CS151B/EE116C - Homework #7, Due 5/21/2018

Reading Assignment:

- *Readings from previous homeworks — Appendix A: pp. A-33 – A-37, Ed4-Chap6: 582-595, Ed2-Chap8: 656-659, 662-673, 676-682*
- Appendix B: B-58 – B-65
- Chapter 5: pp. 374-383
- Preparation for next week's material: Chapter 5: pp. 383-418

Problems:

(1) In the system described in Ed2-Chap8 pp. 665-666, the memory system takes 200 ns to read the first four words, and each additional four words require 20 ns. The memory system is modified so that it takes 120 ns to read the first four words and 15 ns to read each additional four words, find the sustained bandwidth and the latency for a read of 256 words for transfers that use 4-word blocks and for transfers that use 16-word blocks. Also compute the effective number of bus transactions per second for each case.

(2) Is there any situation where interrupt-driven I/O is preferable to DMA-based I/O? Explain briefly.

(3) Design a ''fair'' bus arbiter. There are four devices connected to the bus. The arbiter will have four *''bus request''* lines as inputs. The outputs of the arbiter are four *''bus grant''* lines. When a device wants the bus, it asserts its bus request line. A device gets the bus when its bus grant line is asserted by the arbiter. A device that gets the bus uses it for one cycle. The arbitration for any particular bus cycle occurs during the previous cycle. A device that is currently using the bus *can* contend for use of the bus during the next cycle. At most one device can use the bus during one cycle. Any number of devices (zero to four) can contend for the bus during every cycle.

The arbiter enforces ''fairness'' among the the devices by dynamically changing the relative priority of the devices. Denote the devices: D_0, D_1, D_2, and D_3. When the system is initialized, D_0 has top priority, then D_1 and D_2 with D_3 having the lowest priority. Note that even the lowest priority device may get the bus if none of the other devices request it. When some device D_i ($0 \le i \le 3$) gets the bus, the top priority is changed to device D_j with $j = (i+1) mod 4$. The priority of devices is then $D_{(i+1) mod 4}$, $D_{(i+2) mod 4}$, $D_{(i+3) mod 4}$, and $D_{(i+4) mod 4}$.

A) Design the arbiter. Your design should be at the level of gates and flip-flops. Show your work (truth tables etc.).

B) Show a timing diagram that includes the clock, the bus request lines, the bus grant lines, and the bus data lines for five cycles under the following conditions:

> The system is initialized before the first cycle. During the first cycle D_1 and D_3 request the bus. During the second cycle D_2 requests the bus.

(4) Recall that the MIPS ISA is big endian. Consider the pipelined MIPS implementation shown in Figure 4.60, page 316, with the enhancement for supporting `lw` and `sw` shown in Figure 4.57, page 312. In this implementation, the box labeled ''Data memory'' is a $2^{30} \times 32$ memory — a 32-bit word can be either read from or written to the memory in every access. Hence, the address input to this memory consists of just 30 bits — the 30 most-significant bits of the effective address computed by the ALU.

Your task is to add to this implementation support for the `sb` (store byte) instruction, documented on page A-68 in the book and page 2.22 in the class notes. You **cannot** modify the implementation of the data memory in **any way**. Be sure to consider any interactions between the new `sb` instruction and the already supported instructions that must, of course, continue to work.

Hint: In the existing implementation, for the `sw` instruction no useful work is done in the fifth pipeline stage. For the `sb` instruction, your implementation will require critical work to be performed in the fifth pipeline stage. Obviously, those operations will **not** involve writing anything to the register file.

A) Explain your modifications in 3-5 clear sentences.

B) On a copy of Figure 4.60, show the modifications. If there isn't enough room on the figure to show the modifications, just indicate the locations of the modifications on the figure and show the details separately. If you add any new modules, you must show their implementation in detail. However, you do not need to show the implementation of any new simple MUXes or registers. Clearly state any assumptions you make.

C) List all the required new control signals with an explanation and identify them on the datapath figure.

D) Changes are required to the main *Control* circuit. Show those changes using a table similar to the one shown on slide 7.30 in the class notes.

(5) Consider a $4M \times 1$ DRAM chip. The notation used below is: read 100 is a read operation that reads one bit from address 100, write 350 is a write operation that writes one bit from address 350. One of two possible sequences of operations are performed:

A) read 750, read 6550, read 155550 **B)** write 750, write 6550, write 155550

Which sequence of operations (A or B) can be completed faster? Explain your answer.

(6) You have $2M \times 8$ DRAM chips (as many as you want). You must use these chips to build the data memory for the single-cycle MIPS processor described in chapter 4 of the textbook. What is the minimum size memory you can build? Specify the size in **bits**. Explain your answer.

Practice problems: You do not need to hand in a solution to the problems below.

(7) Here are a variety of building blocks used in an I/O system that has a synchronous processor-memory bus running at 800 MHz and one or more I/O adapters that interface I/O buses to the processor-memory bus.

- *Memory system:* The memory system has a 32-bit interface and handles four-word transfers. The memory system has separate address and data lines and, for writes to memory, accepts a word every clock cycle for 4 clock cycles and then takes an additional 4 clock cycles before the words have been stored and it can accept another transaction.
- *DMA interfaces:* The I/O adapters use DMA to transfer the data between the I/O buses and the processor-memory bus. The DMA unit arbitrates for the processor-memory bus and sends/receives four-word blocks from/to the memory system. The DMA controller can accommodate up to eight disks. Initiating a new I/O operation (including the seek and access) takes 0.1 ms, during which another I/O cannot be initiated by this controller (but outstanding operations can be handled).
- *I/O bus:* The I/O bus is a synchronous bus with a sustainable bandwidth of 100 MB/sec; each transfer is one word long.
- *Disks:* The disks have a measured average seek plus rotational latency of 8 ms. The disks have a read/write bandwidth of 40 MB/sec, when they are transferring.

Find the time required to read a 16 KB sector from a disk to memory, assuming that this is the only activity on the bus.

(8) Recall that the MIPS ISA is big endian. Consider the pipelined MIPS implementation shown in Figure 4.60, page 316, with the enhancement for supporting `lw` and `sw` shown in Figure 4.57, page 312. In this implementation, the box labeled "Data memory" is a $2^{30} \times 32$ memory — a 32-bit word can be either read from or written to the memory in every access. Hence, the address input to this memory consists of just 30

bits — the 30 most-significant bits of the effective address computed by the ALU.

Your task is to add to this implementation support for the lb (load byte) instruction, documented on page A-66 in the book and page 2.22 in the class notes. You **cannot** modify the implementation of the data memory in **any way**. Be sure to consider any interactions between the new lb instruction and the already supported instructions that must, of course, continue to work.

A) Explain your modifications in 3-5 clear sentences.

B) On a copy of Figure 4.60, show the modifications. If there isn't enough room on the figure to show the modifications, just indicate the locations of the modifications on the figure and show the details separately. If you add any new modules, you must show their implementation in detail. However, you do not need to show the implementation of any new simple MUXes or registers. Clearly state any assumptions you make.

C) List all the required new control signals with an explanation and identify them on the datapath figure.

D) Changes are required to the main *Control* circuit. Show those changes using a table similar to the one shown on slide 7.30 in the class notes.