Wahhaj Javed
muj975

[Code Documentation]

- The code uses three timers. Timer 0 for blinking the LED, timer 1 for debouncing SW1 and timer 2 for debouncing SW2

- Timer 0 counts down. Timeout is detected using the RIS register

- Timers 1 and 2 count up and are reset to 0 after timeout

- The button debouncing algorithm is documented in detail in the check_sw1() function. Basically, after the first button release is detected, all signals from that button are ignored for 5ms

[2b]

Blocking code is code that prevents execution of any other code even when its not doing anything useful. Busy waiting or waiting on I/O would be examples of blocking code.

Non-blocking code allows other code to run when it can't proceed. Non-blocking I/O for example would initiate I/O and then allow other code to run while it waits for the I/O to complete.

[2d]

For a 80MHz clock, 1/80E6 = 12.5 ns would pass per clock tick.
16 bit timer would reach max value in (2^16 - 1) / 80E6 = 0.819 ms
32 bit timer would reach max value in (2^32 - 1) / 80E6 = 53687 ms
64 bit timer would reach max value in (2^64 - 1) / 80E6 = 7311.78 years