COURSE TITLE

Course Code	:30102315
Credit Hours	:3
Prerequisite	:30102214

Instructor Information

Name	: Dr. Rushdi Saleem Abu Zneit				
Office No.	**18, building 17 floor 3**				
Tel (Ext)	**Contact Telephone(Department Telephone)**				
E-mail	dr.rushdizneit@bau.edu.jo				
Office Hours	**Online**				
Class Times	Building	Day	Start Time	End Time	Room No.
		Sun,Tue, Thu	10:00	11:00	**Online**

Examples

Integrated Circuits Costs

$$\text{IC cost} = \frac{\text{Die cost} + \text{Testing cost} + \text{Packaging cost}}{\text{Final test yield}}$$

$$\text{Die cost} = \frac{\text{Wafer cost}}{\text{Dies per Wafer} \times \text{Die Yield}}$$

$$\text{Dies per wafer} = \frac{\Pi\,(\text{Wafer\_diam}/2)^2}{\text{Die\_Area}} - \frac{\Pi \times \text{Wafer\_diam}}{\sqrt{2 \times \text{Die\_Area}}} - \text{Test\_Die}$$

Die Cost goes up roughly with $(\text{Die\_Area})^2$

Examples of Cost of an IC

- Example 1 (p.22): Find the number of dies per 300 mm (30 cm) wafer for a die that is 1.5 cm on a side.
  - The total die area is 2.25 cm². Thus

$$\#\text{Dies per wafer} = \frac{\pi \times (\text{Wafer radius})^2}{\text{Die area}} - \frac{\pi \times \text{Wafer diameter}}{\sqrt{2 \times \text{Die area}}}$$

$$= \frac{\pi \times (30/2)^2}{2.25} - \frac{\pi \times 30}{\sqrt{2 \times 2.25}} = \frac{706.5}{2.25} - \frac{94.2}{2.12} = 270$$

- Example 2 (p.24): Find the die yield for dies that are 1.5 cm on a side and 1.0 cm on a side, assuming a defect density of 0.4 per cm²and α is 4.
  - The total die areas are 2.25 cm² and 1.00 cm². For the large die the yield is

$$\text{Die yield} = \text{Wafer yield} \times \left(1 + \frac{\text{Detect desity} \times \text{Die area}}{\alpha}\right)^{-\alpha}$$

$$= \left(1 + \frac{0.4 \times 2.25}{4.0}\right)^{-4} = 0.44$$

For the small die, it is

$$\left(\quad 0.4 \times 1.00 \right)^{-4}$$

Die Area and Cost

- **Processor Area:**

$$\text{Die yield} = \text{Wafer yield} \times \left(1 + \frac{\text{Defects per unit area} \times \text{Die area}}{\alpha}\right)^{-\alpha}$$

- **Example:**

Find the die yield for dies that are 1.5 cm on a side and 1.0 cm on a side, assuming a defect density of 0.4 per cm ^{2} and α is 4.

- ***Answer:***

The total die areas are 2.25 cm ^{2} and 1.00 cm ^{2} . For the larger die, the yield is

$$\text{Die yield} = \left(1 + \frac{0.4 \times 2.25}{4.0}\right)^{-4} = 0.44$$

For the smaller die, it is $\text{Die yield} = \left(1 + \frac{0.4 \times 1.00}{4.0}\right)^{-4} = 0.68$

That is, less than half of all the large die are good but more than two-thirds of the small die are good.

Case 1: Wafer 1

Diameter of the wafer = 15 cm
Area of a wafer = $3.14 \times 7.5 \times 7.5 = 176.625$

Number of dies per wafer = 84

Hence, area of 1 die = $\frac{176.625}{84} = 2.10 \text{ cm}^2$ (2 decimal places)

$$\text{Yield} = \frac{1}{\left(1 + \text{defects per area} \times \frac{\text{die area}}{2}\right)^2}$$

$$= \frac{1}{\left(1 + 0.02 \times \frac{2.10}{2}\right)^2}$$

$$= \frac{1}{1.021}$$

$$= 0.9794$$

Hence, yield for first wafer = 0.9794

A program runs on computer A with a 2 GHz clock in 10 seconds. What clock rate must a computer B run at to run this program in 6 seconds? Unfortunately, to accomplish this, computer B will require 1.2 times as many clock cycles as computer A to run the program.

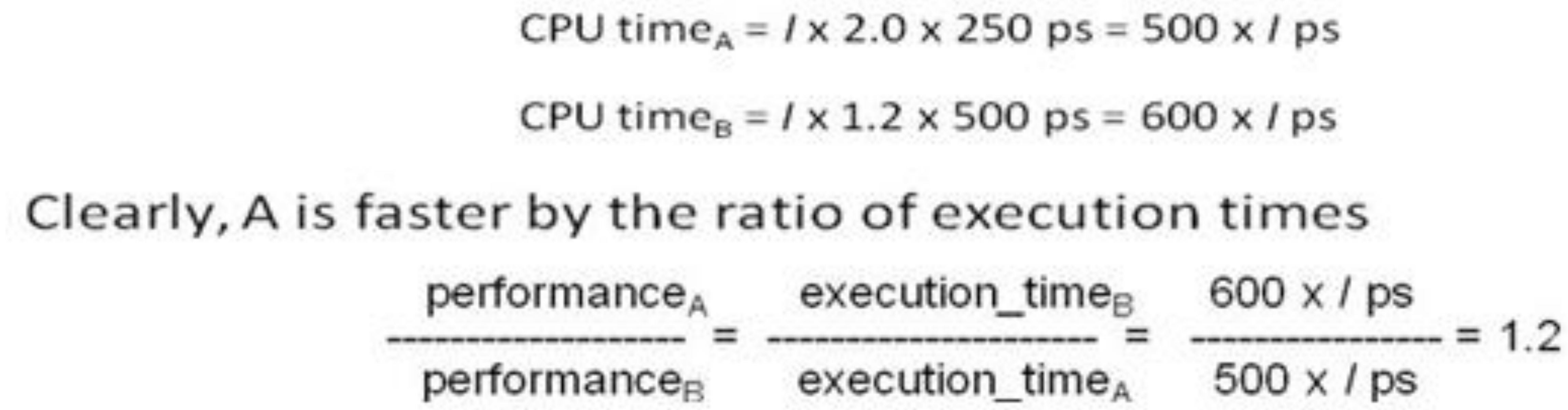
$$\begin{aligned}\text{CPU clock cycles}_A &= 10 \text{ sec} \times 2 \times 10^9 \text{ cycles/sec}\\ &= 20 \times 10^9 \text{ cycles}\end{aligned}$$

$$\text{CPU time}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{clock rate}_B}$$

$$\text{clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = 4 \text{ GHz}$$

Computers A and B implement the same ISA. Computer A has a clock cycle time of 250 ps and an effective CPI of 2.0 for some program and computer B has a clock cycle time of 500 ps and an effective CPI of 1.2 for the same program. Which computer is faster and by how much?

Each computer executes the same number of instructions, I, so



$$\text{CPU time}_A = I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}$$

$$\text{CPU time}_B = I \times 1.2 \times 500 \text{ ps} = 600 \times I \text{ ps}$$

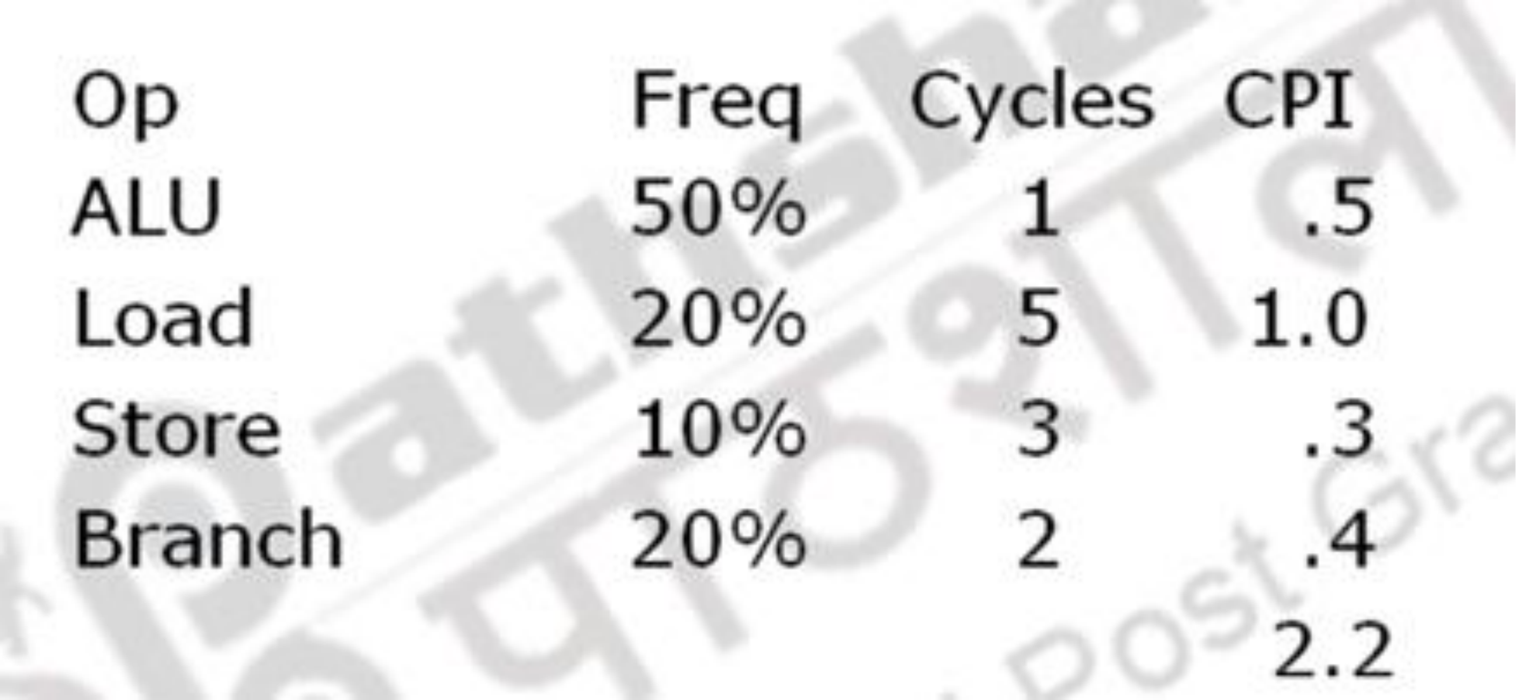
Clearly, A is faster by the ratio of execution times

$$\frac{\text{performance}_A}{\text{performance}_B} = \frac{\text{execution\_time}_B}{\text{execution\_time}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}} = 1.2$$

Computing the overall effective CPI is done by looking at the different types of instructions and their individual cycle counts and averaging.

The overall effective CPI varies by instruction mix – is a measure of the dynamic frequency of instructions across one or many programs.

To look at an example, consider the following instruction mix:



Op	Freq	Cycles	CPI
ALU	50%	1	.5
Load	20%	5	1.0
Store	10%	3	.3
Branch	20%	2	.4
			2.2

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

- Load à 20% x 2 cycles = .4
- Total CPI 2.2 –(1-0.4)= 1.6
- Relative performance is 2.2 / 1.6 = 1.38

How does this compare with reducing the branch instruction to 1 cycle?

- Branch à 20% x 1 cycle = .2
- Total CPI 2.2 –(0.4-0.2)=2.0
- Relative performance is 2.2 / 2.0 = 1.1 We can now write the basic performance equation as:

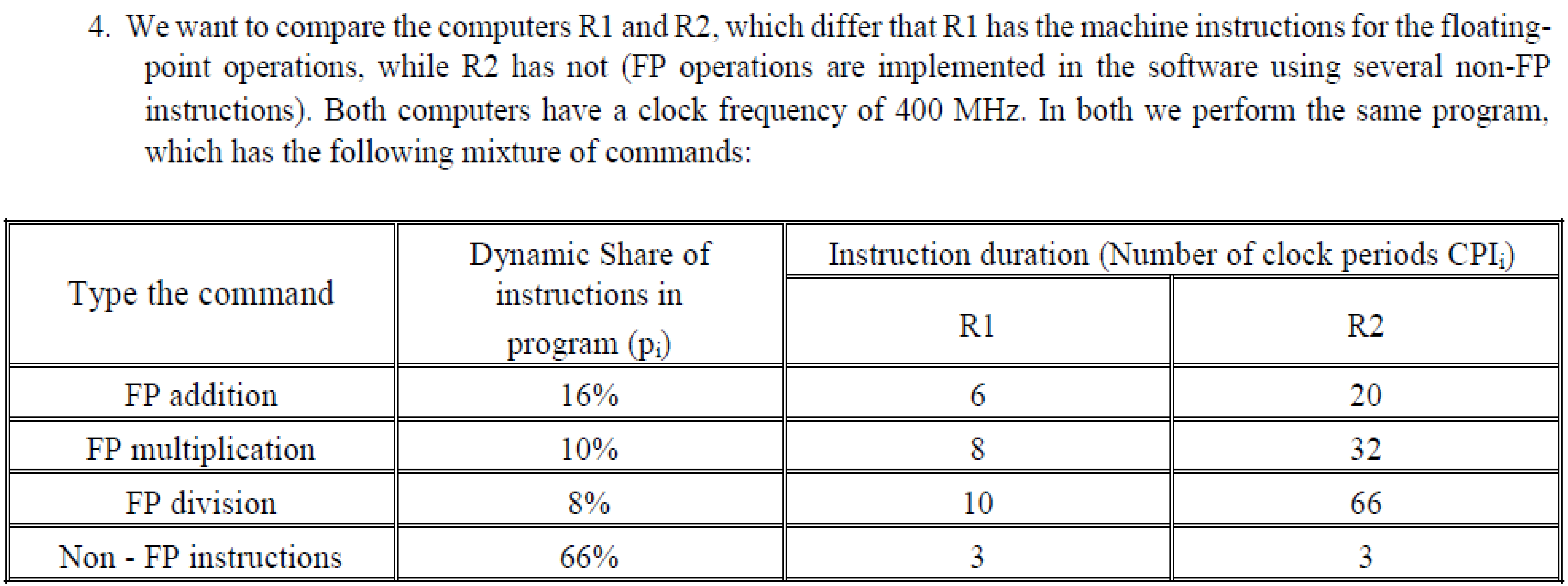
$$\text{CPU time} = \text{Instruction\_count} \times \text{CPI} \times \text{clock\_cycle}$$

or

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

$$\frac{\text{CPU time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{ClockCycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{ClockCycle}}$$

$$\text{ExecutionTime} = \frac{1}{\text{Performance}} = (\text{Instr.Count}) \times (\text{CPI}) \times (\text{cycletime})$$



4. We want to compare the computers R1 and R2, which differ that R1 has the machine instructions for the floating-point operations, while R2 has not (FP operations are implemented in the software using several non-FP instructions). Both computers have a clock frequency of 400 MHz. In both we perform the same program, which has the following mixture of commands:

Type the command	Dynamic Share of instructions in program (p_i)	Instruction duration (Number of clock periods CPI_i)	
		R1	R2
FP addition	16%	6	20
FP multiplication	10%	8	32
FP division	8%	10	66
Non - FP instructions	66%	3	3

a) Calculate the MIPS for the computers R1 and R2.

b) Calculate the CPU program execution time on the computers R1 and R2, if there are 12000 instructions in the program?

c) At what mixture of instructions in the program will both computers R1 and R2 be equally fast?

Solution:

a) CPI = $\sum_{i=0}^{3} CPI_i * p_i$

$$\text{MIPS} = \frac{f_{CPE}}{CPI*10^6}$$

Computer R1:

$$\text{CPI} = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 6 + 0{,}1 * 8 + 0{,}08 * 10 + 0{,}66 * 3 = 4{,}54$$

Computer R1 needs an average of 4.54 clock periods for one instruction

$$\text{MIPS} = \frac{f_{CPE}}{CPI*10^6} = \frac{400*10^6}{4{,}54*10^6} = 88{,}1$$

Computer R1 executes an average of 88 100 000 instructions per second.

Computer R2:

$$\text{CPI} = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 20 + 0{,}1 * 32 + 0{,}08 * 66 + 0{,}66 * 3 = 13{,}66$$

Computer R2 needs an average of 13.66 clock periods for one instruction

$$\text{MIPS} = \frac{\text{f}_{CPE}}{\text{CPI}*10^6} = \frac{400*10^6}{13{,}66*10^6} = 29{,}28$$

Computer R2 executes an average of 29 280 000 instructions per second.

b) $$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS*10^6}}$$

Another form of the equation to calculate the CPU time is:

$$CPU_{time} = Number\_of\_instructions * CPI * t_{CPU}$$

Computer R1:

$$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS * 10^6}} = \frac{12000}{88{,}1 * 10^6} = 136{,}2 * 10^{-6} = 136{,}2\ \mu s$$

Computer R2:

$$\boldsymbol{CPU}_{\text{time}} = \frac{\text{Number\_of\_instructions}}{\boldsymbol{MIPS * 10^6}} = \frac{12000}{29{,}28 * 10^6} = 410 * 10^{-6} = 410\ \mu s$$

$$\text{Overall Speedup} = \frac{\text{Old execution time}}{\text{New execution time}}$$

$$= \frac{1}{\left((1 - \text{Fraction}_{\text{enhanced}}) + \dfrac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)}$$

Amdahl's Law example:
Make the common case fast

- Fraction = 0.1, Speedup = 10

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1-\text{Fraction}_{\text{enhanced}}\right)+\dfrac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$= \frac{1}{\left(1-0.1\right)+\dfrac{0.1}{10}} = \frac{1}{0.91} = 1.1$$

- Fraction = $\underline{0.9}$, Speedup = 10

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1-0.9\right)+\dfrac{0.9}{10}} = \frac{1}{0.19} = 5.3$$

Cache Performance Parameters

- Performance of a cache is largely determined by:
  - Cache miss rate: number of cache misses divided by number of accesses.
  - Cache hit time: the time between sending address and data returning from cache.
  - Cache miss penalty: the extra processor stall cycles caused by access to the next-level cache.

Impact of Memory System on Processor Performance

CPU Performance $_{\text{with Memory Stall}}$ = CPI $_{\text{without stall}}$ + Memory Stall CPI

Memory Stall CPI

= Miss per inst × miss penalty

= % Memory Access/Instr × Miss rate × Miss Penalty

Example: Assume 20% memory acc/instruction, 2% miss rate, 400-cycle miss penalty. How much is memory stall CPI?

Memory Stall CPI= 0.2*0.02*400=1.6 cycles

CPU Performance with Memory Stall

CPU Performance $_{\text{with Memory Stall}}$ = CPI $_{\text{without stall}}$ + Memory Stall CPI

$$\text{CPU time} = \text{IC} \times \left(\text{CPI}_{\text{execution}} + \text{CPI}_{\text{mem\_stall}}\right) \times \text{Cycle Time}$$

$$\text{CPI}_{\text{mem\_stall}} = \text{Miss per inst} \times \text{miss penalty}$$

$$\text{CPI}_{\text{mem\_stall}} = \text{Memory Inst Frequency} \times \text{Miss Rate} \times \text{Miss Penalty}$$

Performance Example 1

- Suppose:
  - Clock Rate = 200 MHz (5 ns per cycle), Ideal (no misses) CPI = 1.1
  - 50% arith/logic, 30% load/store, 20% control
  - 10% of data memory operations get 50 cycles miss penalty
  - 1% of instruction memory operations also get 50 cycles miss penalty
- Compute AMAT.

Performance Example 1

cont…

- CPI = ideal CPI + average stalls per instruction = 1.1(cycles/ins) + [0.30 (DataMops/ins)

  × 0.10 (miss/DataMop) × 50 (cycle/miss)]

  +[1 (InstMop/ins)

  × 0.01 (miss/InstMop) × 50 (cycle/miss)]

  = (1.1 + 1.5 + .5) cycle/ins = 3.1
- AMAT=(1/1.3)×[1+0.01×50]+(0.3/1.3)×[1+0.1×50]=2.54

Example 2

- *Assume* 20% Load/Store instructions
- *Assume CPI without memory stalls is 1*
- Cache hit time = 1 cycle
- Cache miss penalty = 100 cycles
- Miss rate = 1%
- What is:
  - stall cycles per instruction?
  - average memory access time?
  - CPI with and without cache

Example 2: Answer

- Average memory accesses per instruction = 1.2
- AMAT = 1 + 1.2*0.01*100= 2.2 cycles
- Stall cycles = 1.2 cycles
- CPI with cache = 1+1.2=2.2
- CPI without cache=1+1.2*100=121