Problem 3.15.5

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.

- Which processor has the highest performance expressed in instructions per second?
- If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
- 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?

- a)

  CPU time = CPU clock cycles / Clock rate

  CPU clock cycles = instructions * CPI

  CPU time = (instructions * CPI) / Clock rate

  Clock rate / CPI = instructions / CPU time

  Clock rate / CPI = instructions per second

  P1 = 3GHz / 1.5 = (3 x 10^9) / 1.5 = (2 x 10^9) instructions per second

  P2 = 2.5GHz / 1.0 = (2.5 x 10^9) / 1.0 = (2.5 x 10^9) instructions per second

  P3 = 4.0GHz / 2.2 = (4 x 10^9) / 2.2 = (1.82 x 10^9) instructions per second

  P2 has the highest performance because P2 carries out more instructions per second.

- b)

  CPU time = CPU clock cycles / Clock rate

  CPU clock cycles = CPU time x Clock rate

P1 = 10 x (3 x 10^9) = 3 x 10^10 cycles

P2 = 10 x (2.5 x 10^9) = 2.5 x 10 ^10 cycles

P3 = 10 x (4 x 10^9) = 4 x 10^10 cycles

CPU clock cycles = instructions * CPI

Instructions = CPU clock cycles / CPI

P1 = (3 x 10^10) / 1.5 = 2 x 10^10 instructions

P2 = (2.5 x 10 ^10) / 1.0 = 2.5 x 10^10 instructions

P3 = (4 x 10^10) / 2.2 = 1.82 x 10^10 instructions

- c)

  CPU time = (instructions * CPI) / Clock rate

  CPU time (0.7) = (instructions * CPI (1.2)) / Clock rate

  Clock rate = (instructions * CPI (1.2)) / CPU time (0.7)

  New Clock rate = 1.71 * Old Clock Rate

  P1 = 3GHz * 1.71 = 5.13GHz

  P2 = 2.5GHz * 1.71 = 4.27GHz

P3 = 4GHz * 1.71 = 6.84GHz

We multiply the old clock rate by 1.71 to get the new clock rate.

Problem 3.15.7

Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class 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 implementation is faster?

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

- a)

  1.0E6 = 1.0 * 10^6

  CPI

	A	B	C	D
P1	1	2	3	3
P2	2	2	2	2

Instructions

	A (10%)	B (20%)	C (50%)	D (20%)
# Instructions	1.0E5	2.0E5	5.0E5	2.0E5

Global CPI (P1) = (1*1.0E5 + 2*2.0E5 + 3*5.0E5 + 3*2.0E5) / 1.0E6 = 2.6

Global CPI (P2) = (2*1.0E5 + 2*2.0E5 + 2*5.0E5 + 2*2.0E5) / 1.0E6 = 2.0

CPU time = CPU cycles / Clock rate

CPU cycles = instructions * CPI

CPU time = (instructions * CPI) / Clock rate

CPU time (P1) = (1.0E6 * 2.6) / 2.5GHz = (1.0E6 * 2.6) / 2.5*10^9 = .00104s or 1.04 ms

CPU time (P2) = (1.0E6 * 2.0) / 3GHz = (1.0E6 * 2.0) / 3*10^9 = 0.00067 s or 0.67ms

P2 is faster than P1

- b)

  Clock cycles = instructions * CPI

  Clock cycles (P1) = 1E6 * 2.6 = 2.6E6 cycles

  Clock cycles (P2) = 1E6 * 2.0 = 2.0E6 cycles

Problem 3.15.8

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.

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

- a)

  CPU time = CPU clock cycles / Clock rate

  CPU time = CPU clock cycles * Cycle time

CPU clock cycles = instructions * CPI

CPU time = (instructions * CPI) * Cycle time

CPI = CPU time / (instructions * Cycle time)

CPI (A) = 1.1 / (1.0E9 * 1.0E-9) = 1.1

CPI (A) = 1.5 / (1.2E9 * 1.0E-9) = 1.25

- b)

  Clock Cycle = CPU time / (instructions * CPI)

  CPU time is the same so…

  (instructions * CPI) of A = 1.1E9

  (instructions * CPI) of B = 1.5E9

  1.1E9 / 1.5E9 = 1.37

  1.37 times faster

- c)

  Clock Cycles = instructions * CPI

  Clock Cycle (new) = 6.0E8 * 1.1

  CPU time (A) / (CPU time (new) = (1.0E9 * 1.1) / (6.0E8 * 1.1) = 1.67 times faster

  CPU time (B) / (CPU time (new) = (1.2E9 * 1.25) / (6.0E8 * 1.1) = 2.27 times faster

Problem 3.15.10

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 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.

(a)
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 processor result relative to the single processor result.

(b)
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?

(c)
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?

- a)

  Instructions

  Arithmetic = 2.56E9

  Load/Store = 1.28E9

  Branch = 2.56E8

  CPU time = Clock cycles / Clock rate

  Clock cycles = instructions * CPI

  1 Processor

  Clock cycles = (2.56E9 * 1) + (1.28E9 * 12) + (2.56E8 * 5) = 1.92E10

  Time = 1.92E10 / 2.0E9 = 9.6

  2 Processors

  Clock cycles = (2.56E9 * 1) / (.7*2) + (1.28E9 * 12) / (.7*2) + (2.56E8 * 5) = 1.408E10

  Time = 1.408E10/ 2.0E9 = 7.04 sec

  Time (1) / Time (2) = 9.6 / 7.04 = 1.36

  4 Processors

  Clock cycles = (2.56E9 * 1) / (.7*4) + (1.28E9 * 12) / (.7*4) + (2.56E8 * 5) = 7.68E9

Time = 7.68E9/ 2.0E9 = 3.84 sec

Time (1) / Time (2) = 9.6 / 3.84 = 2.5

8 Processors

Clock cycles = (2.56E9 * 1) / (.7*8) + (1.28E9 * 12) / (.7*8) + (2.56E8 * 5) = 4.48E9

Time = 4.48E9/ 2.0E9 = 2.24 sec

Time (1) / Time (2) = 9.6 / 2.24 = 4.29

- b)

  CPU time = (instructions * CPI) / Clock rate

  1 Processor

  Clock cycles = (2.56E9 * 2) + (1.28E9 * 12) + (2.56E8 * 5) = 2.176E10

  Time = 2.176E10 / 2.0E9 = 10.88sec

  2 Processor

  Clock cycles = (2.56E9 * 2) / (.7*2) + (1.28E9 * 12) / (.7*2) + (2.56E8 * 5) = 1.59E10

  Time = 1.59E10/ 2.0E9 = 7.95 sec

  4 Processor

  Clock cycles = (2.56E9 * 2) / (.7*4) + (1.28E9 * 12) / (.7*4) + (2.56E8 * 5) = 8.59E9

  Time = 8.59E9/ 2.0E9 = 4.3 sec

  8 Processor

  Clock cycles = (2.56E9 * 2) / (.7*8) + (1.28E9 * 12) / (.7*8) + (2.56E8 * 5) = 4.94E9

  Time = 4.94E9 / 2.0E9 = 2.47 sec

- c)

  4 Processors

  Clock cycles = (2.56E9 * 1) / (.7*4) + (1.28E9 * 12) / (.7*4) + (2.56E8 * 5) = 7.68E9

  Time = 7.68E9/ 2.0E9 = 3.84 sec

  1 Processor

3.84 sec = [(2.56E9 * 1) + (1.28E9 * x) + (2.56E8 * 5)] / 2E9 GHz

7.68E9 = (2.56E9 * 1) + (1.28E9 * x) + (2.56E8 * 5)

7.68E9 = 3.84E9 + 1.28E9x

3.84E9 = 1.28E9x

3 = x

3/12 = .25

You have to reduce the CPI of load/store by 25%

Problem 3.15.12

The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.

(a)
Find the CPI if the clock cycle time is 0.333 ns.

(b)
Find the SPECratio.

(c)
Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.

(d)
Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.

(e)
Find the change in the SPECratio for this change.

(f)
Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPECratio is 13.7. Find the new CPI.

(g)
This CPI value is larger than obtained in 1.11.1 as the clock rate was increased from 3 GHz

to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?

(h)
By how much has the CPU time been reduced?

(i)
For a second benchmark, libquantum, assume an execution time of 960 ns, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting to the CPI and with a clock rate of 4 GHz, determine the number of instructions.

(j)
Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.

(k)
Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.

- a)

  Instruction = 2.389E12

  CPU time = 750 sec

  Reference time = 9650 sec

  Clock cycle time = .333ns = 3.33E-10

  CPU time = (instructions * CPI) * Cycle time

  750 = (2.389E12 * CPI) * 3.33E-10

  2.23E12 = 2.389E12 * CPI

  0.94 = CPI

- b)

  SPECratio = ref time / exe time

  SPECratio = 9650 / 750 = 12.8

- c)

  CPU time = (instructions * (1.10) * CPI) * cycle time

  CPU time would also increase by 10% = 825 sec

- d)

  CPU time = (instructions * (1.10) * CPI * (1.05)) * cycle time

  CPU time would increase by 15.5% = 866.25 sec

- e)

  SPECratio = 9650 / 866.25

  = 11.14 sec

- f)

  Clock rate = 4GHz = 4E9 cycles/sec

  Instructions = 2.389E12 * .85 = 2.03E12

  Execution time = 700s

  SPECratio = 13.7

  CPI = ?

  CPU time = (instructions * CPI) / Clock rate

  700 = (2.03E12 * CPI) / 4E9

  1.38 = CPI

- g)

If the clock rate increases, the CPI will also increase. If we look at the equation CPU time = (instructions * CPI) / Clock rate, if Clock rate increases, CPU time will also increase. Thus, the greater the CPU time, the greater the CPI.

- h)

700 sec / 750 sec = .93

1- 0.93 = .067 = 6.7%

- i)

  CPU time = (instructions * CPI) / clock rate

  9.6E-7 = (instructions * 1.61) / 3E9

  Instructions = 1788.82

  9.6E-7 * .9 = (instructions * 1.61) / 4E9

  Instructions = 2146.58

- j)

  8.64E-7 * .9 = (2146.58 * 1.61) / clock rate

  Clock rate = 4.44GHz

- k)

  9.6E-7 * .8 = (2146.58 * 1.61 * (.85)) / clock rate

  Clock rate 3.8 GHz

Problem 3.15.15

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.

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

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

(c)
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%?

- a)

  Original Program:

  CPU time = [(50E6 * 1) + (110E6 * 1) + (80E6 * 4) + (16E6 * 2)] / 2E9

  CPU time = .256 sec

  0.256 / 2 = [(50E6 * x) + (110E6 * 1) + (80E6 * 4) + (16E6 * 2)] / 2E9

  0.128 = [(50E6 * x) + (110E6 * 1) + (80E6 * 4) + (16E6 * 2)] / 2E9

  2.56E8= [(50E6 * x) + (110E6 * 1) + (80E6 * 4) + (16E6 * 2)]

  -2.06E8= 50E6 * x

  -4.12 = x = CPI

  CPI is negative, not possible

- b)

  0.128 = [(50E6 *1) + (110E6 * 1) + (80E6 * x) + (16E6 * 2)] / 2E9

  2.56E8 = [(50E6 *1) + (110E6 * 1) + (80E6 * x) + (16E6 * 2)]

  6.4E7 = 80E6 * x

  0.8 = x = CPI

- c)

  INT = 1 * 0.6 = 0.6

  FP = 1 * 0.6 = 0.6

L/S = 4 * 0.7 = 2.8

Branch = 2 * 0.7 = 1.4

CPU time = [(50E6 * 0.6) + (110E6 * 0.6) + (80E6 * 2.8) + (16E6 * 1.4)] / 2E9

CPU time = .1712

0.1712/ 0.256 / = .66875

1 - 0.66875 = .33125

The execution time is improved by about 33%.