Chapter 4. Problems

"All programming problems should include design pseudo code either as a separate design document on embedded comments in the code."

1S. Identify four possible causes that would redirect the processor from executing the next instruction located at (PC+2).

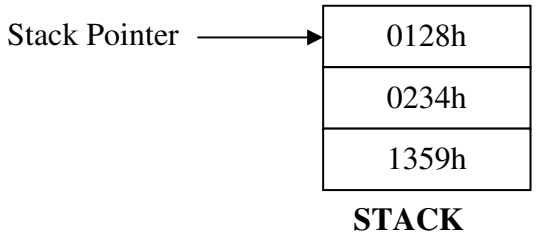
Solution

Go to, Branch, Call, Interrupt, Exception, Reset, Timer …

1U. Identify four instructions that upon execution will result in code located at PC+4 to be executed.

Solution

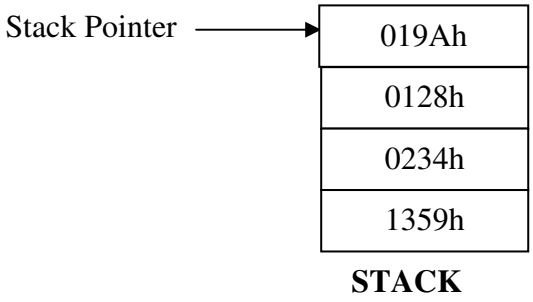
2S. The following diagram shows the current state of the stack.



Show the content of the stack, the content of STKPTR register and TOS registers (TOSU, TOSH, and TOSL) after the execution of following instruction:

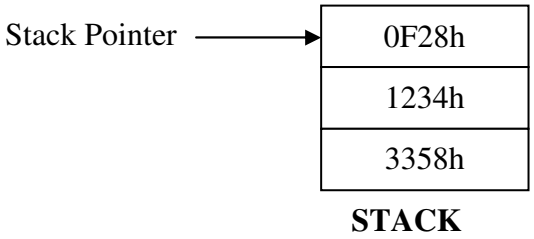
Address	**Content**
0x0198	PUSH

Solution



STKPTR ← 04 ; Stack starts at 1 and each push adds one. (00-00100)
TOSU ← 00 ; PC+2 is pushed on the stack.
TOSH ← 01
TOSL ← 9A

2U. The following diagram shows the current state of the stack.



Show the content of the stack, the content of STKPTR register and TOS registers (TOSU, TOSH, and TOSL) after the execution of following instruction:

Address	**Content**
0x0284	POP

Solution

3S. Assuming "INT1" is set to high priority and "INT2" is set to Low priority, show the first instruction executed after each of "Power on Reset", "INT1" and "INT2" for the following code segment.

```
        org     0x000
        CLRF    PORTA     ; Initialize PORTA
        CLRF    PORTB     ; Initialize PORTB
        BRA     Next_Step
        MOVLW   0x7F      ; Set all A/D Converter Pins as
        MOVWF   ADCON1    ; digital I/O pins
        MOVLW   0x00      ; Value used to initialize data direction
        MOVWF   TRISB     ; Set Port B RB<7:0> as outputs
        MOVLW   0x01      ; Value used to initialize data direction
        MOVWF   TRISA     ; Set Port A Pin 0 RA<0> as input
        MOVLW   0x00      ; W = 0
        MOVWF   COUNT     ; COUNT = WREG
        MOVWF   LASTIN    ; LASTIN = WREG
        BRA     Great_Step
        NOP
        NOP
        NOP
        NOP
        BRA     Another_Step
```

Solution

Location	Program/Code		
	org	0x000	
00 (reset)	CLRF	PORTA	; Initialize PORTA
02	CLRF	PORTB	; Initialize PORTB
04	BRA	Next_Step	
06	MOVLW	0x7F	; Set all A/D Converter Pins as
08 (Int1)	MOVWF	ADCON1	; digital I/O pins
0A	MOVLW	0x00	; Value used to initialize data direction
0C	MOVWF	TRISB	; Set Port B RB<7:0> as outputs
0E	MOVLW	0x01	; Value used to initialize data direction
10	MOVWF	TRISA	; Set Port A Pin 0 RA<0> as input
12	MOVLW	0x00	; W = 0
14	MOVWF	COUNT	; COUNT = WREG
16	MOVWF	LASTIN	; LASTIN = WREG
18 (Int2)	BRA	Great_Step	

```
1A          NOP
1C          NOP
1E          NOP
20          NOP
22          BRA         Another_Step
```

3U. Assuming “INT0” is enabled and “INT1” is set to Low priority, show the first instruction executed after each simultaneous occurance of “INT 0” and “INT1” in the following code segment.

```
            org         0x000
            Goto        Init

            org         0x8
            BRA         Int_first

            org         0x18
            GOTO        Int_2nd

Int_first:  CLRF        PORTA       ; Initialize PORTA
            …

Int_2nd:    CLRF        PORTB       ; Initialize PORTB
            ….

Init:
            MOVLW       0x7F        ; Set all A/D Converter Pins as
            MOVWF       ADCON1      ; digital I/O pins
            MOVLW       0x00        ; Value used to initialize data direction
```

Solution

4S. For the following code show the top of stack content and PC value before and after the execution of CALL and RETURN instruction.

Address		**Content**	
0x0102		CALL	Delay
		MOVFF	0x80, 0x91
0x298	Delay:	CLRF	0x81
	Loop:	NOP	
		NOP	
		NOP	
		INCF	0x81
		BNC Loop	
		RETURN	

Solution

Address		**Content**	
0x0102		CALL	Delay
		MOVFF	0x80, 0x91
0x298	Delay:	CLRF	0x81
0X29A	Loop:	NOP	
0X29C		NOP	
0X29E		NOP	
0X2A0		INCF	0x81
0X2A2		BNC	Loop
0X2A4		RETURN	

	Before Call	After Call	Before Return	After Return
PC	0x102	0x298	0x2A4	0x106
TOS	?	0x106	0x106	?

4U. For the following code show the top of stack content and PC value before and after the execution of CALL and RETURN instruction.

```
Address          Content
0x0AC2           CALL    Delay
                 MOVFF   0x80, 0x91

0x0B12   Delay:  CLRF    0x81
         Loop:   MOVWF   0x92
                 INCF    0x81
                 BNC Loop
                 RETURN
```

Solution

5S. Write a PICmicro assembly code that configures INT1 such that upon an interrupt event, value of PORTB register is set to 0xA.

Show both the address and the instruction for your program.

Solution

```
                 org 0x00
                 BRA     StartL

                 org 0x08
                 BRA     HighPrio

                 org 0x20
HighPrio:        MOVLW   0xA
                 MOVWF   PORTB
                 BCF     INTCON3, INT1IF   ; Clear Interrupt 1 Flag
                 RETFIE

StartL:          CLRF    PORTB     ; Initialize PORTB
                 MOVLW   0x7F
                 MOVWF   ADCON1    ; Set I/O pin to Digital
                 MOVLW   0x00
                 MOVWF   TRISB     ; Set I/O pin to Output
                 BSF     INTCON, PEIE      ; Enable Peripheral Interrupt
                 BSF     INTCON3,INT1IE    ; Enable Interrupt 1
                 BSF     INTCON3, INT1IP   ; Set Interrupt 1 flag to high priority
                 BSF     RCON, IPEN        ; enable priority levels on interrupts
                 BCF     INTCON3, INT1IF   ; clear flag
                 BSF     INTCON,GIE        ; Enable Interrupts Globally
EventL:          BRA  EventL
```

5U. Write a PICmicro assembly code that configures INT2 such that upon an interrupt event, value of PORTA register is set to 0XB. “Include both the address and the instruction in your solution”.

Solution

6S. Write a PICmicro code that uses the Timer1 to implement a clock that calculates second, minute and hour of the day.

Solution

```
;-----------------------------------------------------------------------------------------------
;  Clock Implementations using Timer1:
;    Hour in Register 0x80
;    Min in Register 081
;    Sec in Register 082
;-----------------------------------------------------------------------------------------------
#include p18f1220.inc

Hour        equ   0x80
Min         equ   0x81
Sec         equ   0x82

            org 0x00
            BRA         StartL

            org 0x08
            BRA         HighPrioL

            org ox20
            ; High Priority Interrup Handling code

HighPrioL:
            BTFSS       PIR1, TMR1IF  ; Check for Timer 1 Interrupt Flag
             BRA        DoneHint

Tmr1L:
            INCF        Sec
            MOVLW       60
            CPFSLT      Sec
            BRA         AddMinL
            BRA         DoneHint

AddMinL:    CLRF        Sec
            INCF        Min
            MOVLW       60
            CPFSLT      Min
            BRA         AddHourL
            BRA         DoneHint

AddHourL:   CLRF        Min
            INCF        Hour
            MOVLW       24
            CPFSLT      Hour
            CLRF        Hour
            BRA         DoneHint

DoneHint:
            MOVLW       0xE1                ; reset timer starting value
            MOWF        TMR1H
            MOVLW       0x7B
            MOWF        TMR1L
            BCF         PIR1,TMR1IF         ; Clear Interrupt 1 Flag
```

```
                RETFIE

                ; Initialization code after reset
StartL:         CLRF    Hour
                CLRF    Min
                CLRF    Sec                     BSF     INTCON, PEIE   ;
                ;  Set TMR1L & TMR1H to 0x1E84 → (0xFFFF – 0xE17B)*(4*32 uSec)= 1 Sec.
                MOVLW   0xE1
                MOWF    TMR1H
                MOVLW   0x7B
                MOWF    TMR1L
                MOVLW   0xC9
                MOVWF   T1CON           ; enable timer 1, internal clock "1100 1001"
                BSF     PIE1,TMR1IE      ; Enable Timer 1 Interrupt
                BSF     IPR1,TMR1IP     ; Set Timer 1 flag to high priority
                BCF     PIR1,TMR1IF     ; clear flag
                BSF     INTCON,GIE      ; Enable Interrupts  Globally

EventL:         BRA  EventL
```

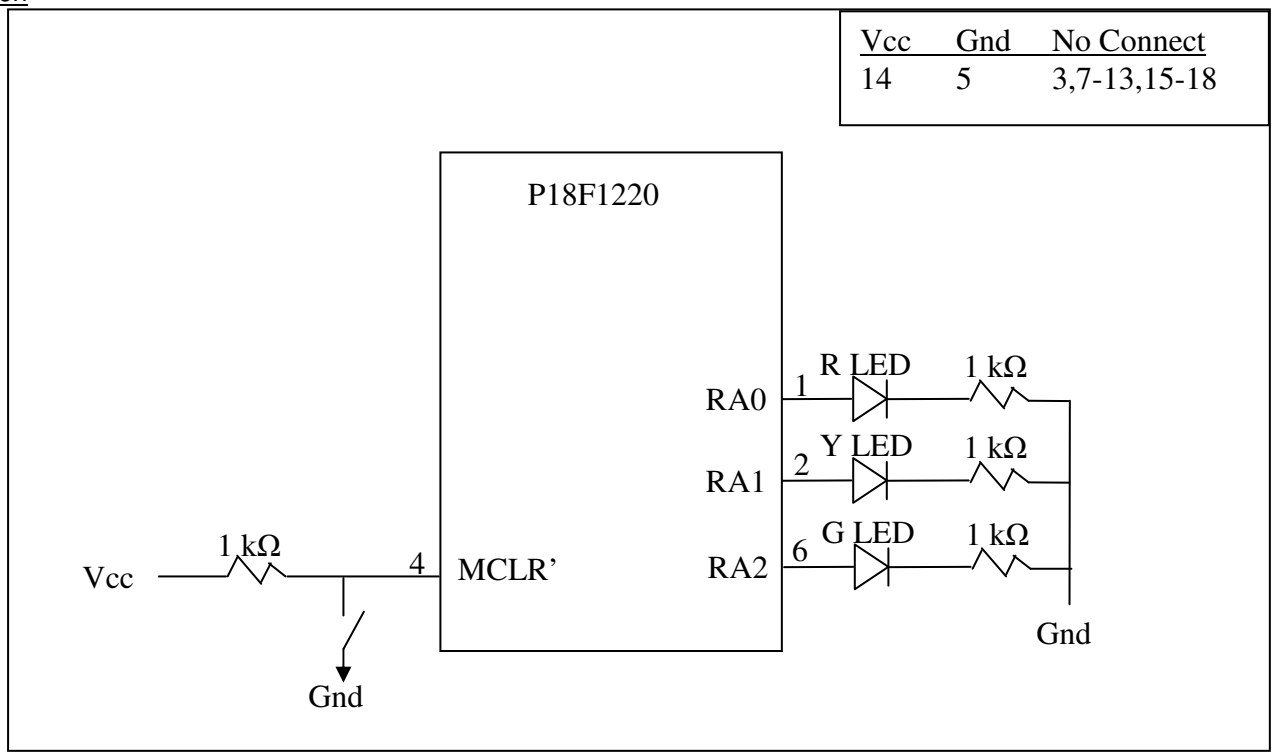
6U. Write a PICmicro code that uses the Timer2 to implement a Stop watch that counts number of seconds from initiation. The stop watch start counting when port A bit 0 is low and resets when port A bit 0 is high.

Solution

7S. Write a PICmicro code that uses the Timer function (not delay loops) to turn on a red LED after 10 milliseconds, a yellow LED after 50 milliseconds and a green LED after 120 milliseconds of Power on Reset (POR).

Show both your code and schematic to implement the functionality described.

Solution



```
;-------------------------------------------------------------------------------------------------
;  INT0 (pin #8) will be the counter reset.
;  RA0 – RA3 will be the output of the counter.
;-------------------------------------------------------------------------------------------------
#include p18f1220.inc

STATE   equ             0x80

        org             0x00            ; Reset
        GOTO            Start

        org             0x08            ; High Priority Interrupt
        GOTO            HPI

        org             0x18            ; Low Priority Interrupt
        GOTO            LPI

        org             0x20            ; start of program

Start:  CLRF            STATE           ; initialize STATE

        ; Set PortA <3:0> to digital output
        MOVLW           0x7F
        MOVWF           ADCON1
        MOVLW           0xF0
        MOVWF           TRISA
        CLRF            PORTA

        ; set internal Oscillator to 500 kHz → Tosc = 2 usec.
        BCF             OSCCON, IRCF2
        BSF             OSCCON, IRCF1
        BSF             OSCCON, IRCF0

        MOVLW           0x08
        MOVWF           T0CON       ; TMR0 off, 16-bit, no pre-scaler → each tick = 4*2 = 8 usec.

        MOVLW           0xF0
        IORWF           INTCON              ; Set GIE, PEIR, TMR0IE, INT0IE
        BCF             INTCON2, TMR0IP ; set to low prio.

        ; set the TMR0 to Interrupt in 10 msec.
        MOVLW           0xFB
        MOVWF           TMR0H
        MOVLW           0x1E
        MOVWF           TMR0L

        BSF             T0CON, TMR0ON       ; Turn on timer

Loop:   BRA             Loop            ; wait loop

        ;High Prio. Int. but we are not using
HPI:    CLRF            STEP
        CLRF            PORTA
        BCF             INTCON, INT0IF
        RETFIE

        ; TIMER 0 Interrupt is the only low prio. Interrupt
LPI:    MOVLW           0
        CPFSEQ          STATE
```

```
        BRA     STATE1
STATE0:  ; STATE0 turns red light on after 10 mec.
        INCF    STATE
        BSF     PORTA,0     ; turn red on.
        ; set the TMR0 for next Interrupt in 40 msec.
        MOVLW   0xEC        ; reset timer
        MOVWF   TMR0H
        MOVLW   0x78
        MOVWF   TMR0L
        BRA     LPI_Done

STATE1:
        MOVLW   1
        CPFSEQ  STATE
        BRA     STATE2
        INCF    STATE
        BSF     PORTA,1     ; turn yellow on.
        ; set the TMR0 for next Interrupt in 70 msec.
        MOVLW   0xDD        ; reset timer
        MOVWF   TMR0H
        MOVLW   0xD2
        MOVWF   TMR0L
        BRA     LPI_Done

STATE2:
        CLRF    STATE
        BSF     PORTA,2     ; turn green
        BCF     T0CON,TMR0ON
LPI_Done:
        BCF     INTCON, TMR0IF
        RETFIE
```

7U. Write a PICmicro code that uses the Timer function (not delay loops) to turn on an LED on for 30 second and off for 40 seconds continuously after Int 0 occurs. Show both your code and schematic to implement the functionality described.

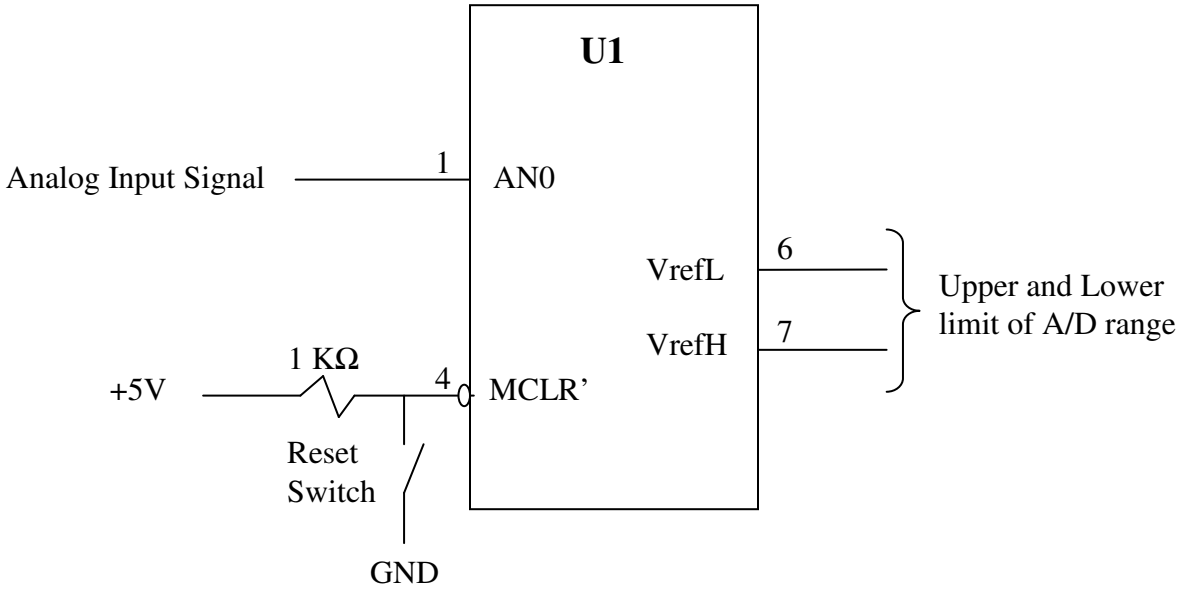
Solution

8S. How may pins are required for A/D conversion. Show a typical circuit diagram for a single A/D converter.

Solution

Maximum of 3 pins are need - One for analog input. If you want to have input range different from 0 to 5V then you need additional two pins for the reference high (V_{refH}) and reference low (V_{refL}).

U#.	Description	+5V	Ground	No Connect
U1	PIC 18F1220	14	5	2,3,8-13, 15-18



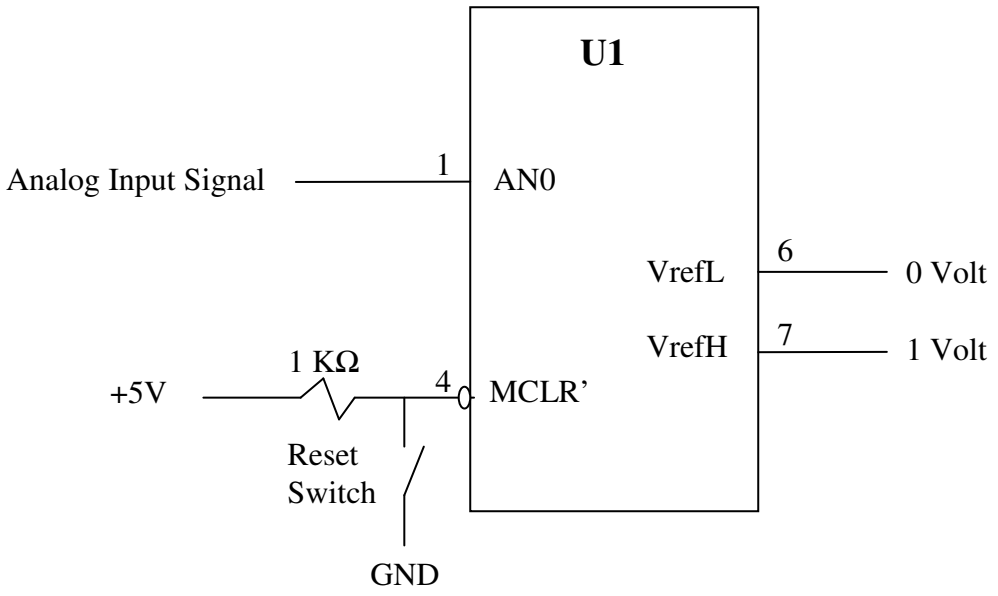
8U. Draw the circuit diagram using PICmicro that would allow for the highest resolution for measuring the input voltage ranging from 2 to 4 volts.

Solution

9S. Configure the A/D module so that it will accept 0 to 1 volt analog input and converts it to an 8 bit digital value that is stored in Wreg. Where 0 volt is represented by 0x00 digital value and 1 volt is represented by 0xFF digital value. The range from 0 to 1 volt is divided into 255 equal segments. Show the solution's circuit diagram and assembly code.

Solution

U#.	Description	+5V	Ground	No Connect
U1	PIC 18F1220	14	5	2,3,8-13,15,16.




```
#include p18f1220.inc                    ; Constants/definitions for PICmicro
        org  0x00
        CLRF    PORTA                   ; Initialize PORTA
        CLRF    PORTB                   ; Initialize PORTB

        ; INITIALIZATION
        ; 1) Use VrefH & VrefL, Select analog input to AN0 (Pin 1), Disable A/D initially
        MOVLW   0xC0                    ; "11 0 000 0 0"
        MOVWF   ADCON0
        ; 2) Configure AN0 as an analog input
        BCF     ADCON1, PCFG0           ; Analog
        BSF     TRISA,0                 ; Input
        ; 3) Set A/D conversion clock (Fosc/8),  acquisition time (2TAD),
            Digital value is left  justified 8-bit results in register ADREFSH
        MOVLW   0x09                    ;"0 0 001 001
        MOWF    ADCON2
        ; 4) enable A/D module
        BSF     ADCON0,ADON

WAITL:  BTFSC   ADCON0,NOT_DONE         ; Wait for A/D module to complete conversion
        BRA     WAITL

        BSF     ADCON0,GO               ; Start the A/D conversion

READL:  BTFSC   ADCON0,NOT_DONE:        ; Wait for A/D module to complete conversion
        BRA     READL

        ;  The value in ADRESH is the highest 8-bit digital value of input analog signal
        MOVF    ADRESH,0
```

```
        BSF     ADCON0,GO       ; Start the Next A/D conversion
        BRA     READL
```

: This method polls the status bit in order to determine when the A/D data is ready to be read.
; Another way is to use the A/D interrupt in order to know when the A/D data is read to be read.

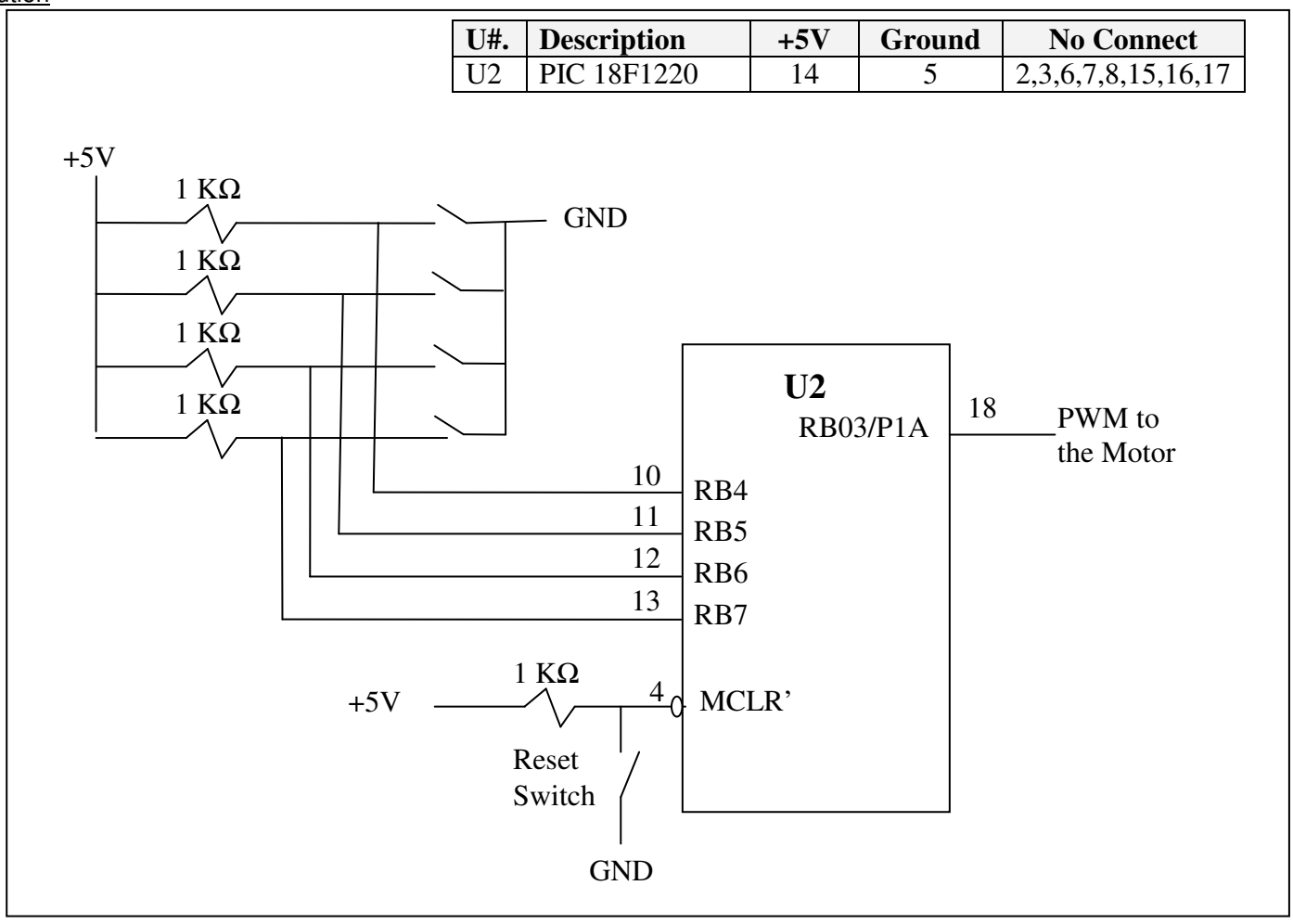
9U. Configure the A/D module so that it will accept 0 to 5 volt analog input and converts it to an 4-bit digital value that is stored in Wreg. Where 0 volt is represented by 0x00 digital value and 5 volt is represented by 0xF digital value. Show the solution's circuit diagram and assembly code.

Solution

10S. Design a system that accept a digital value from 0 to 10 and sets the speed of motor from 0 to maximum speed accordingly. The system is required to use the PICmicro's PWM module. Show the solution's circuit diagram and assembly code.

Solution

U#.	Description	+5V	Ground	No Connect
U2	PIC 18F1220	14	5	2,3,6,7,8,15,16,17




```
#include P18F1220.inc

CURRENT         equ  0x80
NEW             equ  0x81

        CLRF    PORTA           ; Initialize PORTA
        CLRF    PORTB           ; Initialize PORTB
```

```
        ; 1) Set all portB to input
        MOVLW   0xFF
        MOVWF   TRISB

        ; PWM Initialization using TOSC = 32 us, PWM on P1A (pin 18)
        ; 2) PWMperiod = (PR2 + 1) * 4 * TOSC * (TMR2 Prescale)
        ;           = (99 + 1) * 4 * 32 us * 4 = 51 msec
        MOVLW   .99
        MOVWF   PR2

        ; 3) Set PWM  Mode
        MOVLW   0x00C               ; "0000 1100"
        MOVWF   CCP1CON             ; PWM output on P1A (Pin 18)

        ; 4) PWMdutyCycle = (CCPR1L:CCP1CON<5:4>)*TOSC*(TMR2 Pre-scale)
        ;             = (CCPR1L:11)* 32 * 4 usec. where CCPR1L control high value.
        CLRF    CCPR1L              ; Set the duty cycle to 0 for 0% power
                                    ; Set to (51 ms/(4*32)=398) >>2 or 0x63 or 99 for 100% power

        ; 5) Clear and Configure Timer 2 (PWM requires Timer 2)
        CLRF    TMR2                ; Timer 2 Register
        MOVLW   0x05                ; Enable timer and set pre-scale to 4
        MOVWF   T2CON
        BCF     PIR1, TMR2IF        ; Clear Timer 2 flag

        ;6) Set Pin 18 as output after timer 2 overflows once
WAITL:  BTFSS   PIR1,TMR2IF
        BRA     WAITL
        BCF     TRISB,3             ; Set P1A/RB3/CCP1 as an output pin

        CLRF    CURRENT
MAINL:  ; waiting in a loop
        SWAPF   PORTB, 0
        ANDLW   0x0F
        XORWF   CURRENT,0
        BZ      MAINL               ; wait until switches are set to new value
        MOVWF   CURRENT
        MOVLW   10
        MULWF   CURRENT
        MOVFF   PRODL,CCPR1L        ; update the PWM Duty Cycle
        BRA     MainL
```

10U. Design a system that accept a digital value (0 to 255) from port A and sets the speed of motor from 0 (Stopped) to 255 (maximum speed). The system is required to use the PICmicro's PWM module. Show the solution's circuit diagram and assembly code.

Solution