NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
EXAMINATION FOR
Semester 2 AY2003/2004

CS2271: EMBEDDED SYSTEMS

April 2004 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FIVE (5)** questions and comprises **ELEVEN (11)** printed pages, including this page.
2. Answer **ALL** questions within the space provided in this booklet.
3. This is an OPEN BOOK examination.
4. Please write your Matriculation Number Below.

MATRICULATION NO: _____

This portion is for examiner's use only

QUESTION	MARKS	REMARKS
Q1		
Q2		
Q3		
Q4		
Q5		
Total		

QUESTION 1 (Handel-C: 6 Marks)

Suppose we want to design the following filter in hardware. x_i s are assumed to be samples of data taken periodically while the y_i s are the output values. Assume that the samples are coming in periodically and the filter output is computed every time a new sample comes in.

$$y_i = (x_i + 2 . x_{i-1} + x_{i-2}) / 4$$

Write down the Handel-C code to implement this filter in hardware. For your convenience, we have defined two variables `in` and `out` that represent the input and output registers respectively. Your circuit should use minimum number of clock cycles, registers and combinational circuit. **(6 Marks)**

Answer:

```
unsigned int 32 in, out;
```

QUESTION 2 (Assembly Language Programming: 10 Marks)

```
int x;
int fib_iter (int a, int b, int count){
        if (count == 0)
                return b;
        else
                return fib_iter( a+b, a,  count-1);
}
void main(void){
        x = fib_iter(1, 0, 5);
}
```

A straightforward compilation of the above program will use recursive function call which is inefficient. Write an efficient ARM assembly language code for the above which *does not use recursion*. Your code should properly handle the parameter and return value passing between main() and fib_iter(). **(10 Marks)**

BLANK PAGE

QUESTION 3 (Scheduling 12 Marks)

Three tasks have the following timings. Assume all the tasks are ready at the beginning and period is equal to deadline.

Task	Execution Time	Period
T1	1	4
T2	1	5
T3	5	10

A) Write down the EDF schedule for this set of tasks for 20 time units with minimum number of context switches. **(6 Marks)**

Answer:

Time Task

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

B) Let us assume that the task T3 is nonpreemtable (i.e., once the scheduler allows T3 to execute, it cannot be preempted). Will the task set remain schedulable under EDF policy?
If the answer is yes, write down the EDF schedule with T3 as nonpreemptable task.
If the answer is no, then can you design a *custom schedule* with T3 as nonpreemtable task that (1) meets all the deadlines and (2) repeats every 20 time units? **(6 Marks)**

Answer:

Time Task

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

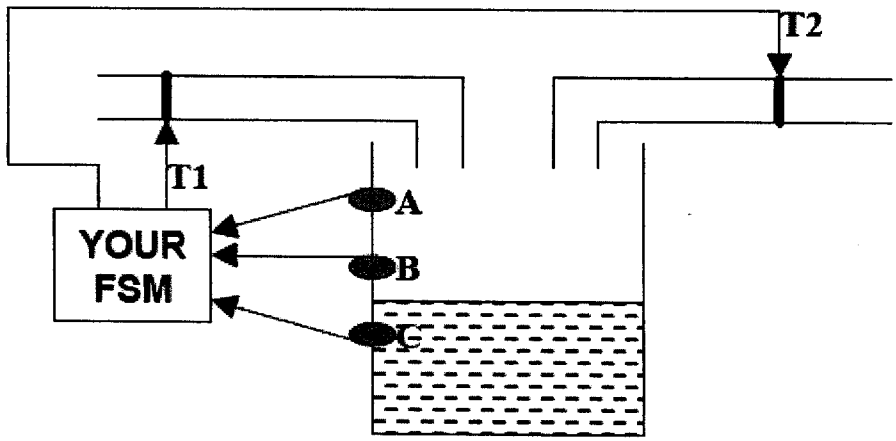
QUESTION 4 (Finite State Machine: 6 Marks)

A water tank has been wired with three water-level detecting sensors (A, B and C in the figure) at different levels. There are two electronically controlled taps that feed into this water tank. You are required to design a finite-state machine to control the taps T1 and T2 according to the requirements stated below.

Tap T1 needs to be opened whenever the water level goes below sensor B. Tap T1 should NOT be closed if the water level reaches sensor B again. T1 should be closed only after the water level reaches sensor A.

Similarly, Tap T2 needs to be opened when the water level goes below sensor C. (At this stage both taps will be open.) Tap T2 will stay open till the water level reaches sensor B and then it is closed.

Your FSM will take the binary inputs from the sensors A, B and C and produce binary outputs to control taps T1 and T2. A sensor input is set to 1 if the water level is at or above the sensor's level. It is 0 if the water level is below the sensor. The tap control should be set to 1 to open the tap and it should be reset to 0 to close the tap. You can use don't care conditions to combine some inputs that produce the same output/next state. **(6 Marks)**



BLANK PAGE

QUESTION 5 (Miscellaneous: 16 Marks)

A) How many bits are required in an address bus to access the following?

(a) Byte-addressable 1 megabyte memory and 100 device registers with memory mapped I/O

(b)Byte-addressable 1 megabyte memory and 100 device registers with I/O mapped I/O

(c)Word-addressable (1 word = 4 bytes) 1 megabyte memory **(3 Marks)**

B) An A/D converter samples the analog input at the rate of 100 KHz. The interrupt handler executes 100 instructions obtaining the sample and passing it to the application routine. The application routine requires 350 instructions to process the sample. What is the minimum required clock speed of the processor assuming it executes 2 instructions per clock cycle?

(4 Marks)

C) Suppose we choose to do DRAM refresh in software by periodically reading the memory. Recall that reading at a particular row address causes the entire row to be refreshed. Assume that the memory is implemented as thirty two 256K x 1-bit DRAM chips where each DRAM chip has 512 rows. The memory is word (4 bytes) addressable; higher order address bits specify rows while lower order address bits specify columns. A timer periodically generates an interrupt to call a refresh handler. The handler should refresh (i.e. read) a row each time it is called. Write C/assembly/pseudo code for the refresh interrupt handler. You should define the variables you use; however, it is not necessary to write the initialization code. **(5 Marks)**

D) The function below combines a name and an integer to form a 12-character filename. The filename is prepared in a buffer whose address is returned as the value of the function. Unfortunately, the function causes run-time errors when called by more than one concurrent process. Explain the cause of the problem and then modify the function to correct the problem.

```
char *make_filename(char *name,  int version){
     static char fname_buffer[13];
     sprintf(fname_buffer, "%.8s.%03d", name, version);
     return fname_buffer;
}
```

(4 Marks)

END OF PAPER