```
movlw   b'00111000'  ;set cpu clock speed
movwf   OSCCON
```

This configures the PIC MCU to run at 500 kHz. The working register (WREG) is used to move bytes into the register. Upon default, if this register was not written to, the PIC16F1829 would also run at 500 kHz. Other PIC devices are different, however, so this should always be written to in the first few lines of code. The PIC MCU will now execute with each instruction taking eight microseconds, as seen in Equation 3-1:

EQUATION 3-1: DELAY SPEED

$$Instruction\ time = \frac{\frac{1}{FOSC}}{4} = \frac{\frac{1}{500kHz}}{4} = 8\ \mu S$$

In order to make the LED blink, the program needs some way of turning on the LED, waiting for a set amount of time, and then turning it off for the same period. This can be achieved by using the on-board RAM.

EXAMPLE 3-9:

```
    cblock 0x70     ;shared memory location that is accessible from all banks
Delay1              ; Define two file registers for the delay loop in shared memory
Delay2
    endc
```

Remember that CBLOCK allocates user memory. The number after CBLOCK is the address of where to put the memory. 0x70 is the address of shared memory across all banks in the enhanced mid-range core. Only 16 bytes can be saved here. Now the program does not need to change banks when using any of these variables. The rest of the lessons will be using variables stored here on the PIC16 and in access RAM for the PIC18. Two variables will be stored here to write the following delay loop.

EXAMPLE 3-10:

```
  bsf    LATC, 0       ; turn LED on
OndelayLoop:
  decfsz Delay1,f      ; Waste time.
  bra    OndelayLoop   ; The Inner loop takes 3 instructions per loop * 256 loops = 768
instructions
  decfsz  Delay2,f     ; The outer loop takes an additional 3 instructions per lap * 256 loops
  bra     OndelayLoop ; (768+3) * 256 = 197376 instructions / 125K instructions per second =
                      1.579 ;sec.
  bcf     PORTC,0      ; Turn off LED C0 - NOTE: do not need to switch banks with 'banksel'
                      since ;bank0 is still selected
OffDelayLoop:
  decfsz  Delay1,f     ; same delay as above
  bra     OffDelayLoop
  decfsz  Delay2,f
  bra     OffDelayLoop
  bra     MainLoop     ; Do it again...
```

The bra Loop backs up and repeats. This loop takes three instruction times; one for the decrement and two for the bra, and the counter will force it to go around 256 times, which takes a total of 768 instruction times to execute. Even that is still too fast for the eye to see. It can be slowed down even more by adding a second loop around this one. The inner loop still takes 768 cycles plus three for the outer loop, but now it is executed another (768+3) * 256 = 197376 instructions/125K instructions per second = 1.579s.

goto and bra instructions take two instructions due to the pipelined design of the processor. The processor fetches the next instruction while executing the current instruction. When a program occurs, the already fetched instruction located after the goto or bra is not executed. Instead, a NOP is executed while the instruction located at the destination is fetched.