CMPE401
Computer Interfacing

MIDTERM EXAMINATION

October 24, 2007

Name: _____
ID: _____

5 questions. 50 minutes.
Allowed material:

- Course notes
- A 68000 Programming Reference Card
- Calculators

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

Marks:
Question 1: _____
Question 2: _____
Question 3: _____
Question 4: _____
Question 5: _____
Total: _____

Question 1 (Introduction to computer interfacing)

Briefly and concisely answer the following questions. Draw neat diagrams if necessary:

a) (5 points) During class we discussed some typical mechanisms at hardware interfaces. Within the context of signal conditioning, explain the effect of:

   a. Filtering

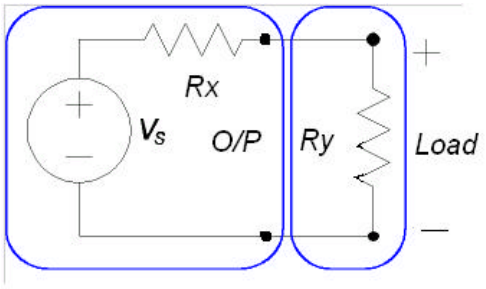
   The intention of filtering is to remove irrelevant information (noise) that affect the signal we are interested in. Band-pass filtering is widely implemented at this point.

   b. Impedance matching

   Impedance matching is performed when a transducer's internal impedance can cause errors in measurements of dynamic signals (variables).

   c. Loading

   Loading occurs when two circuits are connected together, the "load" of the attached circuit can cause the voltage levels in the output of the original circuit to drop. The Thévenin equivalent circuit shown bellow depicts such case.



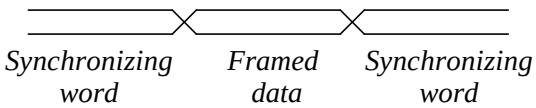
b) (5 points) What do we mean by:

   a. Data synchronization in serial communication.

   Data synchronization is required because of clock drift in computers connected in a serial configuration. The incoming bits have to be read at a precise time so that the message is not disrupted during the reading process. Such process is called data synchronization.

   b. Data framing in serial communication.

   Data framing refers to the process of aligning the data within a specific frame. This step is essential in any asynchronous serial communication. The data transmit/receive is enclosed within synchronizing bits or words, which are unique combination of bits that cannot be mistaken for data by the receiver. The following diagram depicts an aligned frame.



c. Handshaking in serial communication.

It is the *automated process of negotiation to set parameters of a communications channel established between two entities before normal communication over the channel begins. Handshaking may be used to negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel, i.e. transfer rate, coding alphabet, parity, interrupt procedure, and other protocol or hardware features.*

Question 2 (The 68332 microcontroller)

Briefly and concisely answer the following questions:

a) (15 points) Explain in your own words what the System Integration Module (SIM) is, what is its purpose, and how it interacts with the CPU32 in the MC32 microcontroller.

The System Integration Module (SIM) consists of several submodules that allow the MC32 to interact with other devices or modules. Some important components of the SIM include the clock synthesizer, chip select, the external bus interface, and the system protection submodule.

-The clock synthesizer generates the system clock signals using an external crystal oscillator or an external clock circuit. The clock synthesizer is also used to generate other clock signals for "watchdog" timers and a separate periodic interrupt timer on the chip. The periodic interrupt timer can be used to create a real-time clock that generates interrupts to the CPU at fixed time intervals.

-Chip selects and the external bus are used to expand the system by adding circuits external to the microcontroller. A chip select enables a selected peripheral or a memory unit for data transfer. There are 12 chip select signals in the MC32. Five of these signals can be assigned as address signal lines on the external bus if these signals are not used as chip select signals.

-The external bus provides a 16-bit data path and up to 24 address signal lines to external devices. It also contains the controls signal lines for data transfers, interrupt requests, and other functions. The SIM signal lines are controlled by an executing program. Various signals can act as chip selects, as an external bus, or as input and output ports.

-The system protection submodule allows activity on the intermodule bus or the external bus to be monitored electrically. If an operation does not finish within a prespecified allotted time, an error is indicated to the CPU32. This hardware watchdog timer feature can monitor data transfers, interrupt acknowledgement cycles, and other types of bus activities. A software watchdog timer is also available to prevent a program from being trapped in a loop or otherwise exceeding the maximum time allotted to the program. If the program does not complete its execution in the predetermined time, the MC32 is reset.

b) (15 points) Explain the Queued Serial Module (QSM). Indicate its main components/submodules. Specify the general purpose/functions of each of the submodules.

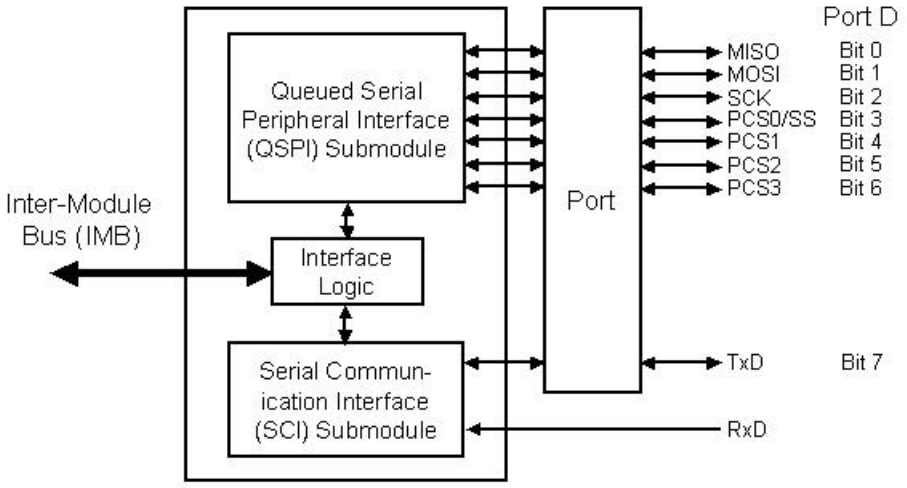
The queued serial module provides the MC32 with two serial communication interfaces. There are two independent submodules:

1. *The Serial Communication Interface (SCI)*
2. *The Queued Serial Peripheral Interface (QSPI)*

The SCI is a Universal Asynchronous Receiver Transmitter (UART) interface. In many applications the SCI transfers ASCII characters between the MC32 and other external devices that also have a UART interface.

The QSPI provides synchronous transfer of data for communication I/O expansion. The QSPI can act as a master or slave device in a network of devices with compatible serial interfaces. As a master, the QSPI controls the data format and timing of transfers between the QSPI and slave devices. This submodule contains a queue that allows up to 16 independent data transfers without intervention of the MC32. As master, the QSPI chip select signals lines select the specified peripheral device involved in each transfer.

The following diagram depicts the QSM and its I/O signal lines, and its submodules:



c) (5 points) Consider the registers of the QSM. Complete the following table with the correct data. The left column indicates the complete register name, the middle column indicates the register mnemonic, and the rightmost column indicates the information such register stores. The first row of the table is already filled as an example for you.

Register Name	**Register Mnemonic**	**Information indicates**
Serial Communication Control Register 0	SCCR0	Baud rate
Serial Communication Control Register 1	*SCCR1*	Data format, parity type
Serial Communication Status Register	*SCSR*	Noise error flag Framing error Parity error
QSM Pin Assignment Register	*QPAR*	QSM pin assignment
Serial Peripheral Control Register 3	SPCR3	*For testing operation of QSPI*
QSM Data Direction Register	QDDR	*Indicates data flow direction of I/O pins of QSM*

Question 3 (Kernels and Operating systems)

a) (5 points) In your own words, briefly and concisely describe the difference between preemptive multitasking and cooperative multitasking. Indicate two disadvantages of cooperative multitasking vs. preemptive multitasking. Give one example of an operating system in each case (preemptive and cooperative).
In preemptive multitasking, a priority level is associated with each task. The highest priority ready-to-run task is *always running on the CPU. If a task was not running previously and for some reason it becomes the highest priority task y, then the currently running task is preempted to allow the highest priority task to start running. In cooperative multitasking, the running task cannot be preempted; it continues to run until it decides to give up the CPU to allow the next task in the queue of tasks to run.*
Some disadvantages of cooperative multitasking:
(1) The system is vulnerable to greedy tasks that hold on to the CPU longer than they perhaps should.
(2) The tasks are not as loosely interacting as they would be if they were strictly prioritized. This makes the system design problem trickier and more difficult to debug and to tune.
(3) The system is less likely to have strictly deterministic behaviour since it is difficult to ensure that high-priority tasks will always get to run promptly.
Examples of a preemptive operating system are Linux, Unix, Windows XP, Mac OS X. An examples of a cooperative operating system is Mac OS 9

b) (5 points) Mention at least four disadvantages of multitasking operating systems.
1. Time is wasted performing the context switches between running tasks.
2. Memory space is required to store the kernel code.
3. *Critical sections and shared resources must be protected. Synchronization mechanisms must be provided.*
4. New problems, such as deadlock, can occur when multiple tasks require exclusive control of multiple shared resources. Precautions must be taken to avoid / detect such problems.
5. Debugging multitasking software can be tricky because it is difficult to recreate and understand complex interactions between the tasks.

c) (5 points) Briefly and concisely describe the synchronization process using counting semaphores.

A counting semaphore is a data structure with two parts:
1. an integer count, which is initialized to the number of available shared resources of a particular kind.
2. a queue of tasks that are blocked and waiting for the semaphore count to exceed a value of 0.
When a task needs exclusive use of a resource, the semaphore count value is decremented by 1. If the new count is nonnegative, then the task is allocated one of the available resources from the shared pool of resources. If the new count is negative, then the task is switched off the CPU and is added to a queue of blocked tasks.
When a task has finished using a resource, then:

1. *If no task is blocked on the semaphore then simply increment the count.*
2. *If tasks are blocked on the semaphore, then the highest priority task is moved to the ready-to-run queue.*

Question 4 (The 68681 DUART)

Briefly and concisely answer the following question:

a) (5 points) What do we mean by "multudrop mode"?

When multiple devices are connected in a network, each "reader" is referred to by a unique address. The addresses of such devices are used by the microcontrollers to talk with each "reader". When a message is sent to a specified device, the message is prefixed by the address of the target device, so only the target device listens to the message. The other devices connected to the network will ignore the message.

b) (25 points)

Write an assembly language program that initializes a DUART for normal duplex serial communications as follows:

Port B is to use interrupts in both the transmit and receiver directions. Receiver interrupts are to be caused whenever FIFO is full. The characters are to contain 7 data bits, with odd parity, and two stop bits. The baud rate is to be 2400 in both directions. Receiver error status is to apply to all characters in the FIFO. The RTS handshake output to be controlled by the program, not controlled automatically by the DUART hardware. The CTS input is to have no automatic effect on the transmitter.

Part of the code is already given to you. You have to fill in the underlined sections of the code.

```
.equ DUART, 0xFF0000          /* DUART base address */
.equ ACR, 0x08
.equ ISR, 0x0A
.equ IMR, 0x0A
.equ MR1B, 0x10
.equ MR2B, MR1B
.equ SRB, 0x12
.equ CSRB, 0x12
.equ CRB, 0x14
.equ RBB, 0x16
.equ TBB, 0x16
Duart_Init:
LEA DUART,A0                  /* point to DUART base address */

/* Ensure that the MR pointers are to MR1B */
MOVE.B #0b00010000,CRB(A0)    /* bits 6-4: reset pointer to MR1B */
CLR.B CRB(A0)                 /* no command to B */

/* Initialize MR1B */
MOVE.B #0b01100110,MR1B(A0)   /* bit 7: RTS controlled by program */
                              /* bit 6: IRQs on FIFO full */
                              /* bit 5: block-mode error status */
                              /* bits 4-2: Odd parity */
                              /* bits 1-0: 7 data */
```

```
/* Initialize MR2B */
MOVE.B #0b00000111,MR2B(A0)   /* bit 5: auto RTS output off (controlled
                              by pogram) */
                              /* bit 4: auto CTS TxB inhibit off */
                              /* bits 0-3: 2.0 stop bits */
                              /* bits 6-7: operating mode is full
                              duplex */

/* Select the baud rates */
MOVE.B #0b10001000,CSRB(A0)   /* bits 0-3 and 4-7 and ACR[7]=0 or 1:
                              2400 used by channel B (transmitter and
                              receiver */

/* Enable the desired interrupts */
MOVE.B #0b00110000,IMR_COPY   /* enable RxRDYB and TxRDYB IRQs */
MOVE.B IMR_COPY,IMR(A0)

/* Issue reset commands to channel B */
MOVE.B #0b00100000,CRB(A0)    /* bits 6-4: reset receiver B */
MOVE.B #0b00110000,CRB(A0)    /* bits 6-4: reset transmitter B */
MOVE.B #0b01000000,CRB(A0)    /* bits 6-4: reset error status B */
MOVE.B #0b01010000,CRB(A0)    /* bits 6-4: reset break interrupt B */
CLR.B CRB(A0)                 /* no command to B */

/* Install the DUART interrupt vector */
MOVE.L #DuartISR,DUART_VEC_ADDR

/* Enable channel B */
MOVE.B #0b00000101,CRB(A0)    /* bits 3-2: enable TxB */
                              /* bits 1-0: enable RxB */

CLR.B CRB(A0)                 /* no command to B */
MOVEM.L (SP)+,A0              /* restore scratchpad registers */
RTS
```

Question 5

Briefly and concisely describe what we mean by the following terminology. Provide a description in each case:

a) (5 points) TCP:

"Transmission Control Protocol" is the transport layer protocol that is used in the Internet to ensure reliable end-to-end transmission of streams of application bytes traveling in both directions between two communicating computers. TCP uses the unreliable datagram services provided by the Inter-networking Protocol (IP) at the next lower layer in the standard TCP/IP communications protocol stack. TCP is in turn used by higher layer application protocols, such as "telnet", "ftp", and "http".

b) (5 points) HTTP:

"Hypertext Transfer Protocol" is a communications protocol used to transfer information in the World Wide Web. Its original purpose was to provide a way to publish and retrieve hypertext pages.

HTTP is a request/response protocol between clients and servers. The client making an HTTP request, i.e. a web browser or a spider, is referred to as the user agent. The responding server is called the origin server. HTTP does not need to use TCP/IP or its supporting layers. HTTP can be implemented on top of any other reliable protocol on the network.

An HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to port 80 on a host. An HTTP server listening to that port waits for the client to send a request message.

Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own, the body of which is perhaps the requested file, an error message, or some other information.