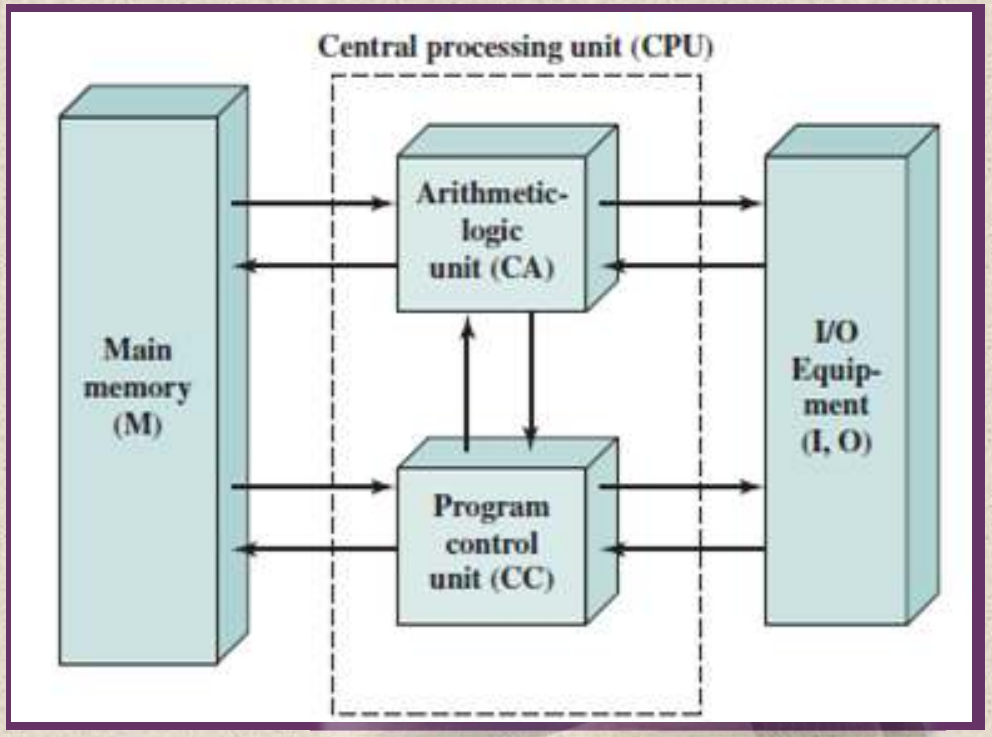
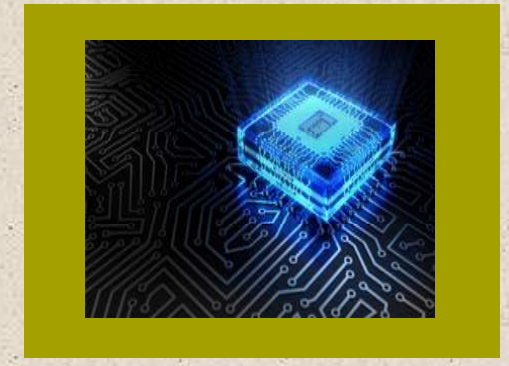
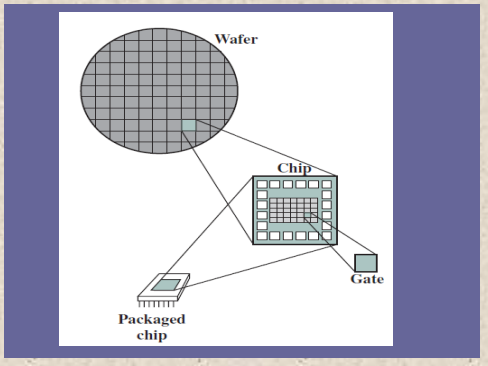
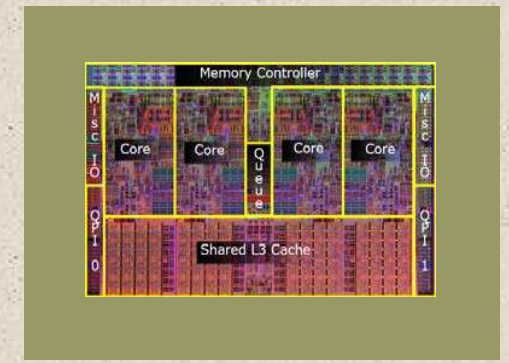
CH02-COA10e Performance

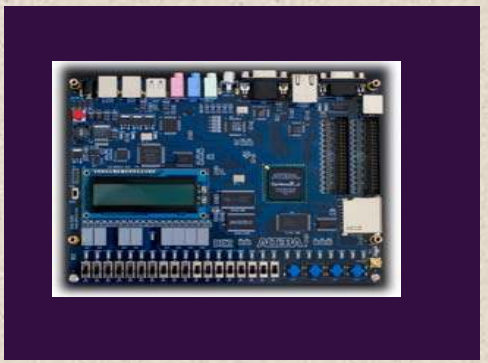
Kiến trúc máy tính _ hợp ngữ (Trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh)





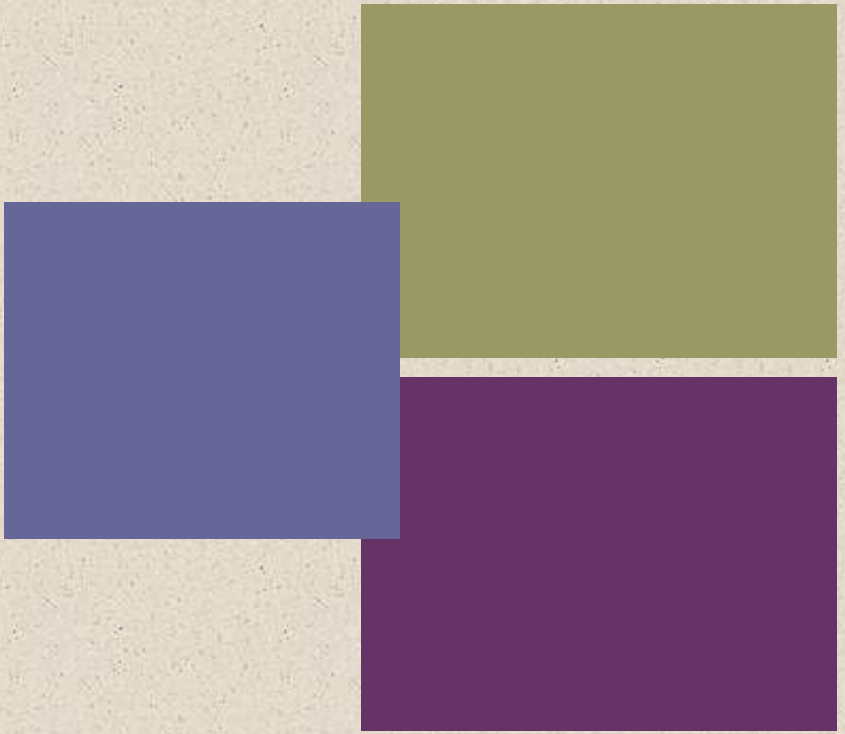






COMPUTER ORGANIZATION & ARCHITECTURE

Van-Khoa Pham (PhD.)

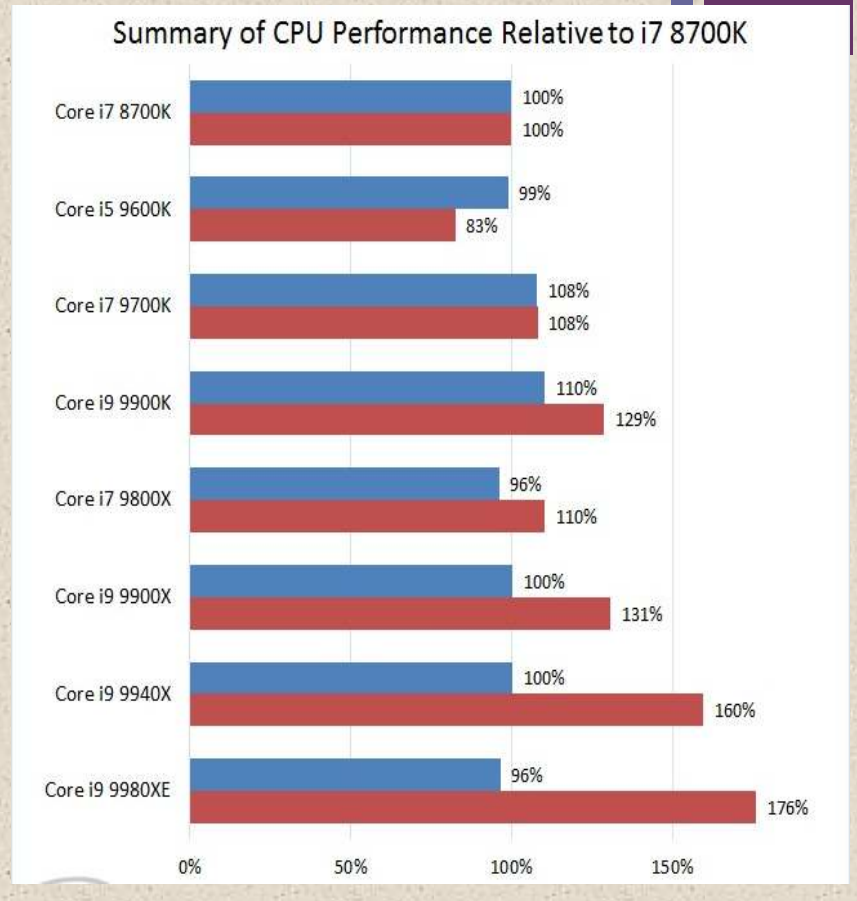


Chapter 2

Performance Issues

Performance

- Trying to choose among different computers, performance is an important attribute.
- Accurately measuring and comparing different computers is critical to purchasers and therefore to designers.



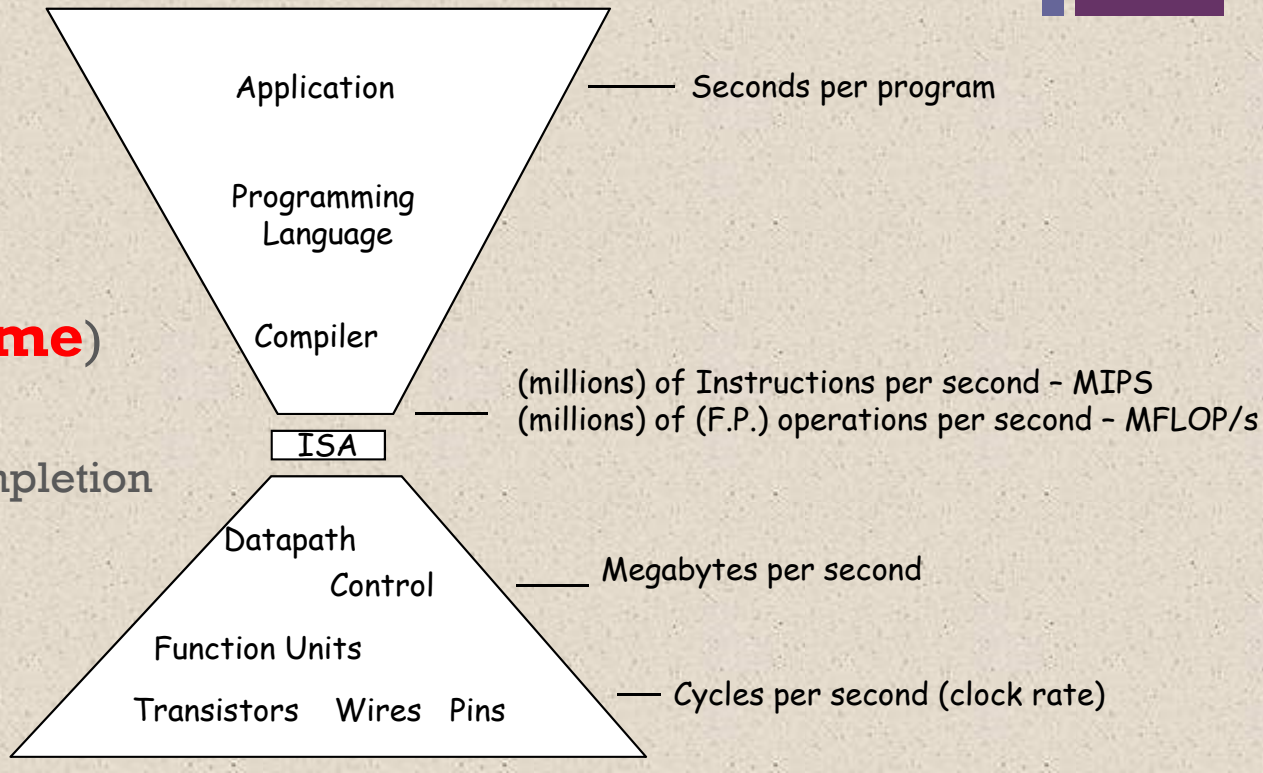
Performance Metrics

- Purchasing perspective: given a collection of machines, which has the
  - best performance ?
  - best cost/performance?
- Design perspective: faced with design options, which has the
  - best performance improvement ?
  - best cost/performance?

understand what factors in the architecture contribute to overall system performance

Metrics of Performance

- Response time (**execution time**)
  - How long it takes to do a task
  - The time between the start and completion of a task



Relative Performance

Performance = 1/Execution Time

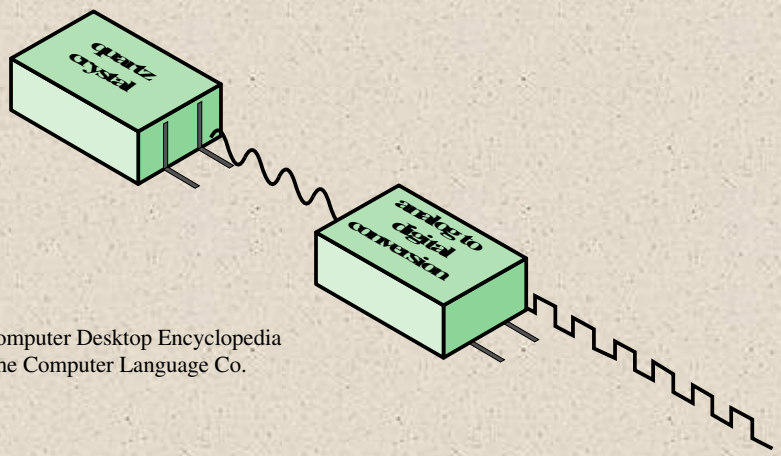
- "A is *n* time faster than B"

$$\text{Performance}_A / \text{Performance}_B = \text{Execution time}_B / \text{Execution time}_A = n$$

time taken to run a program

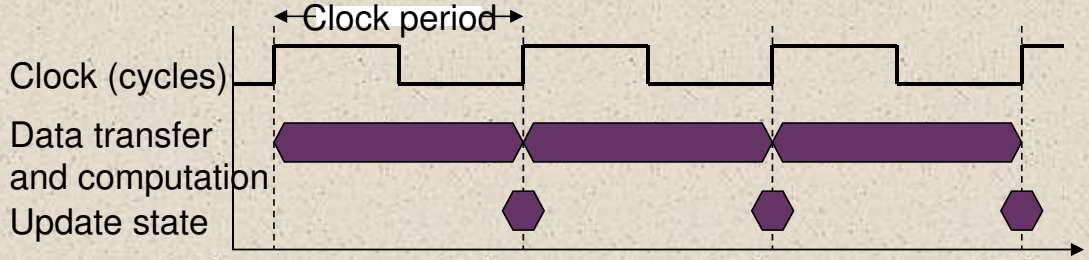
- 10s on A, 15s on B
- Execution Time_B / Execution Time_A = 15s / 10s = 1.5
- So A is 1.5 times faster than B

CPU Clocking



From Computer Desktop Encyclopedia
1998, The Computer Language Co.

Figure 2.5 System Clock



Clock rate is inverse of clock cycle

$$CC = 1 / CR$$

- Clock period (cycle time): duration of a clock cycle
  - e.g., 250ps = 250×10^{-12}s
- Clock frequency (rate): cycles per second
  - e.g., 4.0GHz = 4.0×10^{9}Hz

10 nsec clock cycle => 100 MHz clock rate

200 psec clock cycle => 5 GHz clock rate

CPU Time

$$\text{CPU Time} = \#\text{CPU Clock Cycles} \times \text{Clock Period}$$
$$= \frac{\#\text{ CPU Clock Cycles for a program}}{\text{Clock Rate}}$$

Performance improved by

- Reducing number of clock cycles
- Increasing clock rate

Assume a program (e.g., sorting) is completed in 250 Clock Cycles. What is the total CPU Time used?

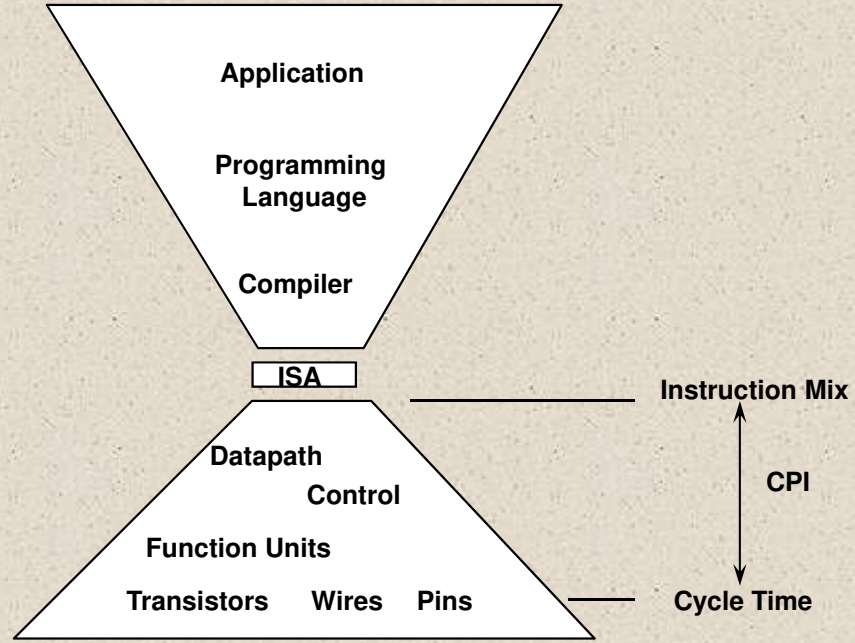
CPU Time Example

- A sorting program runs in ***10 seconds on computer A***, having **4 GHz clock**.
- We are trying to build a ***computer B, that will run this program in 6 seconds***.
- If on Computer B, ***require 1.2 times*** as many clock cycles as computer A for this program.

What clock rate should we tell the designer to target?

- CPU time$_A$ = CPU (clock cycles)$_A$ / (Clock rate)$_A$
- 10 seconds = CPU (clock cycles)$_A$ / 4 x 10^9 cycles/second
- CPU (clock cycles)$_A$ = 10 seconds x 4 x 10^9 cycles/sec = 40 x 10^9 cycles
- CPU time$_B$ = 1.2 x CPU clock cycles$_A$ / (Clock rate)$_B$
- 6 seconds = 1.2 x 40 x 10^9 (clock cycles)$_A$ / (Clock rate)$_B$
- (Clock rate)$_B$ = 1.2 x 40 x 10^9 cycles / 6 seconds = 8 GHz

Metrics of Performance



CPI is a useful design measure relating the **Instruction Set Architecture** with the Implementation of that architecture, and the program measured

I_c : Instruction Count

p : The number of processor cycles needed to decode and execute the instruction

m: The number of memory references needed

k: the ratio between memory cycle time and processor cycle time

τ : cycle time (1/f)

CPI: Clock cycle per instruction

T: The time needed to execute a given programe

$$CPI = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c} \qquad T = I_c \times CPI \times \tau$$

Clock Cycles Per Instruction: CPI

- Multiplication takes more time than addition
- Floating point operations take longer than integer ones
- Accessing memory takes more time than accessing registers

Instruction Type	***CPI***
Arithmetic and logic	1
Load/store with cache hit	2
Branch	4
Memory reference with cache miss	8

average CPI

- If different **instruction classes** take different numbers of cycles (assuming n classes)

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

- average CPI

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\text{Relative frequency}} \right)$$

Clock Cycles Per Instruction: CPI

CPU clock cycles for a program = # Instructions for a program x **Average** clock cycles per instruction

A benchmark program is run on a 40 MHz processor. The executed program consists of 100,000 instruction executions, with the following instruction mix and clock cycle count:

Instruction Type	Instruction Count	Cycles per Instruction
Integer arithmetic	45,000	1
Data transfer	32,000	2
Floating point	15,000	2
Control transfer	8000	2

Average clock cycles per instruction= 1.55

CPI Example

- Alternative compiled code sequences using instructions in classes A, B, C

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

Instruction Count

- Sequence 1: IC = 5
  - Clock Cycles
    = 2×1 + 1×2 + 2×3
    = 10
  - Avg. CPI = 10/5 = 2.0
- Sequence 2: IC = 6
  - Clock Cycles
    = 4×1 + 1×2 + 1×3
    = 9 (Faster)
  - Avg. CPI = 9/6 = 1.5

I_c : Instruction Count
p : The number of processor cycles needed to decode and execute the instruction
m: The number of memory references needed
k: the ratio between memory cycle time and processor cycle time
τ : cycle time (1/f)
CPI: Clock cycle per instruction
T: The time needed to execute a given programe

	I_c	p	m	k	τ
Instruction set architecture	X	X			
Compiler technology	X	X	X		
Processor implementation		X			X
Cache and memory hierarchy				X	X

Table 2.1 Performance Factors and System Attributes

Performance Equation

$$\text{CPU Time} = \#\text{CPU Clock Cycles} \times \text{Clock Period}$$

$$= \frac{\#\text{ CPU Clock Cycles for a program}}{\text{Clock Rate}}$$

CPU time = Instruction_count x CPI x clock_cycle

or

$$\text{CPU time} = \frac{\text{Instruction\_count} \times \text{CPI}}{\text{clock\_rate}}$$

Determinates of CPU Performance

CPU time = Instruction_count x CPI x clock_cycle

	Instruction_ count	CPI	clock_cycle
Algorithm	X	X	
Programming language	X	X	
Compiler	X	X	
ISA	X	X	X
Processor organization		X	X
Technology			X

CPI and Clock Rate

- Computer A: Cycle Time = 250ps (4 GHz clock rate), CPI = 2.0
- Computer B: Cycle Time = 500ps (2 GHz clock rate), CPI = 1.2
- Same ISA
- Which is faster, and by how much?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster…

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

…by this much

A Simple Example

Op	Freq	CPI_i	Freq x CPI_i		
ALU	50%	1	.5	.5	.5
Load	20%	5	1.0	.4	1.0
Store	10%	3	.3	.3	.3
Branch	20%	2	.4	.4	.2
			Σ = 2.2	1.6	2.0

- How much faster would the machine be if a better data cache reduced the average load time to 2 cycles?

  Relative performance is 2.2/1.6 means 37.5% faster

- How does this compare with using branch prediction to shave a cycle off the branch time?

  Relative performance is 2.2/2.0 means 10% faster

MIPS as a Performance Metric

MIPS: Millions of Instructions Per Second

$$\text{IPS} = \frac{\text{Clock rate}}{\text{CPI}} \Rightarrow \textit{Million}\text{IPS} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6} \quad (1)$$

$$\text{Execution time} = \frac{\text{Instruction count} * \text{CPI}}{\text{Clock rate}} \quad (2)$$

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} \quad (3)$$

MIPS is an instruction execution rate, MIPS specifies performance inversely to execution time

MIPS as a Performance Metric: Example

Consider the following performance measurement for a program:

Measurement	Computer A	Computer B
Instruction Count	10 billion	8 billion
Clock Rate	4 GHz	4 GHz
CPI	1.0	1.1

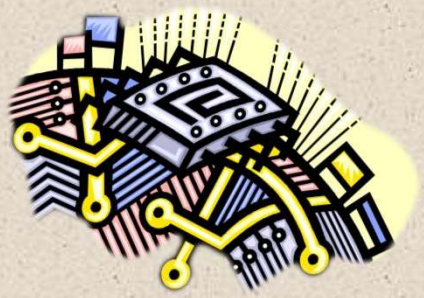
1. **Which computer has the higher MIPS rating?**
2. **Which computer is faster?**

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

1. **Computer A has the higher MIPS rating.**
2. **Computer B is faster**

Improvements in Chip Organization and Architecture

- Increase hardware speed of processor
  - Fundamentally due to shrinking logic gate size
    - More gates, packed more tightly, increasing clock rate
    - Propagation time for signals reduced
- Increase size and speed of caches
  - Dedicating part of processor chip
    - Cache access times drop significantly
- Change processor organization and architecture
  - Increase effective speed of instruction execution
  - Parallelism



Problems with Clock Speed and Login Density

- Power
  - Power density increases with density of logic and clock speed
  - Dissipating heat
- RC delay
  - Speed at which electrons flow limited by resistance and capacitance of metal wires connecting them
  - Delay increases as the RC product increases
  - As components on the chip decrease in size, the wire interconnects become thinner, increasing resistance
  - Also, the wires are closer together, increasing capacitance
- Memory latency
  - Memory speeds lag processor speeds (memory hierarchy)

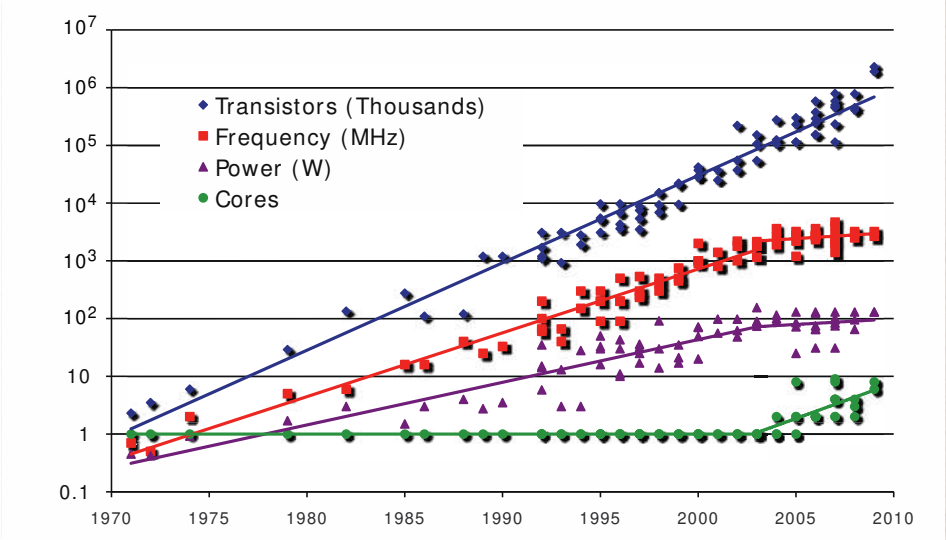


Figure 2.2 Processor Trends

Performance Balance

- Adjust the organization and architecture to compensate for the mismatch among the capabilities of the various components
- Architectural examples include: