CPE 431/531 Technology Trends and Performance Fall 2022

**The University of Alabama in Huntsville**

**ECE Department**

**CPE 431 01, CPE 531 01/01R Fall 2022**

**Technology Trends and Performance Due August 30, 2022**

***You must show your work to get full credit. The number in parentheses is the point value of the problem.***

**1.0(5), 2.0(20), 3.0.0(5), 3.0.1(10), 4.0.0(5), 4.0.1(10), 4.0.1(10), 5.0(5), 6.0.0(10), 6.0.1(5), 7.0(5)**

**\*Answers provided in bold by each problem in case its hard to see in picture (except #7 since it can be seen easily in picture). Work provided below problems starting on Page 3.**

**1.0** The seven great ideas in computer architecture are similar to ideas from other fields. Match the seven ideas from computer architecture, “Use Abstraction to Simplify Design”, “Make the Common Case Fast”, “Performance via Parallelism”, “Performance via Pipelining”, “Performance via Prediction”, “Hierarchy of Memories”, and “Dependability via Redundancy” to the following ideas from other fields:

* + 1. Assembly lines in automobile manufacturing **Performance via Pipelining**
    2. Suspension bridge cables **Dependability via Redundancy**
    3. Aircraft and marine navigation systems that incorporate wind information **Performance via Prediction**
    4. Express elevators in buildings **Common Case Fast**
    5. Library reserve desk **Hierarchy of Memories**
    6. Increasing the gate area on a CMOS transistor to decrease its switching time **Performance via Parallelism**
    7. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into the base vehicle, such as lane departure systems and smart cruise control systems **Use Abstraction to simplify design**

**2.0** Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes of instructions according to their CPI (class A, B, C, and D). P1 with a clock rate of 3.3 GHz and CPIS of 1, 2, 5, and 3, and P2 with a clock rate of 3.1 GHz and CPIS of 1, 4, 2, and 3.

Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 15% class A, 25% class B, 40% class C, and 20% class D:

a. What is the global CPI for each implementation? **P1: 3.25, P2: 2.55**

* + 1. Find the clock cycles required in both cases. **P1: 3250000, P2: 2550000**
    2. Which implementation is faster and by how much? **P1 is faster by 1.275**

**3.0**  Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 8, and 4, 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 3 GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic instructions per processor is divided by 0.75 x p (where p is the number of processors) but the number of branch and load/store instructions per processor remains the same.

Page 1 of 2

CPE 431/531 Technology Trends and Performance Fall 2022

**3.0.0** 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. **See work provided**

**3.0.1** 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 value? **CPI should be 6.66 to match the performance of 4 processors.**

**4.0** 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 3.4 GHz, average CPI of 1.7, and requires the execution of 1.8 E11 instructions. P2 has a clock rate of 2.9 GHz, an average CPI of 0.8, and requires the execution of 0.7 E11 instructions.

**4.0.0** One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2. **The fallacy checks out because P1 has the higher CR but a slower ET than P2 by 0.215.**

**4.0.1** 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. **1.8125 E9 instructions**

**4.0.2** 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. **P2 has a higher MIPS but we know that P1 has a faster ET. So the fallacy checks out.**

**5.0**  Another pitfall cited in Section 1.11 is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 320 s, with 70 s spent executing FP instructions, 85 s executing L/S instructions, and 40 s spent executing branch instructions. Can the total time be reduced by 20 % by reducing only the time for FP instructions? **No you can’t reduce total time by 20% only reducing FP instructions as that only improves the total time by 4.4%. However, if the FP execution time was reduced to 6s that would give a 20% reduction in the overall performance.**

**6.0** Assume a program requires the execution of 50 E6 FP instructions, 110 E6 INT instructions, 80 E6 L/S instructions, and 16 E6 branch instructions. The CPI for each type of instruction is 10, 2, 3, and 5, respectively. Assume that the processor has a 6 GHz clock rate.

**6.0.0** By how much must we improve the CPI of L/S instructions if we want the program to run two times faster? **Not possible**

**6.0.1** By how much is the execution time improved if the CPI of INT and FP instructions is reduced by 45% and the CPI of L/S and Branch is reduced by 35%?  **1.72 time faster**

**7.0** When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.

Assume a program requires t = 250 s of execution time on one processor. When run on p processors, each processor requires t/p s, as well as an additional 5 s of overhead, irrespective of the number of processors. Compute the per-processor execution time for 2, 4, 8, 16, 32, 64, and 128 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead). **See work provided**

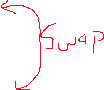
Page 2 of 2

## Work

Sorry I write so light so some things may be hard to see.

A picture containing text

Description automatically generated



A picture containing text

Description automatically generated

Text

Description automatically generated with medium confidence



Text

Description automatically generated with medium confidence

Table

Description automatically generated