

Kevin Kerliu

ECE251 – Computer Architecture

Assignment I – Exercises 1.5 – 1.11

1.5 [4] <§1.6> 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?

P1: Performance₁ = $3 \times 10^9 \text{ Cycles/Second} / 1.5 \text{ Cycles/Instruction} = 2.0 \times 10^9 \text{ Instructions/Second}$

P2: Performance₂ = $2.5 \times 10^9 \text{ Cycles/Second} / 1.0 \text{ Cycles/Instruction} = 2.5 \times 10^9 \text{ Instructions/Second}$

P3: Performance₃ = $4 \times 10^9 \text{ Cycles/Second} / 2.2 \text{ Cycles/Instruction} = 1.8 \times 10^9 \text{ Instructions/Second}$

Thus processor 2 has the highest performance.

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

P1: Number of Cycles = $10 \text{ Seconds} \times 3.0 \times 10^9 \text{ Cycles/second} = 30 \times 10^9 \text{ Cycles}$

Number of Instructions = $30 \times 10^9 \text{ Cycles} / 1.5 \text{ Cycles/Instruction} = 20 \times 10^9 \text{ Instructions}$

P2: Number of Cycles = $10 \text{ Seconds} \times 2.5 \times 10^9 \text{ Cycles/second} = 25 \times 10^9 \text{ Cycles}$

Number of Instructions = $25 \times 10^9 \text{ Cycles} / 1.0 \text{ Cycles/Instruction} = 25 \times 10^9 \text{ Instructions}$

P3: Number of Cycles = $10 \text{ Seconds} \times 4 \times 10^9 \text{ Cycles/second} = 40 \times 10^9 \text{ Cycles}$

Number of Instructions = $40 \times 10^9 \text{ Cycles} / 2.2 \text{ Cycles/Instruction} = 18 \times 10^9 \text{ Instructions}$

Alternative Calculation (Using Part A)

P1: Number of Instructions = $10 \text{ seconds} \times 2.0 \times 10^9 \text{ Instructions/Second} = 20 \times 10^9 \text{ Instructions}$

P2: Number of Instructions = $10 \text{ seconds} \times 2.5 \times 10^9 \text{ Instructions/Second} = 25 \times 10^9 \text{ Instructions}$

P3: Number of Instructions = $10 \text{ seconds} \times 1.8 \times 10^9 \text{ Instructions/Second} = 18 \times 10^9 \text{ Instructions}$

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?

$10 \text{ Seconds} \times 0.70 = 7 \text{ Seconds}$

P1: $20 \times 10^9 \text{ Instructions} \times (1.2 \times 1.5 \text{ Cycles/Instruction}) = 36 \times 10^9 \text{ Cycles}$

Clock Rate = $36 \times 10^9 \text{ Cycles} / 7 \text{ Seconds} = 5.14 \text{ GHz}$

P2: $25 \times 10^9 \text{ Instructions} \times (1.2 \times 1.0 \text{ Cycles/Instruction}) = 30 \times 10^9 \text{ Cycles}$

Clock Rate = $30 \times 10^9 \text{ Cycles} / 7 \text{ Seconds} = 4.29 \text{ GHz}$

P3: $18 \times 10^9 \text{ Instructions} \times (1.2 \times 2.2 \text{ Cycles/Instruction}) = 47.52 \times 10^9 \text{ Cycles}$

Clock Rate = $47.52 \times 10^9 \text{ Cycles} / 7 \text{ Seconds} = 6.79 \text{ GHz}$

1.6 [20] <§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?

P1: Total Number of Cycles = Number of Instructions * Cycles/Instruction
 $= 1.0E6 * (.10 * 1.0 + .20 * 2.0 + .50 * 3.0 + .20 * 3.0) = 2600000$ Cycles
 Time = # of Cycles / Clock Rate = $2600000 \text{ Cycles} / (2.5 * 10^9 \text{ Cycles/Second}) = 1.04 \text{ ms}$

P2: Total Number of Cycles = Number of Instructions * Cycles/Instruction
 $= 1.0E6 * (.10 * 2.0 + .20 * 2.0 + .50 * 2.0 + .20 * 2.0) = 2000000$ Cycles
 Time = # of Cycles / Clock Rate = $2000000 \text{ Cycles} / (3.0 * 10^9 \text{ Cycles/Second}) = 0.67 \text{ ms}$

Thus P2 is the faster processor.

a. What is the global CPI for each implementation?

P1: Global CPI = $(1 + 2 + 3 + 3)/4 = 2.25$ Cycles/Instruction

P2: Global CPI = $(2 + 2 + 2 + 2)/4 = 2.0$ Cycles/Instruction

b. Find the clock cycles required in both cases.

P1: Total Number of Cycles = Number of Instructions * Cycles/Instruction
 $= 1.0E6 * (.10 * 1.0 + .20 * 2.0 + .50 * 3.0 + .20 * 3.0) = 2600000$ Cycles

P2: Total Number of Cycles = Number of Instructions * Cycles/Instruction
 $= 1.0E6 * (.10 * 2.0 + .20 * 2.0 + .50 * 2.0 + .20 * 2.0) = 2000000$ Cycles

1.7 [15] <§1.6> 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 1.0×10^9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2×10^9 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.

Comp. A: $\text{CPI} = 1.1 \text{ Seconds} / (1 \times 10^{-9} \text{ Seconds/Cycle} * 1.0 \times 10^9 \text{ Instructions}) = 1.1 \text{ Cycles/Instruction}$

Comp. B: $\text{CPI} = 1.5 \text{ Seconds} / (1 \times 10^{-9} \text{ Seconds/Cycle} * 1.2 \times 10^9 \text{ Instructions}) = 1.25 \text{ Cycles/Instruction}$

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?

Compiler A: $\text{Time A} = 1.0 \times 10^9 \text{ Instructions} * 1.1 \text{ Cycles/Instruction} * \text{Clock Cycle Time A}$

Compiler B: $\text{Time B} = 1.2 \times 10^9 \text{ Instructions} * 1.25 \text{ Cycles/Instruction} * \text{Clock Cycle Time B}$

$\text{Time A} = \text{Time B}$

$\text{Clock Cycle Time A} * 1.1 \times 10^9 = \text{Clock Cycle Time B} * 1.5 \times 10^9$

$\text{Clock Cycle Time A} = 1.36 * \text{Clock Cycle Time B}$

Thus A is clocking 1.36 times faster than B.

c. A new compiler is developed that uses only 6.0×10^8 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?

$\text{Time A} = 1.0 \times 10^9 \text{ Instructions} * 1.1 \text{ Cycles/Instruction} * 1 \times 10^{-9} \text{ Seconds/Cycle} = 1.1 \text{ Seconds}$

$\text{Time B} = 1.2 \times 10^9 \text{ Instructions} * 1.25 \text{ Cycles/Instruction} * 1 \times 10^{-9} \text{ Seconds/Cycle} = 1.5 \text{ Seconds}$

$\text{Time C} = 6 \times 10^8 \text{ Instructions} * 1.1 \text{ Cycles/Instruction} * 1 \times 10^{-9} \text{ Seconds/Cycle} = 0.66 \text{ Seconds}$

Thus there is a 0.44 second speed up between compilers A and C and a 0.84 second speed up between compilers B and C.

1.8 The Pentium 4 Prescott processor, released in 2004, had a clock rate of 3.6 GHz and voltage of 1.25 V. Assume that, on average, it consumed 10 W of static power and 90 W of dynamic power. The Core i5 Ivy Bridge, released in 2012, has a clock rate of 3.4 GHz and voltage of 0.9 V. Assume that, on average, it consumed 30 W of static power and 40 W of dynamic power.

1.8.1 [5] < §1.7> For each processor find the average capacitive loads.

Pentium 4 Prescott: $C = 2 * 90 \text{ W} / (1.25 \text{ V})^2 * (3.6 \text{ GHz}) = 32 \text{ nF}$

Core i5 Ivy Bridge: $C = 2 * 40 \text{ W} / (0.9 \text{ V})^2 * (3.4 \text{ GHz}) = 24.05 \text{ nF}$

1.8.2 [5] < §1.7> Find the percentage of the total dissipated power comprised by static power and the ratio of static power to dynamic power for each technology.

Pentium 4 Prescott:

Ratio of Static Power to Total Power: $\text{Static Power} / \text{Total Power} = 10 \text{ W} / 100 \text{ W} = .10$

Core i5 Ivy Bridge:

Ratio of Static Power to Total Power: $\text{Static Power} / \text{Total Power} = 30 \text{ W} / 70 \text{ W} = .43$

1.8.3 [15] < §1.7> If the total dissipated power is to be reduced by 10%, how much should the voltage be reduced to maintain the same leakage current? Note: power is defined as the product of voltage and current.

Pentium 4 Prescott:

Power = Current * Voltage

Current = $100 \text{ W} / 1.25 \text{ V} = 80 \text{ A}$

Voltage = $100 \text{ W} * 0.90 / 80 \text{ A} = 1.125 \text{ V}$

Core i5 Ivy Bridge:

Power = Current * Voltage

Current = $70 \text{ W} / 0.90 \text{ V} = 77.78 \text{ A}$

Voltage = $70 \text{ W} * 0.90 / 77.78 \text{ A} = 0.81 \text{ V}$

Thus reduce the voltage by 10% to maintain the same current leakage. This answer makes sense as the relationship $P=I*V$ is linear for power and voltage.

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 2.56×10^9 arithmetic instructions, 1.28×10^9 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 $0.7 \times p$ (where p is the number of processors) but the number of branch instructions per processor remains the same.

1.9.1 [5] < §1.7 > 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 Processor:

of Cycles = # of Instructions * Cycles / Instruction

of Cycles = $2.56 \times 10^9 * 1 + 1.28 \times 10^9 * 12 + 2.56 \times 10^8 * 5 = 1.92 \times 10^{10}$ Cycles

Time = # of Cycles / Clock Rate

Time = 1.92×10^{10} Cycles / 2×10^9 Cycles/Second = 9.6 Seconds

2 Processors:

of Cycles = $2.56 \times 10^9 / 1.4 * 1 + 1.28 \times 10^9 / 1.4 * 12 + 2.56 \times 10^8 * 5 = 1.408 \times 10^{10}$ Cycles

Time = 1.408×10^{10} Cycles / 2×10^9 Cycles/Second = 7.04 Seconds

4 Processors:

of Cycles = $2.56 \times 10^9 / 2.8 * 1 + 1.28 \times 10^9 / 2.8 * 12 + 2.56 \times 10^8 * 5 = 7.68 \times 10^9$ Cycles

Time = 7.68×10^9 Cycles / 2×10^9 Cycles/Second = 3.84 Seconds

8 Processors:

of Cycles = $2.56 \times 10^9 / 5.6 * 1 + 1.28 \times 10^9 / 5.6 * 12 + 2.56 \times 10^8 * 5 = 4.48 \times 10^8$ Cycles

Time = 4.48×10^8 Cycles / 2×10^9 Cycles/Second = 2.24 Seconds

Relative Speedup:

1:2 $7.04 / 9.6 = .73 \rightarrow 27\%$ Faster

1:4 $3.84 / 9.6 = .40 \rightarrow 60\%$ Faster

1:8 $2.24 / 9.6 = .23 \rightarrow 77\%$ Faster

1.9.2 [10] < §§1.6, 1.8 > 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 Processor:

of Cycles = $2.56 \times 10^9 * 2 + 1.28 \times 10^9 * 12 + 2.56 \times 10^8 * 5 = 2.18 \times 10^{10}$ Cycles

Time = 2.1760×10^{10} Cycles / 2×10^9 Cycles/Second = 10.88 Seconds

2 Processors:

of Cycles = $2.56 \times 10^9 / 1.4 * 2 + 1.28 \times 10^9 / 1.4 * 12 + 2.56 \times 10^8 * 5 = 1.59 \times 10^{10}$ Cycles

Time = 1.59×10^{10} Cycles / 2×10^9 Cycles/Second = 7.95 Seconds

4 Processors:

of Cycles = $2.56 \times 10^9 / 2.8 * 2 + 1.28 \times 10^9 / 2.8 * 12 + 2.56 \times 10^8 * 5 = 8.59 \times 10^9$ Cycles

Time = 8.59×10^9 Cycles / 2×10^9 Cycles/Second = 4.30 Seconds

8 Processors:

of Cycles = $2.56 \times 10^9 / 5.6 * 2 + 1.28 \times 10^9 / 5.6 * 12 + 2.56 \times 10^8 * 5 = 4.94 \times 10^9$ Cycles

Time = 4.94×10^9 Cycles / 2×10^9 Cycles/Second = 2.47 Seconds

1.9.3 [10] <§§1.6, 1.8> 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?

Performance of Four Processors using the Original CPI Values: 3.84 Seconds

1 Processor:

of Cycles = $2.56 \times 10^9 \times 1 + 1.28 \times 10^9 \times N + 2.56 \times 10^8 \times 5$

Time = $(2.56 \times 10^9 \times 1 + 1.28 \times 10^9 \times N + 2.56 \times 10^8 \times 5) \text{ Cycles} / 2 \times 10^9 \text{ Cycles/Second} = 3.84 \text{ Seconds}$

For this to occur, N must be 3.0.

1.10 Assume a 15 cm diameter wafer has a cost of 12, contains 84 dies, and has 0.020 defects/cm². Assume a 20 cm diameter wafer has a cost of 15, contains 100 dies, and has 0.031 defects/cm².

1.10.1 [10] <\$1.5> Find the yield for both wafers.

$$\text{Defects} = \text{Area} * \text{Defects/Area}$$

$$\text{Yield} = \# \text{ of Dies} - \text{Defects} / \# \text{ of Dies}$$

Wafer 1: Area = 176.71
 Defects = 3.53
 Yield = 0.96

Wafer 2: Area = 314.15
 Defects = 9.74
 Yield = 0.90

1.10.2 [5] <\$1.5> Find the cost per die for both wafers.

$$\text{Cost / Die} = \text{Cost for Wafer} / \# \text{ of Dies}$$

Wafer 1: Cost / Die = 0.15

Wafer 2: Cost / Die = 0.17

1.10.3 [5] <\$1.5> If the number of dies per wafer is increased by 10% and the defects per area unit increases by 15%, find the die area and yield.

$$\text{Defects / Area} = (1/\sqrt{\text{Yield}} - 1) * 2/\text{Area}$$

Wafer 1: Defects = 2.83
 Yield = 0.97

Wafer 2: Defects = 7.80
 Yield = 0.93

1.10.4 [5] <\$1.5> Assume a fabrication process improves the yield from 0.92 to 0.95. Find the defects per area unit for each version of the technology given a die area of 200 mm².

$$\text{Yield} = 0.95$$

$$\text{Area} = 200$$

$$\text{Defects / Area} = 2.60 * 10^{-4}$$

1.11 The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389×10^{12} , an execution time of 750 s, and a reference time of 9650 s.

1.11.1 [5] <§1.6, 1.9> Find the CPI if the clock cycle time is 0.333 ns.

$\text{CPI} = \text{Execution Time} / \text{Clock Cycle Time} * \text{Instruction Count}$

$\text{CPI} = 0.94 \text{ Cycles / Instruction}$

1.11.2 [5] <§1.9> Find the SPECratio.

$\text{SPECratio} = \text{Reference Time} / \text{Execution Time}$

1.11.3 [5] <§1.6, 1.9> Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.

$\text{CPU Time} = \text{Instruction Count} * \text{CPI} * \text{Clock Cycle Time}$

$\text{Increase in CPU Time} = 1 - \text{New CPU Time} / \text{Old CPU Time}$

$\text{Instruction Count} = 2.63 * 10^{12} \text{ Instructions}$

1.11.4 [5] <§1.6, 1.9> 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%.

$\text{CPU Time} = \text{Instruction Count} * \text{CPI} * \text{Clock Cycle Time}$

$\text{Increase in CPU Time} = 1 - \text{New CPU Time} / \text{Old CPU Time}$

$\text{Instruction Count} = 2.63 * 10^{12} \text{ Instructions}$

$\text{CPI} = 0.99 \text{ Cycles / Instruction}$

$\text{CPU Time} = 866.25$

1.11.5 [5] <§1.6, 1.9> Find the change in the SPECratio for this change.

$\text{SPECratio} = 11.14$

1.11.6 [10] <§1.6> 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.

$\text{Instruction Count} = 2.03 * 10^{12} \text{ Instructions}$

$\text{CPI} = 1.04 \text{ Cycles / Instruction}$

1.11.7 [10] <§1.6> 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?

They are dissimilar because the number of instructions has been altered.

1.11.8 [5] <§1.6> By how much has the CPU time been reduced?

The CPU Time has been reduced by 6.67%.

1.11.9 [10] <\$1.6> 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 the CPI and with a clock rate of 4 GHz, determine the number of instructions.

Original # of Instructions = 1.79×10^3 Instructions

New # of Instructions = 2.15×10^3 Instructions

1.11.10 [10] <\$1.6> 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.

Clock Rate = CPI * # of Instructions / Execution Time * (1 - % Speed Up)

Clock Rate = 4.50×10^9 Cycles/Second

1.11.11 [10] <\$1.6> Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.

Clock Rate = CPI * # of Instructions / Execution Time * (1 - % Speed Up)

Clock Rate = 3.83×10^9 Cycles/Second