



**Embedded Systems Design
(0630470)**

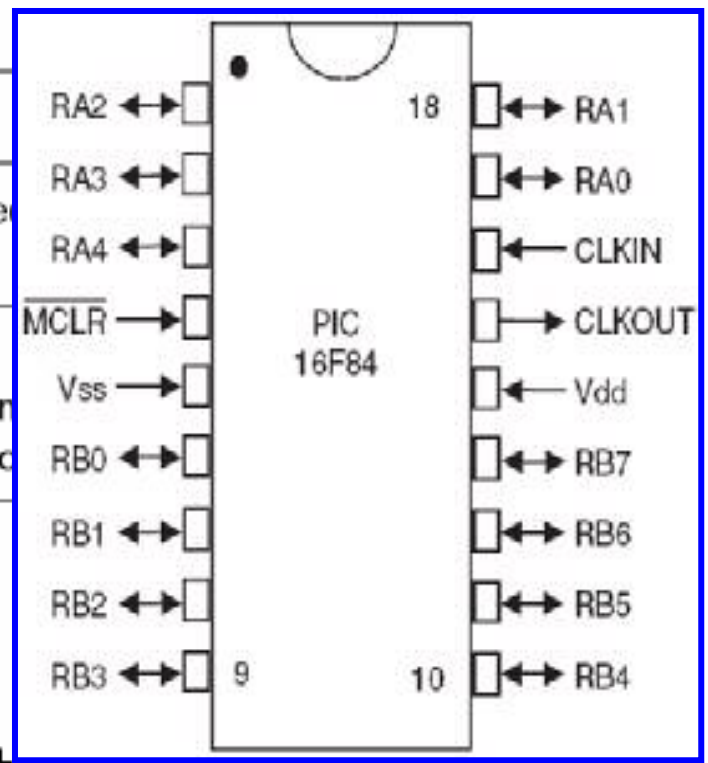
Lecture 13

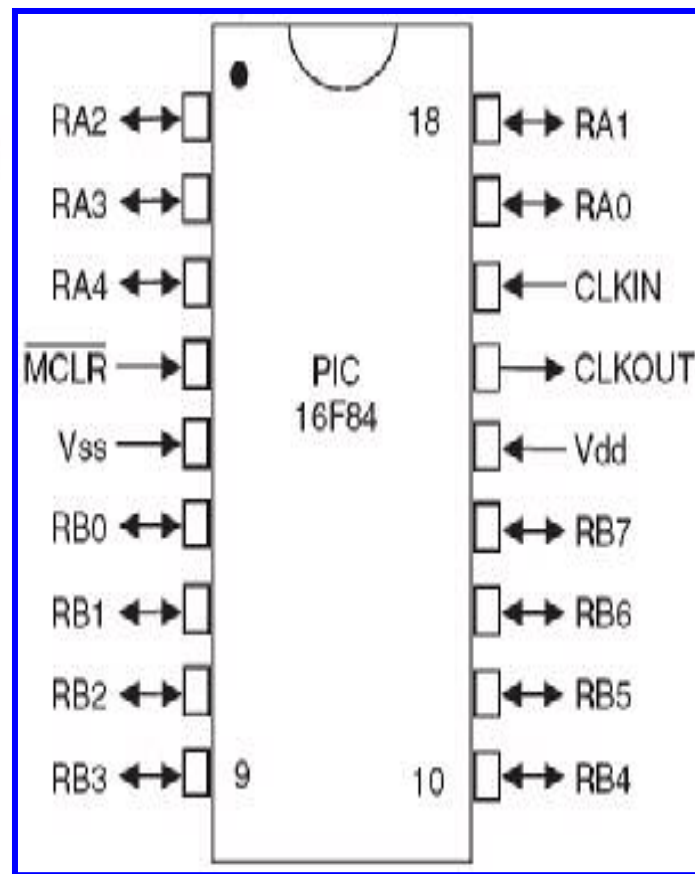
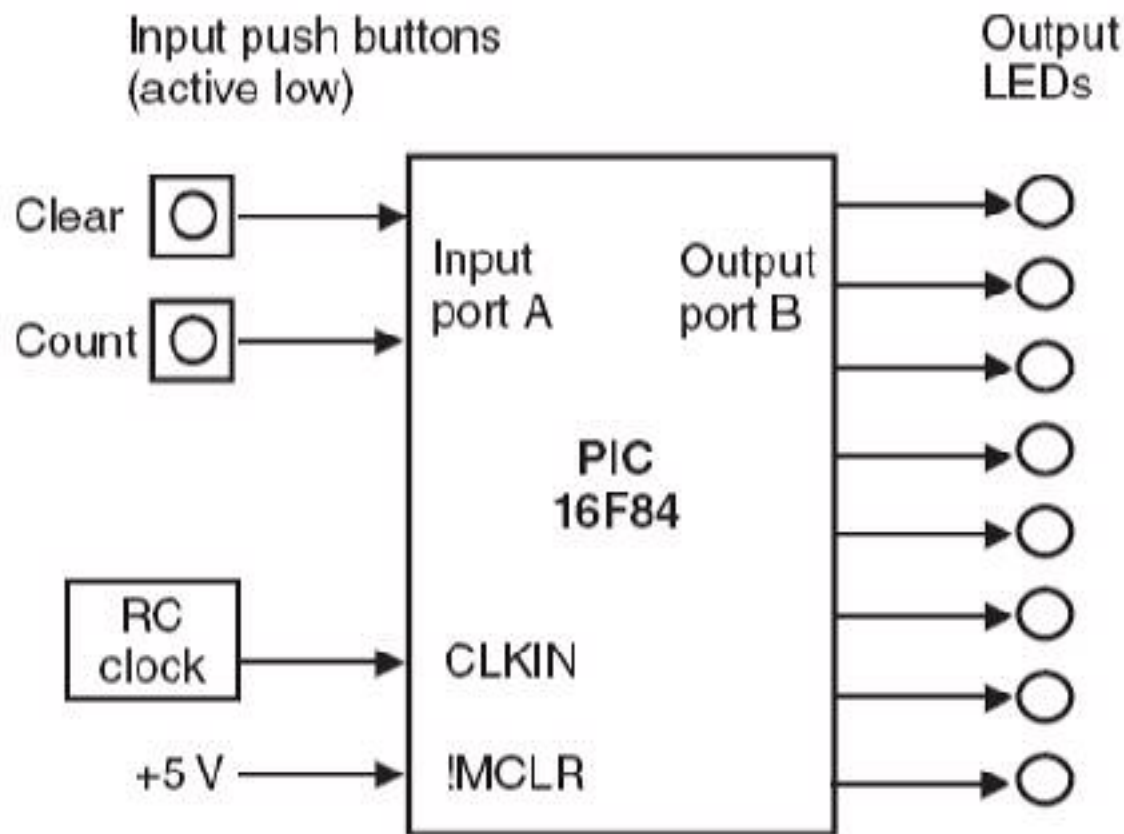
**Microcontroller Interfacing &
Programming**

Prof. Kasim M. Al-Aubidy
Philadelphia University

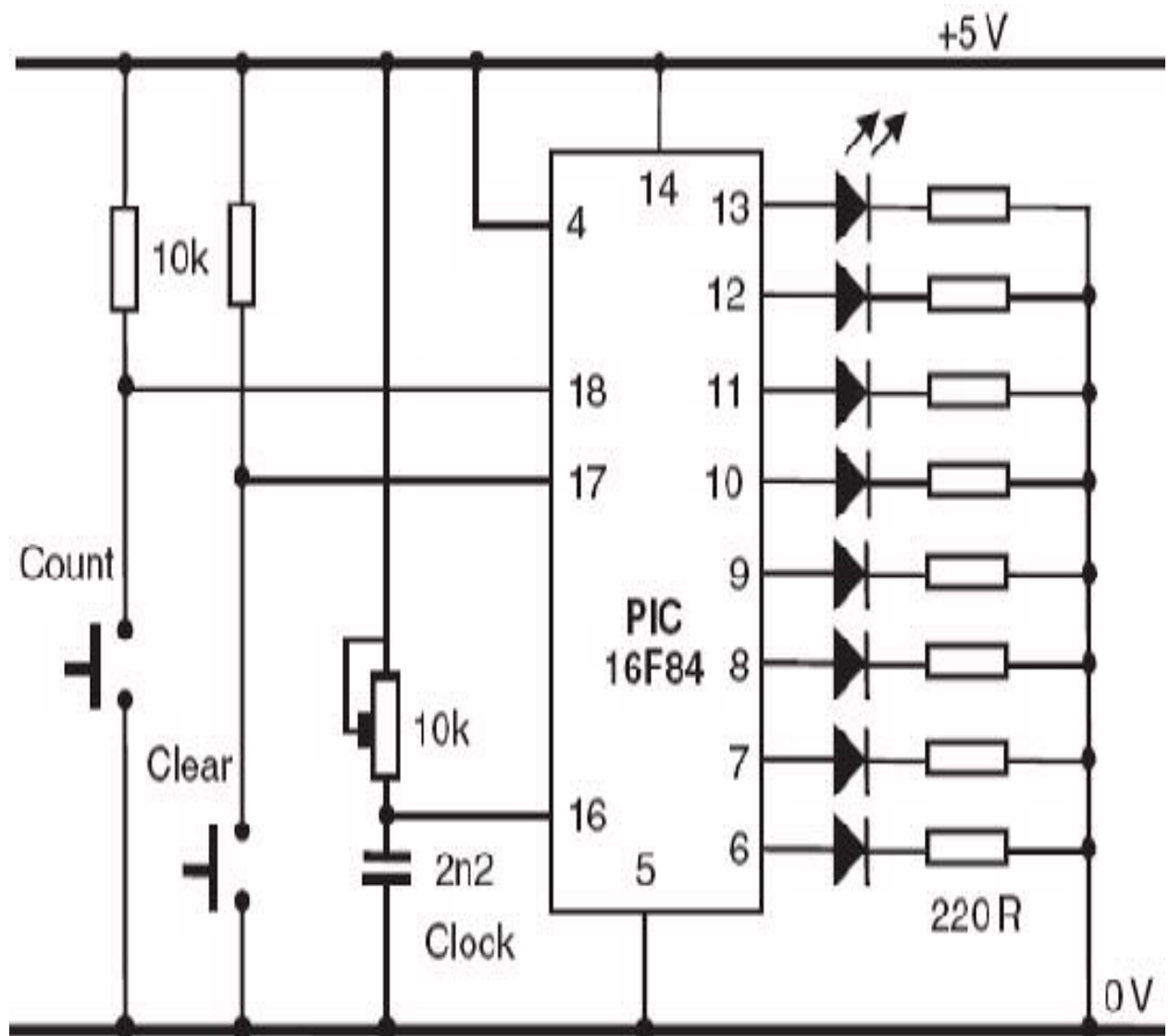
PIC 16F84 pins arranged by function

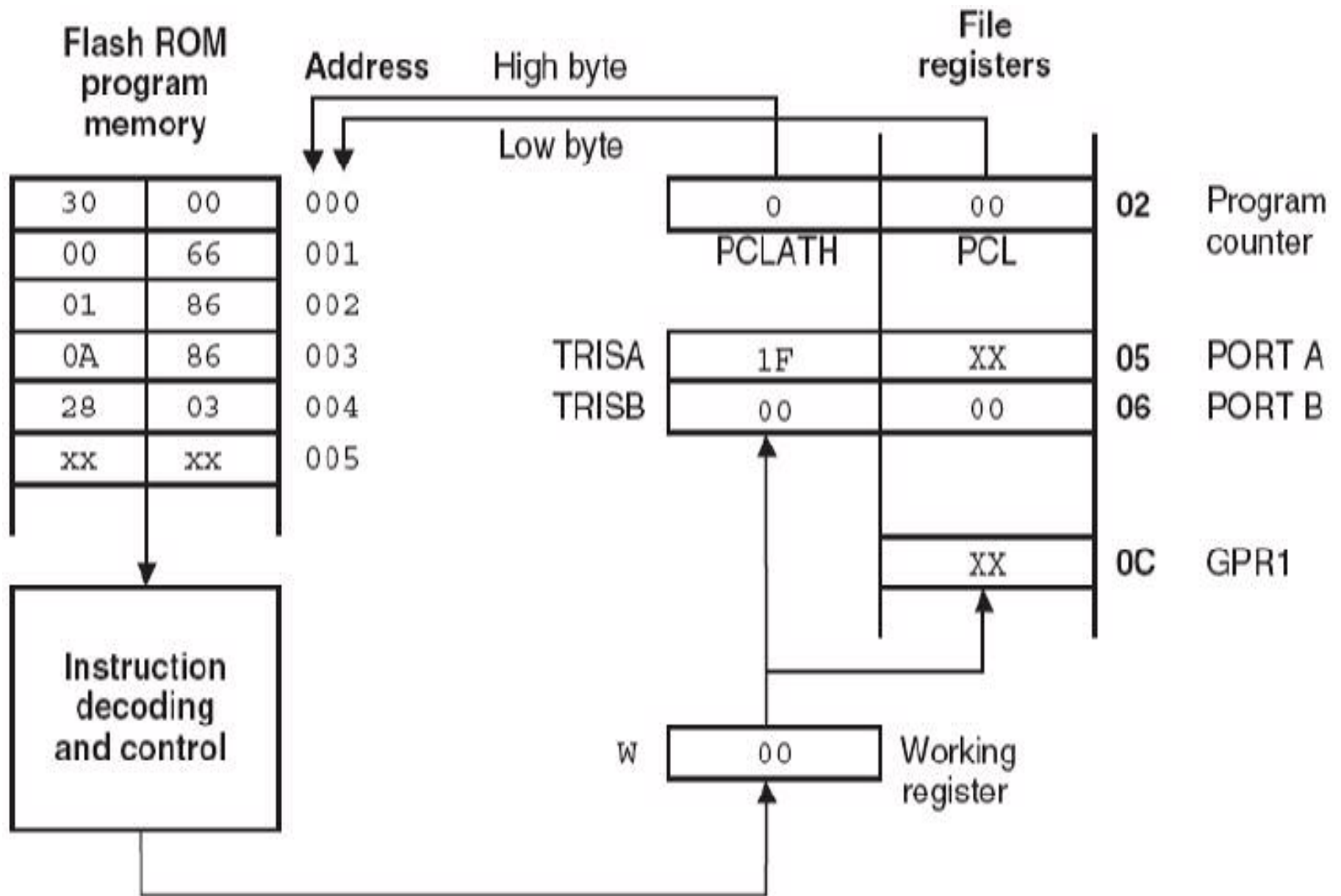
Pin	Label	Function	Comment
14	Vdd	Positive supply	+5V nominal, 3-6 V allowed
5	Vss	Ground supply	0 V
4	MCLR	Master clear	Active low reset input
16	CLKIN	Clock input	Connect RC clock component
15	CLKOUT	Clock output	Connect crystal oscillator to
17	RA0	Port A, Bit 0	Bidirectional Input/Output
18	RA1	Port 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 + make 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





Pin	Connection
V _{ss}	0V
V _{dd}	+5V
IMCLR	+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





<i>Memory address</i>	<i>Machine code instruction</i>	<i>Meaning</i>
000	3000	Load working register (W) with number 00
001	0066	Store W in Port B direction code register
002	0186	Clear Port B data register
003	0A86	Increment Port B data register
004	2803	Jump back to address 0003 above

Top left of edit window
↓

Line number	Column 0	Column 1	Column 2	Column 3
0		MOVLW	00	
1		TRIS	06	
2		CLRF	06	
3		INCF	06	
4		GOTO	03	
5		END		

0	MOVLW	00	Move Literal 00 into W
1	TRIS	06	Move W into TRISB to set Port B as outputs
2	CLRF	06	Clear file register 06 (Port B)
3	INCF	06	Increment file register 06 (Port B)
4	GOTO	03	Jump to address 03 (back to previous instruction)
	END		End of source code - this is not an instruction!

```

;
;           Outputs a binary count at Port B
; .....
allout    EQU      00           ; Define Data Direction Code
portb    EQU      06           ; Declare Port B Address

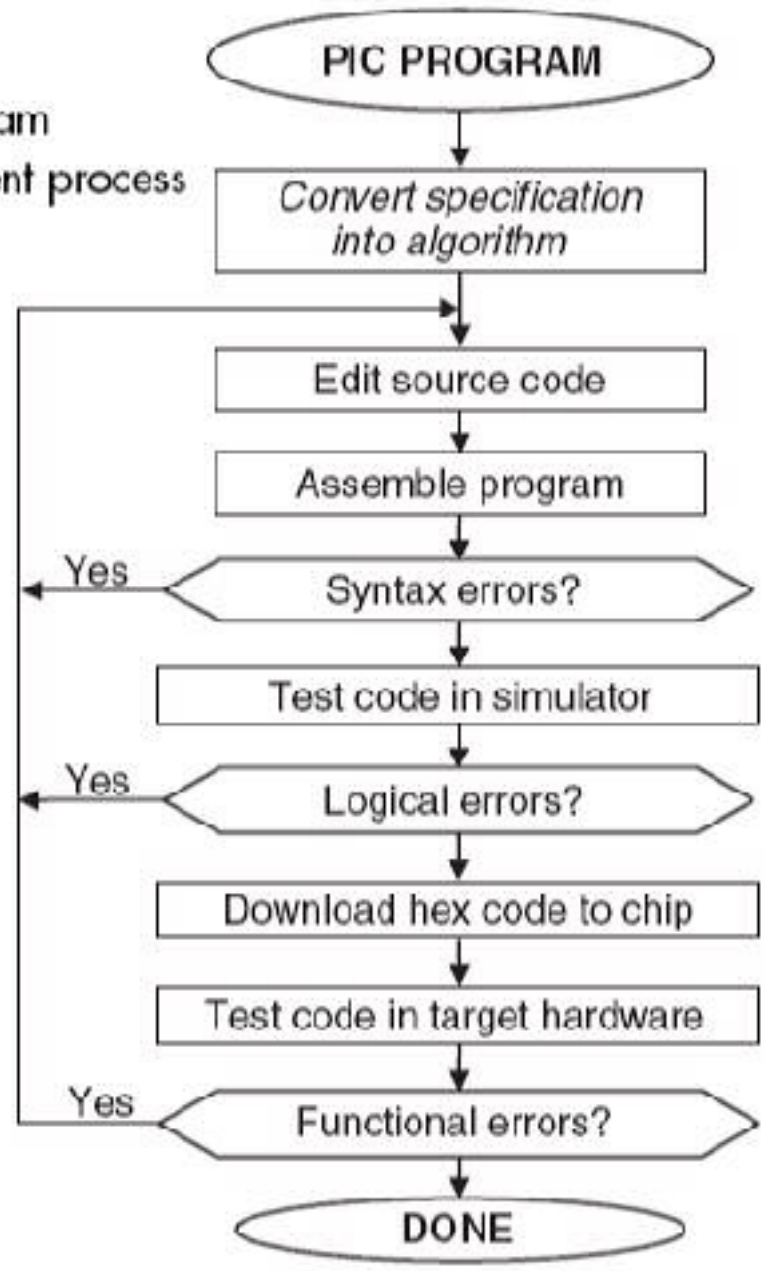
           MOVLW    allout      ; Load W with DDC
           TRIS    portb      ; Set Port B as outputs

again    CLRF    portb      ; Switch off LEDs
           INCF    portb      ; Increment output
           GOTO    again      ; Repeat endlessly

           END              ; Terminate source code

```

PIC program development process



Components of MPLAB development system

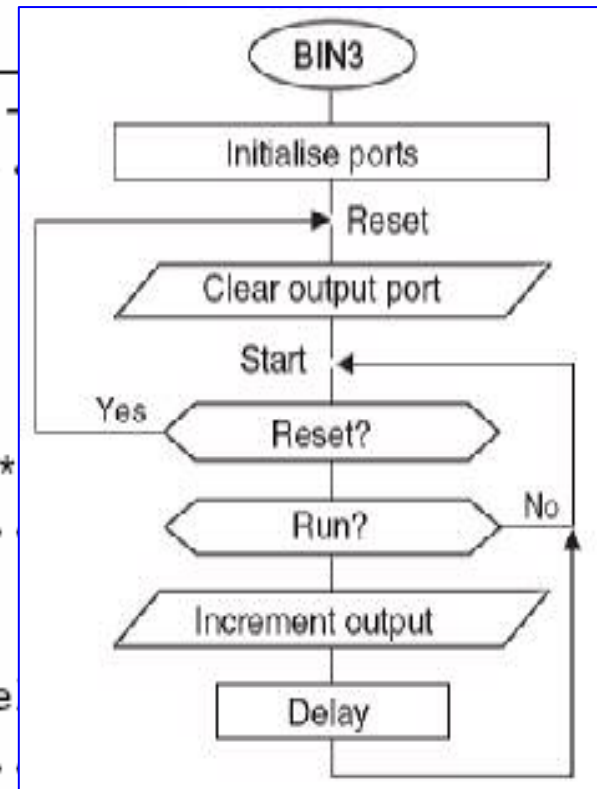
<i>Software tool</i>	<i>Tool function</i>	<i>Files produced or used</i>	<i>File description</i>
Text editor	Used to create and modify source code text file	PROGNAME.ASM	Source code text file
Assembler	Generates machine code from source code, reports syntax errors, generates list and symbol files	PROGNAME.HEX PROGNAME.ERR PROGNAME.LST PROGNAME.COD	Executable machine code Error messages List file with source and machine code Symbol and debug information
Simulator	Allows program to be tested in software before downloading	PROGNAME.HEX PROGNAME.COD	
Programmer	Downloads machine code to chip	PROGNAME.HEX	

BIN3 source code

```

;      BIN3.ASM                M. Bates                12
;.....
;      Slow output binary count is stopped, started
;      and reset with push buttons.
;      Processor = 16F84        Clock = CR, 100kHz
;      Inputs: RA0, RA1        Outputs: RB0 - RB7
;
; *****
; Register Label Equates .....
porta EQU 05                ; Port A Data Register
portb EQU 06                ; Port B Data Register
timer EQU 0C                ; Spare register for de
; Input Bit Label Equates.....
inres EQU 0                 ; 'Reset' input button = RA0
inrun EQU 1                 ; 'Run' input button = RA1
; *****
; Initialise Port B (Port A defaults to inputs).....
        MOVLW 00            ; Port B Data Direction Code
        TRIS portb         ; Load the DDR code into F86

```



; Start main loop

```
reset    CLRF    portb    ; Clear Port B
start    BTFSS   porta,inres ; Test RA0 input bit
        GOTO    reset    ; and reset Port B
        BTFSC   porta,inrun ; Test RA1 input bit
        GOTO    start    ; and run count if
        INCF    portb    ; Increment count
        MOVLW   0FF      ; Delay count literal
        MOVWF   timer    ; Copy W to timer register
down     DECFSZ  timer    ; Decrement timer register
        GOTO    down     ; and repeat until zero
        GOTO    start    ; Repeat main loop always
        END           ; Terminate source code
```

