Understanding the P1.asm template

- Contents
  - Microcontroller ports
  - Macro instructions
  - Timer0 used as counter

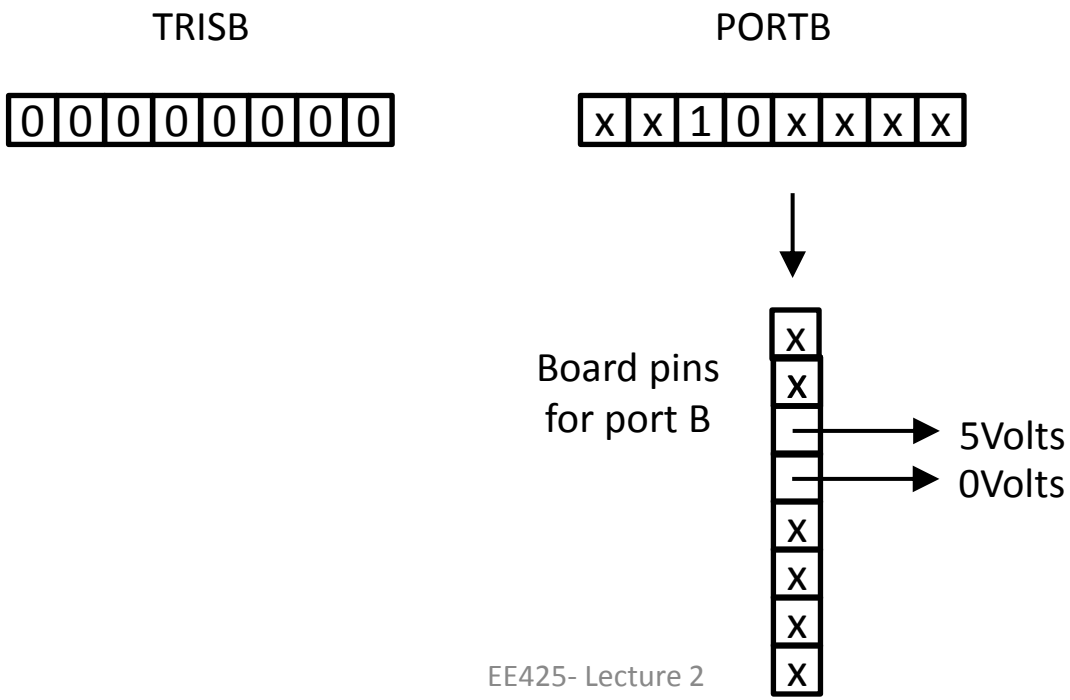
PIC18F4520 I/O Ports

- Five I/O ports
  - PORTA through PORTE
  - Each port has eight (8) available pins
  - Can be set as either input or output
  - Addresses already assigned to these ports
  - Each port is identified by its SFR

  - Set port pins as either input or output by writing to TRISx, where “x” is the port name (A-E).
    - For input, write a ‘1’ to that pin
    - For output, write a ‘0’ to that pin

Example

```
movlw   0               ;Load WREG with 0
movwf   TRISB           ;Set all pins of PORTB as outputs
movlw   B'xx10xxxx'     ;Byte for output at pins B5 and B4
movwf   PORTB           ;Output byte at port B
```



MACROS

- Sequence of instructions assigned by a name that could be used anywhere in the program
- A macro begins with *macro* and *endm* directives

Sintax:

```
Macro_Name  macro parameters
                <macro body>
            endm
```

Example of Macro-instruction definition:

```
MOVLF macro literal, dest
        movlw literal
        movwf dest
        endm
```

.

Usage:

```
MOVLF 250, ALIVECNT
```

```
movlw 250
movwf ALIVECNT
```

Example (cont'd)

Code for output byte at port B:

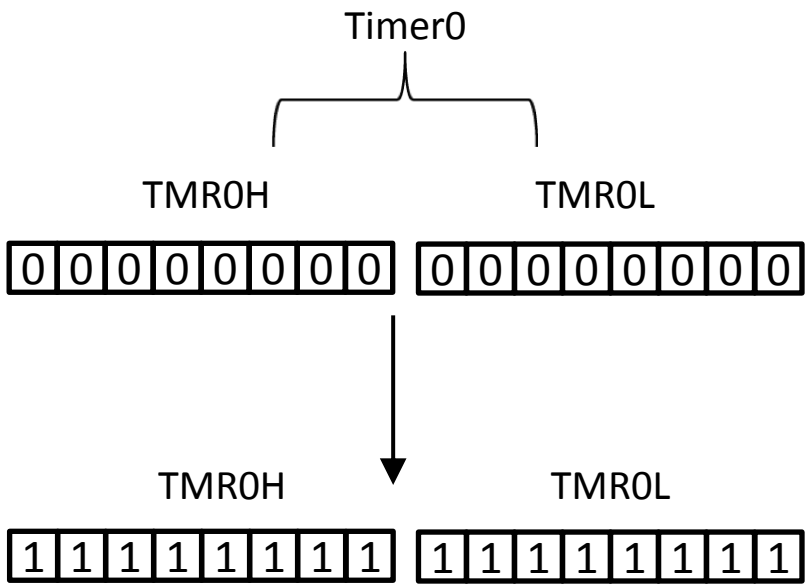
```
movlw   0               ;Load WREG with 0
movwf   TRISB           ;Set all pins of PORTB as outputs
movlw   B'xx10xxxx'     ;Byte for output at pins B5 and B4
movwf   PORTB           ;Output byte at port B
```

Using macros:

```
MOVLF  0, TRISB                 ;Set up PORTB as output
MOVLF B'xx10xxxx', PORTB        ; Output byte at port B
```

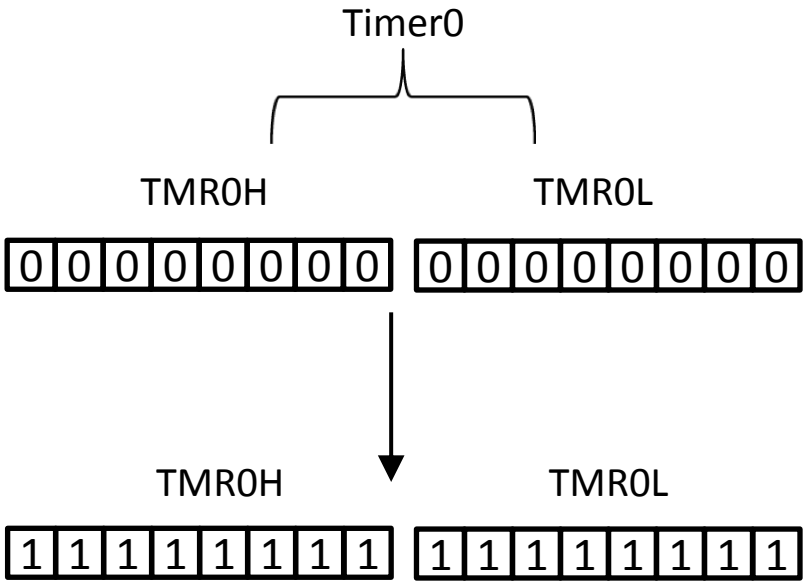
TIMER0 as counter

- 16 bit register:TMR0H:TMR0L
- Timer0 counts from 0x0000 to 0xffff (65,536 clock cycles)
- Bit TMR0IF of register INTCON is set every time timer0 rolls over from 0xffff to 0x000
- T0CON sets the parameters for the Timer0.



TIMER0 as counter (cont'd)

- Synchronize Timer0 with Internal Clock of MCU with freq = 2.5 MHz
- Timer0 will reset in:
  - (0.4 µs)*(65536) = 26214.4 µs
  - Or approximately 26.21ms
- Suppose task requires 10ms rollover
  - 10ms/0.4 µs = 25000 cycles
  - Then remove 65536-25000= 40536 cycles from sequence
- Removal of cycles takes 12+2 cycles
  - Must remove 65536-25000+12+2



TIMER0 configuration