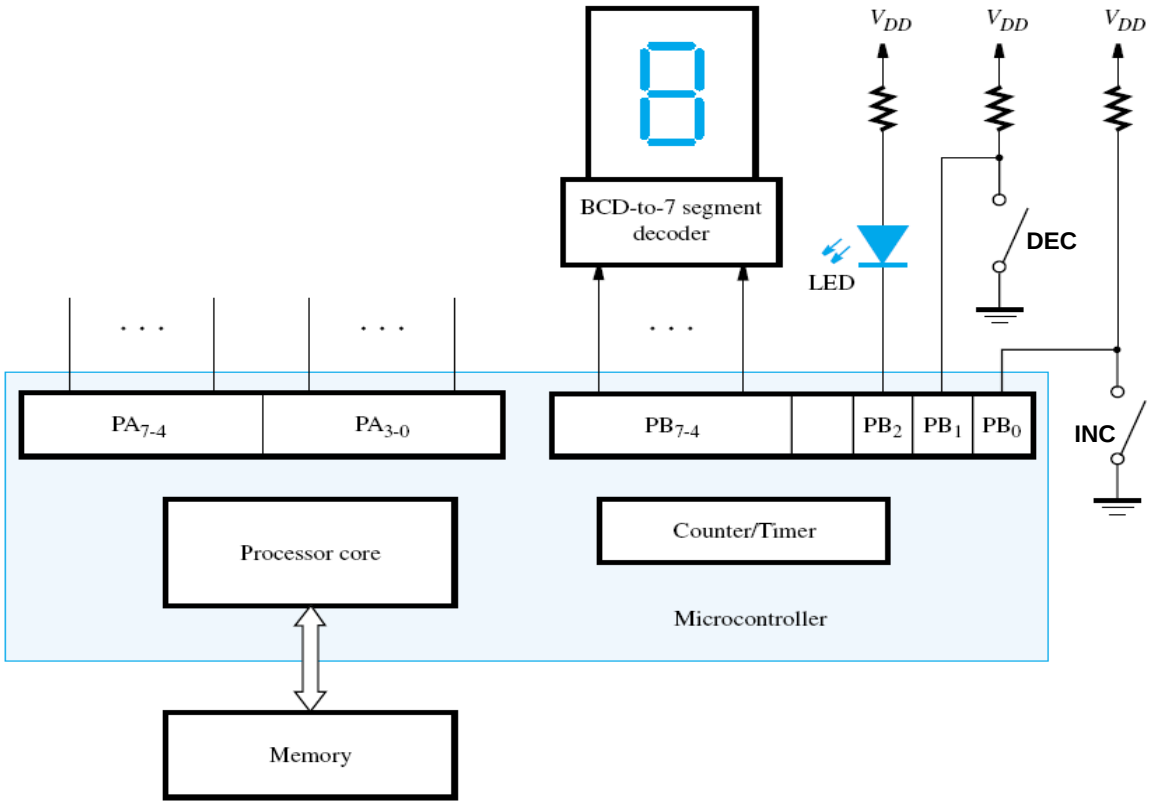
Assignment 1

Due October 3, 13:59pm

NOTE: Late submissions will **NOT** be accepted. Please put your solutions in the CENG 355 **drop-box** (ELW, second floor) – they will be collected at **14:00**.

1. [10 points] The textbook's microcontroller is used in a system shown below and is responsible for 2 tasks: (1) periodically turning the LED on for one second and then off for one second, and (2) incrementing or decrementing the displayed digit when **INC** or **DEC** has been pressed. 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 B** is always ready to receive data from the processor. Initially, the 7-segment display shows digit **0**, and the LED is off.

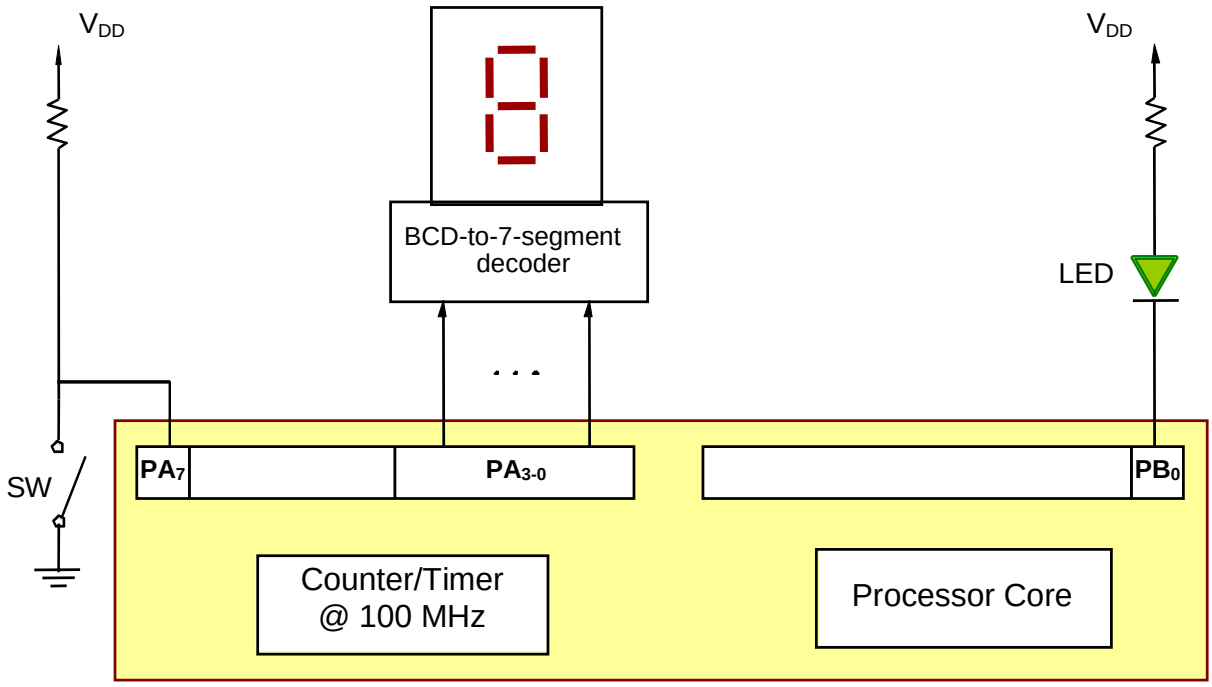
- *Main Program*: Every second the LED state must be flipped: if the LED is on, the program must turn it off, and vice versa. You must use the 100-MHz Counter/Timer to measure one-second delays.
- *ISR*: **Port B** must be configured to generate interrupts when **PBIN** is updated. If **INC** has been pressed, the displayed digit must be incremented; if **DEC** has been pressed, the displayed digit must be decremented. Incrementing **9** gives **0**, and 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) flipping the LED on/off state every second, and (2) incrementing the displayed digit every time the **SW** key has been 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 memory 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 accessed by the processor. Initially, the 7-segment display shows digit **0**, and the LED is off.

• *Main Program*: Every second the LED state must be flipped: if the LED is on, it must be turned off, and vice versa. You must use the 100-MHz Counter/Timer to measure one-second delays.

• *ISR*: **Port A** must be configured to generate interrupts whenever **PAIN** is updated. The ISR must increment the displayed digit, provided that the following condition has been satisfied: **SW** has been pressed and then released (i.e., $\mathbf{PA_7}$ must first become 0 and then 1 again). Note: incrementing **9** gives **0**.



3. [5 points] Assume that some I/O device has the maximum data transfer rate of $\mathbf{R_{I/O} = 4\ MB/s}$. During DMA, the data is transferred in blocks of $\mathbf{d_{I/O\text{-}DMA} = 4\ KB}$ at a time. To initiate a DMA transfer, the CPU takes $\mathbf{N_{DMA\text{-}start} = 1{,}600}$ clock cycles; to complete it, the CPU takes $\mathbf{N_{DMA\text{-}end} = 800}$ clock cycles. If polling is used, the data is transferred in blocks of $\mathbf{d_{I/O} = 32\ B}$ at a time, when the device is ready. To perform a poll, the CPU takes either $\mathbf{N_{poll\text{-}ready} = 800}$ clock cycles (when the device is ready), or $\mathbf{N_{poll\text{-}not\text{-}ready} = 400}$ clock cycles (when the device is not ready). At what activity percentage of the I/O device does the DMA cost become **400 times** cheaper than the polling cost?