Aluno: Carlos Gabriel de Freitas
Matrícula: 19.1.4009
Professor: Ricardo Augusto Rabelo Oliveira
Disciplina: BCC 263 – Arquitetura de Computadores

Lista de Exercícios 1

Questão 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.

Primeiramente, um programador pode usar uma linguagem de alto nível, como C ou Java, para expressar sua linha de raciocínio de forma mais concisa e natural, utilizando palavras e expressões algébricas. Então, um compilador traduzirá este programa em uma linguagem *assembly*, uma outra forma de representar simbolicamente as operações a serem realizadas pela máquina. Por fim, um *assembler* irá traduzir essa linguagem *assembly* em linguagem de máquina, composta por *binary digits*, ou bits, podendo ser estes 0 ou 1. Os bits serão entendidos pelo hardware como sinais elétricos de ligar e desligar, e deste modo, as instruções contidas no programa serão realizadas.

Questão 1.5
Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second?

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

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?

1.5.a) Execution Time (P1) = (I*1.5)/(3E9) = 0,0000000005 * I
Execution Time (P2) = (I*1)/(2.5E9) = 0,0000000004 * I
Execution Time (P3) = (I*2.2)/(4E9) = 0,00000000055 * I
P2 é o processador com maior performance.

1.5.b) 10 = CPU Clock Cycles (P1) / (3E9) → CPU Clock Cycles (P1) = 3E10 ciclos
3E10 = I * 1.5 → I (P1) = 2E10 instruções

10 = CPU Clock Cycles (P2) / (2.5E9) → CPU Clock Cycles (P2) = 2.5E10 ciclos
2.5E10 = I * 1 → I (P2) = 2.5E10 instruções

10 = CPU Clock Cycles (P3) / (4E9) → CPU Clock Cycles (P3) = 4E10 ciclos
4E10 = I * 2.2 → I (P3) = 1.8E10 instruções

1.5.c) 0,7 * CPU Time (P1) = (I*1,8)/Clock Rate (P1) →
0,7 * 0,0000000005 * I = (I*1,8)/Clock Rate (P1) →
Clock Rate (P1) = 5.1 GHz

0,7 * CPU Time (P2) = (I*1,2)/Clock Rate (P2) →
0,7 * 0,0000000004 * I = (I*1,2)/Clock Rate (P2) →
Clock Rate (P2) = 4.3 GHz

0,7 * CPU Time (P3) = (I*2,64)/Clock Rate (P3) →

0,7 * 0,00000000055 * I = (I*2,64)/Clock Rate (P3) →
Clock Rate (P3) = 6.9 GHz

Questão 1.6

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?

a. What is the global CPI for each implementation?
b. Find the clock cycles required in both cases.

1.6) CPU Clock Cycles (P1) = (0.1 * 1 * 1E6) + (0.2 * 2 * 1E6) + (0.5 * 3 * 1E6) + (0.2 * 3 * 1E6)
CPU Clock Cycles (P1) = 2 600 000 ciclos
CPU Time (P1) = 2 600 000/ 2.5E9 = 0,00104 segundos

CPU Clock Cycles (P2) = (0.1 * 2 * 1E6) + (0.2 * 2 * 1E6) + (0.5 * 2 * 1E6) + (0.2 * 2 * 1E6)
CPU Clock Cycles (P2) = 2 000 000 ciclos
CPU Time (P2) = 2 000 000/ 3E9 = 0,00067 segundos

P2 é mais rápido.

1.6.a) CPI (P1) = 2 600 000/1E6 = 2.6
CPI (P2) = 2 000 000/1E6 = 2

1.6.b) CPU Clock Cycles (P1) = 1E6 * 2.6 = 2 600 000 ciclos
CPU Clock Cycles (P2) = 1E6 * 2 = 2 000 000 ciclos

Questão 1.7

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.

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?

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?

1.7.a) 1.1 = CPU Clock Cycles (A) * 1E-9 → CPU Clock Cycles (A) = 1 100 000 000 ciclos
1 100 000 000 = 1E9 * CPI (A) → CPI (A) = 1.1

1.5 = CPU Clock Cycles (B) * 1E-9 → CPU Clock Cycles (B) = 1 500 000 000 ciclos
1 500 000 000 = 1.2E9 * CPI (B) → CPI (B) = 1.25

1.7.b) Execution Time = 1 100 000 000 * Clock Cycle Time (A)
Execution Time = 1 500 000 000 * Clock Cycle Time (B)
1 100 000 000 * Clock Cycle Time (A) = 1 500 000 000 * Clock Cycle Time (B)

Clock Cycle Time (A) = (1 500 000 000 * Clock Cycle Time (B)) / 1 100 000 000
Clock Cycle Time (A) = 1.36 * Clock Cycle Time (B)

O clock do processador quando "roda" o código do compilador A é cerca de 1,36 vezes mais rápido do que quando "roda" o código do compilador B.

1.7.c) CPU Clock Cycles (C) = 6E8 * 1.1 → CPU Clock Cycles (C) = 660 000 000 ciclos
Execution Time (C) = 660 000 000 * 1E-9 → Execution Time (C) = 0,66 segundos

CPU Performance (C) / CPU Performance (A) =
CPU Execution Time (A) / CPU Execution Time (C) = 1.1/0.66 = 1.67

CPU Performance (C) / CPU Performance (B) =
CPU Execution Time (B) / CPU Execution Time (C) = 1.5/0.66 = 2.27

O compilador C é cerca de 1.67 vezes mais rápido que o compilador A e cerca de 2.27 vezes mais rápido que o compilador B, usando o mesmo processador.

Questão 1.9

Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency. Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 × p (where p is the number of processors) but the number of branch instructions per processor remains the same.

1.9.1. Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processors result relative to the single processor result.

1.9.2. If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4, or 8 processors?

1.9.3. To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?

1.9.1)

- Para 1 Processador:

Clock Cycles = (1 * 2.56E9) + (12 * 1.28E9) + (5 * 2.56E8) = 1,92E10 ciclos
Execution Time = 1,92E10/2E9 = 9,6 segundos

- Para 2 Processadores:

Número de instruções aritméticas: 2.56E9 / (0,7 * 2) = 1,83E9 instruções
Número de instruções load/store: 1.28E9 / (0,7 * 2) = 9,14E8 instruções
Clock Cycles = (1 * 1.83E9) + (12 * 9.14E8) + (5 * 2.56E8) = 1,41E10 ciclos
Execution Time = 1,41E10/2E9 = 7,039 segundos
9,6 / 7,039 = 1,36
O programa é 1,36 vezes mais rápido ao utilizar 2 processadores, em relação a utilizar 1 processador.

- Para 4 Processadores:

Número de instruções aritméticas: 2.56E9 / (0,7 * 4) = 9,14E8 instruções
Número de instruções load/store: 1.28E9 / (0,7 * 4) = 4,57E8 instruções
Clock Cycles = (1 * 9.14E8) + (12 * 4.57E8) + (5 * 2.56E8) = 7,68E9 ciclos
Execution Time = 7,68E9/2E9 = 3,839 segundos
9,6 / 3,839 = 2,5

O programa é 2,5 vezes mais rápido ao utilizar 4 processadores, em relação a utilizar 1 processador.

- Para 8 Processadores:

Número de instruções aritméticas: 2.56E9 / (0,7 * 8) = 4,57E8 instruções
Número de instruções load/store: 1.28E9 / (0,7 * 8) = 2,29E8 instruções
Clock Cycles = (1 * 4.57E8) + (12 * 2.29E8) + (5 * 2.56E8) = 4,48E9 ciclos
Execution Time = 4,48E9/2E9 = 2,2425 segundos
9,6 / 2,2425 = 4,28
O programa é 4,28 vezes mais rápido ao utilizar 8 processadores, em relação a utilizar 1 processador.

1.9.2)

- Para 1 Processador:

Clock Cycles = (2 * 2.56E9) + (12 * 1.28E9) + (5 * 2.56E8) = 2,18E10 ciclos
Execution Time = 2,18E10/2E9 = 10,88 segundos
10,88 / 9,6 = 1,13
Ao dobrar o CPI das instruções aritméticas, o programa é executado 1,13 vezes mais lento ao utilizar 1 processador.

- Para 2 Processadores:

Clock Cycles = (2 * 1.83E9) + (12 * 9.14E8) + (5 * 2.56E8) = 1,59E10 ciclos
Execution Time = 1,59E10/2E9 = 7,954 segundos
7,954 / 7,039 = 1,13
Ao dobrar o CPI das instruções aritméticas, o programa é executado 1,13 vezes mais lento ao utilizar 2 processadores.

- Para 4 Processadores:

Clock Cycles = (2 * 9.14E8) + (12 * 4.57E8) + (5 * 2.56E8) = 8,59E9 ciclos
Execution Time = 8,59E9/2E9 = 4,296 segundos
4,296 / 3,839 = 1,12
Ao dobrar o CPI das instruções aritméticas, o programa é executado 1,12 vezes mais lento ao utilizar 4 processadores.

- Para 8 Processadores:

Clock Cycles = (2 * 4.57E8) + (12 * 2.29E8) + (5 * 2.56E8) = 4,94E9 ciclos
Execution Time = 4,94E9/2E9 = 2,471 segundos
2,471 / 2,2425 = 1,1
Ao dobrar o CPI das instruções aritméticas, o programa é executado 1,1 vezes mais lento ao utilizar 8 processadores.

1.9.3) 3,839 = Clock Cycles / 2E9 → Clock Cycles = 7,69E9 ciclos
7,69E9 = (1 * 2.56E9) + (CPI * 1.28E9) + (5 * 2.56E8)
3,84E9 = CPI * 1.28E9 → CPI = 3
Para que a performance de 1 processador se equipare a performance de 4 processadores usando os valores de CPI originais, o CPI das instruções de load/store deve ser mudado para 3.

Questão 1.12

Section 1.10 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 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

1.12.1. 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.

1.12.2. 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.

1.12.3. 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.

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

MFLOPS = Number of FP Opeartions / (Execution Time * 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.

1.12.1) Execution Time (P1) = (5E9 * 0,9) / 4E9 = 1,125 segundos
Execution Time (P2) = (1E9 * 0,75) / 3E9 = 0,25 segundos
Apesar do clock rate de P1 ser maior, P2 possui uma maior performance.

1.12.2) Execution Time = (1E9 * 0,9) / 4E9 = 0,225 segundos
0,225 = (I * 0,75) / 3E9 → I * 0,75 = 675 000 000 → I = 900 000 000 → I = 9E8 Instruções

1.12.3) MIPS (P1) = 5E9 / (1,125 * 1E6) = 4 444
MIPS (P2) = 1E9 / (0,25 * 1E6) = 4000
Apesar do MIPS de P1 ser maior que o MIPS de P2, já foi comprovado que a performance de P2 é maior que a de P1 através do exercício 1.12.1.

1.12.4) MFLOPS (P1) = (0,4 * 5E9) / (1,125 * 1E6) = 1777
MFLOPS (P2) = (0,4 * 1E9) / (0,25 * 1E6) = 1600
Apesar do MFLOPS de P1 ser maior que o MFLOPS de P2, já foi comprovado que a performance de P2 é maior que a de P1 através do exercício 1.12.1.

Questão 1.14

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.

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

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

1.14.3. 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%?

1.14.1) Clock Cycles = (1 * 50E6) + (1 * 110E6) + (4 * 80E6) + (2 * 16E6) = 5,12E8 ciclos
Execution Time = 5,12E8 / 2E9 = 0,256 segundos
Execution Time (FP instructions) = (1 * 50E6) / 2E9 = 0,025 segundos
0,128 = 0,025 / n + (0,256 – 0,025) → 0,128 – 0,231 = 0,025 / n → - 0,103 = 0,025 / n
n = - 0,243
Como o valor de n é negativo, não há valor de CPI para as instruções FP que faça o programa rodar 2 vezes mais rápido.

1.14.2) Execution Time (L/S instructions) = (4 * 80E6) / 2E9 = 0,16 segundos

$0{,}128 = 0{,}16 / n + (0{,}256 - 0{,}16) \rightarrow 0{,}128 - 0{,}096 = 0{,}16 / n \rightarrow 0{,}032 = 0{,}16 / n \rightarrow n = 5$

Para que o programa rode 2 vezes mais rápido, é necessário melhorar o CPI das instruções L/S em 5 vezes.

1.14.3) New Clock Cycles = (0.6 * 50E6) + (0.6 * 110E6) + (2.8 * 80E6) + (1.4 * 16E6) = 3,424E8 ciclos

New Execution Time = 3,424E8 / 2E9 = 0,1712 segundos

0,256 / 0,1712 = 1,49

O tempo de execução é melhorado em cerca de 1,49 vezes ao aplicar as melhorias descritas.