Performance

What is the Performance?

Plane	DC to Paris	Speed	Passengers	passengers X mph
Boeing 747	6.5 hours	610 mph	470	286,700
Concorde	3 hours	1350 mph	132	178,200

Which of the planes has better performance

- The plane with the highest speed is **Concorde**
- The plane with the largest capacity is **Boeing 747**

Performance Example

- Time of Concorde vs. Boeing 747?
  - Concord is 1350 mph / 610 mph = 2.2 times faster

- Throughput of Concorde vs. Boeing 747 ?
  - Boeing is 286,700 pmph / 178,200 pmph = 1.6 times faster

- Boeing is 1.6 times faster in terms of throughput
- Concord is 2.2 times faster in terms of flying time

- When discussing processor performance, we will focus primarily on execution time for a single job - why?

Definitions of Time

- Time can be defined in different ways, depending on what we are measuring:
  - Response time : The time between the start and completion of a task. It includes time spent executing on the CPU, accessing disk and memory, waiting for I/O and other processes, and operating system overhead. This is also referred to as execution time.
  - Throughput :The total amount of work done in a given time.
  - CPU execution time : Total time a CPU spends computing on a given task (excludes time for I/O or running other programs). This is also referred to as simply CPU time.

Performance Definition

- For some program running on machine X,

  Performance = 1 / Execution time$_X$
- "X is n times faster than Y"

  Performance$_X$ / Performance$_Y$ = n

Problem:

- machine A runs a program in 20 seconds
- machine B runs the same program in 25 seconds
- how many times faster is machine A?

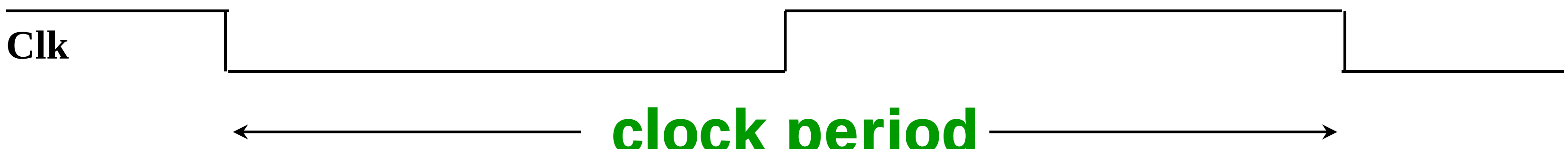
$$\frac{25}{20} = 1.25$$

Basic Measurement Metrics

- Comparing Machines
  - Metrics
    - Execution time
    - Throughput
    - CPU time

Computer Clock

- A computer clock runs at a constant rate and determines when events take placed in hardware.

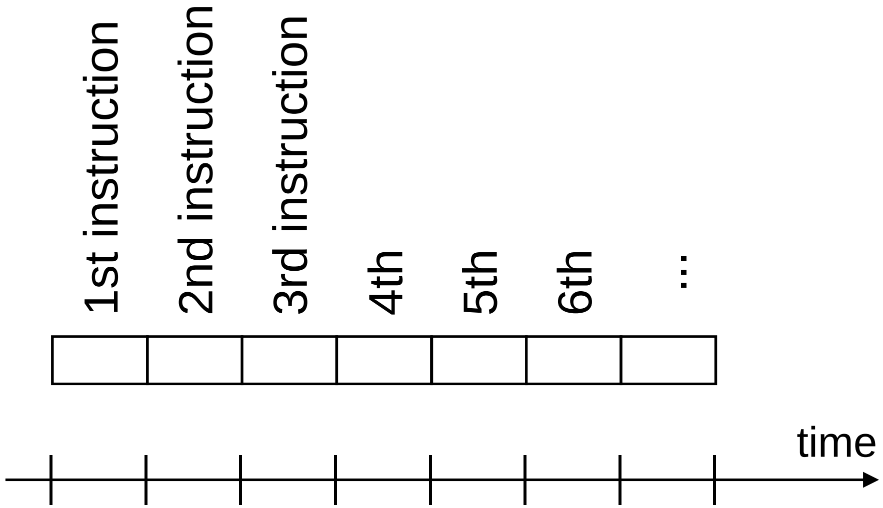


- The clock cycle time is the amount of time for one **clock period** to elapse (e.g. 5 ns).
- The clock rate is the inverse of the clock cycle time.
- For example, if a computer has a clock cycle time of 5 ns, the clock rate is:

$$\frac{1}{5 \times 10^{-9}\ \text{sec}} = 200\ \text{MHz}$$

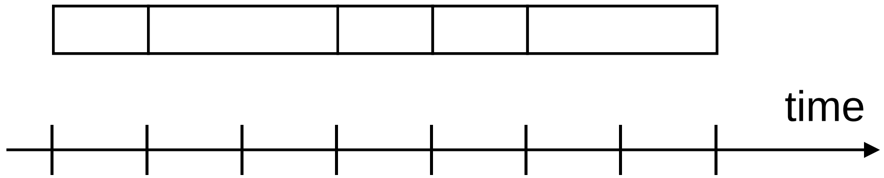
How Many Cycles are Required for a Program?

- Could assume that # of cycles = # of instructions



- This assumption is incorrect, different instructions take different amounts of time on different machines.

Different Numbers of Cycles for Different Instructions



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

Now That We Understand Cycles

- A given program will require
  - some number of instructions (machine instructions)
  - some number of clock cycles
  - some number of seconds
- We have a vocabulary that relates these quantities:
  - clock cycle time (seconds per cycle)
  - clock rate (cycles per second)
  - CPI (cycles per instruction)
    - *a floating point intensive application might have a higher CPI*

Computing CPU Time

- The time to execute a given program can be computed as

  CPU time = CPU clock cycles x clock cycle time

- Since clock cycle time and clock rate are reciprocals

  CPU time = CPU clock cycles / clock rate

- The number of CPU clock cycles can be determined by

  CPU clock cycles = (instructions/program) x (clock cycles/instruction)

  = Instruction count x CPI

  which gives

  CPU time = Instruction count x CPI x clock cycle time

  CPU time = Instruction count x CPI / clock rate

- The units for CPU time are

$$\text{CPU time} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{clock cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

Which factors are affected by each of the following?

	instr. Count	CPI	clock rate
Program	x		
Compiler	x	x	
Instr. Set Arch.	x	x	
Organization		x	x
Technology			x

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

CPU Time Example

- **Example 1:**
  - CPU clock rate is 1 MHz
  - Program takes 45 million cycles to execute
  - What’s the CPU time?

45,000,000 * (1 / 1,000,000) = 45 seconds

- **Example 2:**
  - CPU clock rate is 500 MHz
  - Program takes 45 million cycles to execute
  - What’s the CPU time

45,000,000 * (1 / 500,000,000) = 0.09 seconds

CPI Example

- **Example:** Let assume that a benchmark has 100 instructions:

  25 instructions are loads/stores (each take 2 cycles)

  50 instructions are adds (each takes 1 cycle)

  25 instructions are square root (each takes 50 cycles)

What is the CPI for this benchmark?

$$CPI = ((0.25 * 2) + (0.50 * 1) + (0.25 * 50)) = 13.5$$

Computing CPI

- The CPI is the average number of cycles per instruction.
- If for each instruction type, we know its frequency and number of cycles need to execute it, we can compute the overall CPI as follows:

$$CPI = \sum CPI \times F$$

- For example

Op	F	CPI	CPI x F	% Time
ALU	50%	1	.5	23%
Load	20%	5	1.0	45%
Store	10%	3	.3	14%
Branch	20%	2	.4	18%
Total	100%		2.2	100%

Performance

- Performance is determined by execution time
- Do you think any of the variables is sufficient enough to determine computer performance?
  - # of cycles to execute program?
  - # of instructions in program?
  - # of cycles per second?
  - average # of cycles per instruction?
  - average # of instructions per second

- It is not true to think that one of the variables is indicative of performance.

CPI Example

- Suppose we have two implementations of the same instruction set architecture (ISA).

  For some program,

  Machine A has a clock cycle time of **10 ns**. and a CPI of **2.0**
  Machine B has a clock cycle time of **20 ns**. and a CPI of **1.2**

- Which machine is faster for this program, and by how much?

Assume that # of instructions in the program is 1,000,000,000.

$\text{CPU Time}_A = 10^9 * 2.0 * 10 * 10^{-9} = 20 \text{ seconds}$

$\text{CPU Time}_B = 10^9 * 1.2 * 20 * 10^{-9} = 24 \text{ seconds}$

Machine A is faster

$\frac{24}{20} = 1.2 \text{ times}$

Number of Instruction Example

- A compiler designer is trying to decide between two code sequences for a particular machine. Based on the hardware implementation, there are three different classes of instructions: Class A, Class B, and Class C, and they require one, two, and three cycles (respectively).

The first code sequence has 5 instructions: 2 of A, 1 of B, and 2 of C

The second sequence has 6 instructions: 4 of A, 1 of B, and 1 of C.

- Which sequence will be faster? How much?
- What is the CPI for each sequence?

# of cycles for first code = (2 * 1) + (1 * 2) + (2 * 3) = 10 cycles

# of cycles for second code = (4 * 1) + (1 * 2) + (1 * 3) = 9 cycles

10 / 9 = 1.11 times

CPI for first code = 10 / 5 = 2

CPI for second code = 9 / 6 = 1.5