University of Victoria

UNIVERSITY OF VICTORIA
FINAL EXAM
December 2015

Course Name & No.:	**CENG 355 Microprocessor-Based Systems**
Section(s):	**A01/A02**
CRN:	**10399/10400**
Instructor:	**D.N.Rakhmatov**
Duration:	**3 Hours**

NAME: _____

STUDENT NUMBER: V00 _____

This exam has a total of _7_ pages including this cover page and _0_ separate handout(s).

Students must count the number of pages and report any discrepancy immediately to the Invigilator.

This exam is to be answered:

- On the paper
- In Booklets provided
- NCS Answer sheets

Marking Scheme: This exam has **NINE** questions worth the total of **90** points. Allocation of points per question is indicated at the beginning of each question.

Materials Allowed: Textbook, lecture notes, homework solutions, calculators (nonprogrammable).

University of Victoria

UNIVERSITY OF VICTORIA
FINAL EXAM
December 2015

Course Name & No.:	**CENG 355 Microprocessor-Based Systems**
Section(s):	**A01/A02**
CRN:	**10399/10400**
Instructor:	**D.N.Rakhmatov**
Duration:	**3 Hours**

NAME: _____

STUDENT NUMBER: V00 _____

This exam has a total of _7_ pages including this cover page and _0_ separate handout(s).

Students must count the number of pages and report any discrepancy immediately to the Invigilator.

This exam is to be answered:

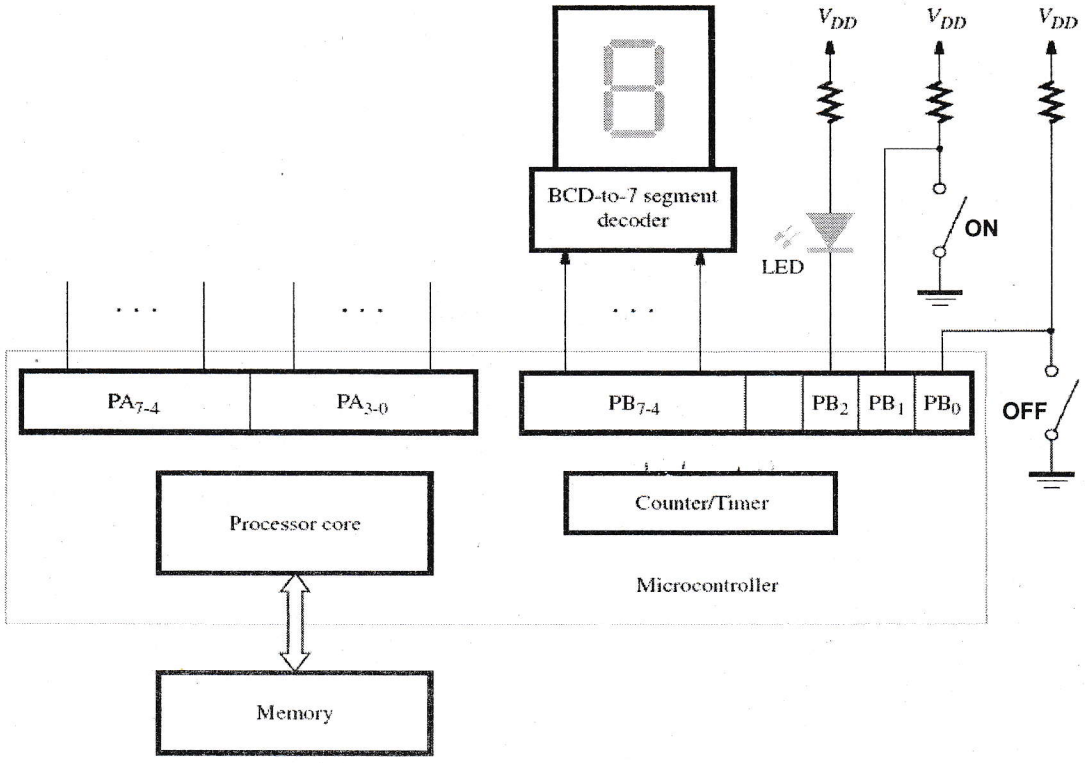
- On the paper
- In Booklets provided
- NCS Answer sheets

Marking Scheme: This exam has **NINE** questions worth the total of **90** points. Allocation of points per question is indicated at the beginning of each question.

Materials Allowed: Textbook, lecture notes, homework solutions, calculators (nonprogrammable).

1. *[15 points]* The textbook's microcontroller is used in a system shown below, which is responsible for 2 tasks: (1) turning the LED on and off when the **ON** and **OFF** switches are pressed, and (2) transferring 4-bit data from **Port A** to **Port B**. Write the corresponding C program, assuming that the second task is the main program, and the first task is an ISR, whose address is stored at memory location **0x20**. Also, assume that bit **6** of the processor status register (**PSR[6]**) is the processor's interrupt-enable bit, and **Ports B** is always ready to receive data from the processor. Initially, the 7-segment display shows digit **0**, and the LED is off. (The Counter/Timer peripheral is not used here.)

- *Main Program*: Whenever **PAIN** is updated with new data, the lower half of **PAIN** (pins $\mathbf{PA_{3-0}}$) must be transferred to the upper half of **PBOUT** (pins $\mathbf{PB_{7-4}}$) for display.
- *ISR*: **Port B** must be configured to generate interrupts whenever **PBIN** is updated. If **OFF** has been pressed ($\mathbf{PB_0}$ equals 0), the LED must be turned off; if **ON** has been pressed ($\mathbf{PB_1}$ equals 0), the LED must be turned on. Otherwise, the LED state remains unchanged.



2. *[15 points]* Consider a byte-addressable computer with 4-KB main memory and 128-byte cache that has 8 blocks, where each block consists of four 32-bit words. While executing some program, the CPU reads 32-bit words from the following sequence of addresses:

080 220 208 444 440 200 24C 44C 448 088

Show the cache contents (e.g., **[000]** = contents stored at address **000**) at the end of this sequence (10 addresses) and calculate the corresponding miss rate given that:

(a) Cache is direct-mapped.

(b) Cache is 4-way set-associative (4 blocks per set) with LRU replacement.

(c) Cache is fully-associative with LRU replacement.

3. *[15 points]* Consider a C code fragment below, working on a given square matrix `float X[N][N]` (stored row by row, i.e., in the row-major order), where `N = 512`:

```
for (i = 0; i < N; i++) {
      float sum = 0;
      for (j = 0; j < N; j++) {
            float temp = X[i][j];
            sum = sum + temp;              /* sum elements of X */
            X[i][j] = temp * temp;         /* square elements of X */
      }
      sum = sum * sum;
      for (j = 0; j < N; j++) {
            X[i][j] = X[i][j]/sum;         /* normalize elements of X */
      }
}
```

Determine the `X`-related page fault rate in the following two cases: (1) the main memory uses **1-KB** paging with four pages allocated for `X`, and (2) the main memory uses **4-KB** paging with only one page allocated for `X`. Initially, no part of `X` is in the main memory.

4. *[5 points]* Table below specifies a set of **3** independent pre-emptive tasks to be executed by a single processor. Show the task schedule based on *Rate Monotonic* (**RM**) prioritization. If needed, break any prioritization ties as you wish.

Task T_i	Period P_i	WCET C_i	Deadline D_i	Initial Delay ϕ_i
T1	25	10	25	0
T2	50	10	50	0
T3	100	20	100	0

5. *[5 points]* Consider a pipelined datapath consisting of five stages:

- **F** – fetch the instruction from the memory,
- **D** – decode the instruction and read the source register(s),
- **C** – execute the ALU operation specified by the instruction,
- **M** – execute the memory operation specified by the instruction,
- **W** – write the result in the destination register.

Identify data hazards in the code below and insert NOP instructions where necessary.

```
ADD   R0, R4, R2        // R2 = R0 + R4
SUB   R1, R3, R0        // R0 = R1 - R3
ADD   R2, R2, R4        // R4 = R2 + R2
ADD   R0, R2, R3        // R3 = R0 + R2
ADD   #4, R4, R4        // R4 = R4 + 4
ADD   #4, R2, R2        // R2 = R2 + 4
MOV   R0, (R4)          // MEMORY[R4] = R0
MOV   (R2), R0          // R0 = MEMORY[R2]
MOV   (R1), R3          // R3 = MEMORY[R1]
SUB   #4, R1, R1        // R1 = R1 - 4
```

6. *[10 points]*

(a) Show **decimal** number $+2^{-128}$ in the 32-bit IEEE-754 floating-point format.

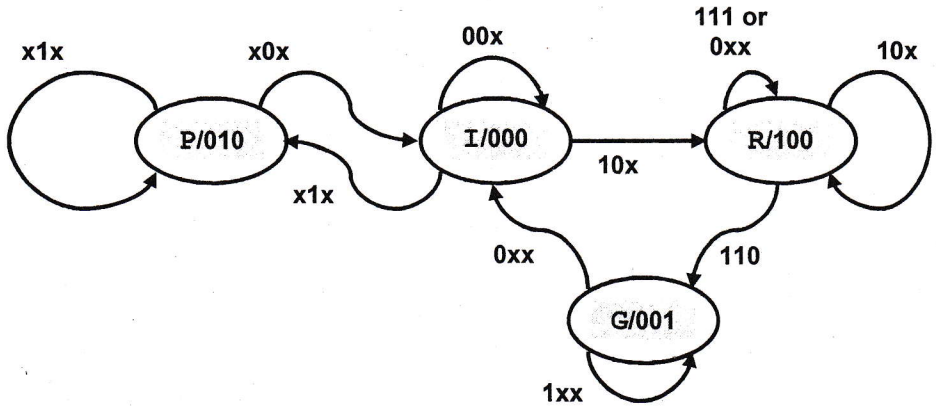
(b) Given two 32-bit IEEE-754 floating-point numbers **X** and **Y** below, calculate (in the binary format) **Z = X–Y**, and then convert **Z** to the decimal format:

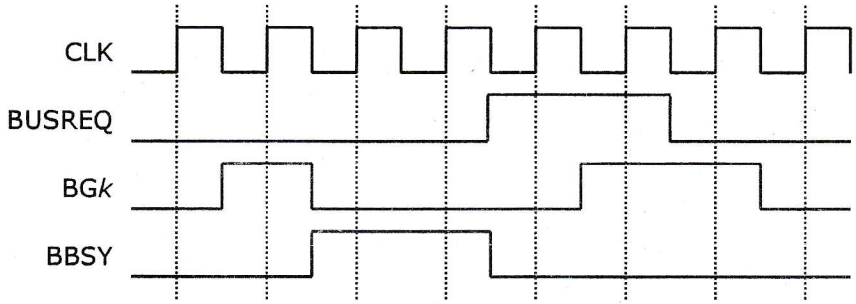
X = 0 01111101 00000000000000000000000,
Y = 1 01111111 10000000000000000000000.

7. *[5 points]* Suppose some **Moore-type** Master FSM has 2 outputs, master-request **mR** and master-waiting **mW**, and 1 input, slave-busy **sB**. Show the state diagram of this FSM, assuming that it implements the following protocol: (1) Initially, the FSM outputs **mR = 1** and **mW = 0**, waiting for **sB** to be asserted; (2) Once **sB = 1** is received, the FSM outputs **mR = 0** and **mW = 1**, waiting for **sB** to be de-asserted; (3) Once **sB = 0** is received, the FSM outputs **mR = 0** and **mW = 0** for one clock cycle, and then goes back to step (1).

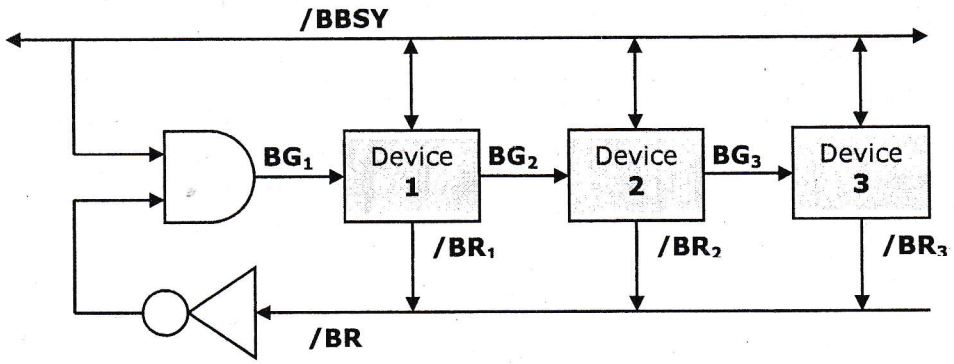
8. *[10 points]* Consider the **Moore-type FSM** state diagram below, where **x** represents **"don't care"**. Given the input waveforms as shown below, draw the corresponding output waveforms. Assume that the FSM is initially in state **I**.

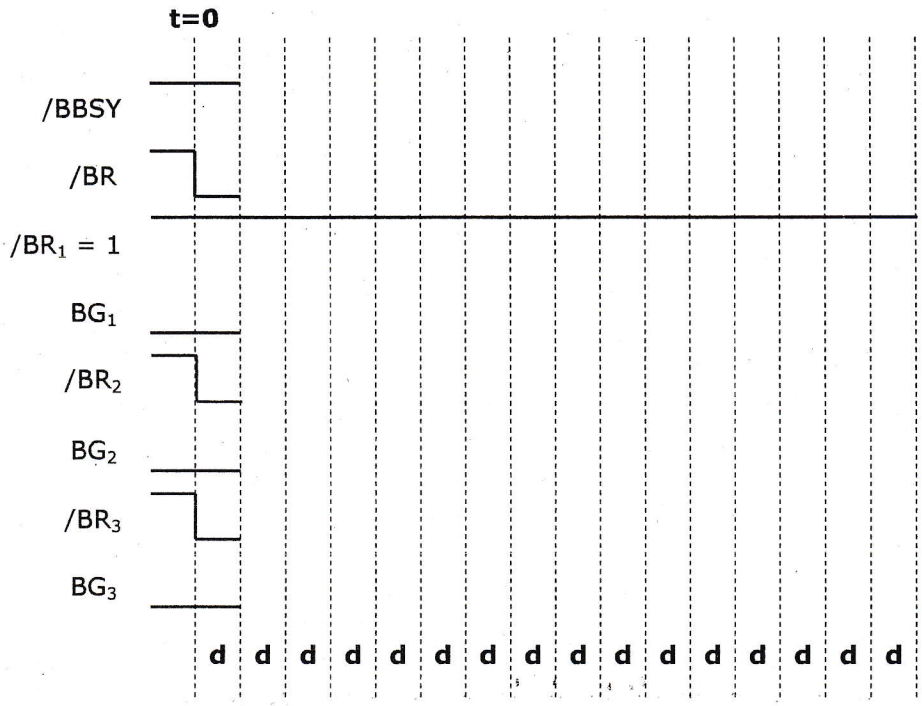
Inputs = BUSREQ, BG*k*, BBSY Outputs = BR, BG(*k*+1), BBSY





9. *[10 points]* Consider the daisy-chain arbitration scheme shown below. Assume that the input-to-output signal propagation delays are the same and equal to **d** for all three devices, the inverter, and the **AND** gate. Also, assume that device **x** is able to start using the bus (making **/BRx = 1** and **/BBSY = 0**) only when it receives a **0-1** transition on its bus-grant input **BGx** and detects that the bus is not currently busy (i.e., **/BBSY = 1**). Also, assume that device **x** lets the bus-grant propagate through only when it is neither requesting nor using the bus. Finally, assume that any of the three devices will need to use the granted bus for only **3d** time units. Complete the timing diagram shown on the next page, where Device 2 and Device 3 request the bus at the same time **t = 0**.





NOTE: If answering Question 9 here, please write down your name below and insert this page in your Booklet.

NAME: _____

END