Embedded Systems Design (0630470)

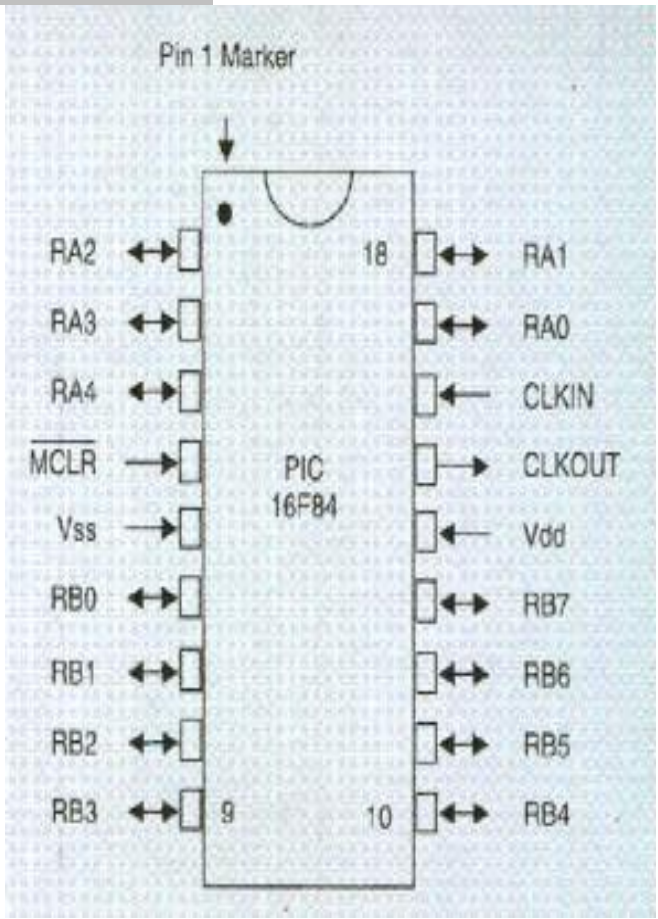
Lecture 11

Microcontroller Interfacing

Prof. Kasim M. Al-Aubidy
Philadelphia University-Jordan

Hardware Design

- PIC 16F84 pin out:
  - It is supplied in an 18-pic DIL chip.
  - Some of the pins have dual functions.
  - It has TWO ports, Port A & Port B.
  - A summary of the pin functions is provided in the next table.



Pin	Label	Function	Comment
14	Vdd	Positive supply	+5 V nominal, 3–6 V allowed
5	Vss	Ground supply	0 V
4	!MCLR	Master clear	Active low reset input
16	CLKIN	Clock input	Connect RC clock components to 16
15	CLKOUT	Clock output	Connnect crystal oscillator to 15 and 16
17	RA0	Port A, Bit 0	Bidirectional Input/Output
18	RA1	Port A, Bit 1	Bidirectional Input/Output
1	RA2	Port A, Bit 2	Bidirectional Input/Output
2	RA3	Port A, Bit 3	Bidirectional Input/Output
3	RA4	Port A, Bit 4	Bidirectional Input/Output + TMR0 Input
6	RB0	Port B, Bit 0	Bidirectional Input/Output + Interrupt Input
7	RB1	Port B, Bit 1	Bidirectional Input/Output
8	RB2	Port B, Bit 2	Bidirectional Input/Output
9	RB3	Port B, Bit 3	Bidirectional Input/Output
10	RB4	Port B, Bit 4	Bidirectional Input/Output + Interrupt Input
11	RB5	Port B, Bit 5	Bidirectional Input/Output + Interrupt Input
12	RB6	Port B, Bit 6	Bidirectional Input/Output + Interrupt Input
13	RB7	Port B, Bit 7	Bidirectional Input/Output + Interrupt Input

- **NOTES:**

- Timer Zero Register (TMR0): File Register # 01
- Program Counter: File Register # 02
- Status or Flag Register: File Register # 03

- Port A Data Register: File Register # 05
- Port B Data Register: File Register # 06
- Interrupt Control Register (INTCON): File Register # 0B
- General Purpose Register 1: File Register # 0C
- General Purpose Register 2: File Register # 0D

.

.

- General Purpose Register 68: File Register # 4F

- Working Register: W register: it is 8-bit data register used to hold data that is currently being worked. It is separate from the file register set.

- W register and General Purpose Registers are in RAM Bank (bank 0). These registers can be directly accessed.
- Bank 1 Registers (TRISA, TRISB and PCLATH) can not be directly accessed. In this case, special instructions are needed to load them. There are TWO ways to do this:

1. Simple Method: it requires the 8-bit code to be loaded to be placed in W first, and then moved into Bank 1.
2. Bank Selection Method: using operation BANKSEL which selects the bank that the specified register is in.

Example:

```
BANKSEL   TRISB    ;select bank containing TRISB bank 1.
CLRW               ;load code for all outputs.
MOVWF     TRISB    ;set port B as output.
BANKSEL   PORTB    ;re-select bank containing PORTB, bank 0
```

TRISA: Port A Data Direction Register, File Register # 85.

TRISB: Port B Data Direction Register, File Register # 86.

•The data direction register TRISA is loaded by placing the required code in W register and then using instruction TRIS05 OR TRIS06 for port A & port B respectively.

•The program instruction 'TRIS06' moves the data direction code from W to TRISB register.

STATUS: Status or flag register : File Register #03

Bit	Label	Name	Function
0	C	Carry flag	Set if register operation causes a carry out of bit 8 of the result (8-bit operations)
1	DC	Digit carry flag	Set if register operation causes a carry out of bit 3 of the result (4-bit operations)
2	Z	Zero flag	Set if the result of a register operation is zero
3	PD	Power down	Cleared when the processor is in Sleep mode
4	TO	Time out	Cleared when Watchdog Timer (WDT) times out
5	RP0	Register bank	RP0 selects File Registers 00–7F or 80–FF
6	RP1	Select bits	RP1 not used
7	IRP		IRP not used

- **TMR0: Timer Zero Register: File Register # 01**
- **OPTION: Option Register: File Register # 81**

Bit	Label	Name	Function
0	PS0	Prescaler rate Select bit 0	These 3 bits form a 3-bit code to select one of 8 prescale values for the counter/timer TMR0 or WDT
1	PS1	Prescaler rate Select bit 1	
2	PS2	Prescaler rate Select bit 2	
3	PSA	Prescaler Assignment	Assigns prescaler to WDT or TMR0
4	T0SE	Timer zero Source edge select	Select rising or falling edge trigger for T0CKI input at RA4
5	T0CS	Timer zero Clock source select	Select timer/counter input as RA4 or internal clock
6	INTEDG	Interrupt edge select	Select rising or falling edge trigger for RB0 interrupt input
7	RBPU	Port B pull-up enable	Enable pull-ups on Port B pins so input data defaults to '1'

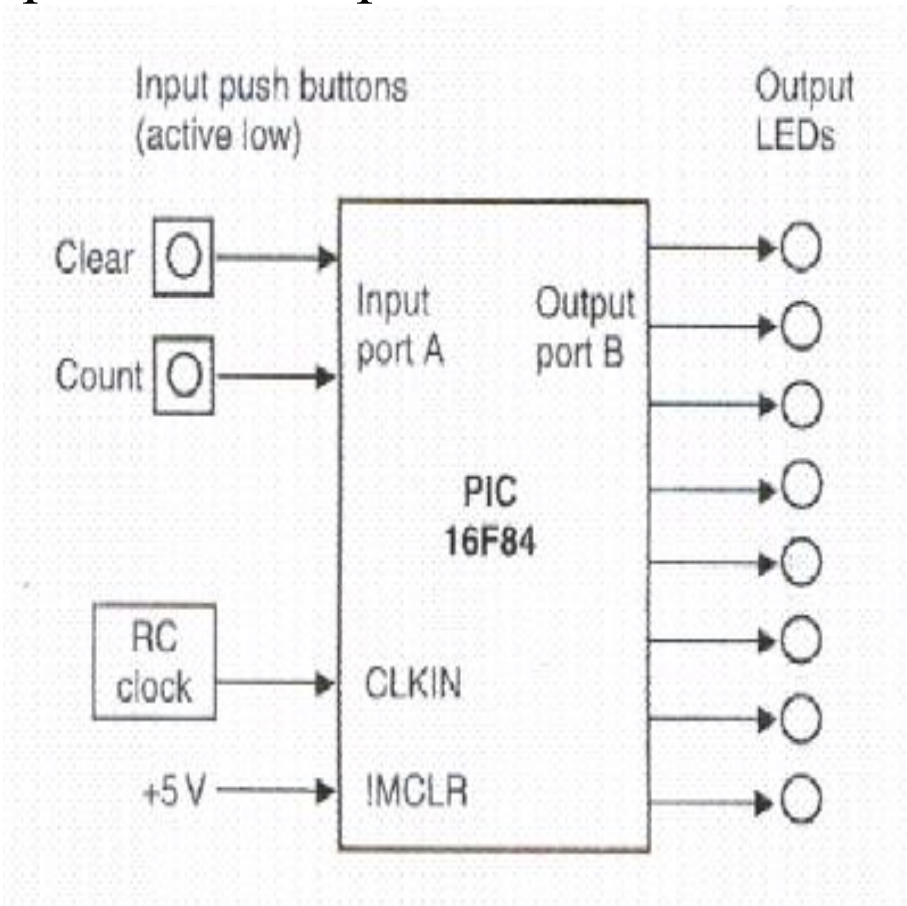
•INTCON: Interrupt control Register: File Register # 0B

Bit	Label	Name	Function
0	RBIF	Port B change Interrupt flag	Set when any one of RB4–RB7 changes state
1	INTF	RB0 pin Interrupt flag	Set when RB0 detects interrupt input
2	T0IF	Timer overflow Interrupt flag	Set when timer TMR0 rolls over from FF to 00
3	RBIE	Port B change Interrupt enable	Set to enable Port B change interrupt
4	INTE	RB0 pin Interrupt enable	Set to enable RB0 interrupt
5	T0IE	Timer overflow Interrupt enable	Set to enable timer overflow interrupt
6	EEIE	Data EEPROM write Interrupt enable	Set to enable interrupt on completion of write operation to non-volatile data memory
7	GIE	Global interrupt Enable	Enable all interrupts which have been selected

Example: Simple PIC Application

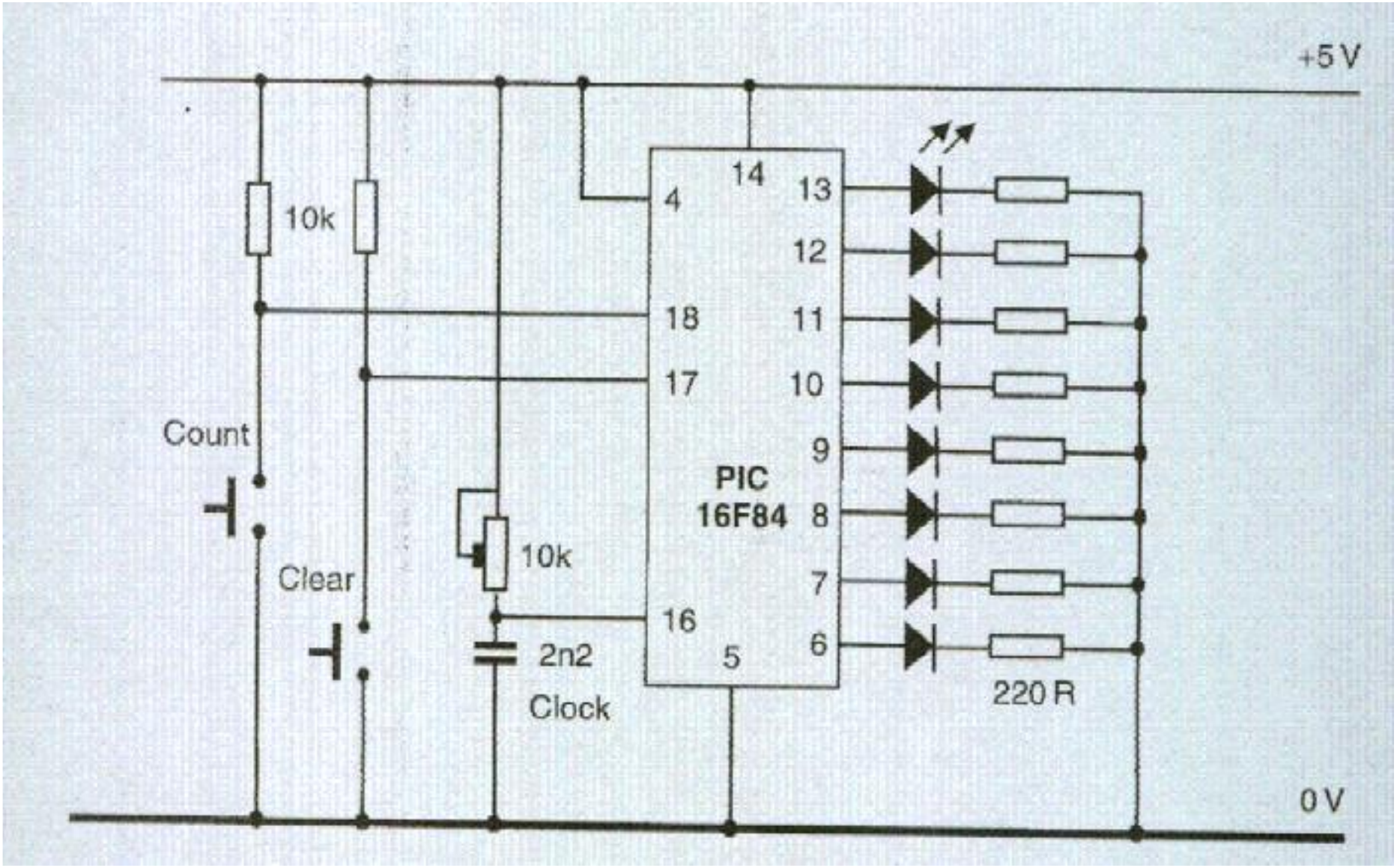
Design a PIC-based system that displays a binary count of a sequence of pulses under the control of two push button switches. One input will start the output sequence when pressed. The other input will clear the output.

Pin	Connection
Vss	0V
Vdd	+5V
!MCLR	+5V
CLKIN	CR clock circuit
CLKOUT	Not connected (n/c)
RA0	Reset switch
RA1	Count switch
RA2	n/c
RA3	n/c
RA4	n/c
RB0	LED bit 0
RB1	LED bit 1
RB2	LED bit 2
RB3	LED bit 3
RB4	LED bit 4
RB5	LED bit 5
RB6	LED bit 6
RB7	LED bit 7



SOLUTION:

1. Hardware Design



Software Design:

Assembly Program:

```
Porta     EQU    05     ;port A data register
Portb     EQU    06     ;port B data register
timer     EQU    0C     ;register for delay
resetin   EQU    0      ;reset i/p=RA0
runin     EQU    1      ;run i/p=RA1
```

Prot B initialization as input port

```
          MOVLW    B'00000000'     ;port b data direction code
          TRIS     Port B          ;Load the DDR code into F86
          GOTO     reset
reset:    CLRF     PortB           ;clear port b data
start:    BTFSS    Porta,  resetin ;test RA0 input button
          GOTO     reset           ;and reset Port B if pressed
          BTFSC    porta,  runin   ;test RA1 input button

          GOTO     start           ;and run count if pressed
          INCF     portb           ;increment count at port B
          MOVLW    OFF             ;delay count literal

          CALL     delay           ;jump to delay subroutine
          GOTO     start           ;repeat main loop always
```

CONTINUE

```
delay:    MOVWF    timer      ;copy W to timer register

down:     DECFSZ   timer      ;decrement timer register

          GOTO     down       ;and repeat until zero

          RETURN              ;jump back to main program

          END                 ;terminate source code
```