UNIVERSITY OF ALBERTA

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CMPE 401 – Computer Interfacing

Midterm Examination

Instructor: B. F. Cockburn
Exam date: October 28, 2011
Exam duration: 50 minutes
Aids permitted: A hardcopy of the course overheads can be freely consulted.
Electronic calculators are permitted.
Model solutions from this year's assignments are permitted.

Instructions:
1. Please enter your printed name, signature and I.D. number on this page.
2. Verify that this booklet contains 6 pages (including the cover page).
3. Neatly enter your answers in the spaces provided.
4. Use the reverse sides of the pages for rough work.
5. Take into account the marks per question when budgeting your time.

Student name: _____**Model**_____**,**__**Solutions**_____
Family name **Given name**

Signature: _____

Student I.D.: _____

Question	Time	Worth	Mark	Subject
1.	8	15		Basic Concepts
2.	12	25		Microcomputer Hardware
3.	18	35		Microcomputer Software
4.	12	25		Computer Networking
Total	**50 mins**	**100**		**—**

Question #1 (Basic Concepts)

Briefly answer the following questions:

(a) What exactly is a "critical section"? Give two different situations where critical sections can arise in a microcontroller. What can happen if a critical section is not handled properly?

[3 marks] A critical section is a segment of program code that must be executed to completion by one task, without being interrupted by the execution of any other code.

[2 marks] One kind of critical section is an update operation to a data structure, like a linked list. The update requires two or more pointer update operations, which much be completed together to preserve the correctness of the data structure.

[2 marks] A second kind of critical section is an update operation to hardware registers where the update requires at least two different read or write operations, which must all be performed to (a) avoid stale information from reads, and/or (2) to ensure correct operation of the peripheral hardware after write operations to peripheral registers.

[3 marks] If a critical section for a data structure update is not handled properly, then the data structure could be left in inconsistent states, and the resulting incorrect information could be read by software and acted upon. If a critical section for a register update is not handled properly, then the hardware could be operated in an unintended and perhaps dangerous mode of operation.

(b) What is an "Integrated Development Environment" and how is an IDE supposed to increase engineering productivity?

[5 marks] An integrated development environment is a coordinated set of software development tools (e.g., compilers, linkers, debuggers, project management, source file version control) that are intend to assist software developer. The user interfaces of the various tools are made similar to minimize the mental effort that would otherwise be required to master different user interfaces for the different tools. The desired result is increased programmer productivity and higher quality software (software with fewer design and implementation defects).

Question #2 (Microcomputer Hardware)

(a) What is meant by a "memory-mapped interface"? What benefits are provided by memory mapping?

[4 marks] A memory-mapped interface is one that is controlled through registers that are given addresses in the microcomputer's memory map. The registers can be read and written like memory words (say using MOVE instructions); however, unlike for memory words, such operations memory-mapped registers can have side-effects on the interface hardware.

[4 marks] Memory-mapping allows interfaces to be controlled using existing memory word operations (reads, writes, read-modify-writes). No special instructions are required to control interfaces. The memory map can be re-organized as desired in different microcomputers, allowing different memory-mapped interfaces to be accommodated.

(b) What is a FIFO buffer and what is it used for? Give two examples of FIFOs that were mentioned in the lectures.

[6 marks] A First-In First-Out (FIFO) is a memory that temporarily holds data that is written into one data port, and will be read out later from a second data port. The order of the stored data is not modified when it passes through the FIFO. The FIFO allows short-term fluctuations in data writes and reads to be accommodated without the risk of data loss.

[2 marks] One FIFO buffer was the instruction queue that is present in the V2 ColdFire CPU. The queue is used to temporarily store machine language instructions between the circuitry that fetches instructions from memory, and the CPU circuitry that decodes and executes the instructions. The FIFO allows instructions to be "pre-fetched" to speed up the average execution time of software.

[2 marks] A second FIFO appears in the Fast Ethernet Controller (FEC) block of the MCF5234 microcontroller. It is used to hold data bytes that are (1) being transferred in the transmit direction from data buffers to the PHY hardware, and (2) being transferred in the receive direction from the PHY to data buffers.

(c) What is an Ethernet PHY and what is its main purpose in a microcomputer?

[7 marks] An Ethernet PHY (for physical layer) is a subsystem that interfaces the analog signals on a communication medium with the digital signals handled by the Medium Access Control (MAC) subsystem. Different PHYs can be used to interface the same MAC to different communication media (e.g., twisted pair, coaxial cable, optical fibre, wireless). The PHY takes care of all of the complexities of transmitting and receiving analog signals onto the communications medium. This simplifies the design of the MAC. Also, the PHY can be implemented on a different process than the MAC hardware. One PHY design can be made in great volumes and then used with many different MACs.

Question #3 (Microcomputer Software)

(a) In a pre-emptive multitasking environment, the highest-priority ready-to-run task always gets to run on the CPU. The kernel must enforce this rule when the currently running task gets blocked on some condition and when hardware interrupts occur. Briefly describe the checks that must be performed by the kernel to ensure that the priority rule is maintained at all times.

The priority rule must be enforced by having the kernel check every time some event happens that could cause a new task, other than the currently running task, to be the highest priority ready-to-run task.

[8 marks] When the currently running task gets blocked on some event (e.g., waiting for a semaphore to go to 1, or waiting for a message to arrive) or terminates or suspends itself, the kernel must save the context of the blocked task, and then select the next running task from the available ready-to-run tasks. (A lowest-priority idle task will always be ready-to-run.) The kernel must apply the priority rule by selecting the highest priority ready-to-run task to run on the CPU.

[8 marks] When an interrupt occurs, the act of servicing the interrupt may unblock one or more unblocked tasks (e.g., the interrupt signals the arrival of a message, and a task was blocked waiting for that message). These newly unblocked tasks may have higher priorities that than the currently running (but presently interrupted) task. The kernel must apply the priority rule after the interrupt has been serviced, but before the interrupted task is restarted on the CPU. The kernel must cause a context switch to the highest-priority ready-to-run task if that task is among the newly unblocked tasks.

Question #3 (Microcomputer Software, cont'd)

(b) Implementing a real-time embedded computer system using a multitasking software architecture has both advantages and disadvantages. Briefly describe three important advantages and five important challenges that arise when a software system is partitioned into interacting tasks.

Some of the major advantages: [9 marks for 3 advantages]

Advantage #1: Each task can have a simple structure that is dedicated to solving one aspect of the problem. Task design and verification is thus simpler.

Advantage #2: The tasks only interact loosely with each other, which further simplifies the design (following divide and conquer strategy).

Advantage #3: New tasks can be readily added to an existing system, so the system is more scalable.

Advantage #4: If priorities with preemption are used, then the kernel is responsible for enforcing the priorities, not the tasks. This helps to ensure that higher-priority events are dealt with appropriately compared to lower-priority events.

Advantage #5: If time-slicing is used, then the kernel is responsible for the fair sharing of the CPU's time.

Some of the major challenges: [10 marks for 5 challenges]

Challenge #1: It can be hard to understand and debug inter-task interactions that are causing problems.

Challenge #2: New phenomena, such as deadlock, can occur that must be handled properly.

Challenge #3: Memory space is required to store the kernel code that performs task context switching.

Challenge #4: Critical sections and shared resources must be protected properly.

Challenge #5: Task synchronization mechanisms must be provided.

Challenge #6: Time is wasted performing context switches.

Question #4 (Computer Networking)

(a) Computer communications interfaces are typically partitioned into layers. Briefly summarize the benefits of layering in this context. What do you think would be some of the major potential disadvantages of introducing too much layering into a communications interface?

[8 marks] Partitioning a communications interface into layers is the major strategy for managing design complexity. Each layer provides a simplified service to the next layer lying above. Low-level implementation details are thus hidden from the higher layers. Each layer interacts "horizonally" with its peer(s) at the same layer on other nodes. Each layer can be specified, designed, verified, and modified independently from the other layers.

[5 marks] Each new layer typically contributes additional cost in terms of processing time, software storage space, and data storage space. This cost is additional overhead that adds to the implementation cost and can reduce operating efficiency. If too much layering is introduced (or if the layered interface is implemented inefficiently) then these costs and inefficiencies can hurt the economic competitiveness of an implementation of an interface.

(b) The UDP and TCP transport layer protocols introduce port numbers at both ends of a connection over the Internet. What is the purpose of these port numbers? Which nodes along a route through a data network need to examine the port numbers?

[6 marks] The port numbers are used to associate TCP/IP and UDP/IP connections with applications that run on the networked nodes. For example, all calls to Port 80 on a node are routed to the HTTP server on that node.

[6 marks] Port numbers for the source and destination nodes are contained in the headers of the UDP and TCP segment. Such segments are carried in the payload portion of IP datagrams. Thus the port numbers are not considered and processed by the IP entities in nodes along the route from a source node to a destination node.