of binary storage cells
- The smallest unit of memory is called a bit
- Binary information is stored in groups of bits
- Each group is called a word
- Word is a fixed size of data (a group of bits)
- Each word has a unique address to identify it

Main Memory

Data
Or
Instructions

7	6	5	4	3	2	1	0	Address
								0
								1
								2
								3
								4
								5
								6
								7
								....
								M

8 bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address
																0
																1
																2
																3
																4
																5
																6
																7
																....
																N

16 bits

Different Attributes of Memory

- Address Length and Address Range
- Word Size
  - Word Addressable
  - Byte Addressable

Example: Address Length

- The address length of a memory system is 10 bits and it is a byte-addressable memory.
  - *What is the maximum number of addresses this system can have?*
  - *What is the maximum size of memory this system can have?*
  - *What is the address range?*

Example: Address Length

- The address length of a memory system is 10 bits and it is a byte-addressable memory.
  - *What is the maximum number of addresses this system can have?*
    - $\mathbf{2^{10} = 1024}$
  - *What is the maximum size of memory this system can have?*
    - $\mathbf{2^{10} \times 1 byte = 1024\ bytes}$
  - *What is the address range?* $\mathbf{0\ to\ 2^{10} - 1}$

Example: Address Length

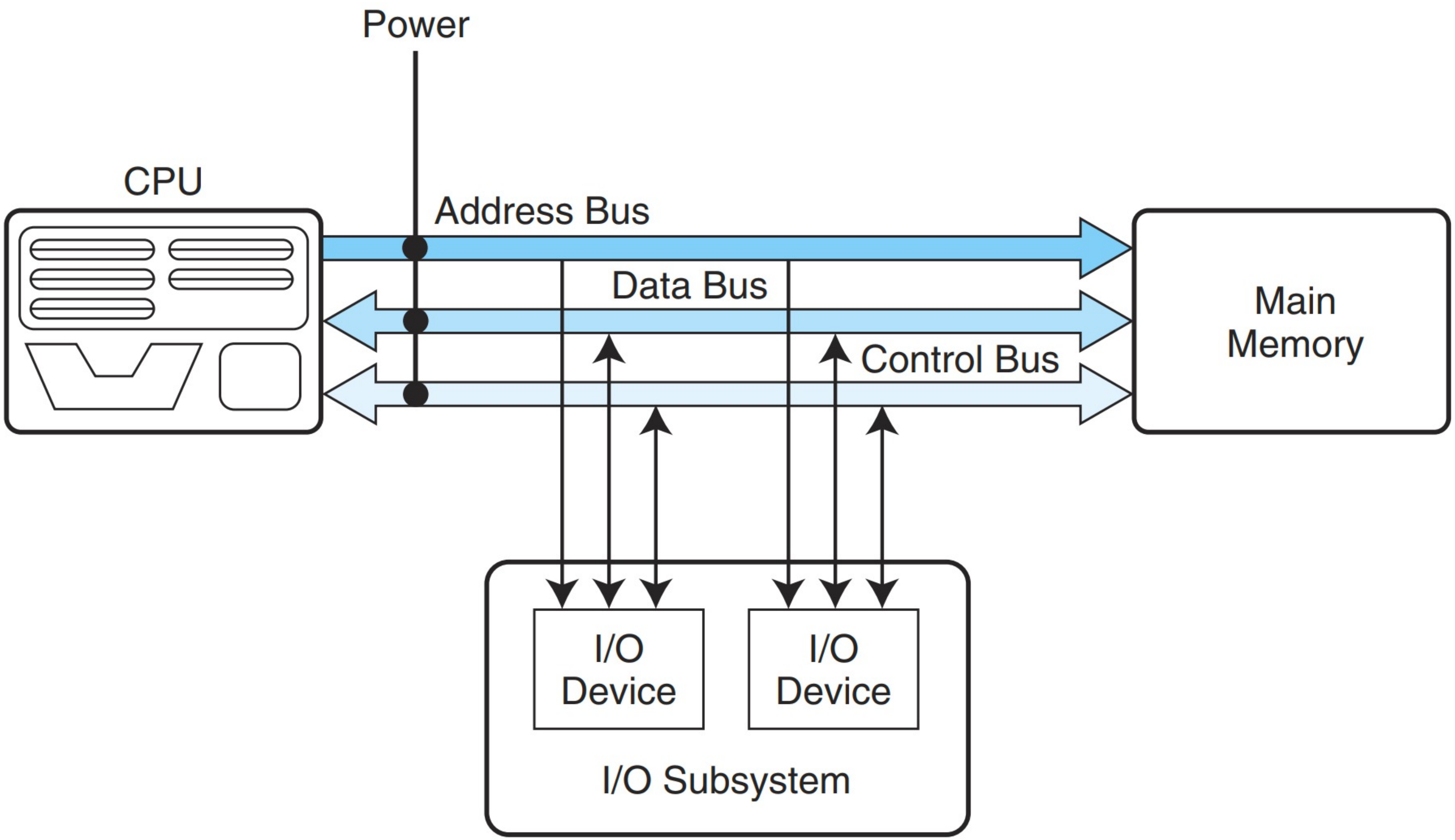
- The address length of a memory system is 10 bits and it is a word-addressable memory with word size = 32 bits.
  - *What is the maximum number of addresses this system can have?*
  - *What is the maximum size of memory this system can have?*
  - *What is the address range?*

Example: Address Length

- The address length of a memory system is 10 bits and it is a word-addressable memory with word size = 32 bits.
  - *What is the maximum number of addresses this system can have?*
    - $\mathbf{2^{10} = 1024}$
  - *What is the maximum size of memory this system can have?*
    - $\mathbf{2^{10} \times 4\ \boldsymbol{bytes} = 4096\ \boldsymbol{bytes}}$
  - *What is the address range?* $\mathbf{0\ \boldsymbol{to}\ 2^{10} - 1}$

The Bus

- A bus is a set of wires that connects multiple subsystems to form a shared data path.
  - Parallel movement of data
- At anytime, only one device may use the bus
  - Sharing creates a bottleneck
- The bus speed is affected by,
  - Width of the bus
  - Number of devices sharing it


Power
CPU
Address Bus
Data Bus
Control Bus
Main Memory
I/O Device
I/O Device
I/O Subsystem

Address Bus

- The address bus is a set of lines that the CPU uses to send the memory address of the data it wants to read or write.
- The memory controller uses this address to locate the correct memory location where the data is stored.

Data Bus

- CPU uses to send and receive data.
  - When the CPU wants to read data from memory, it sends the memory address on the address bus and the memory controller responds by sending the data on the data bus.
  - When the CPU wants to write data to memory, it sends the data on the data bus and the memory address on the address bus.

Control Bus

- CPU uses to send control signals to the memory controller and other components in the system.
- These signals can include memory read or write commands, interrupt requests, and other types of control signals.
- The control bus also enables the CPU to control the flow of data on the data and address buses.

Memory Controller

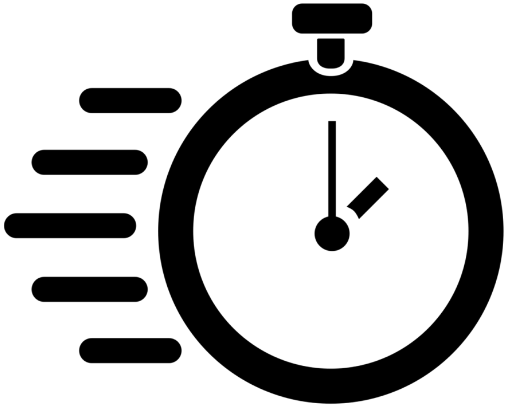
- Manages the flow of data going to and from the computer's main memory.
- Responsible for
  - coordinating communication between the CPU and the memory modules
  - managing tasks such as error checking and correction
- Typically integrated into the CPU or the chipset of a computer's motherboard.

Clock

- Clock generates pulses (ticks)
- A pulse sets the pace for everything happens in the computer
- Therefore, it determines how quickly instructions are executed
- Fixed number of clock pulses are required to complete each instruction
- Clock frequency is measured in hertz (Hz)

Clock Speed

- Clock Speed (Clock Rate) – The number of pulses per second
- 1 Million pulses per second = 1MHz



Measuring Time

- Duration
  - $1\ second$
  - ${}^{1}/{}_{1000}\ second = 1\ millisec = 1\ ms$
  - ${}^{1}/{}_{1,000,000}\ second = 1\ microsec = 1\ \mu s$
  - ${}^{1}/{}_{1,000,000,000}\ second = 1\ nanosec = 1\ ns$
- Frequency
  - $1\ cycle\ per\ second\ =\ 1\ Hertz\ =\ 1\ Hz$
  - $1,000,000\ cycles\ per\ second\ =\ 1\ MHz$
  - $1,000,000,000\ cycles\ per\ second\ =\ 1\ GHz$

Clock Cycle Time

- Clock cycle time is the period of the clock that synchronizes the circuits in a processor.
- It is the reciprocal of the clock frequency

$$Clock\ Cycle\ Time = \frac{1}{Clock\ Speed}$$

- What is the clock cycle time of a CPU with 1MHz clock speed?

Clock Rate Vs. Clock Cycle

- The clock rate is the reciprocal of clock cycle time
  - Frequency = How many cycles per second
- Clock cycle time $20ns$
  - The clock rate is ${}^{1}/_{20\times10^{-9}}\,Hz = 0.5\times10^{8}Hz = 500MHz$
- Clock rate of $2.5GHz$
  - Clock cycle time is ${}^{1}/_{2.5\times10^{9}}\,s = 0.4\,ns$

Example

- What is the clock cycle time of a 750 MHz CPU ?

$$Clock\ Cycle\ Time = \frac{1}{750{,}000{,}000} s$$

CPU Execution Time

- CPU execution time of a program (T)

$$T = No.of\ Clock\ Cycles \times Clock\ Cycle\ Time$$

$$T = No.of\ Clock\ Cycles \times \frac{1}{Clock\ Rate}$$

CPU Execution Time (Cont.)

- Does execution time depend on the number of instructions ?
  - Different Instructions take different number of cycles
- Therefore,

$$CPU\ Clock\ Cycles = N \times CPI$$

- N – Number of Instructions
- CPI – Average clock cycles per instruction

CPI – Cycles Per Instruction

- CPI - the average number of cycles per instruction
- For each instruction type, if we know its **frequency** ($\boldsymbol{F_i}$) and the **number of cycles** need to execute it ($\boldsymbol{CPI_i}$), we can compute the **overall CPI** as follows:

$$Overall\ CPI = \sum_i CPI_i \times F_i$$

CPI – Cycles Per Instruction (Cont.)

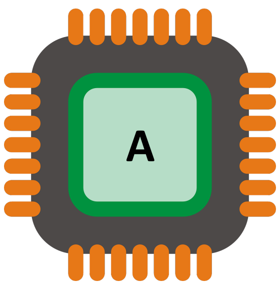
- A benchmark program has 100 instructions.
  - 25 instructions are **`load/store`** operations, and each takes 2 cycles
  - 50 instructions are **add** operations and each takes 1 cycle
  - 25 instructions are **`square root`** operations, and each takes 50 cycles
- What is the CPI of this program ?

$$CPI = (0.25\times 2) + (0.5\times 1) + (0.25\times 50)$$

$$CPI = 13.5$$

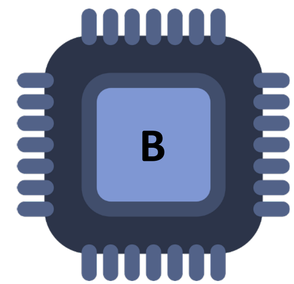
Example

- We have **two machines** with different implementations of the same ISA. Machine ***A*** has a **clock cycle time of 10ns** and a **CPI of 2.0** for program P, while machine ***B*** has a **clock cycle time of 20ns** and a **CPI of 1.2** for the same program.



Machine A
- Clock Cycle time 10ns
- CPI 2.0

Machine B
Clock Cycle time 20ns -
CPI 1.2 -



Which machine is faster?

Example

- Total number of cylces for $A \; = 2.0 \times N$
- Total number of cycles for $B \; = 1.2 \times N$

- CPU Time for $A \; = 2N \times 10ns = 20N\ ns$
- CPU Time for $B \; = 1.2N \times 20ns = 24N\ ns$

- Machine A is fater…!

Equations

$$CPU\ Time = N \times CPI \times t$$

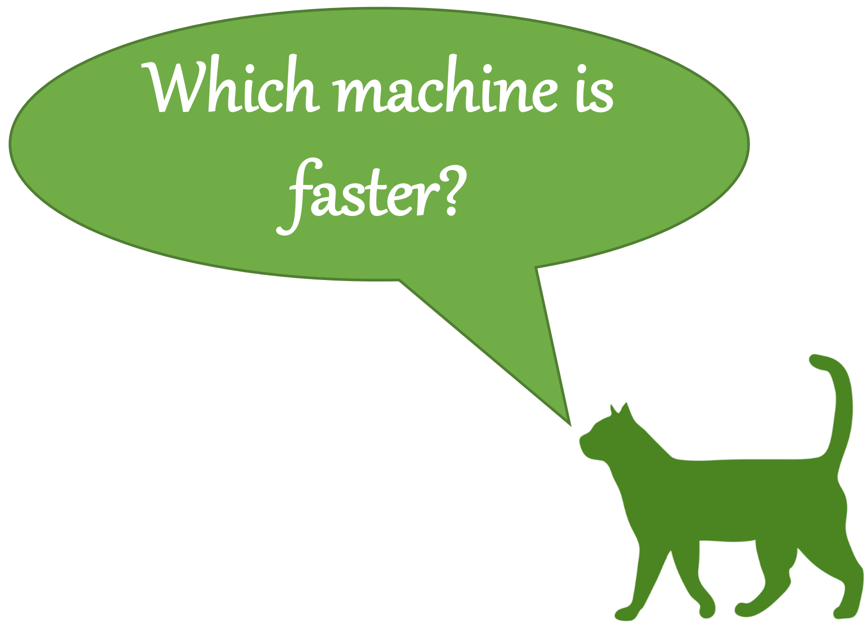
$$CPU\ Time = N \times CPI \times \frac{1}{r}$$

- N = number of instructions in program
- CPI = average cycles per instruction
- t = clock cycle time
- r = clock rate

Homework

For a given program,

- Machine A
  - Clock cycle time 10ns
  - CPI 2.0
- Machine B
  - Clock cycle time 30ns
  - CPI 0.5



Instruction Count

```
250 instructions
for i = 1, 100 do
20 instructions
    for j = 1, 100 do
    40 instructions
        for k = 1, 100 do
        10 instructions
        endfor
    endfor
endfor
```

- Number of instructions to be executed by the CPU is not the static count of instructions in the program

Instruction Count (Cont.)

```
250 instructions
for i = 1, 100 do
20 instructions
    for j = 1, 100 do
    40 instructions
        for k = 1, 100 do
        10 instructions
        endfor
    endfor
endfor
```

- Number of instructions to be executed by the CPU is not the static count of instructions in the program
- *Let's try to find the dynamic instructions count here.*
- *Assume each for counts for two instructions, one for index increment and another to check exit condition*

Instruction Count (Cont.)

250 instructions
for i = 1, 100 **do**
20 instructions
    for j = 1, 100 **do**
    40 instructions
        for k = 1, 100 **do**
        10 instructions
        endfor
    endfor
endfor

$$(2 + 10)\ instructions$$
$$in\ 100\ iterations = 1{,}200$$

Instruction Count (Cont.)

250 instructions
for i = 1, 100 **do**
20 instructions
  for j = 1, 100 **do**
  40 instructions
    for k = 1, 100 **do**
    10 instructions
    endfor
  endfor
endfor

$(2+10)\ instructions$
$in\ 100\ iterations = 1{,}200$

$(2+40+1200)\ instructions$
$in\ 100\ iterations = 124{,}200$

Instruction Count (Cont.)

250 instructions
for i = 1, 100 **do**
20 instructions
    for j = 1, 100 **do**
    40 instructions
        for k = 1, 100 **do**
        10 instructions
        endfor
    endfor
endfor

$(10 + 2)\ instructions$
$in\ 100\ iterations = 1{,}200$

$(2 + 40 + 1200)\ instructions$
$in\ 100\ iterations = 124{,}200$

$(2 + 20 + 124{,}200)\ instructions$
$in\ 100\ iterations = 12{,}422{,}200$

Instruction Count (Cont.)

```
250 instructions
for i = 1, 100 do
20 instructions
    for j = 1, 100 do
    40 instructions
        for k = 1, 100 do
        10 instructions
        endfor
    endfor
endfor
```

$(10 + 2)\ instructions$
$in\ 100\ iterations = 1{,}200$

$(2 + 40 + 1200)\ instructions$
$in\ 100\ iterations = 124{,}200$

$(2 + 20 + 124{,}200)\ instructions$
$in\ 100\ iterations = 12{,}422{,}200$

$(250 + 12{,}422{,}200)\ instructions$
$in\ 100\ iterations = 12{,}422{,}450$

Instruction Count (Cont.)

```
250 instructions
for i = 1, 100 do
20 instructions
    for j = 1, 100 do
    40 instructions
        for k = 1, 100 do
        10 instructions
        endfor
    endfor
endfor
```

$(10 + 2)\ instructions$
$in\ 100\ iterations = 1{,}200$

$(2 + 40 + 1200)\ instructions$
$in\ 100\ iterations = 124{,}200$

$(2 + 20 + 124{,}200)\ instructions$
$in\ 100\ iterations = 12{,}422{,}200$

$(250 + 12{,}422{,}200)\ instructions$
$in\ 100\ iterations = 12{,}422{,}450$

$Static\ instruction\ count\ is = 326$

Impact on CPU Throughput

- CPU Throughput Impact factors
  - The number of instructions in a program
  - The number of cycles per instruction
  - The clock cycle time (duration of a single cycle)
- Common Limitations
  - Multiplications takes longer than addition
  - Floating point operations > Integer operations
  - Accessing main memory > Accessing registers

Activity

- A compiler writer must choose between two code sequences for a certain high level language statement. Instruction counts for the two sequences are as follows:

Sequence	Instruction count per type		
	A	**B**	**C**
1	2	1	2
2	4	1	1

CPIs given in the Hardware Spec	
Instruction Type	**CPI per instruction type**
A	1
B	2
C	3

- Which sequence executes more instructions?
- Which is faster?
- Which has lower CPI?

Activity

- Let's calculate it,

	Sequence 1	Sequence 2
Instruction Count	2+1+2 = 5	4+1+1 = 6
Total Cycles	$(2\times1) + (1\times2) + (2\times3) = 10$	$(4\times1) + (1\times2) + (1\times3) = 9$

- Since total number of cycles for Sequence 2 is lower than Sequence 1, the faster sequence will be the Sequence 2.
- Sequence 2 has the lower CPI
  - $CPI_1 = {}^{10}/_{5} = 2$ $\qquad$ $CPI_2 = {}^{9}/_{6} = 1.5$

Thank You..!