Q1

1. We are considering an enhancement to a processor. The new CPU is 10x faster on computation than the original CPU. Assuming that the CPU is busy with computation 40% of the time and waiting for I/O 60% of the time. What is the overall speedup gained by incorporating the enhancement?

2. What if the enhancement can be applied system-wide?

3. What if the enhancement is nearing infinity?

Soln

1. Fraction enhanced = 0.4

Speedup enhanced = 10

Speed up = ?

2. Fraction enhanced = 1

Speedup enhanced = 10

Speedup =

3. Fraction enhanced = 0.4

Speedup enhanced = Infinite

Soln

1. Fraction enhanced = 0.4

Speedup enhanced = 10

Speed up = 1.56

2. Fraction enhanced = 1

Speedup enhanced = 10

Speedup = 10

3. Fraction enhanced = 0.4

Speedup enhanced = Infinite

Speedup = 1.67 --> Implication??

Soln

1. Fraction enhanced = 0.4

Speedup enhanced = 10

Speed up = 1.56

2. Fraction enhanced = 1

Speedup enhanced = 10

Speedup = 10

If an enhancement is applicable only for a fraction of a task, then we can't speed up the task by more than the reciprocal of 1 minus that fraction

3. Fraction enhanced = 0.4

Speedup enhanced = Infinite

Speedup = 1.67 --> Implication??

Q2

Eg – Compare 2 code sequences

- Assume that an ISA has 3 classes of instructions:
  - Class A: CPI =1 (eg- Add)
  - Class B: CPI = 2 (eg- Shift)
  - Class C: CPI = 3 (eg- branch)
- Assume a program having 2 code sequences with the following instruction counts per sequence

	A	B	C
Sequence 1	2	1	2
Sequence 2	4	1	1

- Which sequence is faster? (calculate the number of clock cycles)
- What is the CPI for each sequence?
  - CPI = Clock cycles / Instruction count

Solution

- Which sequence is faster?
  - Seq1 = 5 inst, Seq2= 6 instr
  - CPU clk cycles
    - Seq1 – 2*1 + 1*2 + 2*3 = 10 cycles
    - Seq 2- 9 cycles - faster

Solution

- What is the CPI for each sequence?
  - CPI (1) = 10/5 = 2
  - CPI (2) = 9/6 = 1.5 --> more instructions--> leads to less CPI?

$$\text{CPI} = \frac{\text{CPU clock cycles}}{\text{Instruction count}}$$

Q3

Eg - Instructions and CPI

	No of instructions	Cycles
ALU (add, mul etc)	650	1
Load	600	5
Store	100	5
Branch	50	2

For a 2GHz processor, what is the execution time and CPI?

Soln

$(650 \times 1 + 100 \times 5 + 600 \times 5 + 50 \times 2) \times$

$0.5\ e\text{-}9 = 2125$ ns

CPI = time × clock rate/No. Instr

$2125 \times e - 9 \times 2 \times 10^9 / 1{,}400 = 3.03$

Q4 – Program, Assembly and CPI

for (i=0; i<100; i=i+1)

{ x[i] = x[i] + s; }

```
       LW $r3, 10($r1)
       Add $r6, $r2, 100
Loop:  LW $r3, 12 ($r2)
       Add $r6, $r2, $10
       SW $r3, 12 ($r12)
       Add $r6, $r2, $10
       BNE $r6, $r2, loop
```

Assume

	CPI
ALU (add, mul etc)	**1**
Load	**5**
Store	**3**
Branch	**2**

- Total number of instructions = ?
- Fraction of each instruction type = ?
- Total CPU cycles?
- Average CPI?
- What is the execution time for a clock frequency of 2 Ghz?

..cont

	CPI
ALU (add, mul etc)	**1**
Load	**5**
Store	**3**
Branch	**2**

```
        LW $r3, 10($r1)
        Add $r6, $r2, 100
Loop:   LW $r3, 12 ($r2)
        Add $r6, $r2, $10
        SW $r3, 12 ($r12)
        Add $r6, $r2, $10
        BNE $r6, $r2, loop
```

- Total number of instructions = 2 (before loop) + (5* 100) = 502
- Fraction:
  - ALU = 1 + (2*100) = 201 -- > fraction = 201/502 =
  - Load = 1 + (1 * 100) = 101 -- > fraction = 101/502
  - Store = 100 --> fraction = 100/502
  - Branch = 100 --> fraction = 100/502

..cont

	CPI
ALU (add, mul etc)	**1**
Load	**5**
Store	**3**
Branch	**2**

- ALU = 1 + (2*100) = 201
- Load = 1 + (1 * 100) = 101
- Store = 100
- Branch = 100

- CPU clock cycles = Instruction count * CPI
  - (201 * 1) + (101*5) + (100 * 3) + (100 * 2) = 1206 cycles
- Average CPI = CPU clock cycles/Instruction count
  - CPI = 1206/502= 2.4

..cont

	CPI
ALU (add, mul etc)	**1**
Load	**5**
Store	**3**
Branch	**2**

- What is the execution time for a clock frequency of 2 Ghz?

- CPU execution time = I * CPI * Clock period
  - 502 * 2.4 * (1/2G)

Exercise 1.3

Consider three different processors P1, P2, and P3 executing the same instruction set with the clock rates and CPIs given in the following table.

	Processor	Clock Rate	CPI
a.	P1	3 GHz	1.5
	P2	2.5 GHz	1.0
	P3	4 GHz	2.2
b.	P1	2 GHz	1.2
	P2	3 GHz	0.8
	P3	4 GHz	2.0

1.3.1 [5] <1.4> Which processor has the highest performance expressed in instructions per second?

1.3.2 [10] <1.4> If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

For problems below, use the information in the following table.

	Processor	Clock Rate	No. Instructions	Time
a.	P1	3 GHz	20.00E+09	7 s
	P2	2.5 GHz	30.00E+09	10 s
	P3	4 GHz	90.00E+09	9 s
b.	P1	2 GHz	20.00E+09	5 s
	P2	3 GHz	30.00E+09	8 s
	P3	4 GHz	25.00E+09	7 s

1.3.4 [10] <1.4> Find the IPC (instructions per cycle) for each processor.

1.3.5 [5] <1.4> Find the clock rate for P2 that reduces its execution time to that of P1.

1.3.6 [5] <1.4> Find the number of instructions for P2 that reduces its execution time to that of P3.

Solution

1.3.1 P2 has the highest performance.

Instr/sec = f/CPI

a.	performance of P1 (instructions/sec) = $3 \times 10^9/1.5 = 2 \times 10^9$ performance of P2 (instructions/sec) = $2.5 \times 10^9/1.0 = 2.5 \times 10^9$ performance of P3 (instructions/sec) = $4 \times 10^9/2.2 = 1.8 \times 10^9$
b.	performance of P1 (instructions/sec) = $2 \times 10^9/1.2 = 1.66 \times 10^9$ performance of P2 (instructions/sec) = $3 \times 10^9/0.8 = 3.75 \times 10^9$ performance of P3 (instructions/sec) = $4 \times 10^9/2 = 2 \times 10^9$

1.3.2 No. cycles = time × clock rate

time = (No. Instr × CPI)/clock rate, then No. instructions = No. cycles/CPI

a.	cycles(P1) = $10 \times 3 \times 10^9 = 30 \times 10^9$ s cycles(P2) = $10 \times 2.5 \times 10^9 = 25 \times 10^9$ s cycles(P3) = $10 \times 4 \times 10^9 = 40 \times 10^9$ s No. instructions(P1) = $30 \times 10^9/1.5 = 20 \times 10^9$ No. instructions(P2) = $25 \times 10^9/1 = 25 \times 10^9$ No. instructions(P3) = $40 \times 10^9/2.2 = 18.18 \times 10^9$
b.	cycles(P1) = $10 \times 2 \times 10^9 = 20 \times 10^9$ s cycles(P2) = $10 \times 3 \times 10^9 = 30 \times 10^9$ s cycles(P3) = $10 \times 4 \times 10^9 = 40 \times 10^9$ s No. instructions(P1) = $20 \times 10^9/1.2 = 16.66 \times 10^9$ No. instructions(P2) = $30 \times 10^9/0.8 = 37.5 \times 10^9$ No. instructions(P3) = $40 \times 10^9/2 = 20 \times 10^9$

1.3.3 $time_{new} = time_{old} \times 0.7 = 7$ s

Solution

a.	$CPI_{new} = CPI_{old} \times 1.2$, then CPI(P1) = 1.8, CPI(P2) = 1.2, CPI(P3) = 2.6 f = No. Instr × CPI/time, then f(P1) = 20 × 10^9 × 1.8 / 7 = 5.14 GHz f(P2) = 25 × 10^9 × 1.2 / 7 = 4.28 GHz f(P1) = 18.18 × 10^9 × 2.6 / 7 = 6.75 GHz
b.	$CPI_{new} = CPI_{old} \times 1.2$, then CPI(P1) = 1.44, CPI(P2) = 0.96, CPI(P3) = 2.4 f = No. Instr × CPI/time, then f(P1) = 16.66 × 10^9 × 1.44/7 = 3.42 GHz f(P2) = 37.5 × 10^9 × 0.96/7 = 5.14 GHz f(P1) = 20 × 10^9 × 2.4/7 = 6.85 GHz

1.3.4 IPC = 1/CPI = No. instr/(time × clock rate)

a.	IPC(P1) = 0.95 IPC(P2) = 1.2 IPC(P3) = 2.5
b.	IPC(P1) = 2 IPC(P2) = 1.25 IPC(P3) = 0.89

1.3.5

a.	$Time_{new}/Time_{old}$ = 7/10 = 0.7. So $f_{new} = f_{old}/0.7$ = 2.5 GHz/0.7 = 3.57 GHz.
b.	$Time_{new}/Time_{old}$ = 5/8 = 0.625. So $f_{new} = f_{old}/0.625$ = 4.8 GHz.

1.3.6

a.	$Time_{new}/Time_{old}$ = 9/10 = 0.9. Then $Instructions_{new} = Instructions_{old}$ × 0.9 = 30 × 10^9 × 0.9 = 27 × 10^9.
b.	$Time_{new}/Time_{old}$ = 7/8 = 0.875. Then $Instructions_{new} = Instructions_{old}$ × 0.875 = 26.25 × 10^9.

Exercise 1.4

Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

		Clock Rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
a.	P1	2.5 GHz	1	2	3	3
	P2	3 GHz	2	2	2	2
b.	P1	2.5 GHz	2	1.5	2	1
	P2	3 GHz	1	2	1	1

1.4.1 [10] <1.4> Given a program with 10^6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?

1.4.2 [5] <1.4> What is the global CPI for each implementation?

1.4.3 [5] <1.4> Find the clock cycles required in both cases.

Solution

1.4.1

Class A: 10^5 instr.
Class B: 2×10^5 instr.
Class C: 5×10^5 instr.
Class D: 2×10^5 instr.

Time = No. instr × CPI/clock rate

a.	Total time P1 = $(10^5 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3)/(2.5 \times 10^9) = 10.4 \times 10^{-4}$ s Total time P2 = $(10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2)/(3 \times 10^9) = 6.66 \times 10^{-4}$ s
b.	Total time P1 = $(10^5 \times 2 + 2 \times 10^5 \times 1.5 + 5 \times 10^5 \times 2 + 2 \times 10^5)/(2.5 \times 10^9) = 6.8 \times 10^{-4}$ s Total time P2 = $(10^5 + 2 \times 10^5 \times 2 + 5 \times 10^5 + 2 \times 10^5)/(3 \times 10^9) = 4 \times 10^{-4}$ s

1.4.2 CPI = time × clock rate/No. instr

a.	CPI (P1) = $10.4 \times 10^{-4} \times 2.5 \times 10^9/10^6 = 2.6$ CPI (P2) = $6.66 \times 10^{-4} \times 3 \times 10^9/10^6 = 2.0$
b.	CPI (P1) = $6.8 \times 10^{-4} \times 2.5 \times 10^9/10^6 = 1.7$ CPI (P2) = $4 \times 10^{-4} \times 3 \times 10^9/10^6 = 1.2$

1.4.3

a.	clock cycles (P1) = $10^5 \times 1 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 3 + 2 \times 10^5 \times 3 = 26 \times 10^5$ clock cycles (P2) = $10^5 \times 2 + 2 \times 10^5 \times 2 + 5 \times 10^5 \times 2 + 2 \times 10^5 \times 2 = 20 \times 10^5$
b.	clock cycles (P1) = 17×10^5 clock cycles (P2) = 12×10^5

		No. Instructions				
		Compute	Load	Store	Branch	Total
a.	Program1	600	600	200	50	1450
b.	Program 2	900	500	100	200	1700

1.5.4 [5] <1.4> Assuming that computes take 1 cycle, loads and store instructions take 10 cycles, and branches take 3 cycles, find the execution time on a 3 GHz MIPS processor.

1.5.5 [5] <1.4> Assuming that computes take 1 cycle, loads and store instructions take 2 cycles, and branches take 3 cycles, find the execution time on a 3 GHz MIPS processor.

Final Solution:

1.5.4

a.	2.91 μs
b.	2.50 μs

1.5.5

a.	0.78 μs
b.	0.90 μs

H&P 4th edition

Exercise 1.6

Compilers can have a profound impact on the performance of an application on given a processor. This problem will explore the impact compilers have on execution time.

	Compiler A		Compiler B	
	No. Instructions	**Execution Time**	**No. Instructions**	**Execution Time**
a.	1.00E+09	1.8 s	1.20E+09	1.8 s
b.	1.00E+09	1.1 s	1.20E+09	1.5 s

1.6.1 [5] <1.4> For the same program, two different compilers are used. The table above shows the execution time of the two different compiled programs. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.

1.6.2 [5] <1.4> Assume the average CPIs found in 1.6.1, but that the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A's code versus the clock of the processor running compiler B's code?

Solution

1.6.1 $CPI = T_{exec} \times f/\text{No. Instr}$

	Compiler A CPI	Compiler B CPI
a.	1.8	1.5
b.	1.1	1.25

1.6.2 $f_A/f_B = (\text{No. Instr}(A) \times CPI(A))/(\text{No. Instr}(B) \times CPI(B))$

a.	$f_A/f_B = 1$
b.	$f_A/f_B = 0.73$