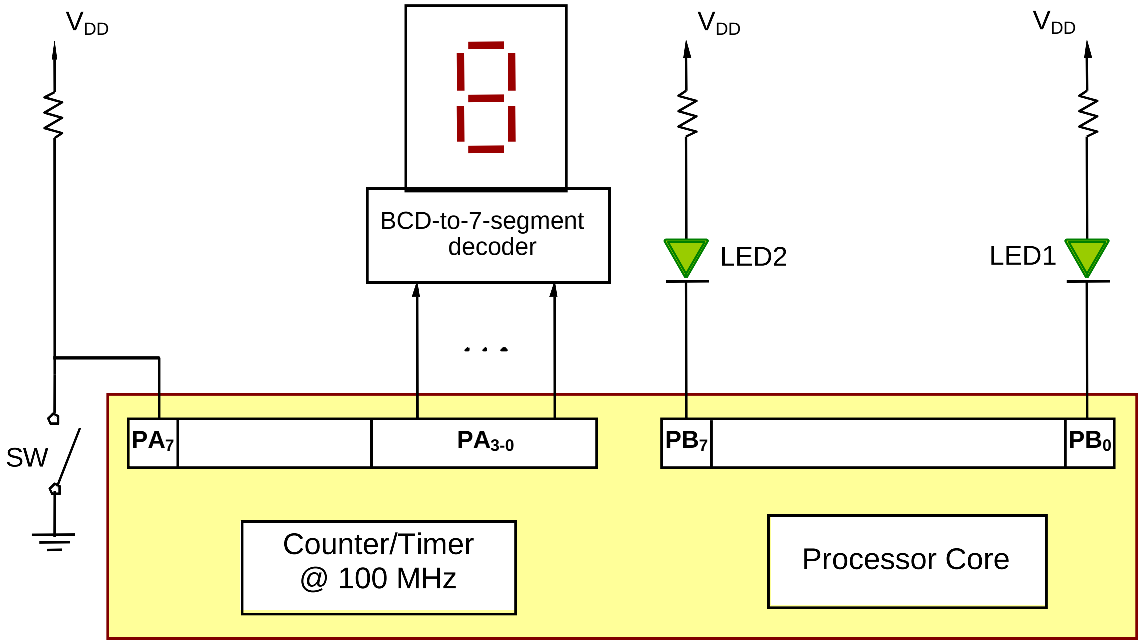
Assignment 2

Due October 9, 17:00

NOTE: Late submissions will **NOT** be accepted. Please submit a single PDF file with your answers via the **ECE 355 Brightspace** webpage.

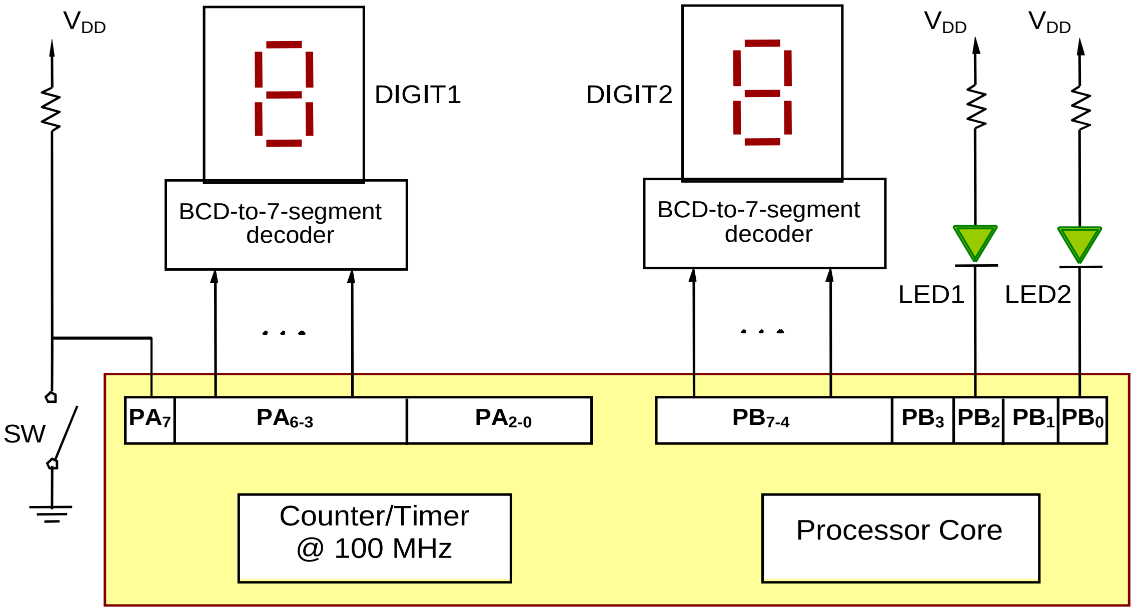
1. [10 points] The textbook's microcontroller is used in a system shown below and is responsible for two tasks: (1) decrementing the displayed digit every second, and (2) alternating between **LED1** and **LED2** being on, whenever the **SW** key has been hit (i.e., pressed and then released). Write the corresponding C program, assuming that the **first task** is an ISR, whose address is stored at memory location **0x20**, and the **second task** is the main program. Also, assume that **PSR[6]** is the processor's interrupt-enable bit, and both **Ports A** and **B** are always ready to be accessed by the processor. Initially, **LED1** is on, **LED2** is off, and the 7-segment display shows **0**.

- *Main Program*: Every time the **SW** key is hit, i.e., pressed and then released (**PA_7** must first become 0 and then 1 again), **LED1** and **LED2** must swap their states: if **LED1** is on and **LED2** is off, then **LED1** becomes off and **LED2** becomes on, or if **LED1** is off and **LED2** is on, then **LED1** becomes on and **LED2** becomes off. **Note: LED1** and **LED2** are never both on, or both off.
- *ISR*: The 100-MHz Counter/Timer must be configured to generate interrupts every second, and its ISR must decrement the displayed digit (decrementing **0** gives **9**).



2. [10 points] The textbook's microcontroller is used in a system below and is responsible for two tasks: (1) incrementing either **DIGIT1** (when **LED1** is on) or **DIGIT2** (when **LED2** is on) every second, and (2) alternating between **LED1** and **LED2** being on, whenever the **SW** key is hit (i.e., pressed and then released). 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 location **0x20**. Also, assume that **PSR[6]** is the processor's interrupt-enable bit, and **Ports A** and **B** are always ready to be accessed by the processor. Initially, **LED1** is on, **LED2** is off, and both **DIGIT1** and **DIGIT2** show **0**.

- *Main Program*: The 100-MHz Counter/Timer must be used to measure one-second delays. Every second, the main program must increment **DIGIT1** if **LED1** is on, or increment **DIGIT2** if **LED2** is on. (Note: incrementing **9** gives **0**.)
- *ISR*: Port A must be configured to generate interrupts when **PAIN** is updated. Whenever **PA$_7$** first becomes 0 and then 1 again, **LED1** and **LED2** must swap their states: if **LED1** is on and **LED2** is off, then **LED1** becomes off and **LED2** becomes on, and vice versa. (Note: **LED1** and **LED2** are never both on, or both off.)



3. [5 points] Table below specifies a set of **4** independent pre-emptive tasks to be executed by a single processor. Show the task schedule using *Earliest Deadline First* (**EDF**) priority assignment. 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	20	4	16	0
T2	40	4	32	0
T3	40	12	35	0
T4	80	24	70	0