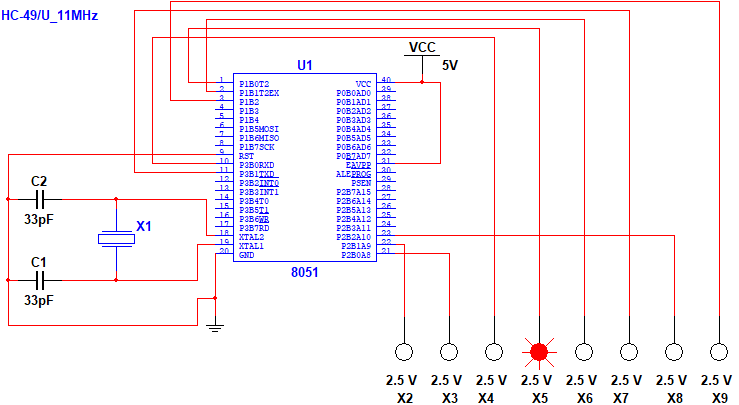
1. **Write a code of LED interface in Assembly and C through different P1, P2, P3**

**Block Diagram With 8 Led**

1. **Perform Rotation operation:**

**C code:**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

} }

void main() {

P21=0x00;

P20=0x00;

P30=0x00;

P10=0x00;

P11=0x00;

P31=0x00;

P22=0x00;

P12=0x00;

while (1) {

P21 = 1; // Turn on P2.1 (LED1)

delay(1); // Delay to keep LED1 on

P21 = 0; // Turn off LED1

P20 = 1; // Turn on P2.0 (LED2)

delay(1); // Delay to keep LED2 on

P20 = 0; // Turn off LED2

P30 = 1; // Turn on P3.0 (LED3)

delay(1); // Delay to keep LED3 on

P30 = 0; // Turn off LED3

P10 = 1; // Turn on P1.0 (LED4)

delay(1); // Delay to keep LED4 on

P10 = 0; // Turn off LED4

P11 = 1; // Turn on P1.1 (LED5)

delay(1); // Delay to keep LED5 on

P11 = 0; // Turn off LED5

P31 = 1; // Turn on P3.1 (LED6)

delay(1); // Delay to keep LED6 on

P31 = 0; // Turn off LED6

P22 = 1; // Turn on P2.2 (LED7)

delay(1); // Delay to keep LED7 on

P22 = 0; // Turn off LED7

P12 = 1; // Turn on P1.2 (LED8)

delay(1); // Delay to keep LED8 on

P12 = 0; // Turn off LED8

}

}

**Assembly Code:**

#include <htc.h>

delay: ; Delay function

movlw .100 ; Load W with 100

delay\_loop:

decfsz W, F ; Decrement W and skip if zero

goto delay\_loop ; Repeat delay\_loop

return ; Return from the delay function

main: ; Main program

clrf PORTA ; Initialize PORTA to 0x00

clrf PORTB ; Initialize PORTB to 0x00

clrf PORTC ; Initialize PORTC to 0x00

main\_loop:

bsf PORTB, 1 ; Turn on P2.1 (LED1)

call delay ; Delay to keep LED1 on

bcf PORTB, 1 ; Turn off LED1

bsf PORTB, 0 ; Turn on P2.0 (LED2)

call delay ; Delay to keep LED2 on

bcf PORTB, 0 ; Turn off LED2

bsf PORTC, 0 ; Turn on P3.0 (LED3)

call delay ; Delay to keep LED3 on

bcf PORTC, 0 ; Turn off LED3

bsf PORTA, 0 ; Turn on P1.0 (LED4)

call delay ; Delay to keep LED4 on

bcf PORTA, 0 ; Turn off LED4

bsf PORTA, 1 ; Turn on P1.1 (LED5)

call delay ; Delay to keep LED5 on

bcf PORTA, 1 ; Turn off LED5

bsf PORTC, 1 ; Turn on P3.1 (LED6)

call delay ; Delay to keep LED6 on

bcf PORTC, 1 ; Turn off LED6

bsf PORTB, 2 ; Turn on P2.2 (LED7)

call delay ; Delay to keep LED7 on

bcf PORTB, 2 ; Turn off LED7

bsf PORTA, 2 ; Turn on P1.2 (LED8)

call delay ; Delay to keep LED8 on

bcf PORTA, 2 ; Turn off LED8

goto main\_loop ; Repeat main\_loop

1. **Perform four on and off.**

**C code:**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void main() {

P21=0x00;

P20=0x00;

P30=0x00;

P10=0x00;

P11=0x00;

P31=0x00;

P22=0x00;

P12=0x00;

while (1) {

P21 = 1; // Turn on P2.1 (LED1)

P20 = 1; // Turn on P2.0 (LED2)

P30 = 1; // Turn on P3.0 (LED3)

P10 = 1; // Turn on P1.0 (LED4)

delay(1); // Delay to keep LED1 on

P21 = 0; // Turn off LED1

P20 = 0; // Turn off LED2

P30 = 0; // Turn off LED3

P10 = 0; // Turn off LED4

P11 = 1; // Turn on P1.1 (LED5)

P12 = 1; // Turn on P1.2 (LED8)

P31 = 1; // Turn on P3.1 (LED6)

P22 = 1; // Turn on P2.2 (LED7)

delay(1); // Delay to keep LED5 on

P11 = 0; // Turn off LED5

P31 = 0; // Turn off LED6

P22 = 0; // Turn off LED7

P12 = 0; // Turn off LED8

}

}

**Assembly Code:**

; Define constants and variables

LED1 equ P21 ; Define the symbolic names for LEDs

LED2 equ P20

LED3 equ P30

LED4 equ P10

LED5 equ P11

LED6 equ P31

LED7 equ P22

LED8 equ P12

; Delay function

delay:

movlw .100 ; Load W with delay value

delay\_loop:

decfsz W, F ; Decrement W and skip if zero

goto delay\_loop

return

; Main program

main:

clrf LED1 ; Turn off LED1

clrf LED2 ; Turn off LED2

clrf LED3 ; Turn off LED3

clrf LED4 ; Turn off LED4

clrf LED5 ; Turn off LED5

clrf LED6 ; Turn off LED6

clrf LED7 ; Turn off LED7

clrf LED8 ; Turn off LED8

main\_loop:

bsf LED1, 0 ; Turn on LED1

bsf LED2, 0 ; Turn on LED2

bsf LED3, 0 ; Turn on LED3

bsf LED4, 0 ; Turn on LED4

call delay ; Delay to keep LED1 to LED4 on

bcf LED1, 0 ; Turn off LED1

bcf LED2, 0 ; Turn off LED2

bcf LED3, 0 ; Turn off LED3

bcf LED4, 0 ; Turn off LED4

bsf LED5, 0 ; Turn on LED5

bsf LED8, 0 ; Turn on LED8

bsf LED6, 0 ; Turn on LED6

bsf LED7, 0 ; Turn on LED7

call delay ; Delay to keep LED5 to LED8 on

bcf LED5, 0 ; Turn off LED5

bcf LED6, 0 ; Turn off LED6

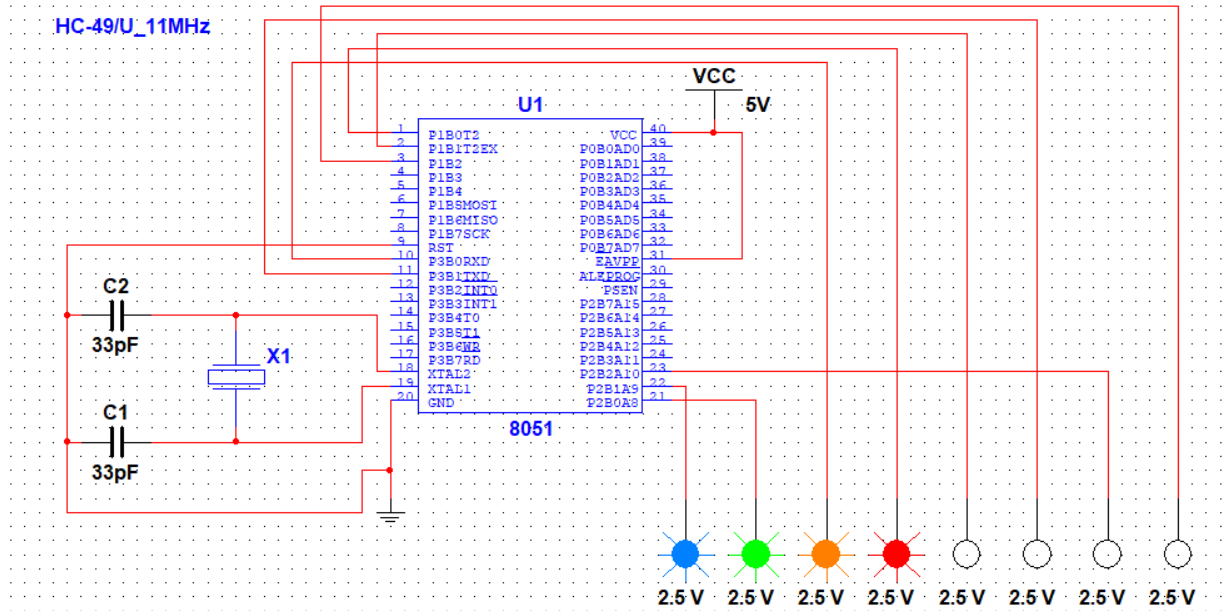
bcf LED7, 0 ; Turn off LED7

bcf LED8, 0 ; Turn off LED8

goto main\_loop ; Repeat main\_loop

**Output:**

A diagram of a circuit board

Description automatically generated

1. **Two on and six off**

**C Code:**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void main() {

P21=0x00;

P20=0x00;

P30=0x00;

P10=0x00;

P11=0x00;

P31=0x00;

P22=0x00;

P12=0x00;

while (1) {

P21 = 1; // Turn on P2.1 (LED1)

P20 = 1; // Turn on P2.0 (LED2)

delay(1);

P21 = 0; // Turn off LED1

P20 = 0; // Turn off LED2

P30 = 1; // Turn on P3.0 (LED3)

P10 = 1; // Turn on P1.0 (LED4)

P11 = 1; // Turn on P1.1 (LED5)

P31 = 1; // Turn on P3.1 (LED6)

P22 = 1; // Turn on P2.2 (LED7)

P12 = 1; // Turn on P1.2 (LED8)

delay(1);

P30 = 0; // Turn off LED3

P10 = 0; // Turn off LED4

P11 = 0; // Turn off LED5

P31 = 0; // Turn off LED6

P22 = 0; // Turn off LED7

P12 = 0; // Turn off LED8

}

}

**Assembly Code:**

; Define constants and variables

LED1 equ P21 ; Define the symbolic names for LEDs

LED2 equ P20

LED3 equ P30

LED4 equ P10

LED5 equ P11

LED6 equ P31

LED7 equ P22

LED8 equ P12

; Delay function

delay:

movlw .100 ; Load W with delay value

delay\_loop:

decfsz W, F ; Decrement W and skip if zero

goto delay\_loop

return

; Main program

main:

clrf LED1 ; Turn off LED1

clrf LED2 ; Turn off LED2

clrf LED3 ; Turn off LED3

clrf LED4 ; Turn off LED4

clrf LED5 ; Turn off LED5

clrf LED6 ; Turn off LED6

clrf LED7 ; Turn off LED7

clrf LED8 ; Turn off LED8

main\_loop:

bsf LED1, 0 ; Turn on LED1

bsf LED2, 0 ; Turn on LED2

call delay ; Delay to keep LED1 and LED2 on

bcf LED1, 0 ; Turn off LED1

bcf LED2, 0 ; Turn off LED2

bsf LED3, 0 ; Turn on LED3

bsf LED4, 0 ; Turn on LED4

bsf LED5, 0 ; Turn on LED5

bsf LED6, 0 ; Turn on LED6

bsf LED7, 0 ; Turn on LED7

bsf LED8, 0 ; Turn on LED8

call delay ; Delay to keep LED3 to LED8 on

bcf LED3, 0 ; Turn off LED3

bcf LED4, 0 ; Turn off LED4

bcf LED5, 0 ; Turn off LED5

bcf LED6, 0 ; Turn off LED6

bcf LED7, 0 ; Turn off LED7

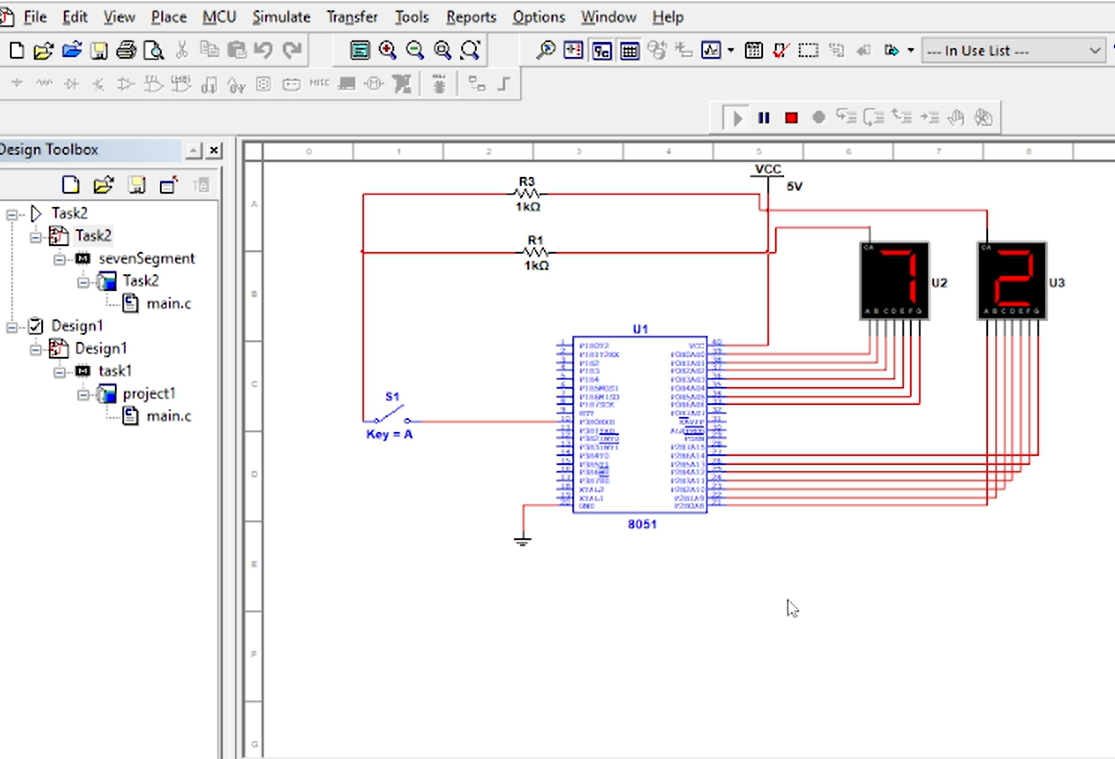
bcf LED8, 0 ; Turn off LED8

goto main\_loop ; Repeat main\_loop

A diagram of a circuit board

Description automatically generatedA diagram of a circuit board

Description automatically generated**Output:**

1. **Draw block diagram of two seven segment display. Write a program in Assembly and C to display a digit from 0 to 9 in both seven segment display**

**Both Seven segment displays are working From 0-9**

**C Code:**

#include <htc.h>

#define Port\_0 P0

#define Port\_1 P2

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void display(int a){

switch(a){

case 0:

Port\_0=0x40;

Port\_1=0x10;

break;

case 1:

Port\_0=0x79;

Port\_1=0x00;

break;

case 2:

Port\_0=36;

Port\_1=0x78;

break;

case 3:

Port\_0=48;

Port\_1=0x02;

break;

case 4:

Port\_0=0x19;

Port\_1=0x12;

break;

case 5:

Port\_0=0x12;

Port\_1=0x19;

break;

case 6:

Port\_0=0x02;

Port\_1=48;

break;

case 7:

Port\_0=0x78;

Port\_1=36;

break;

case 8:

Port\_0=0x00;

Port\_1=0x79;

break;

case 9:

Port\_0=0x10;

Port\_1=0x40;

break;

}

}

void main() {

int num=0;

while (1) {

display(num);

delay(1);

if(num<10){

num++;

}

else{

num=0;

}

}

}

**Assembly Code:**

#include <htc.h>

#define Port\_0 P0

#define Port\_1 P2

; Delay function

delay:

movlw .100 ; Load W with delay value (adjust as needed)

delay\_loop:

decfsz W, F ; Decrement W and skip if zero

goto delay\_loop

return

; Display function

display:

movf \_display\_value, W ; Load the value to display into W

movwf Port\_0 ; Set Port\_0

movf \_display\_value, W ; Load the value again into W

sublw 0x0A ; Check if the value is greater than or equal to 10

btfss STATUS, Z ; Skip if not

goto not\_overflow ; If it is, jump to not\_overflow

clrf \_display\_value ; If it's greater or equal to 10, reset the display value to 0

not\_overflow:

movwf Port\_1 ; Set Port\_1

return

; Variables

\_display\_value equ 0x20 ; Define the variable to store the value to display

; Main program

main:

clrf \_display\_value ; Initialize the display value to 0

bsf STATUS, RP0 ; Bank 1

movlw 0x07 ; Set the W register to 7

movwf TRISA ; Configure TRISA as output (RA0 to RA2)

bcf STATUS, RP0 ; Bank 0

clrf TRISB ; Configure TRISB as output (RB0 to RB7)

main\_loop:

call display ; Display the current value

call delay ; Delay for a short time

incf \_display\_value, F ; Increment the display value

goto main\_loop ; Repeat the loop

end

**THE END**