CMPE401
Computer Interfacing

Quiz #1

October 17, 2007

Name: _____SOLUTION_____
ID: _____

2 questions. 10 minutes.
Allowed material:

- Course notes
- Calculators
- Textbooks

Model solutions of midterms, finals, quizzes, and assignments are NOT allowed.
Answer the questions in the space provided.
Write clearly otherwise your answer will not be marked.

Marks:
Question 1: _____
Question 2: _____
Total: _____

Question 1 (The MC32 and Interfaces)
Briefly and concisely answer the following questions:

a) (20 points) What is the difference between "user mode" and "supervisor mode" in the MC32 microprocessor?

The difference between user mode and supervisor mode is controlled by the contents of the S bit in the CPU's status register. When S=1 the CPU is said to be in the more privileged supervisor mode. In supervisor mode the running program can execute all of the instructions and can access all of the CPU register bits. When S=0 the CPU is said to be in the less privileged user mode. In user mode the running program has slightly restricted access to the CPU:

- *It cannot execute certain instructions (e.g. STOP, MOVE to SR) that could potentially have far-reaching impact on the system*
- *It cannot access the system byte of the status register*
- *It cannot access the supervisor stack pointer (SSP).*
- *User mode programs cannot use the MOVEA instruction with the user stack pointer (USP) as the source or destination.*

b) (20 points) How does the microprocessor make the transition between these two modes?

The processor transitions automatically from user mode to supervisor mode when the CPU handles exceptions, such as software traps or interrupts. There are two ways of transitioning from supervisor mode to user mode:

1. *An instruction can be executed that clears the S bit in the status register*
2. *A return from exception (RTE) instruction can be executed when the mode before the exception was handled was user mode.*

c) (10 points) Consider the interface chip MC6843 Floppy Disk Controller (FDC). It serves as the interface between the microcontroller and a floppy disk drive. During class we mentioned few tasks it can perform. Mention three of those tasks.

- *Some tasks that the MC6843 is capable of performing include:*
- *Reading and writing from and to the floppy disks.*
- *Decode and execute commands such as seek and read information contained in the floppy disk.*
- *Execution of algorithms for error detection.*
- *Data synchronization*

Question 2 (Operating systems)
Briefly and concisely answer the following questions:

a) (20 points) What do we mean by "critical section"?

A critical section is a segment of code that must be executed until completion, without being interrupted by any context switches. For certain critical sections, not even hardware interrupts are allowed to run.

Examples of critical sections include the following:

- *A routine that updates a data structure (e.g. a linked list, or a binary tree) consisting of records that are linked together by pointers typically has critical sections if two or more pointers must be changed when the data structure is changed from one consistent state to the next consistent state.*
- *A hardware register (e.g. the interrupt status register in the DUART) may need to be read and then written in a certain sequence of operations before any other program can be allowed use the hardware. Such a situation commonly occurs during hardware initialization routines or interrupts service routines.*

The first kind of critical section could probably be handled by simply disabling context switches. However, if interrupt service routines could potentially need to access the data structure in question, then those interrupts would need to be masked out as well.

The second kind of critical section could also be handled by disabling context switches and interrupts. However, depending on the way the interrupts are prioritized, in may be sufficient to rely on the interrupt priority systems to protect the critical section.

b) (20 points) Describe the two common strategies used for protecting critical sections.

The first strategy implements the following steps:

1. Mask out interrupts (i.e. disable multitasking) at the start of the critical section.
2. Restore the interrupts (i.e. enable multitasking) at the end of the critical section.

The second strategy implements the following steps:

1. Mask out interrupts, then check on the value of the semaphore
2a. If the flag is already set, then some other task is in the critical section. Add the new task to a queue of tasks that are blocked on the flag. Switch to another ready-to-run task. Restore interrupts.
2b. If the flag is cleared, then the data structure is available. Set the flag, restore interrupts, and proceed into the critical section. When exiting the critical section, mask out interrupts. If no other task is blocked on the flag, then clear the flag. If at least one task is blocked on the flag, then unblock the highest priority task by removing its TCB to the ready-to-run queue. Restore interrupts.

c) (10 points) What do we mean by “deadlock”?

It is a state in an operating system environment in which two or more tasks become permanently blocked from running because they are waiting for resources, which are held by the other task, to become available.

The four necessary conditions to deadlock are:

1. *Resources can be assigned exclusively to tasks*
2. *One task can request two or more resources*
3. *A task cannot be forced to release a resource*
4. *Two or more tasks for a circular chain where each task is waiting for a resource that is held by the next task in the chain*