CS 250 Spring 2017 Homework 11 SOLUTION and Grading Guide

GTAs: Please enter scores in Blackboard by April 24, 2017.

Maximum score = 15

1. **[5 pts. total]** A serial interface operates at a throughput of 10 million bits per second and requires 100 microseconds to configure (prepare) a packet of bits to be transmitted regardless of the size of the packet.
   a. **[4 pts., 1 pt. each for highlighted answers]** What is the effective throughput, expressed first in packets/second and then in bits/second, for this interface for an infinite series of identically-sized packets for sizes 1 bit, 100 bits, 10^4 bits, and 10^6 bits? Answer: Raw throughput is $1x10^7$ bits/second and packet latency is 100×10^{-6} seconds.
      Total time per packet = Packet_latency + Bits_per_packet / Raw_throughput
      = 100×10^{-4} sec + Bits_per_packet / 1×10^{-7} bits/sec
      The following table shows the calculation for each packet size.

Bits/packet	Time (sec/packet)	Throughput (packets/sec)	Throughput (bits/sec)
1	$10^{-4} + 10^{-7} = 0.0001001$	**9,990.01**	**9,990.01**
100	$10^{-4} + 10^{-5} = 0.00011$	9,090.91	909,091
10^4	$10^{-4} + 10^{-3} = 0.0011$	**909.091**	**9,090,910**
10^6	$10^{-4} + 10^{-1} = 0.1001$	9.99001	9,990,010

   b. **[1 pt.]** What is being amortized?
      Answer: **Time to configure a packet,** *which might also be viewed as the latency of this interface.* The table shows that the throughput measured in bits per second shifts dramatically higher as the size of the packet increases. This reflects amortization of the time to construct a packet across the bits contained in the packet. Because this packet configuration time is independent of packet size, we can reduce the latency cost per bit significantly by making packets significantly larger.

2. How many simultaneous transfers can occur over a crossbar switching fabric of N inputs and M outputs?
   Answer: minimum(N, M).

3. **[2 pts.]** Assume that a RISC processor takes two microseconds to execute each instruction and an I/O device can wait at most 1 millisecond before its interrupt is serviced. What is the maximum number of instructions that can be executed with interrupts disabled?
   Answer: Applying the CPU time equation, we have
      CPU Time = Instruction/Program x Clocks_per_instruction X clock_cycle_time
   which for this question and its given information reduces to
      1 millisecond = Max_instructions_with_interrupts_disabled x (1.0) x $2x10^{-6}$ seconds
   thus,
      Max_instructions_with_interrupts_disabled = (10^{-3} sec) / (1.0)x $2x10^{-6}$ sec = **500**.

4. **[2 pts., 1 pt. each answer]** Suppose a user installs ten devices that all perform DMA into a single computer and attempts to operate the devices simultaneously. What components of the computer might become a bottleneck?
   Answer: These ten DMA devices all connect to the computer's memory bus alongside the computer's memory module(s). Each of the ten devices needs to use the memory bus, which, if the devices are fast, may give the bus a workload at or beyond its maximum capacity to transfer bits. So the **memory bus may become a bottleneck**. Also, each DMA device, by definition, will try to move information in/out of the memory module(s). The throughput of the memory module(s) may be exceeded by the combined throughput capability of these ten DMA devices, **thus making the memory module(s) another potential bottleneck**.

5. A user invokes an app that writes a file. The app displays a progress bar that shows how much of the file has been written. Just as the progress bar reaches 50%, the power fails and the computer crashes. When the power is restored and the computer rebooted, the user discovers that less than 20% of the file was actually written. Why did the app report 50%?
   Answer: The progress bar was showing the amount of the data that the upper half of the device driver had placed into the shared data area. The actual amount of the file written is the amount removed from the shared area by the lower half, which sends data to the file storage device.

6. A user invokes an app that writes a file. The app displays a progress bar that shows how much of the file has been written. The progress bar has been advancing quickly, but just as the progress bar reaches 99%, the progress bar seems to freeze, then after a delay the bar show 100%, then disappears, and the app GUI moves on to another phase of activity. What might explain why the progress bar seemed to freeze at 99% for a time before reporting 100%?
   Answer: As in Question 5, the progress bar was showing the amount of data written into the shard data area of the driver by the upper half. However, the progress bar programmer decided to design the progress bar to stop upon reaching 99% written into shared space and wait until receiving a done signal from the lower half of the device driver before reporting 100%, or done, to the app user.

7. Discuss the advantages of multiplexing with respect to buses. What are the two major disadvantages? **[1pt. for identifying complexity of bus controller hardware as being minimized due to effect of Moore's Law]** Which of these two disadvantages is minimized, even rendered moot, by Moore's Law?
   Answer: The advantages include use of fewer signal lines, and if the total number of lines is fixed, multiplexing can increase performance because no one line need be dedicated to one purpose: all lines can be used to best advantage at any time by the designer.
   The two disadvantages are that a *store* operation takes two bus cycles instead of just one cycle, and **multiplexing requires a more complex bus protocol, which increases the complexity of the bus interface hardware. Moore's Law drives down the cost of hardware, so the second disadvantage becomes less important with time,** eventually becoming of no practical consequence, or moot.

8. **[2 pts., 1pt. each answer]** What are the two types of bus error?
   Answer: **Address conflict** and an **unassigned address**. See our text, page 298.

9. **[3 pts., 1 pt. per bus with allowance for making a good case that there are fewer than 3 buses in the student's computer, meaning a 2-bus answer can earn 3 pts.]**
   Run a system info command on your computer, and use its output to find at least three different buses that your computer contains. For each bus, (1) describe in specific detail the physical hardware from which it would be constructed, (2) name the units within your computer that are connected to the bus (if any) and the external units (if any) typically connected to the bus, (3) classify the bus as serial or parallel, (4) state if the bus is proprietary or if is it standard and when was its standard specification released and what entity released the spec, and (5) state how the bus addresses are configured (manually per I/O device, by hardwiring on the bus, or automatically), and (6) state for automatically configured buses, state whether the bus is hot-pluggable or not. The web, especially Wikipedia, can be a helpful resource.

   Answer: For my laptop: mid-2012 11-inch MacBook Air. "Your mileage may vary."

   Memory bus: (1) copper wires within the circuit board of the motherboard, (2) connecting the processor to the DIMM modules of the DRAM memory and to the I/O controller(s) with no external units typically, (3) typically a wide parallel bus, (4) a proprietary/standard mix, (5) with addresses configured by bus wiring, and (6) not/applicable.

   USB bus: (1) 4 copper wires from the USB hub on the motherboard (2) to the internal USB socket(s) in the side of the laptop and to built-in USB devices, such as the keyboard, trackpad, and Bluetooth, plus connecting the internal USB hub to the various external USB peripherals, such as flash drives, etc., (3) a serial bus, (4) of standard configuration defined for USB2 April 2000 released by the USB Implementers Forum, (5) with bus addresses configured automatically, and (6) this bus is hot-pluggable.

   PCI bus: (1) copper wires on the motherboard, (2) connecting the processor to internal Ethernet and Thunderbolt I/O ports, (3) a parallel bus, (4) of standard specification defined in 1992 by Intel, (5) with addresses typically configured manually, but (6) also optionally hot-pluggable.