UNIVERISTY OF VICTORIA

CENG 355 MICROPROCESSOR-BASED SYSTEMS

MIDTERM EXAMINATION

27 OCTOBER 2014

NAME:_____ STUDENT NO._____

INSTRUCTOR: D.N.RAKHMATOV DURATION: 80 MINUTES

TO BE ANSWERED IN THE BOOKLET.

STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER, AND REPORT ANY DISCREPANCY IMMEDIATELY TO THE INVIGILATOR.

THIS EXAMINATION PAPER HAS **3** PAGES AND **3** QUESTIONS.

In taking this examination, you agree that all work recorded herein is your own. A student caught in the act of cheating will be given a grade of **F** on this examination.

Show your work.

Read the questions carefully. If something appears ambiguous, write down your assumptions.

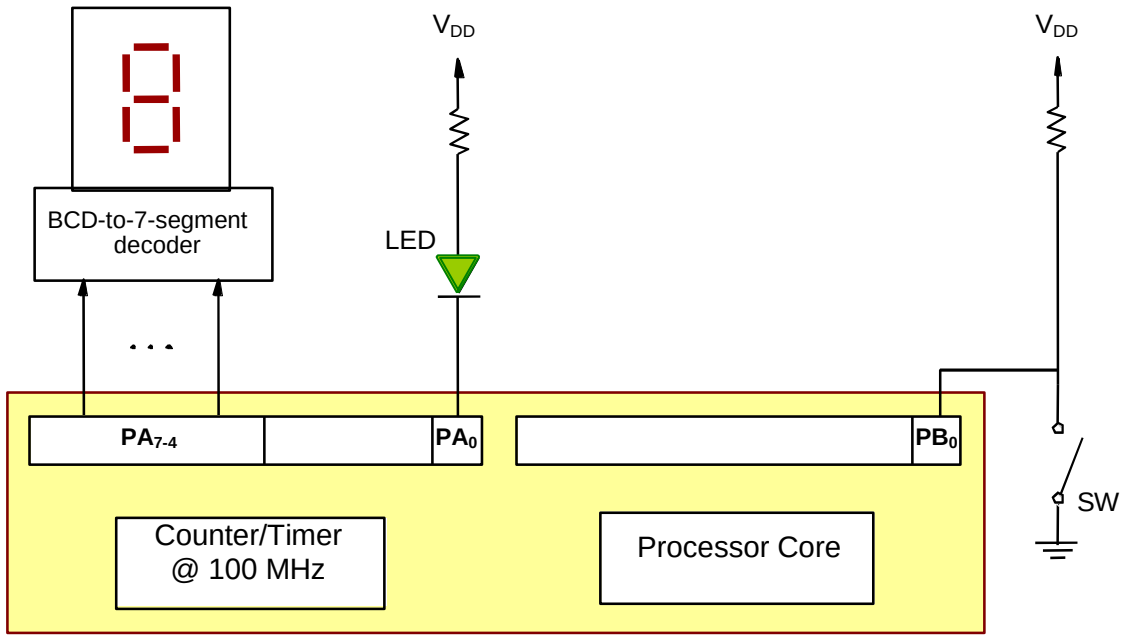
You are allowed to use books, notes, and/or calculators during this examination.

GOOD LUCK!

1. [15 points] The textbook's microcontroller is used in a system shown below and is responsible for 2 tasks: (1) flipping the LED on/off state every time the **SW** key has been hit, and (2) incrementing the displayed digit every second, i.e., displaying the numerical sequence **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **0**, **1**, … . Write the corresponding C program, assuming that the **first task** is the main program, and the **second 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 **Port A** is always ready to receive data from the processor. Initially, the 7-segment display shows digit **0**, and the LED is off.

• *Main Program*: Every time the **SW** key is hit, i.e., pressed and then released (bit $\mathbf{PB_0}$ must first become 0 and then 1 again), the LED state must be flipped: if the LED is on, it must be turned off, and vice versa.

• *ISR*: The 100-MHz Counter/Timer must be configured to generate interrupts every second, and its ISR must increment the displayed digit. Incrementing **9** gives **0**.



2. [10 points] Table below specifies a set of **4** independent pre-emptive tasks to be executed by a single processor. Show the task schedule using *Rate Monotomic* (**RM**) priority assignment. **Note:** If needed, break any prioritization ties as you wish.

Task T_i	**P_i**	**C_i**	**D_i**	**ϕ_i**
T1	**30**	**5**	**20**	**0**
T2	**30**	**5**	**30**	**0**
T3	**45**	**15**	**40**	**0**
T4	**90**	**15**	**90**	**0**

3. [15 points] Assume a byte-addressable computer has 4-KB main memory and 128-byte cache with **four blocks**, where each block has **eight 32-bit words**. While executing some program, the CPU reads 32-bit words from the following sequence of 10 addresses (in hexadecimal format):

088 090 250 09C 240 200 220 210 230 080

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

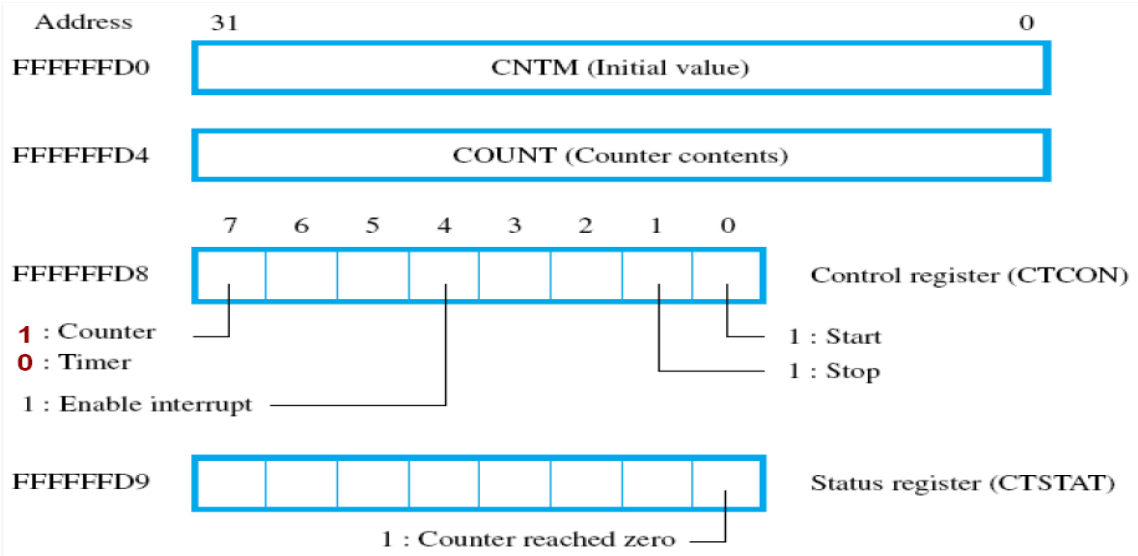
(a) Cache is direct-mapped.

(b) Cache is 2-way set-associative (2 blocks per set) with LRU replacement.

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

END

Counter/Timer Registers



Parallel Port Registers