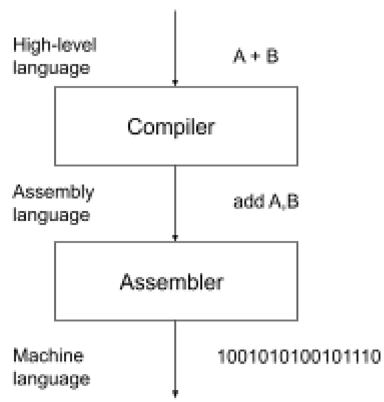
COEN 210 Computer Architecture: Assignment 1

Name: Inchara Raveendra

SCU ID: 00001653600

1.3 Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor.

A program written in a high-level language such as C is compiled by the compiler into an assembly language. The assembler then converts it into a machine language which is executed by a computer processor.



1.4 Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024.

a. What is the minimum size in bytes of the frame buffer to store a frame?

Number of bits per pixel = 8 x 3 = 24 bits

Minimum size (in bytes) of the frame buffer to store a given sized frame = (24 x 1280 x 1024)/8 = 3932160 bytes/frame

---

b. How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?

Minimum time for a frame to be sent over the given network = (3932160 x 8)/(100 x 10^6) = 0.315s

---

1.6 Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3GHz clock rate and a CPI of 1.5. P2 has a 2.5GHz clock rate and a CPI of 1.0. P3 has a 4.0GHz clock rate and has a CPI of 2.2.
a. Which processor has the highest performance expressed in instructions per second?

CPU time = (Instruction Count x CPI)/Clock Rate
Assuming Instruction Count = 1 based on given

Performance = Clock Rate/ CPI
Performance_{P1} = (3 x 10^9)/1.5 = 2 x 10^9 instructions/second
Performance_{P2} = (2.5 x 10^9)/1 = 2.5 x 10^9 instructions/second
Performance_{P3} = (4 x 10^9)/2.2 = 1.8181 x 10^9 instructions/second

Process P2 has the highest performance of the three.

---

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions

CPU time = (Instruction Count x CPI)/Clock Rate
10 = (I_{P1} x 1.5)/3
I_{P1} = 30/1.5
I_{P1} = 20 x 10^9 instructions

10 = (I_{P2} x 1)/2.5
I_{P2} = 25/1
I_{P2} = 25 x 10^9 instructions

10 = (I_{P3} x 2.2)/4
I_{P3} = 40/2.2
I_{P3} = 18.182 x 10^9 instructions

Clock Cycle = Instruction Count x CPI

Clock Cycle_{P1} = 20 x 10^9 x 1.5 = 30x10^9 cycles

Clock Cycle_{P2} = 25 x 10^9 = 25x10^9 cycles

Clock Cycle_{P3} = 18.182 x 2.2 x 10^9= 40x10^9 cycles

---

c. We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

CPU time = (Instruction Count x CPI)/Clock Rate

$7 = (20 \times 10^9 (1.5 \times 1.2))/\text{Clock Rate}_{P1}$

$\text{Clock Rate}_{P1} = (20 \times 10^9 \times 1.5 \times 1.2)/7 = 5.1428\text{GHz}$

$\text{Clock Rate}_{P2} = (25 \times 10^9 \times 1 \times 1.2)/7 = 4.2857\text{GHz}$

$\text{Clock Rate}_{P3} = (18.18 \times 10^9 \times 2.2 \times 1.2)/7 = 6.8564\text{GHz}$

1.7 Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.

Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which is faster: P1 or P2?

CPU time = (Instruction Count x CPI)/Clock Rate

$\text{CPU time}_{P1A} = (\text{Instruction Count}_{P1A} \times \text{CPI}_{P1A})/\text{Clock Rate}_{P1}$

$\text{CPU time}_{P1A} = (10^5 \times 1)/(2.5 \times 10^9) = 4 \times 10^{-5}$

$\text{CPU time}_{P1B} = (2 \times 10^5 \times 2)/(2.5 \times 10^9) = 16 \times 10^{-5}$

$\text{CPU time}_{P1C} = (5 \times 10^5 \times 3)/(2.5 \times 10^9) = 60 \times 10^{-5}$

$\text{CPU time}_{P1D} = (2 \times 10^5 \times 3)/(2.5 \times 10^9) = 24 \times 10^{-5}$

$\text{CPU time}_{P1} = \text{CPU time}_{P1A} + \text{CPU time}_{P1B} + \text{CPU time}_{P1C} + \text{CPU time}_{P1D} = 1.04\text{^}10\text{-}3$

$\text{CPU time}_{P2A} = (\text{Instruction Count}_{P2A} \times \text{CPI}_{P2A})/\text{Clock Rate}_{P2}$

$\text{CPU time}_{P2A} = (10^5 \times 2)/(3 * 10^9) = 6.67 \times 10^{-5}$

$\text{CPU time}_{P2B} = (2 \times 10^5 \times 2)/(3 * 10^9) = 13.33 \times 10^{-5}$

CPU $time_{P2C}$= (5x10^5 x 2)/(3*10^9) = 33.33x10^-5

CPU $time_{P2D}$= (2x10^5 x 2)/(3*10^9) = 13.33x10^-5

CPU $time_{P2}$= CPU $time_{P2A}$+CPU $time_{P2B}$+CPU $time_{P2C}$+CPU $time_{P2D}$ = 66.66x10^-5

CPU $time_{P1}$>CPU $time_{P2}$

Therefore, P2 is faster.

a. What is the global CPI for each implementation?

Global CPI = (CPU $time_{P1}$xClock $Rate_{P1}$)/Instruction Count

Global CPI_{P1} = (1.04 x 10^-3 x 2.5 x 10^9)/(1x 10^6) = 2.6

Global CPI_{P2} = (66.667 x 10^-5 x 3 x 10^9)/(1x 10^6) = 2

---

b. Find the clock cycles required in both cases.

Clock Cycle = Instruction Count x CPI

Clock $Cycle_{P1}$ = 1 x 10^6 x 2.6 = 2.6x10^6 cycles
Clock $Cycle_{P2}$ = 1 x 10^6 x 2 = 2x10^6 cycles

1.8 Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.

a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.

CPU time = Instruction count x CPI x Clock cycle time

CPU $time_A$ = 1 x 10^9 x CPI_A x 10^-9 = 1.1s
CPI_A = 1.1

CPU $time_B$ = 1.2 x 10^9 x CPI_B x 10^-9 = 1.5s
CPI_B = 1.5/1.2 = 1.25

---

b. Assume 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?

CPU $time_A$ = CPU $time_B$

Instruction count x CPI x Clock cycle time
1 x 10^9 x 1.1 x Clock cycle $time_A$ = 1.2 x 10^9 x 1.25 x Clock cycle $time_B$

Clock cycle $time_A$/Clock cycle $time_B$ = (1.2 x 1.25)/1.1 = 1.3636

---

c. A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?

CPU $time_{new}$ = Instruction count x CPI x Clock cycle time

CPU $time_A$/CPU $time_{new}$ = (1 x 10^9 x 1.1)/(6 x 10^8 x 1.1)
CPU $time_A$/CPU $time_{new}$ = 1.67

CPU $time_B$/CPU $time_{new}$ = (1.2 x 10^9 x 1.25)/(6 x 10^8 x 1.1)
CPU $time_B$/CPU $time_{new}$ = 2.27

1.13 Section 1.11 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

1.13.1 One usual fallacy is to consider the computer with the largest clock rate as having the highest performance. Check if this is true for P1 and P2.

CPU $time_{P1}$ = Instruction count x CPI / Clock rate
CPU $time_{P1}$ = (5 x 10^9 x 0.9)/ (4x10^9) = 1.125s

CPU $time_{P2}$ = (1 x 10^9 x 0.75)/ (3x10^9) = 0.25s

Fallacy: $Performance_{P1}$ > $Performance_{P2}$, if clock $rate_{P1}$> clock $rate_{P2}$

This is not true.
Since CPU $time_{P2}$ < CPU $time_{P1}$ and performace is inversely proportional to CPU time,
$Performance_{P2}$ > $Performance_{P1}$

1.13.2 Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.

CPU time = Instruction count x CPI / Clock rate

CPU time$_{P1}$ = (10^9 x 0.9) / (4x10^9) = 0.225s

CPU time$_{P2}$ = (Instruction count$_{P2}$ x 0.75) / (3x10^9) = 0.225
Instruction count$_{P2}$ = 9x10^8 instructions

1.13.3 A common fallacy is to use MIPS (*millions of instructions per second*) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

MIPS = Instruction Count/ (Execution time x 10^6)

MIPS$_{P1}$ = (5 x 10^9) / (1.125 x 10^6) = 4.44 x 10^3

MIPS$_{P2}$ = (1 x 10^9) / (0.25 x 10^6) = 4 x 10^3

According to the fallacy, Performance$_{P1}$ > Performance$_{P2}$ if MIPS$_{P1}$> MIPS$_{P2}$. However, this is not true.

We have previously proven in **1.13.1** that **Performance$_{P2}$ > Performance$_{P1}$**

1.13.4 Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as

MFLOPS = No. FP operations /(execution time x 1E6)

but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the processors.

MFLOPS = No. FP operations /(execution time x 1E6)

MFLOPS$_{P1}$ = (0.4 x 5 x 10^9) /(1.125 x 10^6) = 1.778 x 10^3

$MFLOPS_{P2}$ = (0.4 x 1 x 10^9) /(0.25 x 10^6) = 1.6 x 10^3

According to the fallacy, $Performance_{P1}$ > $Performance_{P2}$ if $MFLOPS_{P1}$ > $MFLOPS_{P2}$. However, this is not true.

We have previously proven in **1.13.1** that **$Performance_{P2}$ > $Performance_{P1}$**

1.15 Assume a program requires the execution of 50 × 10^6 FP instructions, 110 × 10^6 INT instructions, 80 × 10^6 L/S instructions, and 16 × 10^6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.

1.15.1 By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

CPU time = Instruction count x CPI / Clock rate

CPU $time_{old}$ = ((50 × 10^6 x 1) + (110 × 10^6 x 1) + (80 × 10^6 x 4) + (16 × 10^6 x 2))/ (2 x 10^9) = 0.256

CPU $time_{new}$ = CPU $time_{old}$ /2

((50 × 10^6 x CPI_{FP}) + (462 x 10^6)) = 0.128

CPI_{FP} = (0.128 - 462 x 10^6)/ (50 × 10^6)

Since CPI_{FP} < 0, the program cannot be made to run two times faster.

1.15.2 By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?

CPU $time_{old}$ = ((50 × 10^6 x 1) + (110 × 10^6 x 1) + (80 × 10^6 x 4) + (16 × 10^6 x 2))/ (2 x 10^9) = 0.256

CPU $time_{new}$ = CPU $time_{old}$ /2

((80 × 10^6 x $CPI_{L/s}$) + (0.192 x 10^9)) = 0.128 x 2 x 10^9

$CPI_{L/s}$ = (64 x 10^6)/(80 × 10^6)

$CPI_{L/s}$= 0.8

1.15.3 By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%?

CPU $time_{old}$ = ((50 × 10^6 x 1) + (110 × 10^6 x 1) + (80 × 10^6 x 4) + (16 × 10^6 x 2))/ (2 x 10^9) = 0.256

CPU $time_{new}$ = ((50 × 10^6 x 1 x 0.6) + (110 × 10^6 x 1 x 0.6) + (80 × 10^6 x 4x 0.7) + (16 × 10^6 x 2 x 0.7))/ (2 x 10^9) = (342.4 x 10^6)/ (2 x 10^9)

CPU $time_{new}$ = 0.1712s

Improvement in CPU time = 100 - (0.1712 x 100) / 0.256 = 33.125

CPU time improved by 33.125%
