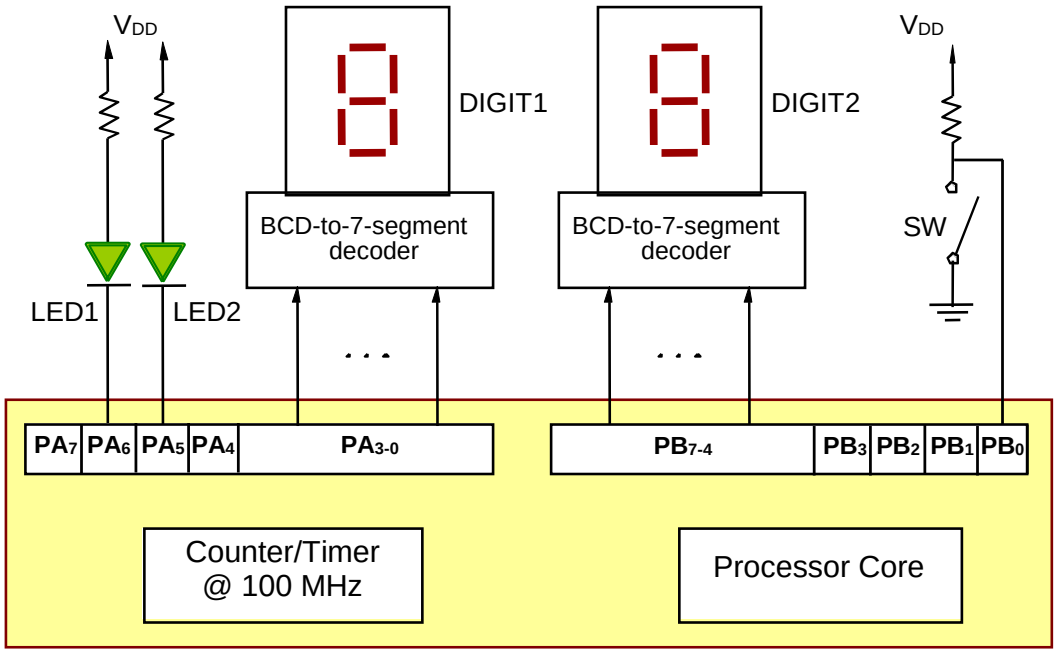
Assignment 2

Due October 9, 23:59

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 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 **second task** is the main program, and the **first task** is an ISR whose address is stored at location **0x20**. Also, assume that bit **6** of the processor status register (i.e., **PSR[6]**) is the processor's interrupt-enable bit, and **Ports A** and **B** are always ready to be written by the processor. Initially, **LED1** is on, **LED2** is off, and both **DIGIT1** and **DIGIT2** show **0**.

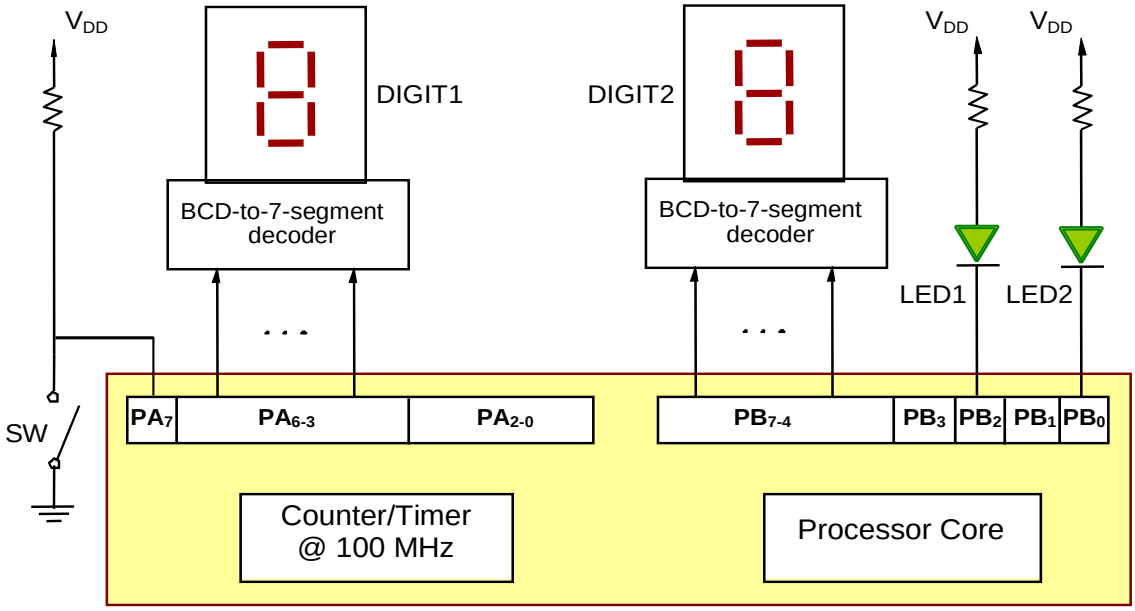
- *Main Program*: Whenever **PB_0** first becomes 0 and then 1 again, **LED1** and **LED2** must swap their states: if **LED1** is on and **LED2** is off, then **LED1** becomes off and **LED2** becomes on, and vice versa. (Note: **LED1** and **LED2** should never be both on, or both off.)
- *ISR*: The 100-MHz Counter/Timer must be configured to generate interrupts every second. Its ISR must decrement **DIGIT1** if **LED1** is on, or decrement **DIGIT2** if **LED2** is on. (Note: decrementing **0** gives **9**.)



2. [10 points] The textbook's microcontroller is used in a system shown 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 bit **6** of the processor status register (i.e., **PSR[6]**) is the processor's interrupt-enable bit, and **Ports A** and **B** are always ready to be written 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 $\mathbf{PA_7}$ 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] The table below specifies a set of independent pre-emptive tasks to be executed by a single processor. Show the task schedule using the **Earliest Deadline First** (EDF) priority assignment. Note: If some tasks happen to have the same EDF priority, break such ties using **Rate Monotonic** (RM) prioritization.

Task $\mathbf{T_i}$	Period $\mathbf{P_i}$	WCET $\mathbf{C_i}$	Deadline $\mathbf{D_i}$	Initial Delay ϕ_i
T1	30	10	30	0
T2	40	10	40	0
T3	60	10	50	0
T4	120	15	100	0