**ĐỀ CƯƠNG ÔN THI MÔN VI XỬ LÝ 2022**

**(ĐỀ ĐÓNG)**

***PHẦN 1A. LÝ THUYẾT***

**Câu 1:** Trình bày các thành phần cơ bản của hệ thống vi xử lý?

**Câu 2:** Phân loại và mô tả vai trò của từng loại BUS trong hệ thống vi xử lý ?

**Câu 3:** Trình bày vai trò của bộ nhớ ngăn xếp trong hệ thống vi xử lý?

**Câu 4:** Tần số xung nhịp ảnh hưởng như thế nào đế quá trình thực hiện một lệnh?

**Câu 5:** Nêu vai trò của bộ đếm chương trình trong hoạt động của vi xử lý?

**Câu 6:** Thanh ghi lệnh có vai trò gì trong hoạt động của vi xử lý?

**Câu 7:** Thế nào là địa chỉ và không gian địa chỉ ?

**Câu 8:** Ngăn xếp (stack) là gì ? Ứng dụng cơ bản của ngăn xếp?

**Câu 9:** Phân biệt chu kì máy, chu kì lệnh trong hoạt động của vi xử lý?

**Câu 10:** Trình bày các pha trong hoạt động của vi xử lý?

**Câu 11:** Thế nào là dữ liệu và bus dữ liệu?

***PHÂN 1B: BÀI TẬP***

**Câu 1:** Mã hóa các lệnh trong bộ nhớ và giải thích các bước khi thực hiện lệnh bằng mạch dãy tuần tự phép tính [(6 or 1) and 4] + 4

**Câu 2:** Mã hóa các lệnh trong bộ nhớ và giải thích các bước khi thực hiện lệnh bằng mạch dãy tuần tự phép tính [(7 and 3) + 2] – 1

**Câu 3:** Mã hóa các lệnh trong bộ nhớ và giải thích các bước khi thực hiện lệnh bằng mạch dãy tuần tự phép tính [(6 or 1) - 2] or 4

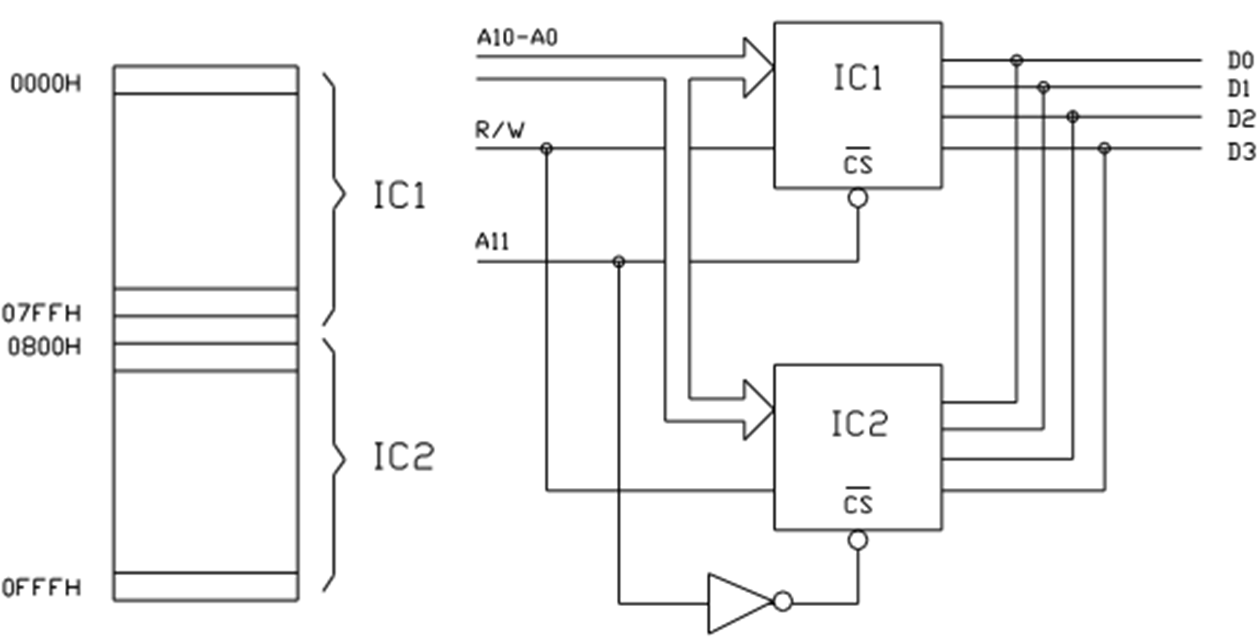
**Câu 4:** Mã hóa các lệnh trong bộ nhớ và giải thích các bước khi thực hiện lệnh bằng mạch dãy tuần tự phép tính [(3-1) and 3] and 4

**Câu 5:** Mã hóa các lệnh trong bộ nhớ và giải thích các bước khi thực hiện lệnh bằng mạch dãy tuần tự phép tính [(6 or 1) and 4] + 4

***PHẦN 2: GHÉP NỐI BỘ NHỚ (Tăng dung lượng)***

**Ví dụ 2:**Thiết kế bộ nhớ có dung lượng 4K x 4bit dùng các IC nhớ 2K x 4bit.

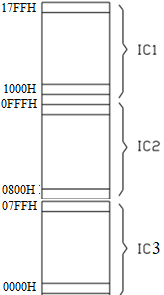
* **Giải:** Bộ nhớ 4K = 4x2^10 = 2^12 nên cần 12 bit địa chỉ A0÷ A11. Dùng 2 IC này đấu song song các đầu dữ liệu và các đầu địa chỉ từ A10÷A0.



**Câu 1:** Ghép nối bộ nhớ cho 8085 với các yêu cầu sau:  
- Vùng nhớ có dung lượng 7 KBytes (7K=23\*210 nên cần 13 bit địa chỉ A0-A12) từ các IC nhớ 4 Kbytes (cần 2 IC nhớ) bắt đầu từ địa chỉ thấp nhất trong không gian bộ nhớ (IC1:0000h-07FFh; IC2:0800h-0FFFh)

**Câu 2:** Ghép nối bộ nhớ cho 8085 với các yêu cầu sau:

- Vùng nhớ có dung lượng 5 Kbytes (23\*210)từ các IC nhớ 8 Kbytes→2 Kbytes (cần 3IC nhớ) bắt đầu từ địa chỉ cao nhất trong không gian bộ nhớ (IC1: 17FFFh-1000h, IC2: 0FFFh-0800h, IC3: 07FFh-0000h).



**Câu 3:** Ghép nối bộ nhớ cho 8085 với các yêu cầu sau:  
- Vùng nhớ có dung lượng 4 KBytes từ các IC nhớ 2 Kbytes bắt đầu từ địa chỉ 2000h(IC1: 2000h – 27FFh, IC2: 2800h-2FFFh)

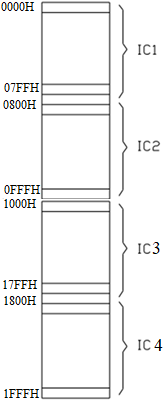
- Và vùng nhớ có dung lượng 8 KBytes từ các IC nhớ 2 Kbytes (cần 4 IC nhớ) bắt đầu từ địa chỉ cao nhất trong không gian bộ nhớ.

**Câu 4:** Ghép nối bộ nhớ cho 8085 với các yêu cầu sau:  
- Vùng nhớ có dung lượng 4 KBytes từ các IC nhớ 1 Kbytes (cần 4 IC nhớ) bắt đầu từ địa chỉ 4000h

- Và vùng nhớ dung lượng 4 Kbytes từ IC nhớ 2 Kbytes bắt đầu từ địa chỉ cao nhất trong không gian bộ nhớ

**Câu 5**: Ghép nối bộ nhớ cho 8085 với các yêu cầu sau:

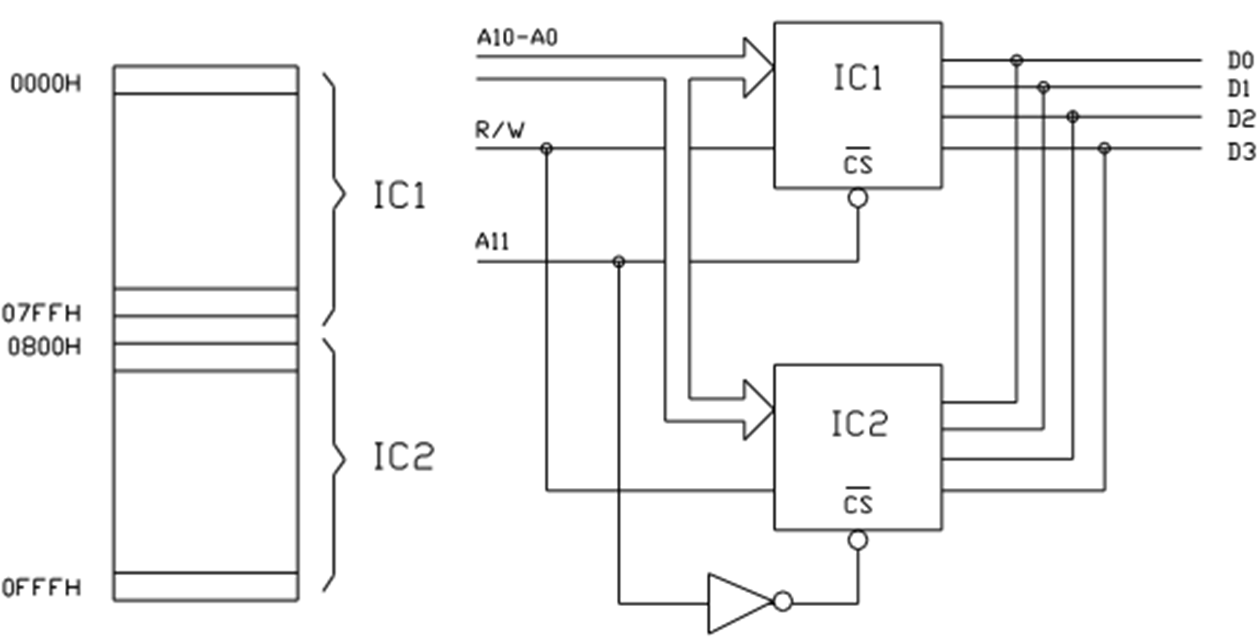
- Vùng nhớ có dung lượng 8 KBytes từ các IC nhớ 2 Kbytes bắt đầu từ địa thấp nhất trong không gian bộ nhớ

  
- Và vùng nhớ dung lượng 4 Kbytes từ IC nhớ 2 Kbytes bắt đầu từ địa chỉ cao nhất trong không gian bộ nhớ

* **:** Bộ nhớ 4K = 4x2^10 = 2^12 nên cần 12 bit địa chỉ A0÷ A11. Dùng 2 IC này đấu song song các đầu dữ liệu và các đầu địa chỉ từ A10÷A0.

IC1:0FFFH-0800H

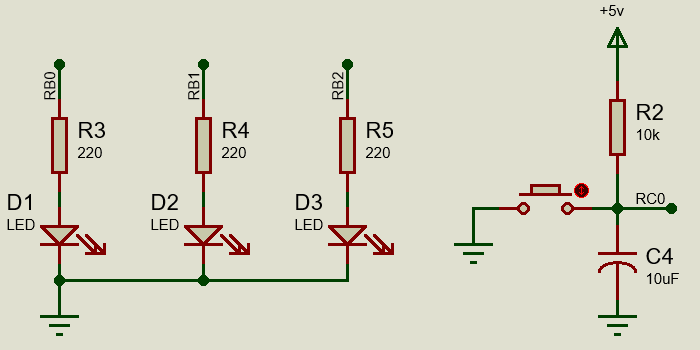
IC2:07FFH-0000H

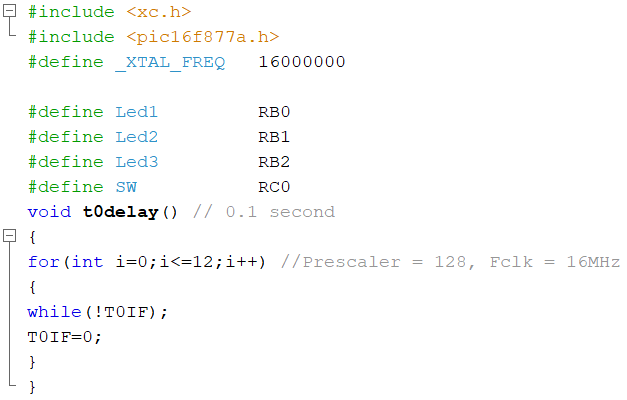


***PHẦN 3: BÀI TẬP LẬP TRÌNH***

**Câu 1:** Hãy thiết kế sơ đồ vi xử lý ghép nối với 3 Led đơn (Led1, Led2, Led3) và 1 phím ấn. Viết chương trình thực hiện đồng thời các công việc sau:

* Khi phím ấn: Led2 tắt, Led1 sáng nhấp nháy theo chu kỳ 0.5s sáng, 0.5s tắt.
* Khi nhả phím: Led1 tắt, Led2 sáng nhấp nháy theo chu kỳ: sáng 0.2s, 0.8s tắt.
* Led3 sáng lần lượt theo chu kỳ: sáng 0.5s, tắt 0.5s, sáng 0.3s, tắt 0.3s và lặp lại.





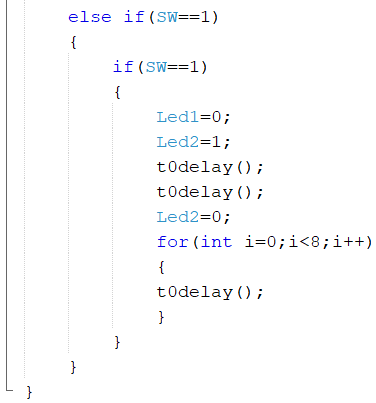
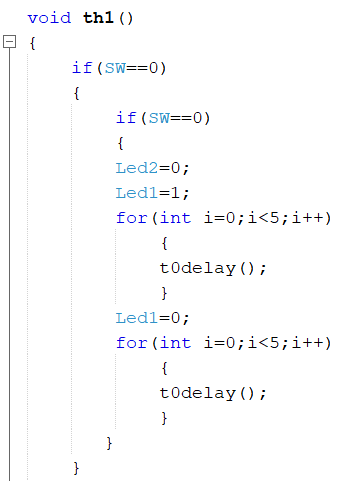
***Ví dụ:*** *Tạo xung vuông có tần số 10Hz cho PORTC sử dụng Timer0.*

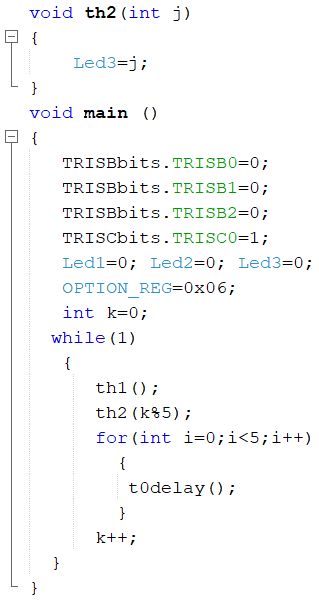
* *Tính độ trễ trong 0.1 giây sử dụng Timer0:*

*suy ra*

*Trong đó:*

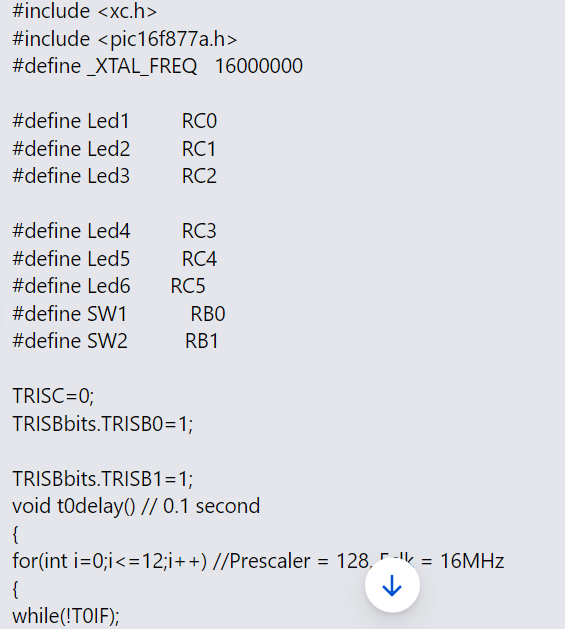
* *fclk là tần số của mạch dao động thạch anh, giả sử chọn thạch anh 16MHz thì ta có fclk=16MHz.*
* *=128 (tùy chọn)*
* *TMR0 = 0 (giá trị TMR=0)*
* *Vì mong muồn fout=10Hz nên Count = 16\*106/[(4\*128\*(256-0)\*10]≈12*

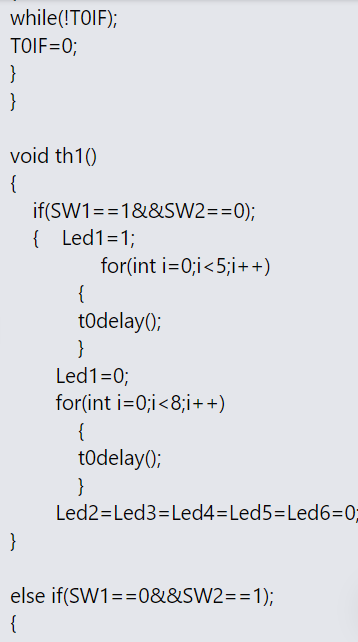


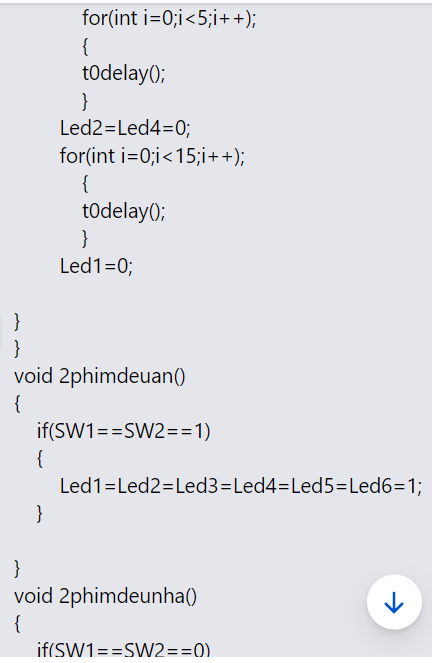
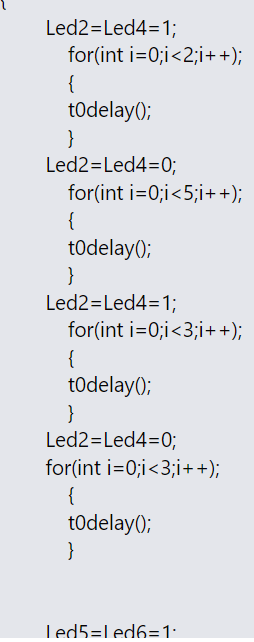


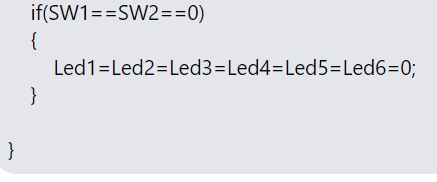
**Câu 2*:*** Hãy thiết kế sơ đồ vi xử lý ghép nối 6 Led đơn và 2 phím ấn. Viết chương trình thực hiện đồng thời các công việc sau:

* Khi phím 1 ấn, phím 2 nhả: Led1 sáng nhấp nháy với chu kỳ 1.3s (0.5s sáng, 0.8s tắt), Led 2 đến Led 6 tắt
* Khi phím 2 ấn, phím 1 nhả: Led2 và Led4 sáng theo chu kỳ: sáng 0.2s, tắt 0.5s, sáng  
  0.3s, tắt 0.3s. Led5 và Led6 sáng sáng theo chu kỳ 2 giây (0.5s sáng, 1.5s tăt), Led1 tắt.
* Khi cả 2 phím đều ấn: cả 6 Led đều sáng.
* Khi cả 2 phím đều nhả: cả 6 Led đều tắt.
* Count=(16.10^6/4.128.256)=12

******

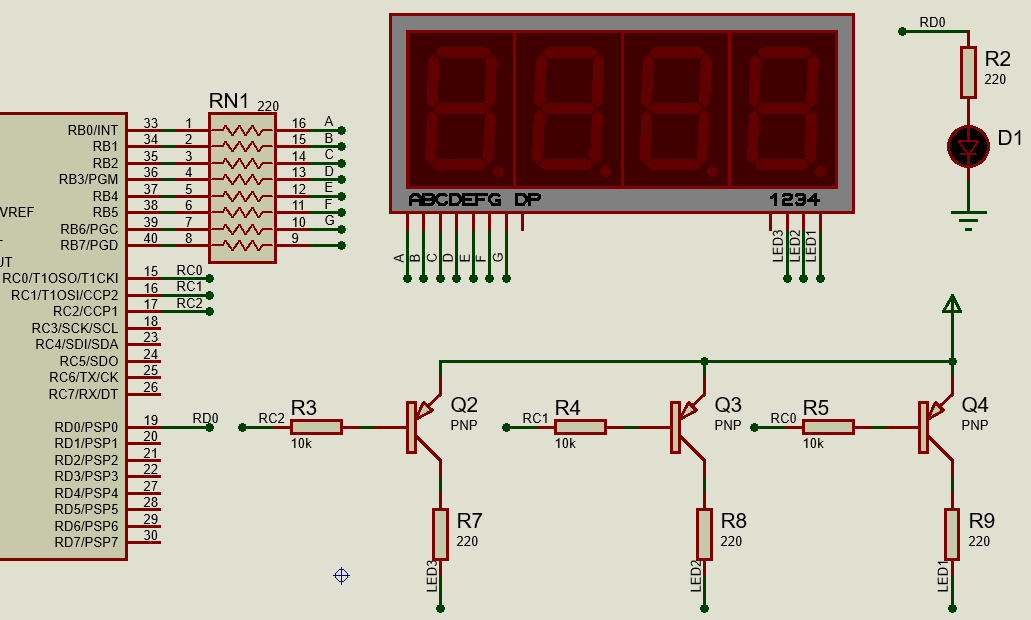
******

******

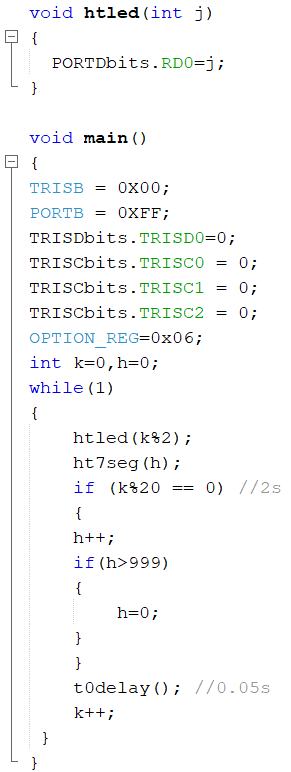
******

**Câu 3:** Hãy thiết kế sơ đồ vi xử lý ghép nối với 3 Led 7 thanh và 1 Led đơn. Viết chương trình thực hiện đồng thời các công việc sau:

* Hiển thị số đếm trên Led 7 thanh với thời gian 2 giây tăng 1 số. Khi tăng đến giá trị 999, giá trị quay về giá trị 0.
* Led đơn sáng nhấp nháy với tần số 10Hz (thời gian sáng là 50% chu kỳ xung)



****



**Câu 4:** Hãy thiết kế sơ đồ vi xử lý ghép nối với 3 Led 7 thanh và 2 Led đơn (Led1, Led2) và 1 phím bấm. Viết chương trình thực hiện đồng thời các công việc sau:

* Khi ấn và giữ phím giá trị số đếm trên Led 7 tăng theo chu kỳ 0.5s tăng 1 giá trị (giá trị đếm từ 0 đến 999). Nếu giá trị đếm ≤100, Led1, Led2 tắt. Nếu giá trị đếm >100 Led1 sáng nhấp nháy với chu kỳ 0.5s (0.2s sáng, 0.3s tắt), Led2 tắt
* Khi nhả phím giá trị đếm xóa về 0, đồng thời Led1 tắt, Led2 sáng.

**// PIC16F877A Configuration Bit Settings**

**// 'C' source line config statements**

**// CONFIG**

**#pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)**

**#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)**

**#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)**

**#pragma config BOREN = ON // Brown-out Reset Enable bit (BOR enabled)**

**#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)**

**#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)**

**#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write protection off; all program memory may be written to by EECON control)**

**#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)**

**#include <xc.h>**

**#include <pic16f877a.h>**

**#define \_XTAL\_FREQ 16000000**

**#define Led RD0**

**unsigned int dv,chuc, tram;**

**unsigned int count=0;**

**unsigned char time;**

**const unsigned char ma7doan[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};**

**void t0delay() // 0.05 second, f=10Hz**

**{**

**for(int i=0;i<=6;i++) //Prescaler = 128, Fclk = 16MHz**

**{**

**while(!T0IF);**

**T0IF=0;**

**}**

**}**

**void ht7seg (unsigned int count)**

**{**

**tram = count/100;**

**chuc = (count - tram\*100)/10;**

**dv = count - tram\*100 - chuc\*10;**

**PORTB = ma7doan[dv];**

**PORTCbits.RC0 = 0;**

**\_\_delay\_ms(1);**

**PORTCbits.RC0 = 1;**

**PORTB = ma7doan[chuc];**

**PORTCbits.RC1 = 0;**

**\_\_delay\_ms(1);**

**PORTCbits.RC1 = 1;**

**PORTB = ma7doan[tram];**

**PORTCbits.RC2 = 0;**

**\_\_delay\_ms(1);**

**PORTCbits.RC2 = 1;**

**}**

**void htled(int j)**

**{**

**PORTDbits.RD0=j;**

**}**

**void main()**

**{**

**TRISB = 0X00;**

**PORTB = 0XFF;**

**TRISDbits.TRISD0=0;**

**TRISCbits.TRISC0 = 0;**

**TRISCbits.TRISC1 = 0;**

**TRISCbits.TRISC2 = 0;**

**OPTION\_REG=0x06;**

**int k=0,h=0;**

**while(1)**

**{**

**htled(k%2);**

**ht7seg(h);**

**if (k%20 == 0)**

**{**

**h++;**

**}**

**t0delay(); //0.05s**

**k++;**

**}**

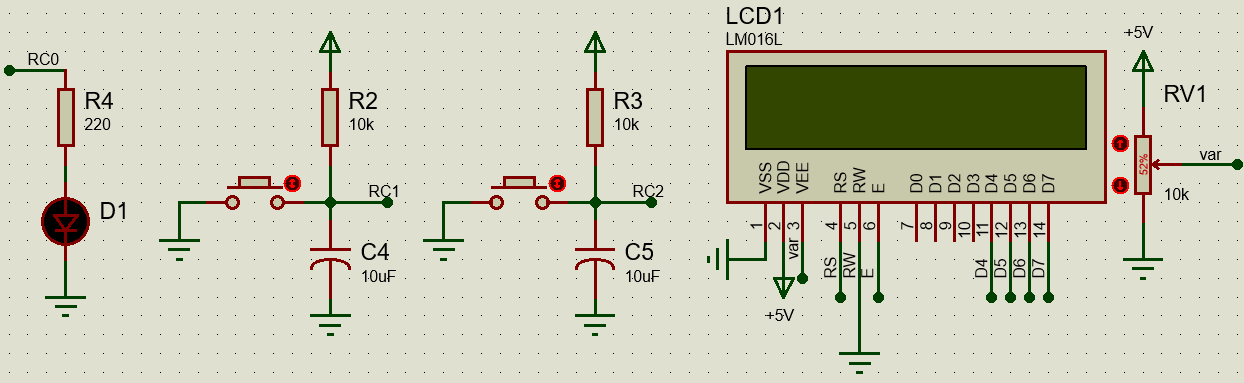
**}**

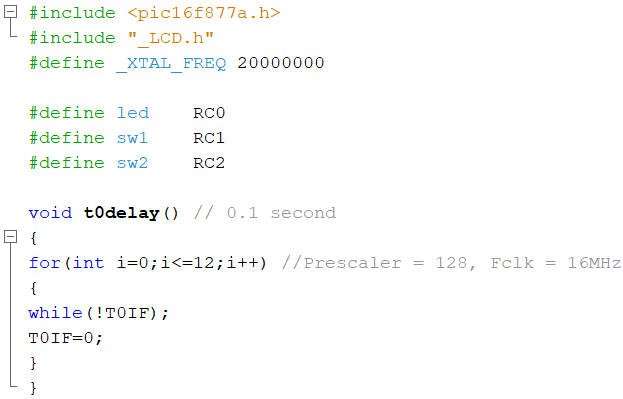
**Câu 5:** Hãy thiết kế sơ đồ vi xử lý ghép nối với 2 Led 7 thanh, 1 Led đơn và 2 phím ấn. Viết chương trình thực hiện đồng thời các công việc sau:

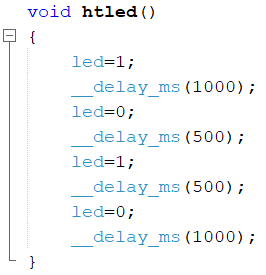
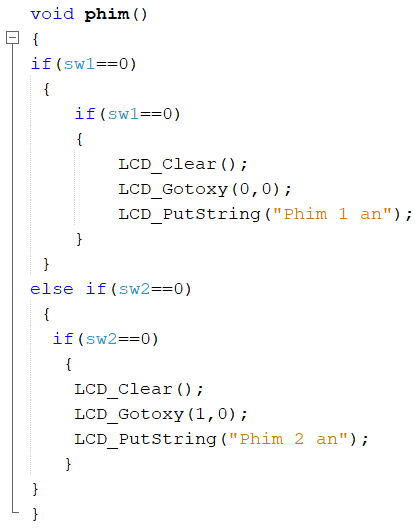
* Cứ mỗi lần phím 1 ấn, giá trị đếm trên Led 7 thanh tăng 1 giá trị. Cứ mỗi lần phím 2 ấn, giá trị đếm trên Led 7 thanh giảm 1 giá trị. Giá trị đếm nằm trong khoảng từ 0 đến 99
* Led đơn sáng nhấp nháy với chu kỳ 2 giây (1 giây sáng, 1 giây tắt)

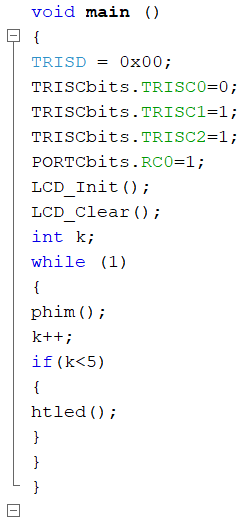
**Câu 6:** Hãy thiết kế sơ đồ vi xử lý ghép nối với LCD loại 2 hàng 16 cột, 2 phím ấn và 1 Led đơn. Viết chương trình thực hiện đồng thời các công việc sau:

* Khi phím 1 ấn: Hiển thị ở hàng 1 LCD dòng chữ “Phim 1 an”, hàng 2 bị xóa
* Khi phím 2 ấn: Hiển thị ở hàng 2 của LCD dòng chữ : “Phim 2 an”, hàng 1 bị xóa
* Led đơn sáng lần lượt theo chu kỳ: sáng 1s, tắt 0.5s, sáng 0.5s, tắt 1s.



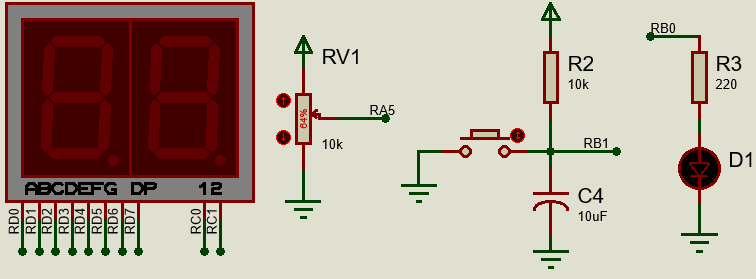


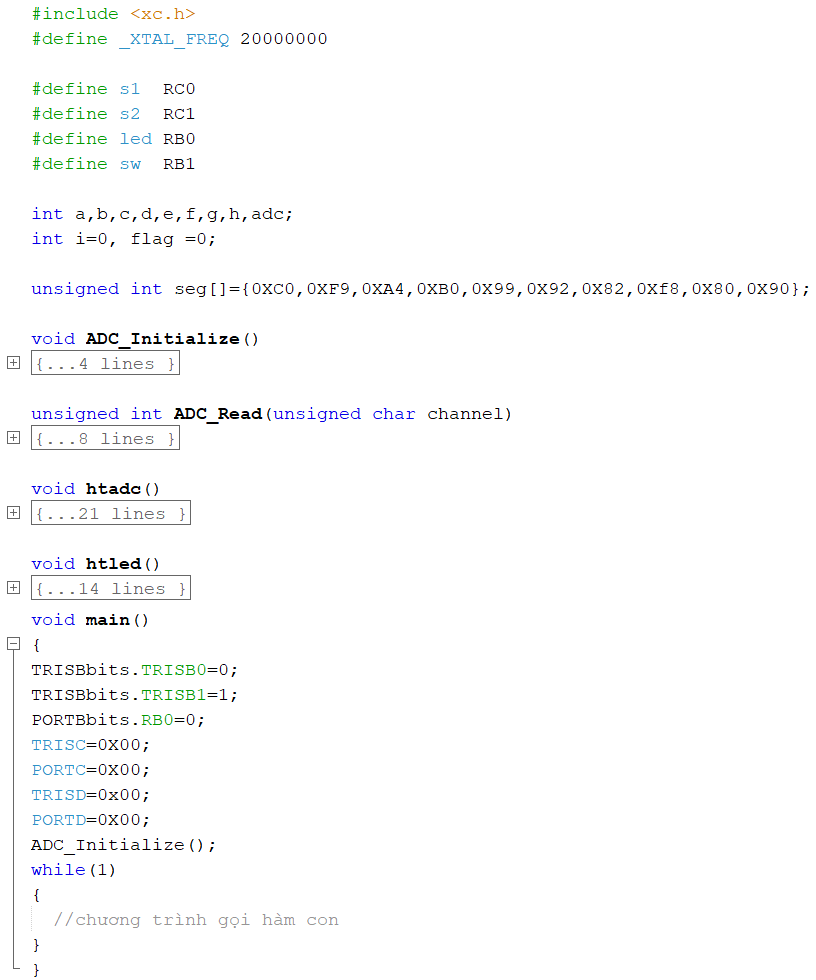
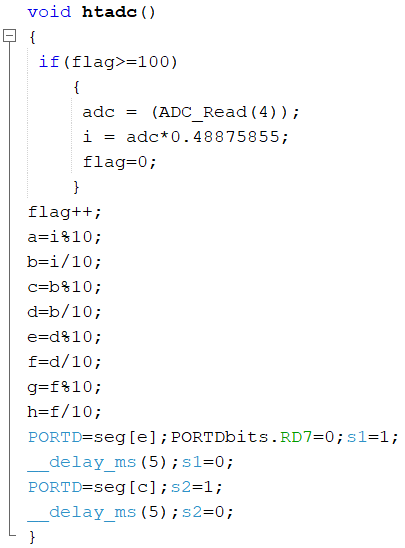
 



**Câu 7:** Hãy thiết kế sơ đồ vi xử lý đo giá trị điện áp tương tự dải đo từ 0-5V có thể hiển thị dữ liệu lên cụm led 7 thanh gồm 2 digit, và ghép nối với 1 phím ấn và 1 Led đơn. Viết chương trình thực hiện đồng thời các công việc sau:

* Xác định giá trị đo với tần xuất 2 giây/1 lần đọc hiển thị trên led 7 thanh.
* Khi ấn phím Led đơn sẽ sáng, khi nhả phím sau 0.5s Led đơn sẽ tắt





*Gợi ý chương trình htadc():*

**Câu 8:** Hãy thiết kế sơ đồ vi xử lý ghép nối đo giá trị điện áp tương tự dải đo từ 0-5V và hệ thống hiển thị LCD, 1 phím ấn và 1 Led đơn. Viết chương trình thực hiện đồng thời các công việc sau:

* Mỗi lần ấn phím thực hiện đọc tín hiệu tương tự và hiển thị giá trị ADC nhận được lên LCD.
* Led đơn sáng nhấp nháy với chu kỳ 1 giây (0.5 giây sáng, 0.5 giây tắt)

#pragma config FOSC = XT // Oscillator Selection bits (XT oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)

#pragma config BOREN = OFF // Brown-out Reset Enable bit (BOR disabled)

#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)

#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)

#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write protection off; all program memory may be written to by EECON control)

#pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)

#include <xc.h>

#define \_XTAL\_FREQ 4000000

#include "\_Library\_LCD.h"

#include "ADC.h"

int dienap=0;

unsigned char Digit[3] = { 0,0,0 };

unsigned int ADC\_Value = 0;

void hienthi\_dienap()

{

LCD\_Clear();

dienap = ADC\_Read(1)\*5/10.23;

ADC\_Value=(unsigned int)dienap;

Digit[2] = (ADC\_Value/100);

Digit[1] = (ADC\_Value%100)/10;

Digit[0] = (ADC\_Value%100)%10;

LCD\_Gotoxy(0,0);

LCD\_PutString("Gia tri U= ");

LCD\_PutChar(Digit[2]+0x30);

LCD\_PutChar('.');

LCD\_PutChar(Digit[1]+0x30);

LCD\_PutChar(Digit[0]+0x30);

}

void main()

{

ADCON1 = 0x06;

TRISD = 0x00;

TRISA = 0xFF;

LCD\_Init();

ADC\_Init();

int test;

while(1)

{

test=ADC\_Read(1)\*5/10.23;

if( dienap != test)

{

hienthi\_dienap();

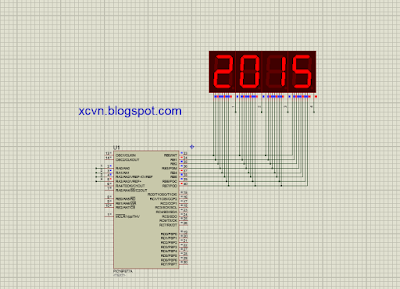
}

}

}

Trong bài này mình sẽ hướng dẫn các bạn hiển thị số 2015 lên 4 LED 7SEG.

- Đây là ảnh mô phỏng protues.

[](https://3.bp.blogspot.com/-s3cGeiljDIA/Vkl5NxH19eI/AAAAAAAAAM8/2cFyGMP4ifk/s1600/2015.png)

- Đây là code chương trình.

#include <xc.h>  
#include <stdio.h>  
#include <stdlib.h>  
#define \_XTAL\_FREQ 8000000  
// CONFIG  
#pragma config FOSC = HS        // Oscillator Selection bits (HS oscillator)  
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled)  
#pragma config PWRTE = OFF      // Power-up Timer Enable bit (PWRT disabled)  
#pragma config BOREN = OFF      // Brown-out Reset Enable bit (BOR disabled)  
#pragma config LVP = OFF        // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)  
#pragma config CPD = OFF        // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)  
#pragma config WRT = OFF        // Flash Program Memory Write Enable bits (Write protection off; all program memory may be written to by EECON control)  
#pragma config CP = OFF         // Flash Program Memory Code Protection bit (Code protection off)  
const unsigned char Anode[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};  
const unsigned char Cathode[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};  
void main(void)  
{  
    TRISB = 0X00;  
    PORTB = 0X00;  
    TRISAbits.TRISA0 = 0;  
    TRISAbits.TRISA1 = 0;  
    TRISAbits.TRISA2 = 0;  
    TRISAbits.TRISA3 = 0;  
    while(1)  
    {  
        // xuat so 2 len led 7seg  
        PORTB = 0XA4;  
        PORTAbits.RA0 = 1;  
        \_\_delay\_ms(4);  
        PORTAbits.RA0 = 0;  
        // xuat so 0 len led 7 seg  
        PORTB = 0XC0;  
        PORTAbits.RA1 = 1;  
        \_\_delay\_ms(4);  
        PORTAbits.RA1 = 0;  
        // xuat so 1 len led 7seg  
        PORTB = 0XF9;  
        PORTAbits.RA2 = 1;  
        \_\_delay\_ms(4);  
        PORTAbits.RA2 = 0;  
        // xuat so 5 len led 7 seg  
        PORTB = 0X92;  
        PORTAbits.RA3 = 1;  
        \_\_delay\_ms(4);  
        PORTAbits.RA3 = 0;  
    }  
  
}