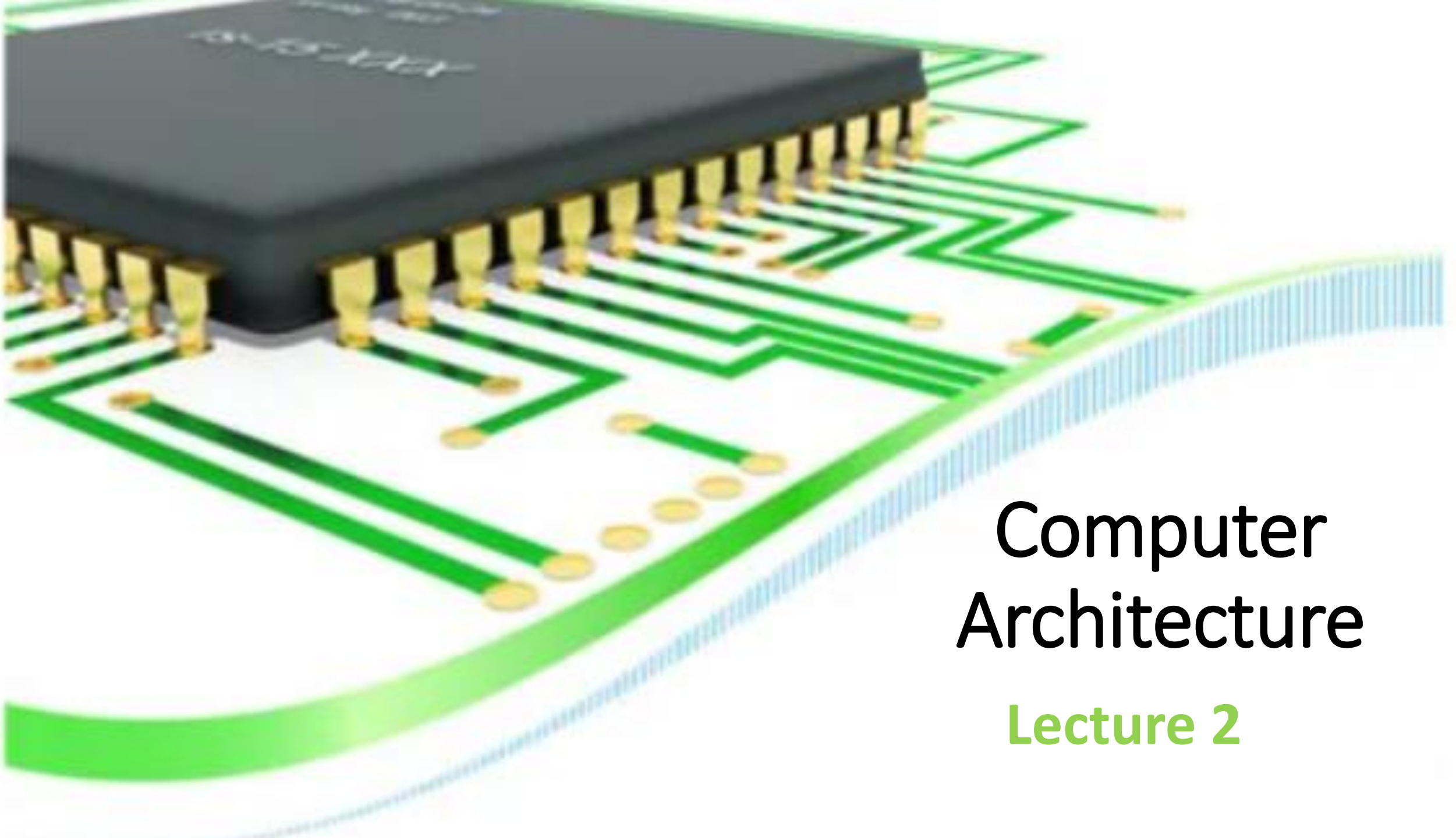


Computer Architecture

Lecture 2

Computer Performance

- Measuring and Comparing different computers is critical to purchasers and therefore to designers.

- When could we say that a computer has better performance than another?

Computer Performance

- Computer performance is the amount of useful work accomplished by a computer system.

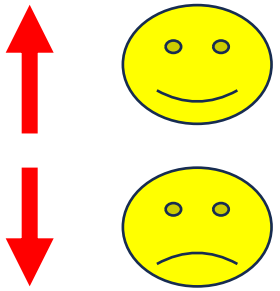
- Computer performance is estimated in terms of accuracy, efficiency, and speed of executing computer program instructions.

MIPS

Warning:
We will meet another MIPS in the module

- *Million instructions per second (MIPS)* is a measure of a computer's speed, providing a standard for representing the number of instructions that a central processing unit (CPU) can process in 1 second.

MIPS



The computer that has the higher MIPS value is the one with the highest performance

Two problems with MIPS

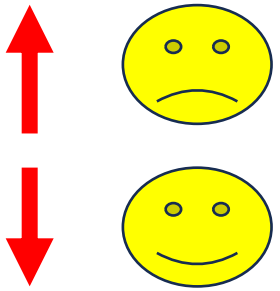
- There are two problems (of our interest) with using MIPS as a measure for comparing computers:
  - ❑MIPS doesn’t take into account the capabilities of instructions.
  - ❑MIPS varies between programs on the same computer

    Thus, a computer can not have a single MIPS rating.

Execution Time (Response Time)

- Execution time is the total time required for the computer to complete a task.

Execution time



Performance is inversely proportional to Execution time

If given the same task on two different computers, the computer that has the higher performance is the one that executes the task in less execution time

Relative Performance

- If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

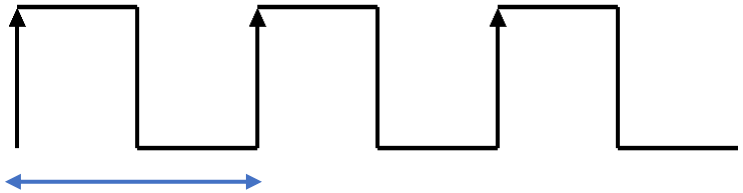
$$\text{Speed Up} = \frac{Performance_A}{Performance_B} = \frac{Execution\ time_B}{Execution\ time_A}$$

$$= \frac{15}{10} = 1.5$$

A is therefore 1.5 times as fast as B

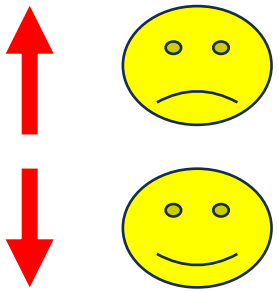
Execution time based on Clock cycles

- To calculate the execution time, one needs to calculate the average clock cycles per instruction (CPI).



Clock cycle time varies from one system to the other

$$CPI = \frac{Total\ clock\ cycles}{Instruction\ count}$$



$$Execution\ time = Total\ clock\ cycles \times clock\ cycle\ time$$

$$Execution\ time = Instruction\ count \times CPI \times clock\ cycle\ time$$

CPI (average clock cycles per instruction)

A given program consists of 40 instruction loop that is executed 10 times, and 20 load and store instructions. If it takes 7,000 cycles to execute the program on a given system, what is the system CPI value for the program? What is the execution time if the clock cycle time is 2ns?

Solution:

- $CPI = \frac{Total\ clock\ cycles}{Instruction\ count}$
- Instruction Count (IC) = (40x10) + 20 = 420 instructions.
- CPI = 7000/420 = 16.67 clock cycles/instruction
- $Execution\ time = Total\ clock\ cycles \times clock\ cycle\ time$ = 7000 x 2 = 14000 ns

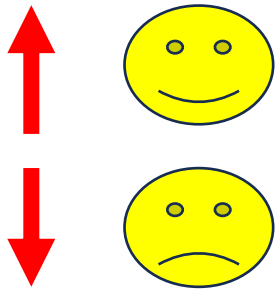
OR

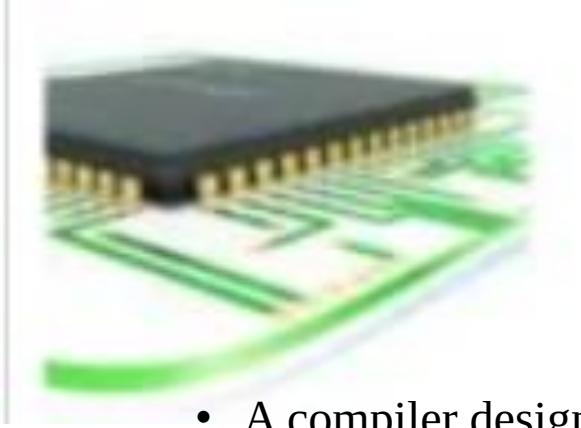
- $Execution\ time = Instruction\ count \times CPI \times clock\ cycle\ time$ = 420 x 16.67x 2 = 14000 ns

IPC (average number of instructions per clock cycle)

- Some systems could accomplish several instructions per cycle. For these systems, we use an average number of instructions per cycle (IPC)

$$IPC = \frac{Instruction\ count}{Total\ clock\ cycles}$$





Example

- A compiler designer is trying to decide between two code sequences for a particular machine. The hardware designers have supplied the following facts:

Instruction class	Clock cycle of the instruction class
A	4
B	2
C	3

- For a particular high-level language, the compiler writer is considering two sequences that require the following instruction counts:

Code sequence	Instruction Counts in Millions		
	A	B	C
1	5	4	3
2	2	6	2

What is the CPI for each sequence? Which code sequence is faster according to CPI?

Example

Instruction class	Clock cycle of the instruction class
A	4
B	2
C	3

Code sequence	Instruction Counts in Millions		
	A	B	C
1	5	4	3
2	2	6	2

- $CPI = \frac{Total\ clock\ cycles}{Instruction\ count}$
- $CPI_1 = \frac{(4\times5)+(2\times4)+(3\times3)\times10^6}{(5+4+3)\times10^6} = 3.08$
- $CPI_2 = \frac{(4\times2)+(2\times6)+(3\times2)\times10^6}{(2+6+2)\times10^6} = 2.6$

Code sequence 2 is faster than code sequence 1 according to CPI measurement.

Check your understanding

- Consider a machine with four instruction classes (A,B,C and D) with clock cycles 3,4,2, and 3 respectively. For a particular high-level language, the compiler writer is considering two sequences that require the following instruction counts:

Code sequence	Instruction Counts			
	A	B	C	D
1	10	20	50	40
2	15	10	60	30

What is the CPI for each sequence? Which code sequence is faster according to CPI?

- $CPI = \frac{Total\ clock\ cycles}{Instruction\ count}$
- $CPI_1 = \frac{(10\times3)+(20\times4)+(50\times2)+(40\times3)}{(10+20+50+40)} = 2.75$
- $CPI_2 = \frac{(15\times3)+(10\times4)+(60\times2)+(30\times3)}{(15+10+60+30)} = 2.56$

Code sequence 2 is faster than code sequence 1 according to CPI measurement.

If an Enhancement is introduced to a system.

How to measure the Speed Up gained?

Using Amdahl's law

Amdahl’s law

- It states that “the overall performance improvement gained by optimizing a single part of a system is limited by the fraction of time that the improved part is actually used”. The formula used to measure such improvement as follows:

$$SpeedUp_{overall} = \frac{1}{Frac_{unused} + \frac{Frac_{used}}{Speedup_{used}}}$$

Amdahl's Law

Suppose a given architecture does not have hardware support for division, so divisions have to be done through repeated subtractions. If it takes 300 cycles to perform a division in software and 5 cycles to perform a division in hardware if a hardware for performing a division is added.

What's the overall speedup from hardware for division if a program spends 30% of its time doing divisions?

$$SpeedUp_{overall} = \frac{1}{Frac_{unused} + \frac{Frac_{used}}{Speedup_{used}}}$$

$Speedup_{used}$ = 300/5 = 60 (ratio of time to do a division without the hardware to time with the hardware).

$Frac_{used}$ = 0.3

$Frac_{unused}$ = 1 - 0.3 = 0.7

Using Amdahl's law Speed $up_{overall}$ = 1 / [0.7 + (0.3/60)] = 1.48

Check your understanding

An enhancement is introduced to a given architecture. If this enhancement caused a speed up of 30 for the part affected by this enhancement.

What's the overall speedup of the system if the programs on this system has 20% of its instructions affected by this enhancement?

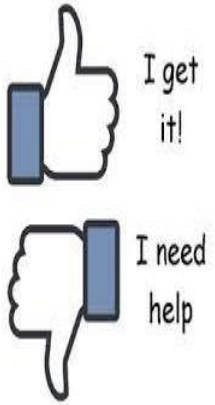
$$SpeedUp_{overall} = \frac{1}{Frac_{unused} + \frac{Frac_{used}}{Speedup_{used}}}$$

$Speedup_{used} = 30$

$Frac_{used} = 0.2$

$Frac_{unused} = 1 - 0.2 = 0.8$

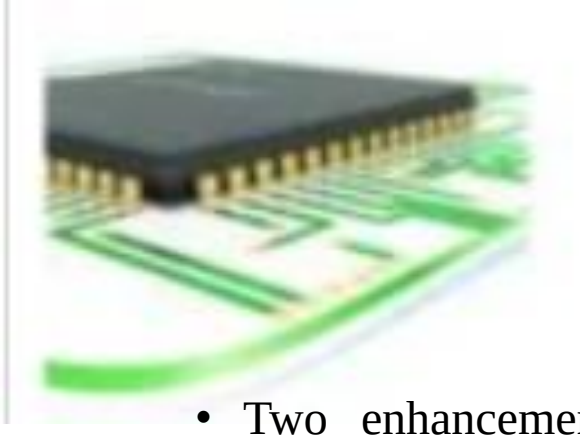
Using Amdahl's law Speed $up_{overall} = 1 / [0.8 + (0.2/30)] = 1.23$



What if more than one enhancement is introduced?
How to measure the Speed Up gained?

Using Amdahl's law

$$SpeedUp_{overall} = \frac{1}{1 - \sum Frac_{used} + \frac{\sum Frac_{used}}{\sum Speedup_{used}}}$$



Amdahl's law

- Two enhancements with the following speedups are proposed for a new machine: Speedup from enhancement(a) = 40, Speedup from enhancement(b) = 20. Assume that for some set of programs, the fraction of use is 15% for enhancement (a), and 20% for enhancement (b). Calculate the Speedup overall if the two enhancements are applied.

$$SpeedUp_{overall} = \frac{1}{1 - \sum Frac_{used} + \frac{\sum Frac_{used}}{\sum Speedup_{used}}}$$

$$SpeedUp_{overall} = \frac{1}{1-(0.15+0.2) + \frac{(0.15+0.2)}{(40+20)}} = 1.52$$

Next Week's Lab Topics

- Problems on Performance

- Bring a Sheet of paper, a pen or a pencil with you

Thank You