Important Equations for Performance Measurements:

$$\text{MIPS} = \frac{Instruction\ count}{Execution\ Time}$$ MIPS (Million Instructions per second)

$$\text{CPI} = \frac{Total\ number\ of\ Clock\ Cycles}{Instruction\ count}$$ CPI (average clock cycles per instruction)

$$\text{IPC} = \frac{Instruction\ count}{Total\ number\ of\ Clock\ Cycles}$$ IPC (average num. of instructions per cycle)

Amdahl's law: $\text{Speed up} = \frac{Execution\ Time_{old}}{Execution\ Time_{new}} = \frac{1}{Frac_{unused} + \frac{Frac_{used}}{Speedup_{used}}}$

Problem 1:

If a program takes 35 seconds to run on one version of an architecture and 15 seconds to run on a new version, what is the overall Speed up?

Solution:

$\text{Speed up} = \frac{Execution\ Time_{old}}{Execution\ Time_{new}} =$ 35/15 = 2.33

Problem 2:

A given program consists of 100-instruction loop that is executed 42 times. If it takes 16,000 cycles to execute the program on a given system, what are that system's CPI and IPC values for the program?

Solution:

Total number of instruction executed = 100x42 = 4200

Total number of clock cycles for the program = 16000

CPI = Total number of clock cycles / instruction count = 16000/4200 = 3.81

IPC = Instruction count/ Total number of clock cycles = 4200/16000 = 0.26

Problem 3:

A given program consists of 50-instruction loop that is executed 20 times, 30 load and store instructions. If it takes 5,000 cycles to execute the program on a given system, what are that system's CPI and IPC values for the program?

Solution:

Total number of instruction executed = 50x20 +30 = 1030

Total number of clock cycles to for the program = 5000

CPI = Total number of clock cycles / Instruction count = 5000/1030 = 4.85

IPC = Instruction count/ Total number of clock cycles = 1030/5000 = 0.20

Problem 4:

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	1
B	3
C	4

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	2	1	2
2	4	3	1

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

Solution:

$$CPI_1 = \frac{Total\ Clock\ Cycles}{Instrcution\ count} = \frac{(2\times1+1\times3+2\times4)\times10^6}{(2+1+2)\times10^6} = 2.6$$

$$CPI_2 = \frac{Total\ Clock\ Cycles}{Instrcution\ count} = \frac{(4\times1+3\times3+1\times4)\times10^6}{(4+3+1)\times10^6} = 2.125$$

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

Problem 5:

Consider a machine with three instruction classes and Clock cycle as follows:

Instruction class	Clock cycle of the instruction class
A	2
B	5
C	7

Suppose that we measured the code for a given program in two different compilers and obtained the following data:

Code sequence	Instruction Counts in Millions		
	A	B	C
Compiler 1	15	5	3
Compiler 2	25	2	2

Which code sequence will execute faster according to CPI?

Solution:

$$CPI_1 = \frac{(15\times2+5\times5+3\times7)\times10^6}{23\times10^6} = 3.3$$

$$CPI_2 = \frac{(25\times2+2\times5+2\times7)\times10^6}{29\times10^6} = 2.55$$

Code sequence on Compiler2 will be faster than on Compiler 1 according to CPI measurement.

Problem 6:

We are interested in two different implementations of a machine, one called MFP with special floating point hardware and one called MNFP without special floating point hardware.

Consider a program P with the following mix of arithmetic operations:

Floating-point multiply	15%
Floating-point add	15%
Floating-point divide	5%
Integer Instructions	65%

Machine MFP (Machine with Floating Point) has floating point hardware and can therefore implement the floating point operations directly. It requires the following number of clock cycles for each instruction class:

Floating-point multiply	6
Floating-point add	4
Floating-point divide	20
Integer Instructions	2

Machine MNFP has no floating point hardware and so must emulate the floating point operations using integer instructions. The integer instructions all take 2 clock cycles. The number of integer instructions needed to implement each of the floating point operations as follows:

Floating-point multiply	90
Floating-point add	20
Floating-point divide	50

a) Calculate the CPI for both machines.
b) If the machine MFP needs 300 million instructions for program P, how many integer instructions does MNFP require for the same program?

Solution:

a) <u>For Machine MFP:</u>

$CPI = 0.15 \times 6 + 0.15 \times 4 + 0.05 \times 20 + 0.65 \times 2 = 3.8$

For Machine MNFP:

CPI = 2

b) For Machine MFP:

Instruction Count = 300×10^6

For Machine MNFP:

Instruction Count = $300 \times 10^6 \times [0.15 \times 90 + 0.15 \times 20 + 0.05 \times 50 + 0.65 \times 1]$

$= 5895 \times 10^6$ instructions

Problem 7:

Suppose that a given architecture that does not have a hardware support for multiplication, so multiplications have to be done through repeated addition. If it takes 200 cycles to perform a multiplication in software and 4 cycles to perform a multiplication in hardware.

a) What's the overall speedup from hardware for multiplication if a program spends 10% of its time doing multiplications?
b) What's the overall speedup from hardware for multiplication if a program spends 40% of its time doing multiplications?

Solution:

Speed up = 200/4 = 50 (ratio of time to do a multiplication without the hardware to time with the hardware).

Using Amdahl's law

a) Frac used = 0.1
Frac unused = 1 - 0.1 = 0.9
Speed up = 1 / [0.9 + (0.1/50)] = 1.11

b) Frac used = 0.4
Frac unused = 1 - 0.4 = 0.6
Speed up = 1 / [0.6 + (0.4/50)] = 1.64

Problem 8:

Three enhancements with the following speedups are proposed for a new machine: Speedup (a) = 30, Speedup (b) = 20, and Speedup (c) = 15. Assume that for some set of programs, the fraction of use is 25% for enhancement (a), 30% for enhancement (b), and 45% for enhancement (c). If only one enhancement can be implemented, which should be chosen to maximize the speedup? If two enhancements can be implemented, which should be chosen, to maximize the speedup?

Solution:

One Enhancement:

a) $SU_a = \frac{1}{(1-F_a)+\frac{F_a}{SU_1}} = \frac{1}{(1-0.25)+\frac{0.25}{30}} = 1.3186$

b) $SU_b = \frac{1}{(1-F_b)+\frac{F_b}{SU_2}} = \frac{1}{(1-0.3)+\frac{0.3}{20}} = 1.3986$

c) $SU_c = \frac{1}{(1-F_c)+\frac{F_c}{SU_3}} = \frac{1}{(1-0.45)+\frac{0.45}{15}} = 1.7241$

If one enhancement can be implemented, enhancement (c) is the best.

Two Enhancements:

a) a & b:

$SU_{a\text{-}b} = \frac{1}{(1-F_a-F_b)+\frac{F_a+F_b}{SU_1+SU_2}} = \frac{1}{(1-0.25-0.3)+\frac{0.25+0.3}{30+20}} = 2.17$

b) a & c:

$$SU_{a\text{-}c} = \frac{1}{(1 - F_a - F_c) + \frac{F_a + F_c}{SU_1 + SU_3}} = \frac{1}{(1 - 0.25 - 0.45) + \frac{0.25 + 0.45}{30 + 15}} = 3.17$$

c) b & c:

$$SU_{b\text{-}c} = \frac{1}{(1 - F_b - F_c) + \frac{F_b + F_c}{SU_2 + SU_3}} = \frac{1}{(1 - 0.3 - 0.45) + \frac{0.3 + 0.45}{20 + 15}} = 3.68$$

If two enhancements can be implemented, implementing enhancements (b and c) is the best.