Name: ____________________ ID: ____________

COMP5201 Assignment 1 Fall 2023

Computer Organization & Design Prof. D.K. Probst
Issued: September 12, 2023 Due: September 28, 2023
Submit electronically to Moodle. No extension will be granted.

1. [20 marks] Amdahl's Law

i) On a serial computer, portion A of program Q takes 13% of the time, while portion B takes 87% of the time. On a parallel computer, portion A speeds up by a factor of 3.75, while portion B speeds up by the number of processors P.

a) What is the minimum number of processors P to achieve at least 1/4 of the theoretical maximum speedup on program Q?

b) What is the minimum number of processors P to achieve at least 4/5 of the theoretical maximum speedup on program Q?

ii) On a serial computer, portion A of program Q takes integral 'x' s of time, while portion B takes '100 - x' s of time. On a parallel computer, portion A speeds up by a factor of 1.57, while portion B speeds up by the number of processors P. Program Q is run on P = 105 processors and then on P = 315 processors. The difference in run time is approximately 0.495238095 s.

What are the values of 'x' and 'y = 100 - x''?

2. [20 marks] Power and Performance.

A processor die has enough room for 50 processor cores. All cores have the same fixed square size. The die lies on top of a cooling pad that can extract 46 W of heat. The cooling pad must be able to extract _all_ the power dissipated on the die.

a) 'H' cores have a peak performance of 7 TFs/s and dissipate 1.5 W of power. How many 'H' cores can be put on (i.e., powered on) the processor die without overwhelming the cooling pad? What is the peak performance of this core configuration?

b) 'L' cores have a peak performance of 5 TFs/s and dissipate 0.9 W of power. How many 'L' cores can be put on (i.e., powered on) the processor die without overwhelming the cooling pad? What is the peak performance of this core configuration?

3. [20 marks] RISC-V Assembly Language.

Array 'a' contains 1,024 8-byte floating-point numbers. Write a RISC-V assembly-language program that, proceeding linearly through the array, inverts groups of 'a' elements four at a time. Thus, in an array of size 8, 12345678 becomes 43218765. You may not use auxiliary memory, only array 'a', two 'r' registers, and some number of 'f' registers. You must initialize the two 'r' registers. Array 'a' starts at memory address 0. Use only the following instructions. The examples have particular register names in them, but that is only to show syntax. For example, 'f2', 'f4', 'f6', etc., are equally acceptable. In this question, the integer 'n' must be nonnegative.

Hint: Program inverting four adjacent array elements as straight-line code.

```
set  r1,n        // set 'r1' to integer 'n'
fld  f0,n(r1)    // load to 'f0' eight bytes starting at Mem[r1+n]
fsd  f0,n(r1)    // store 'f0's eight bytes to Mem starting at Mem[r1+n]
addi r1,r1,n     // add integer 'n' to 'r1'
bne  r1,r2,loop  // if 'r1 /= r2' then goto 'loop'
```

4. [20 marks] Latency Challenge

A RISC-1.0 processor is attached to some memory and to storage. The clock rate is 4 GHz. When it is not waiting for data, the processor completes one floating-point operation in every processor cycle; this is the processor's peak performance. When a processor _logically_ blocks waiting for a data request to be completed, it has two options. It can either i) twiddle its thumbs by executing an idle loop until the request operation completes, or ii) context switch to another thread that, in this question, never blocks. The total cost for the two context switches is 5 us.

In the examples below, assume that program execution of the original thread consists of two floating-point operations followed by one data request, repeated indefinitely.

a) The cost of a disk access is 7 ms. When faced with a data request from disk, which option should the processor choose? Explain with numbers.

b) The cost of a DRAM memory access is 100 ns. When faced with a data request from DRAM, which option should the processor choose? Explain with numbers.

5. [20 marks] Arithmetic Intensity.

Every computer has a peak floating-point _processor_ performance and a peak memory bandwidth. Every program has an arithmetic intensity (ai), measured in flops/byte, which allows you to calculate the peak computer performance for a given computer and a given program. The formula is:

  peak computer performance = min{ai * peak b/w, peak processor performance}

a) Computer X1 has a peak floating-point processor performance of 289 TFs/s, and a peak memory bandwidth of 2,000 TBs/s. What is the minimum 'ai' in a program running on X1 so that the peak computer performance equals the peak processor performance?

b) Computer X2 has a peak floating-point processor performance of 357 TFs/s, and a peak memory bandwidth of 1,500 TBs/s. What is the minimum 'ai' in a program running on X2 so that the peak computer performance equals the peak processor performance?

c) Suppose a program has an arithmetic intensity of 0.13 Fs/B. Will it run faster on X1 or on X2? State the resulting two computer performances.

d) Suppose a program has an arithmetic intensity of 0.21 Fs/B. Will it run faster on X1 or on X2? State the resulting two computer performances.

--