THE UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
EECE 485 : Digital Instrumentation for Mechanical Systems
2000/2001 Winter Session Term 1

MID-TERM EXAMINATION
8:30 am – 9:20 am
October 23, 2000

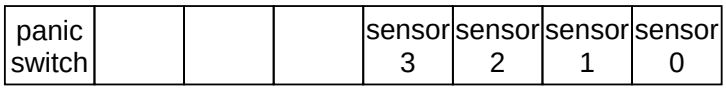
This exam has two (2) questions on two (2) pages. The marks for each question are as indicated. There are a total of 26 marks. Answer all questions. Write your answers in the exam book provided. Show your work. You may answer the questions in any order. Books, notes and calculators are allowed. You may keep this exam paper.

Question 1 (9 marks)

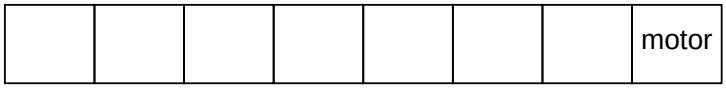
This question asks you to write a C program that implements a computer-controlled safety system for a machine.

Your program can determine the status of four sensors and one "panic button" by reading a data port at address 0x300. The four sensors are connected to the four least-significant bits of the data port. The panic button is connected to the most-significant bit. The values of the other three bits should be ignored.

data port (0x300)



control port (0x301)



If the panic button bit or any of the sensor bits are set to 1 then the controller should turn the motor off, otherwise it should turn the motor on. The motor can be controlled by writing 1 (on) or 0 (off) to a control port at address 0x300.

The following two functions, described in Lab 2, can be used by your C program to access these ports:

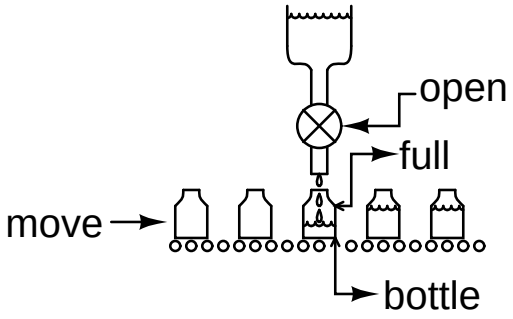
```
int speek( int address ) ;
int spoke( int address, int value ) ;
```

Write a function `check()` that takes no arguments and returns no value. This function should check the sensor status once and then turn the motor on or off as specified above.

Write a `main()` function that calls `check()` repeatedly (forever).

Question 2 (a) (11 marks)

This question asks you to design a controller for a machine that fills milk bottles. Bottles move down a line to a filling station where they are filled. The controller has two outputs: when `move` is 1 a motor moves the bottles down a line; when `open` is 1 a valve opens and milk flows into the bottle at the filling station. The controller has two inputs: if `bottle` is 1 a bottle is in position to be filled; if `full` is 1 the bottle is full of milk.



The controller should repeatedly carry out the following three steps:

- move the bottles down the line until a bottle is in position to be filled
- open the valve until the bottle is full
- move the bottles down the line until there is no bottle in the filling station

The valve should not be open when the bottles are moving and the bottles should not be moving when the valve is open.

Design a state machine for this machine's controller. List the inputs and outputs. Choose a sufficient number of states and give a name to each state. Write a table showing the outputs for each state. Write the state transition table showing the following columns: starting state, input, next state.

You may use "X" to indicate that an *input* has no effect.

Hint: You may need more states than there are combinations of outputs.

(b) (6 marks)

Consider a (different) state machine with three states. These three states are encoded using two binary values (*A* and *B*). This state machine is part of a controller with two outputs (*x*, and *y*). The following table shows the states, their binary encodings and the corresponding outputs.

state	*A*	*B*	*x*	*y*
S1	0	0	1	0
S2	0	1	0	1
S3	1	0	1	1

Give the boolean logic equations for the outputs *x* and *y* as a function of the state variables *A* and *B*. Use AND, OR and NOT gates to draw a schematic diagram of a circuit that generates *x* and *y* from *A* and *B*. You need not simplify your solution.