UNIVERSITY OF ALBERTA

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CMPE 401 – Computer Interfacing

Final Examination

Instructor: B. F. Cockburn
Exam date: December 11, 2009
Exam duration: 120 minutes
Aids permitted: A hardcopy of the course overheads can be freely consulted.
Model solutions for the Fall 2009 assignments and midterm can be consulted.
No other model solutions are to be consulted.
Electronic calculators (all kinds) are permitted.

Instructions:
1. Enter your printed name, signature and I.D. number on this cover page.
2. Verify that this booklet contains 10 pages (including this cover page).
3. Neatly enter your answers in the spaces provided.
4. Use the reverse sides of the pages for extra space or rough work.

Student name: _____,_____
Last name **First name**

Signature: _____

Student I.D.: _____

Question	Time	Worth	Mark	Subject
1.	15	12		Fundamental Concepts
2.	30	25		Task Priorities
3.	15	12		Interfacing to Ethernet
4.	10	9		Client-Server Communication
5.	20	18		Buffering and Flow Control
6.	15	12		Improved Half-Stepping Waveforms
7.	15	12		Semisynchronous PCI Bus
Total	**120 mins**	**100**		---

Question #1 (Fundamental Concepts) [12 marks]

In your own words briefly define each of the following concepts. Be sure to explain why each concept is important in Computer Interfacing.

(a) Big Endian byte order

[2 marks] Big Endian byte order refers to the byte address ordering that is used when, in each memory word, the byte addresses increase going from the most significant byte position to the least significant byte position. So, if a 32-bit word is at address A, the byte addresses (from the most to the least significant byte) are A, A+1, A+2 and A+3.

[2 marks] Big Endian byte order is one of two possible byte address orderings that are used in industry. Thus it is important to know (when solving some computer interfacing problems) which ordering is being used by a given processor. IBM, Freescale, HP and Sun Microsystems are notable companies that use the Big Endian byte address order in their processors.

(b) Counting semaphore

[3 marks] A counting semaphore is a data structure that includes an integer count and a queue of blocked tasks. Typically the count is initialized to the number of available shared resources of a particular type. As resources are allocated to requesting tasks, the count is decremented. After all resources have been allocated, further requests for the resources cause the requesting task to be blocked and moved to the queue, and the count is further decremented to negative values. When a resource is released by a task and no tasks are on the queue, then the count is simply incremented. If there are tasks waiting in the queue, then the highest priority task is taken off the queue, allocated the resource, and placed on the ready-to-run queue.

[1 mark] Counting semaphores are useful in computer interfacing for managing shared resources (e.g., access to a pool of external interfaces) in a multitasking system.

(c) Two-address DMA

[2 marks] Two-address direct memory access (DMA) is a type of DMA where data is between transferred from one region in memory to a second region in memory. In DMA a DMA controller generates the bus signals that transfer the bytes so that many MOVE instructions do not have to be executed on the CPU. The DMA controller uses a read address pointer for the first region and a write address pointer for the second region. Both pointers automatically increment as the bytes are transferred.

[2 marks] Two-address DMA is the fastest way of transferring data from one memory buffer to a second memory buffer. It is a widely used technique in computer software. Some microprocessors (e.g. the MCF5234) have DMA controllers with built-in support for two address DMA.

Question #2 (Task Priorities) [25 marks]

(a) In pre-emptive multitasking kernels, each task is assigned a priority. The task priorities are used by the kernel to decide which task should be allowed to run on the CPU at any given time. What would be some important considerations when assigning priorities to tasks? Which tasks should be assigned the higher priorities, and which tasks should be assigned the lower priorities? Justify your recommendation.

[8 marks] Two important considerations are:
(1) How soon a task's deadline is for getting its work completed. If the deadline is soon, then the task's priority should be greater. This way each system deadline is more likely to be met.

(2) How frequently a task is scheduled to run. A task that must run more frequently should have a higher priority. A frequently running task is more likely to be more sensitive to the times when it is scheduled to run because delays in scheduling will be larger relative to the interval between runs.

(b) In some pre-emptive multitasking kernels, the priority that is assigned to each task when the task is created is fixed; in other kernels, task priorities can be changed. Give at least two good reasons for having the ability to change the priority of a task. What are at least two disadvantages to allowing task priorities to be changed?

[4 marks] Two good reasons for wanting to change the priority of a task are: (1) the task's deadline may change significantly at different stages of its existence, and (2) the frequency at which a task is running may change significantly during its existence.

[4 marks] There are disadvantages to changing task priorities. When task priorities are changing over time, it becomes even more difficult to predict system behaviour. Also, it becomes important to verify that any new priority that a task may be changed to is not already in use by another task. When a task lowers its own priority, it risks losing the CPU immediately, and this possibility must be taken in to account when designing the system.

Question #2 (Task Priorities, cont'd)

(c) One of the potential drawbacks of priorities is that the high priority tasks might prevent the low priority tasks from having enough execution time on the CPU. Assume that the tasks in a given system have been split into two groups: high-priority tasks with priorities that lie within one range of priority values, and low-priority tasks with priorities that lie within a second range of priority values. Briefly outline a way in which a strictly pre-emptive kernel (like MicroC/OS) could be modified to use timer interrupts to ensure that low-priority tasks always get some minimum fraction of the available execution time compared to the high-priority tasks.

[9 marks] A time-sharing scheme similar to that used in the eTPU could be used in a modified kernel. The timer interrupts would define the boundaries of two interleaved kinds of intervals of time. In one kind of interval, there would be pre-emptive scheduling of both the high and low-priority tasks; in the second kind of interval, there would pre-emptive scheduling of only the low-priority tasks (high-priority tasks would be allowed if no low-priority tasks were ready-to-run). By adjusting the relative amount of time between the two kinds of intervals, one could ensure that the low-priority tasks always get some appropriate minimum amount of time.

Question #3 (Interfacing to Ethernet) [12 marks]

(a) In the context of an Ethernet interface on a microcomputer, what is the PHY and what is the MAC? Why are these two subsystems often implemented as separate devices?

[3 marks] The PHY is the physical layer transceiver subsystem that translates between digital signals on the computer side and the analog signals that propagate on the physical medium. It performs necessary equalization and signal processing to ensure that digital bits are recovered in the receive direction, and that the correct analog pulses are sent in the transmit direction.

[3 marks] The MAC is the Medium Access subsystem that translates between data packets on the computer side and the digital bits on the PHY side. The MAC is responsible for operating the data link protocol that gains access to the physical medium, contacts the other party over the medium, and sends the bits of the packet to that party over the medium. The MAC typically adds extra bits to the data for packet framing and synchronization, physical layer addressing, and possibly error detection and correction.

(b) Ethernet local area networks (LANs) use a physical layer addressing scheme where each device on a LAN has a 6-byte (48-bit) address. However, when an Ethernet LAN is connected to the Internet, 4-byte (32-bit) IP addresses must be assigned to each node. Why do you suppose both of these addressing schemes are being used? Hint: What is the primary purpose of Ethernet physical addresses and Internet IP addresses?

[6 marks]
The main reason for the two addressing schemes is both historical and because the two schemes solve different problems.

The 6-byte Ethernet (or MAC) addresses were developed as part of that standard, before the Internet had appeared on a large scale. The Ethernet addresses are assigned by the manufacturer of each Ethernet device, and are supposed to be tied permanently to the hardware. The primary purpose is to uniquely identify the communications interface for use by Ethernet networking.

The 4-byte Internetworking Protocol (IP) addresses were developed as part of the TCP/IP protocol suite. The main priority for IP was to accommodate the differences between all the underlying data networking protocols, and so it needed its own addressing scheme: one that could be scaled up to a global Internet. The IP addresses are therefore assigned more on a geographical and organization-based manner as opposed to the details of the hardware interface (like MAC addresses).

Question #4 (Client-Server Communication) [9 marks]

(a) What is meant by "client-server" communication over the Internet?

[4 marks]
Client-server communication is a very common mode of communication over the Internet where one node (the server) stores information and waits for connections, and client nodes make connections to the server to access the information.

Application programming interfaces (APIs) have been developed to simplify the implementation of clients and servers on Unix systems and hosts running other operating systems (e.g., MicroC/OS).

(b) Briefly describe how could the client-server model, together with HTTP, be used for organizing a system that monitors the status and controls a distributed network of data loggers?

[5 marks]
The data loggers could be provided with webserver functionality. The webserver would implement dynamic webpages, encoded in HTML, that permit clients to log in and monitor the status of the data logger. Using HTML forms the remote clients could issue commands and data that control the operation of the data logger. Since the data logger would be accessible over the Internet (or even a private network), security would be required to ensure that only authorized users would be permitted to access the webserver. SHTTP, the secure version of HTTP, would be convenient to use since it has a built-in standard security mechanism.

Question #5 (Buffering and Flow Control) [18 marks]

(a) The flow of data in a computer system is typically managed using both data buffering (for example, FIFO queues or circular buffers) and flow control. Briefly explain the major purposes of data buffering and flow control. Be sure to explain the differences between these two mechanisms. Why are both mechanisms often required to handle the flow of data from a data source/producer to a data sink/consumer?

[8 marks]
Data buffering is the temporary storage of arriving data in either (1) a storage region in computer memory, or (2) a separate hardware memory device. Data buffering accommodates short-term fluctuations in the arrival rate of data. By accumulating a buffer of data, the next stage of data processing can more conveniently and efficiently access a single larger group of data items. The timing of the transmitter and receiver can be de-coupled.

Flow control is a mechanism where a transmitter of data is prevented from sending too much data and thus overwhelming the capacity of a receiver of data. For example, there is typically a signal that is sent back from the receiver to the transmitter to slow down or halt the transmitter's ability to send more data. There must also be a signal to cause the transmitter to speed up or restart its transmissions.

Data buffering accommodates short-term mismatches in the data rates of the transmitter and receiver. Flow control accommodates longer-term mismatches in the two data rates.

Both mechanisms are useful and help to solve different problems. Data buffering cannot overcome a longer-term mismatch in data rates because the finite buffers will fill up. Flow control is an inefficient way of handling small-term mismatches in data rates because the flow control mechanisms would have to be used more frequently than otherwise necessary.

Question #5 (Buffering and Flow Control, cont'd)

(b) What is the "window size" in window-based flow control? What would be some reasonable factors to consider when choosing the window size in a particular situation?

[4 marks]
The window size is the amount of data (e.g., number of data bytes) that can be sent by the transmitter without acknowledgement from the receiver. The window size allows flexibility to efficiently accommodate the travel time of the data and small processing delays along the communication path.

(c) One of the challenges of implementing the ENQ/ACK flow control method is to design a method of recovering from lost "enq" and "ack" characters. Propose a strategy that could be used to recover from this error situation.

[6 marks]
A lost "enq" character would prevent the transmitter from ever transmitting again. To fix the problem, the transmitter could start a timer every time that it sends an "enq". If the timer expires before an "ack" character is received back from the receiver, then the transmitter should assume that the "enq" was lost and should try sending again with a new time-out.

A lost "ack" character will also prevent transmitting. The simplest solution would be to let the transmitter discover and handle the problem. If a transmitted "enq" fails to produce an "ack" in return, then the "enq" timer will time-out, and a new "enq" will be sent. The transmitter should ignore duplicate "ack" characters that are received because of slow transmission delays.

Question #6 (Improved Half-Stepping Waveforms) [12 marks]

Assume that you are given a new kind of bridge circuit that receives three digital inputs (E, D & M) to control each of two stepper motor windings, identified as "a" and "b". Inputs "Ea" and "Eb" enable (for input 1) or disable (for input 0) the current flow in windings "a" and "b", respectively. Inputs "Da" and "Db" give the current direction, either forward (0) or reverse (1), for the two windings. Finally, inputs "Ma" and "Mb" give the magnitude, either 0.707 times full strength (0) or 1.0 times full strength (1), for the two windings. Construct a table that illustrates an "improved" half-step drive waveform sequence. Your table should have seven columns (labeled angle, Ea, Da, Ma, Eb, Db, Mb) and eight rows of data. In what ways would this new half-stepping drive waveform be improved over standard half-stepping?

Angle	Ea	Da	Ma	Eb	Db	Mb	Current strengths
1	1	0	0	1	0	0	reduced & reduced
2	1	0	1	0	X	X	full & off
3	1	0	0	1	1	0	reduced & reduced
4	0	X	X	1	1	1	off & full
5	1	1	0	1	1	0	reduced & reduced
6	1	1	1	0	X	X	full & off
7	1	1	0	1	0	0	reduced & reduced
8	0	X	X	1	0	1	off & full

The new drive waveforms should produce more uniform torque on the rotor. There will be two different nominal torque values as follows:
(Angles 2, 4, 6 & 8) One winding carries full current, while the second winding carries no current.
(Angles 1, 3, 5 & 7) Both windings carries 0.707 times full current.

The magnitude of the reduced current should be adjusted to ensure that the magnitude of their vector sum produces the same torque as the full magnitude current.

Question #7 (Semisynchronous PCI Bus) [12 marks]

(a) The Peripheral Communications Interconnection (PCI) Bus uses a semisynchronous bus, which includes a clock signal, and two active-low handshake signals: IRDY and TRDY. Briefly explain how these IRDY and TRDY signal are used to accommodate different peripheral response times in a read operation to a memory-mapped peripheral register.

[6 marks]
The target ready (TRDY) signal is asserted low by the slave peripheral device (or some other suitably configured address decoder) when the requested contents of the peripheral register have been fetched and are being driven as stable values on the address-data (AD) bus.

The initiator ready (IRDY) signal is asserted low by the reading master (e.g., the CPU) around rising clock edges when it is prepared to latch in the fetched data from the AD bus.

The new data transfer occurs at the next rising clock edge when both TRDY and IRDY are asserted low.

(b) The PCI Bus uses "Reflected Wave Switching", where the bus drivers produce half-height signals, and reflections at the end of the bus are used to double up the amplitude of the signals. What would be some of the major advantages and disadvantages of using this kind of scheme on a microcomputer bus?

[6 marks]
Major advantages include:
(1) The drivers require less power to drive half-height signals.

Major disadvantages include:
(1) Series termination is required at the output of all drivers.
(2) The PCI backplane must be more carefully designed to have good transmission line properties, with an even characteristic impedance.
(3) The signal propagation time is longer to all for the incident wavefront to propagate to the far end of the backplane, then reflect back along the whole transmission line.