CS151B/EE116C - Homework #6, Due 5/14/2018

Reading Assignment:

- Chapter 6, of the **4th Edition**: pp. 570-572, 575-595 (input-output).
  Note: This is from the **Fourth** Edition of the Patterson & Hennessy book. This chapter is available in the "Supplementary Readings" part of the class web site.
- Chapter 7, in the textbook by Hamacher, et al.: pp. 228-229.
  Note: This is available in the "Supplementary Readings" on the class website, under "I/O Bus Structure and PCI Express."
- Chapter 5: pp. 381-383 ("Disk Memory").
- Chapter 4: pp. 325-332 ("implementing exceptions in the pipelined MIPS").
- Appendix A: pp. A-33 – A-37 (Exceptions and Interrupts on MIPS).
- We will use some material from chapter 8 of the 2nd edition of the Patterson and Hennessy book. We will refer to this as *Ed2-Chap8*. This material is available on the class web page in the "Supplementary Readings" section.
  - Ed2-Chap8: pp. 646-647 ("Magnetic Disks")
  - Ed2-Chap8: 656-659 (ignore figures 8.7 and 8.8)
  - The examples in Ed2-Chap8: pp. 676-682
  - Ed2-Chap8: pp. 662-673

Problems:

(1) A disk has an average seek time of 4ms and rotates at 5,400 RPM. There are 1100 sectors per track. How long (in seconds) does it take to read the data from three consecutive sectors on the same track? Explain your answer.

(2) Consider a system where all I/O is done using programmed I/O. The I/O device, CPU, and memory are connected to the same data bus. The data bus is 64 bits wide. It takes 12 nanoseconds for one transfer across the bus. What is the maximum possible bandwidth (in bytes per seconds) at which data can be transferred from the I/O device to memory? Explain your answer!

(3) A program repeatedly performs a three-step process: it reads in a 4 KB block of data from disk, does some processing on that data, and then writes out the result as another 4 KB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 12,000 RPM, has an average seek time of 4.5 ms, and has a transfer rate of 80 MB/sec. The controller overhead is 0.3 ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 20 million clock cycles, and the clock rate is 5 GHz. What is the overall speed of the system in blocks processed per second?

(4) An important advantage of interrupts over polling is the ability of the processor to perform other tasks while waiting for communication from an I/O device. Suppose that a 1.8 GHz processor needs to read 1000 bytes of data from a particular I/O device. The I/O device supplies 1 byte of data every 0.02 ms. The code to process the data and store it in a buffer takes 1000 cycles.

A) If the processor detects that a byte of data is ready through polling, and a polling iteration takes 60 cycles, how many cycles does the entire operation take?

B) If instead, the processor is interrupted when a byte is ready, and the processor spends the time between interrupts on another task, how many cycles of this other task can the processor complete while the I/O communication is taking place? The overhead for handling an interrupt is 200 cycles.

Clearly state your assumptions.

(5) A pipelined MIPS implementation with support for overflow exceptions is described in the book, pp. 327-331. Consider the following claim: The description in the book is incomplete since it fails to modify the operation of the *Forwarding unit* when an exception occurs.

A) Is the claim above correct? Your answer must be **Yes** or **No**.

B) Explain your answer to Part A.

(6) Consider the pipelined MIPS implementation shown in slides 9.30-9.32 and explained in the book, pp. 327-331. Your task is to enhance this implementation to add support for the illegal opcode exception. As described on page 327, the *Cause* register must be modified based on whether the exception is an illegal opcode (value 10) or an overflow (value 12). Your implementation must work correctly even for the case where one instruction causes an overflow while the very next instruction contains an illegal opcode.

A) Provide a list of your modifications, where each one of the modifications is described in 1-2 clear sentences.

B) The existing datapath implementation is shown on slide 9.30 (available in the *Useful Figures* section of the course website). Are any modification to this figure required? If so, you can show them on a copy of the figure from slide 9.30. If there isn't enough room on the figure to show the modifications, just indicate the location of the modifications on the figure and show the details separately.

C) Are any new control signals required? If so, list them with an explanation and identify them on the datapath diagram.

D) Changes are required to the main *Control* circuit. Show those changes using a table similar to the one shown on slide 9.32. The contents of the table you provide must reflect correct operation when there are no exceptions, when there is an overflow exception, when there is an illegal opcode exception, and when overflow and illegal opcode exceptions are detected simultaneously.

Practice problems: You do <u>not</u> need to hand in a solution to the problems below.

(7) Consider the factors that affect the performance of a disk. All other parameters (including total disk capacity) being equal, would you expect better performance from a disk with fewer platters or more platters. Explain your answer.

(8) The rotational speed of a disk is doubled. How does that affect the access time of the disk? Your answer must be as detailed and quantitative as possible.

(9) Consider a disk with the following parameters: 512-byte sectors, rotation speed of 10,000 RPM, average seek time of 6 ms, transfer rate of 50 MB/s, and controller overhead of 0.2 ms. The system described in Ed2-Chap8, page 665 is used to handle disk accesses. If the I/O is allowed to consume 100% of the bus and memory bandwidth, what is the maximum number of simultaneous disk transfers that can be sustained for the two block sizes?

(10) Consider the MIPS implementation shown in Figure 4.66 (page 329) in the book. Due to a fault in the main *Control* circuit, whenever the input to the circuit is the opcode of `sw` (i.e., `101011`), regardless of any other input, the `EX.Flush` control output is asserted.
The *Control* circuit and the rest of the processor operate normally in every other respect.

Explain **in full detail** what will be the consequences of this fault when the processor executes programs — how will it change the behavior of the processor **as observed by a user/programmer** who does not know and does not care how the processor is implemented internally? Be sure to clearly identify **each and every** consequence of this fault.