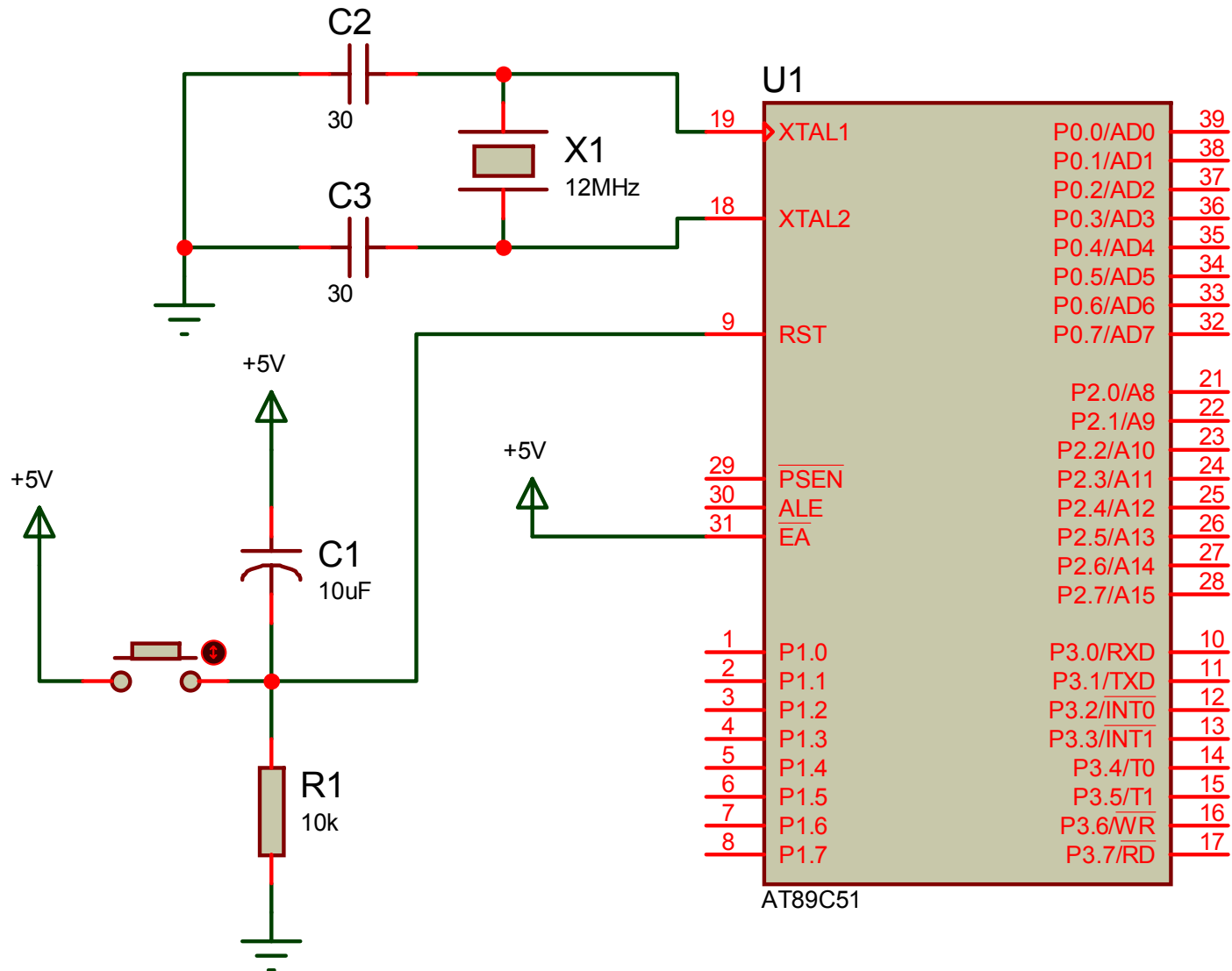


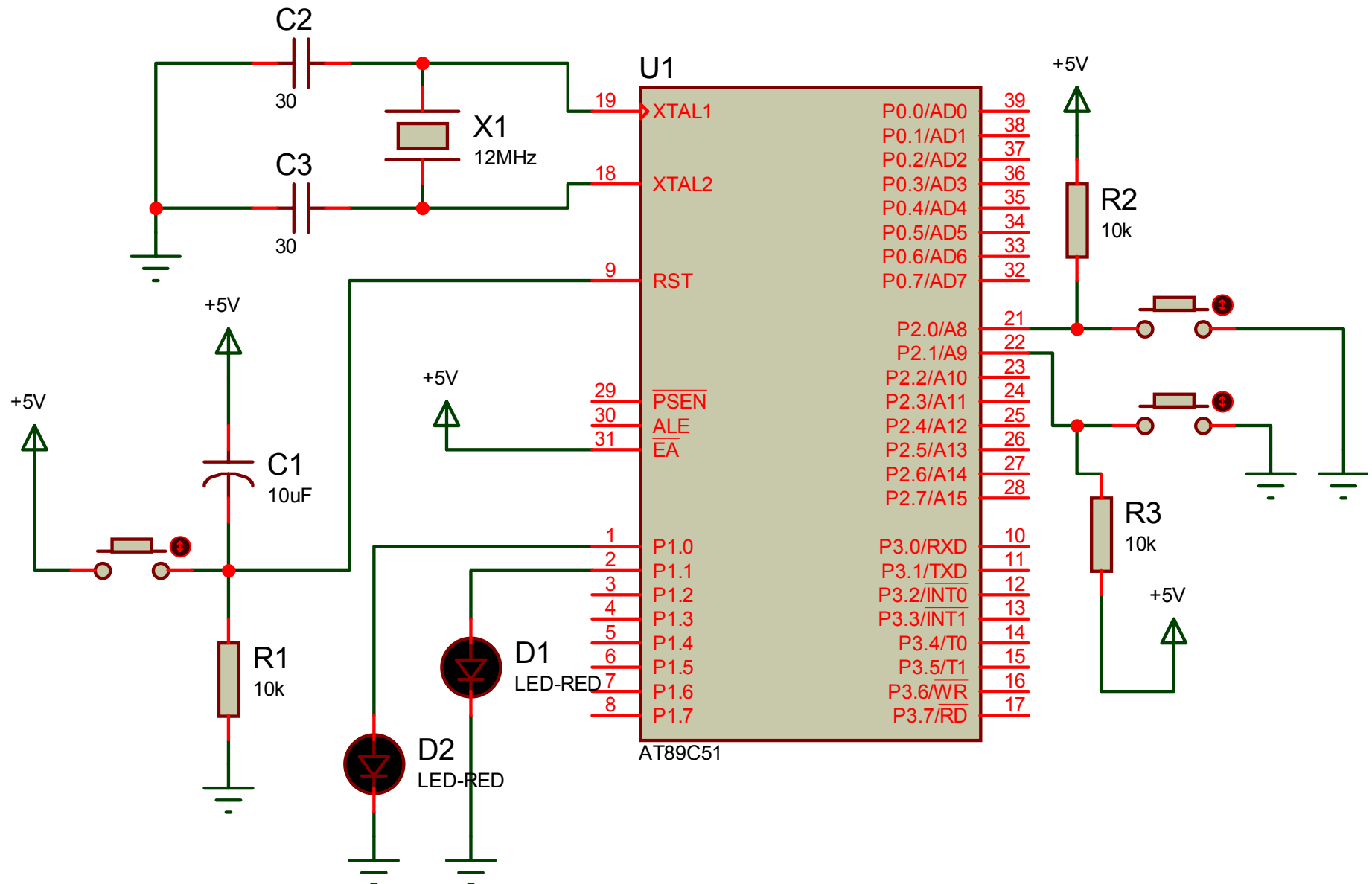
Chương 02

I/O

1. Mạch cơ bản nhất



Mạch xuất nhập cơ bản



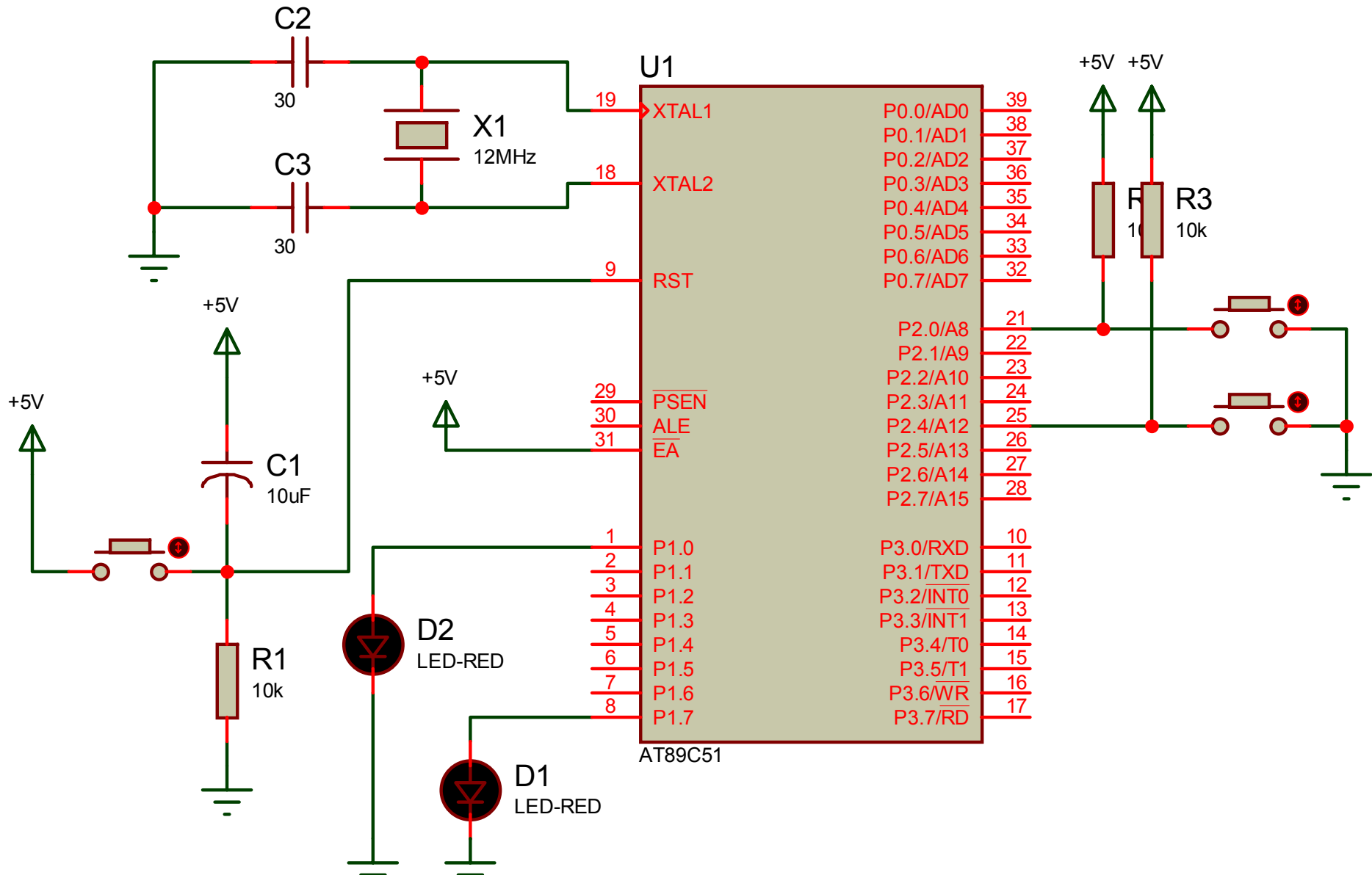
2. Xuất nhập Port (byte)

```
#include "regx51.h"
unsigned char giatri;
void main (void)
{
    while (1)
    {
        P2 = 0xFF;
        giatri = P2;
        P1 = giatri;
    }
}
```

/ khai bao thu vien AT89c51 */*
/ Bien trung gian */*
/ Chuong trinh chinh */*

/ Port 2 nhap */*
/ nhap gia tri Port 2 vao bien giatri */*
/ xuat gia tri vao Port 1 */*

3. Xuất nhập Pin (bit)



```

#include "regx51.h"
bit x; bit y;
sbit input01 = P2^0;
sbit input02 = P2^4;
sbit led01 = P1^0;
sbit led02 = P1^7;
void main (void)
{
    while (1)
    {
        P2 = 0xFF;
        x = input01;
        y = input02;
        led01 = x;
        led02 = y;
    }
}

```

/ khai bao thu vien AT89c51 */*

/ Bien trung gian */*

/ khai bao Pin */*

/ Chuong trinh chinh */*

/ Port 2 nhap */*

/ nhap gia tri Port 2 vao bien giatri */*

/ xuat gia tri vao Port 1 */*

4. Chương trình delay bằng phần mềm

```
/* Chương trình delay ngắn*/
```

```
Loop_Delay( )
```

```
{
```

```
    unsigned int x,y;
```

```
    for (x=0; x <= 65535; x++)
```

```
        {
```

```
            y++;
```

```
        }
```

```
}
```

```
/* Chương trình delay dài*/
```

```
Longer_Loop_Delay( )
```

```
{
```

```
  unsigned int x, y, z;
```

```
  for (x=0; x<=65535; x++)
```

```
  {
```

```
    for (y=0; y<=65535; y++)
```

```
    {
```

```
      z++;
```

```
    }
```

```
  }
```

```
}
```


Chương trình con delay us

```
#include <intrins.h>
```

```
void DelayUS(unsigned char us)
```

```
{
```

```
while (us--)
```

```
{
```

```
  _nop_();
```

```
}
```

```
}
```

Chương trình con delay ms

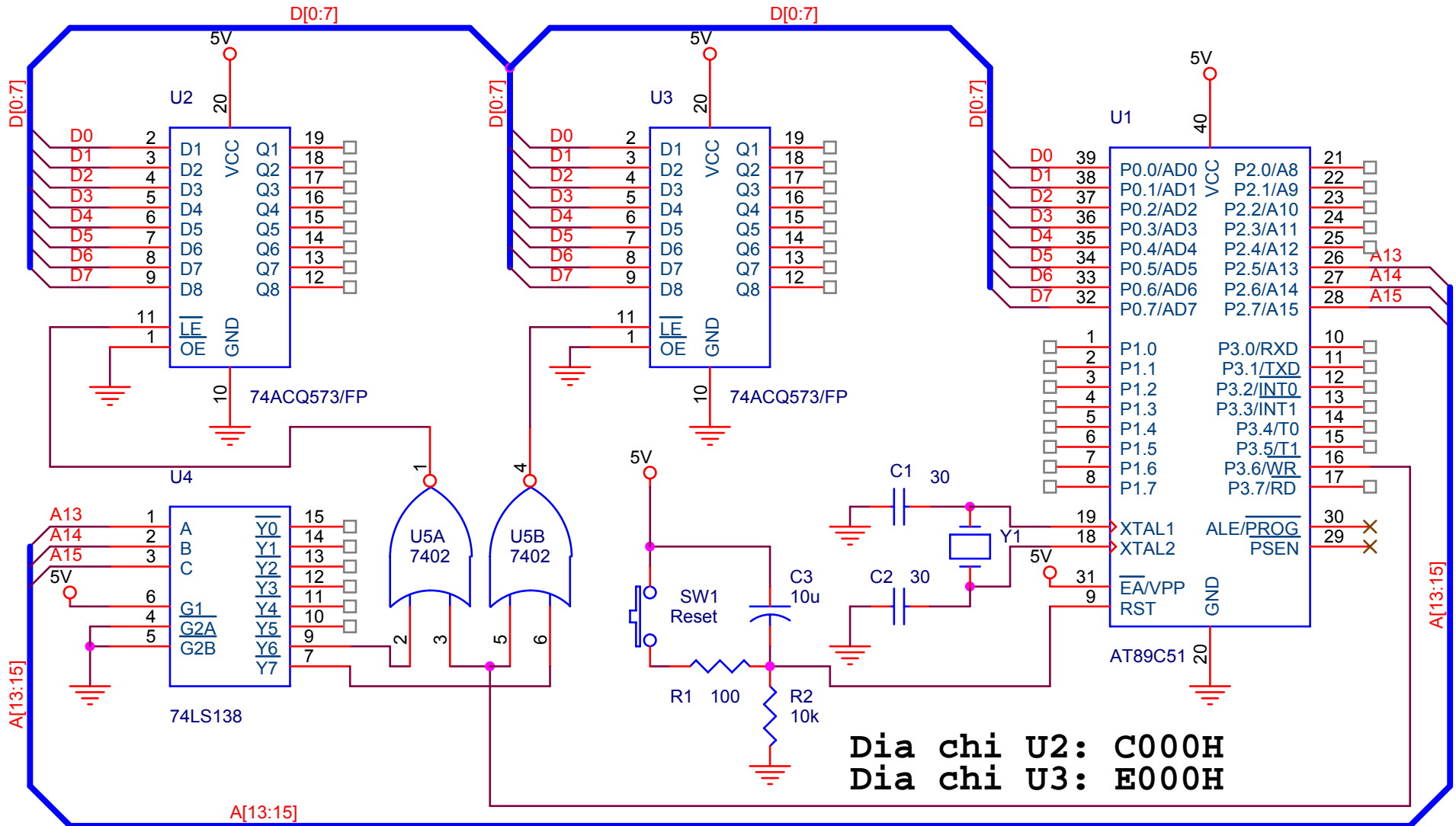
```
#include <intrins.h>
void DelayMS(unsigned char ms)
{
    unsigned long us = 1000*ms;
    while (us--)
    {
        __nop_();
    }
}
```

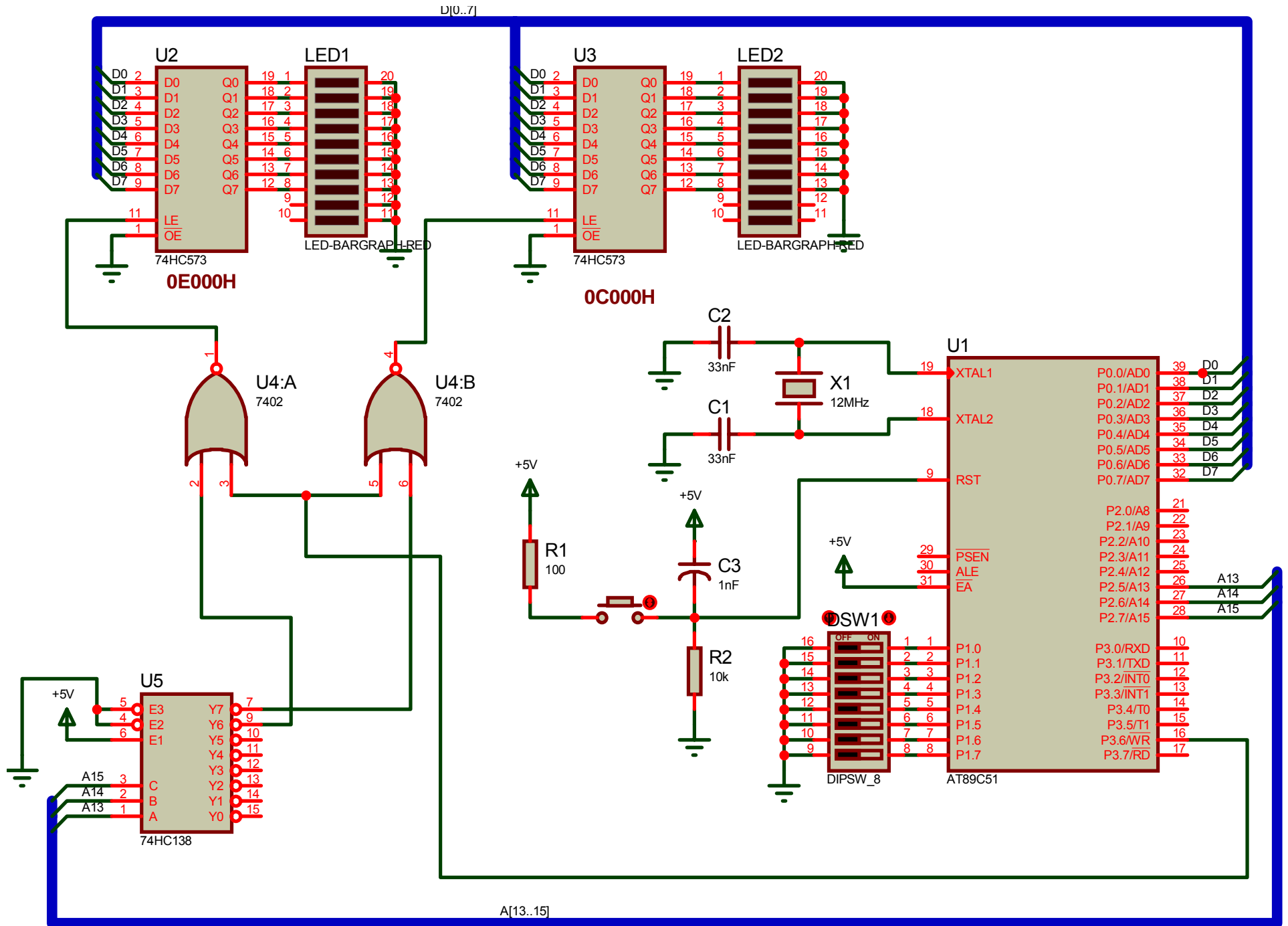
```

#include "regx51.h"    #include "intrins.h"
bit x; bit y;          /* Bien trung gian */
sbit input01 = P2^0; sbit input02 = P2^4; sbit led01 = P1^0; sbit led02 = P1^7;
void DelayMS(unsigned char ms); /* khai bao chuong trinh con*/
void main (void)      /* Chuong trinh chinh */
    {
    while (1)
        {
        P2 = 0xFF;    /* Port 2 nhap */
        x = input01; /* nhap gia tri Port 2 vao bien giatri */
        y = input02;
        led01 = x;    /* xuat gia tri vao Port 1 */
        led02 = y;
        DelayMS(1000);
        }
    }
void DelayMS(unsigned char ms)
    {
    unsigned long us = 1000*ms;
    while (us--)
        {
        _nop_();
        }
    }

```

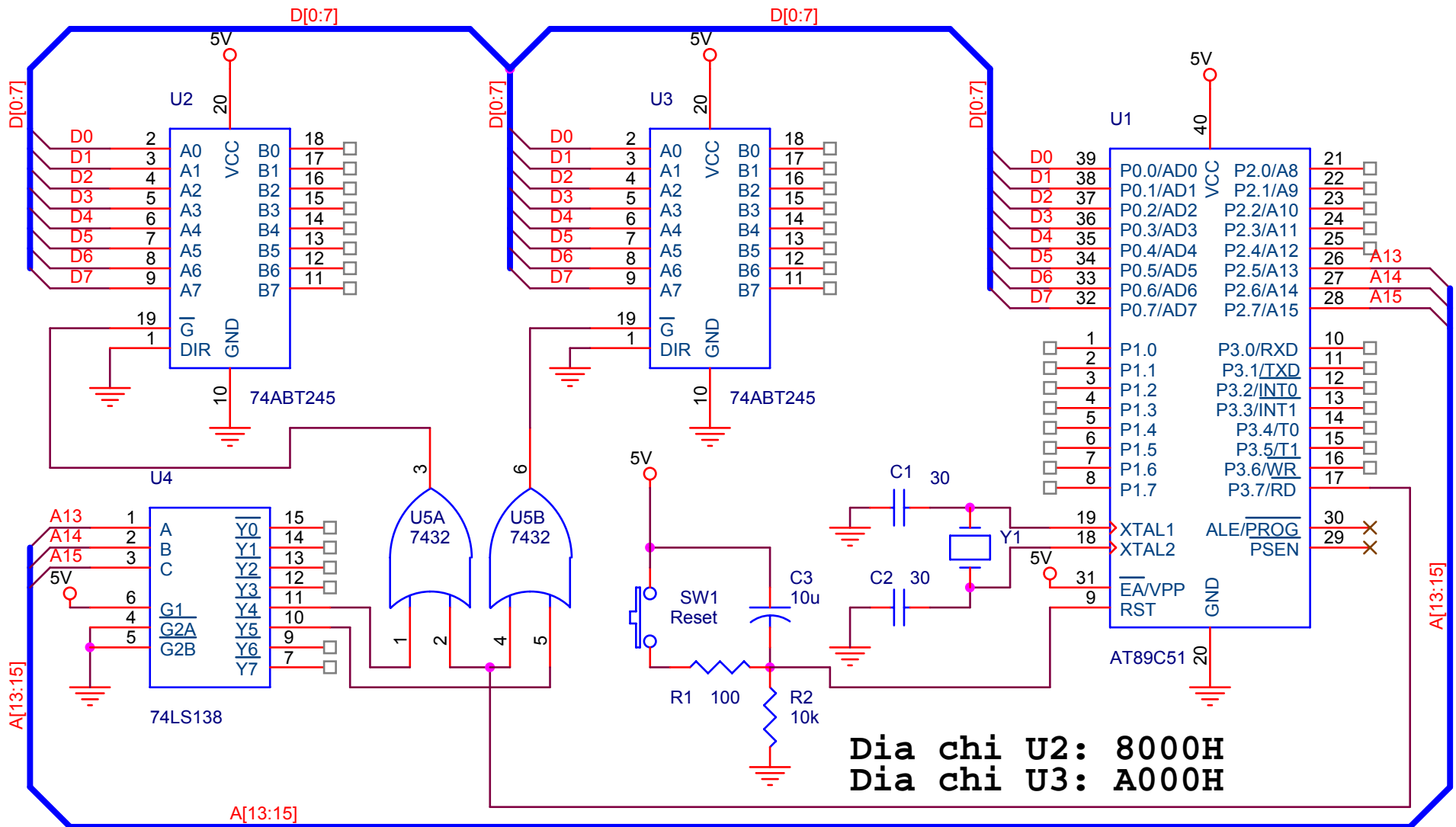
5. Mở rộng Port xuất

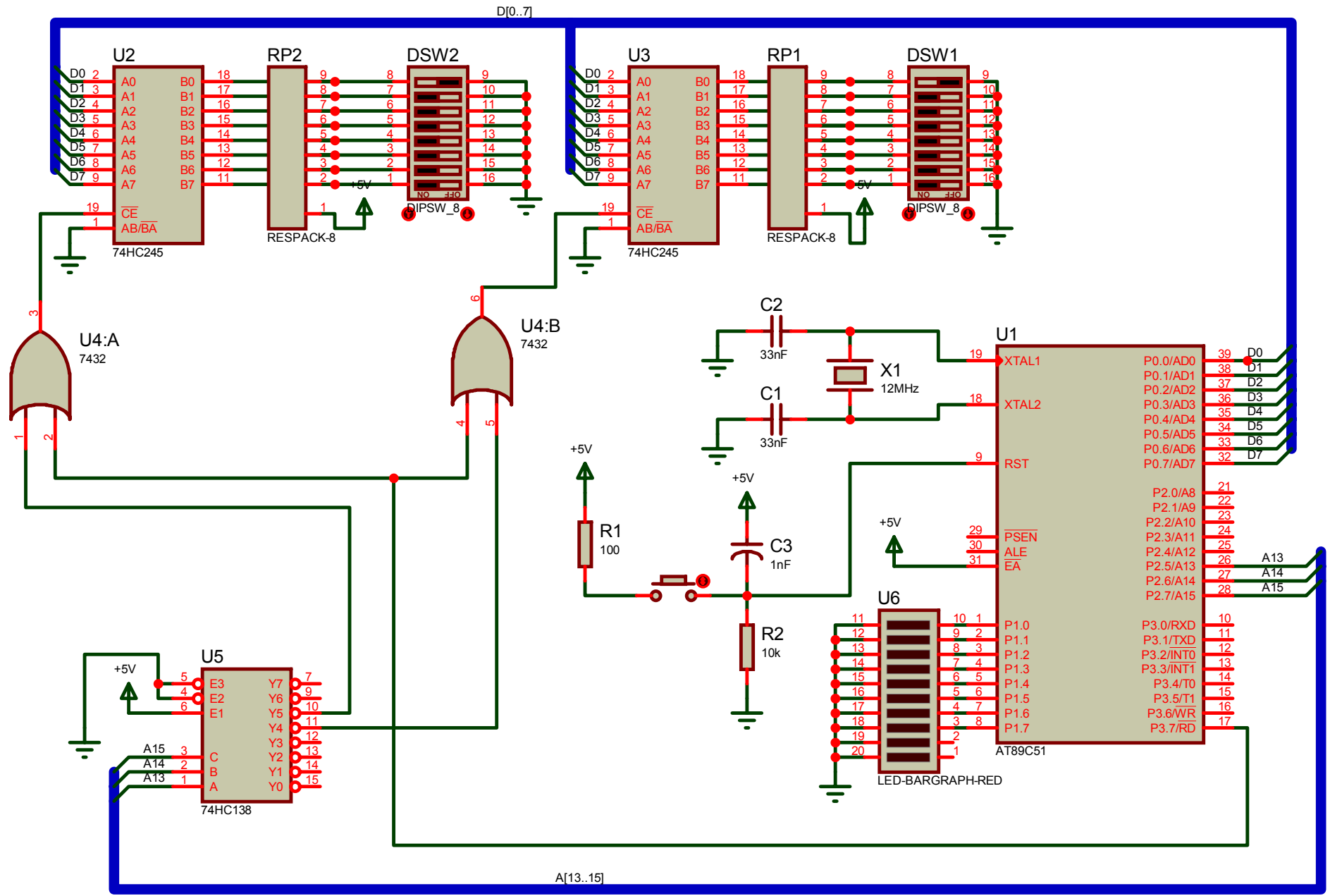




```
#include <regx51.h>
#include <absacc.h>
unsigned char giatri;
void main (void)
{
    while (1)
    {
        P1=0xFF;
        giatri=P1;
        XBYTE[0xC000] = giatri;
        XBYTE[0xE001] = giatri;
    }
}
```

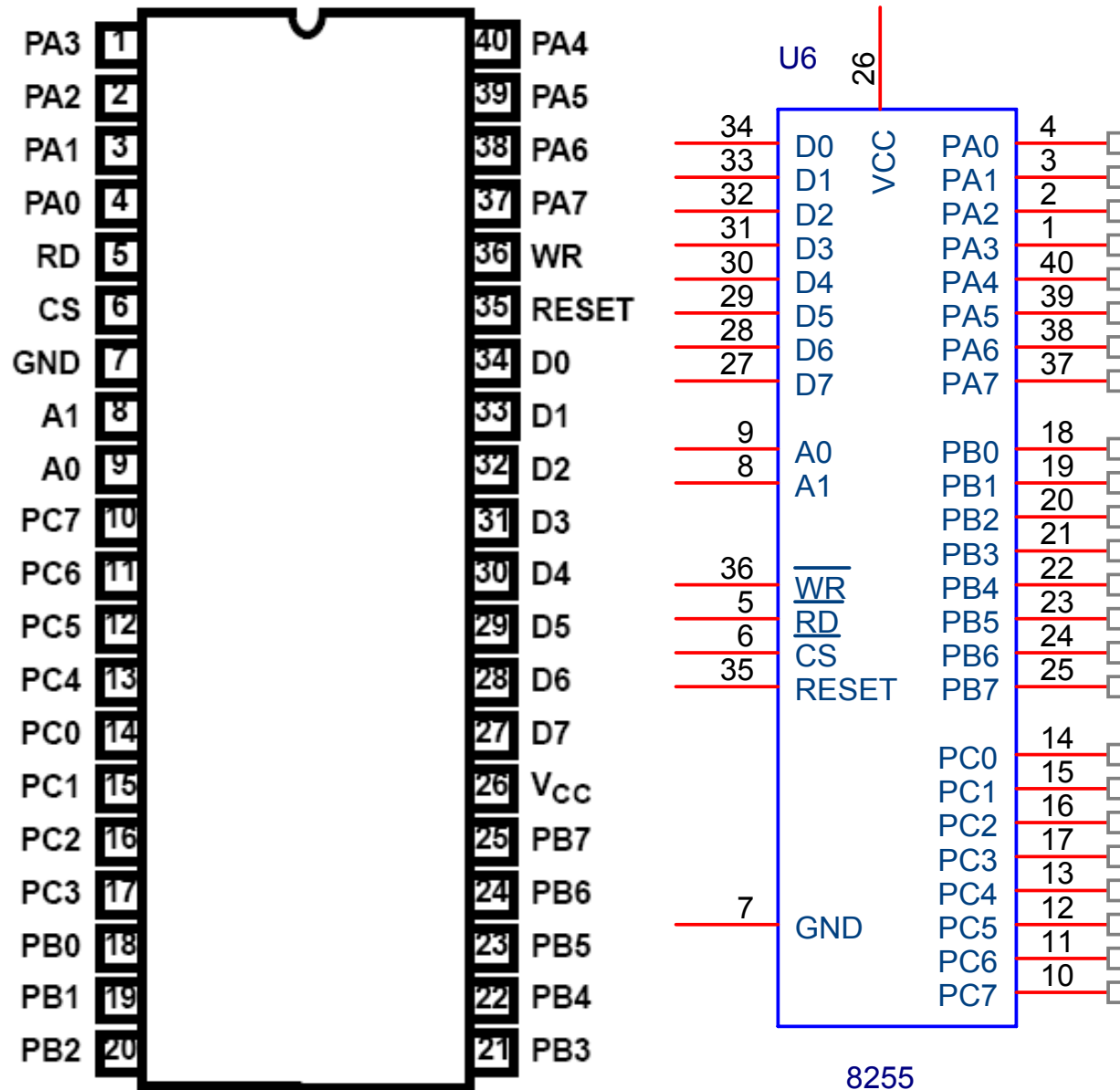
6. Mở rộng Port nhập






```
#include <regx51.h>
#include <absacc.h>
unsigned int giatri1, giatri2, giatri;
void main (void)
{
    while (1)
    {
        giatri1 = XBYTE[0x8000];
        giatri2 = XBYTE[0xA001];
        giatri= giatri1 & giatri2;
        P1=giatri;
    }
}
```

7. Mở rộng port xuất/nhập sử dụng PPI 8255A



7. Mở rộng port xuất/nhập sử dụng PPI 8255A

- PPI: programmable peripheral interface – giao tiếp ngoại vi lập trình được.
- Dùng 8255 để mở rộng I/O. Từng port có thể được lập trình là input hay output một cách linh hoạt bằng phần mềm (so sánh với việc thiết kế I/O port dùng 74LS244 và 74LS373 ở chương 1 → input hay output được thiết kế “cứng”, cố định).

7. Mở rộng port xuất/nhập sử dụng PPI 8255A

- Các chân:

D0÷D7: bus dữ liệu 2 chiều.

PA0÷PA7: port A.

PB0÷PB7: port B.

PC0÷PC7: port C.

/RD: Read. (Nối với /RD (P3.7) của 8051.)

/WR: Write. (Nối với /WR của 8051.)

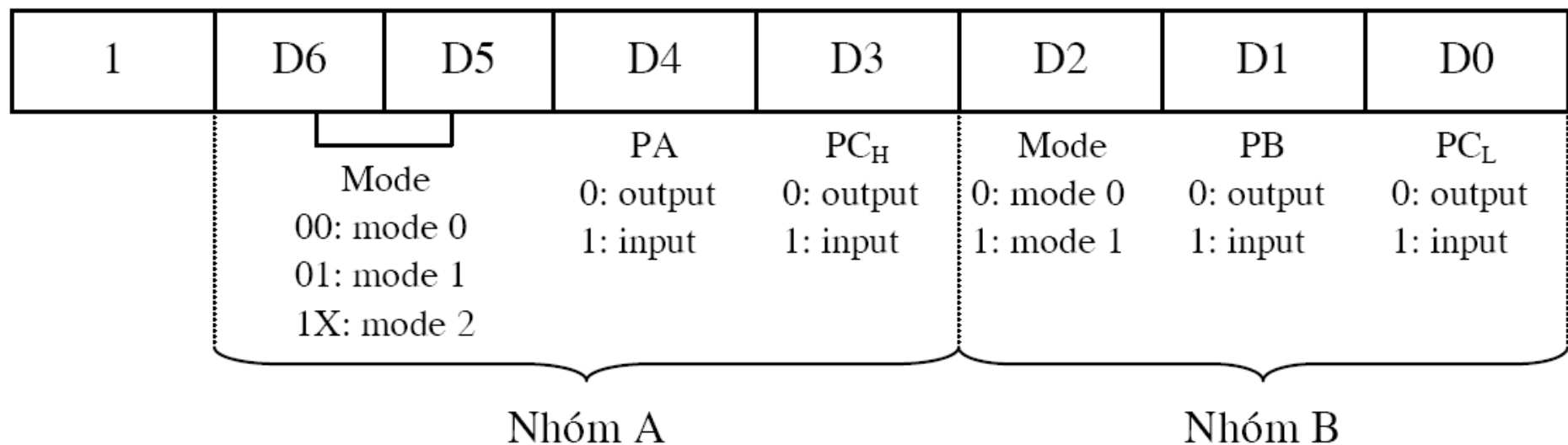
RESET: khởi động lại 8255. (Thường được nối với mạch reset của 8051 hoặc GND.

/CS: chọn chip.)

A0, A1: địa chỉ port. (Nối với bus địa chỉ.)

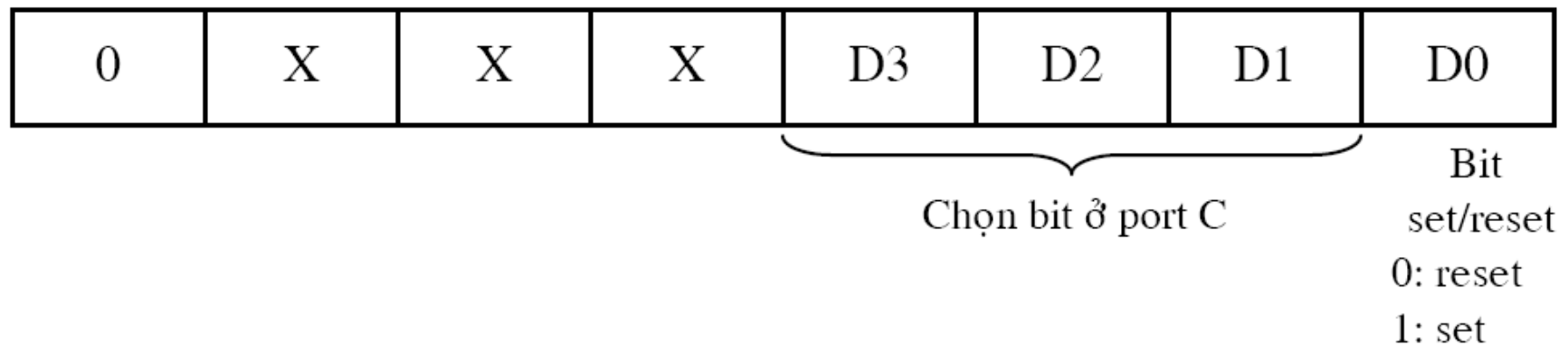
/CS	A1	A0	Mô tả
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Từ điều khiển (control word)
1	×	×	8255 không được chọn

- Thanh ghi điều khiển:
 - Hoạt động I/O (D7 = 1)



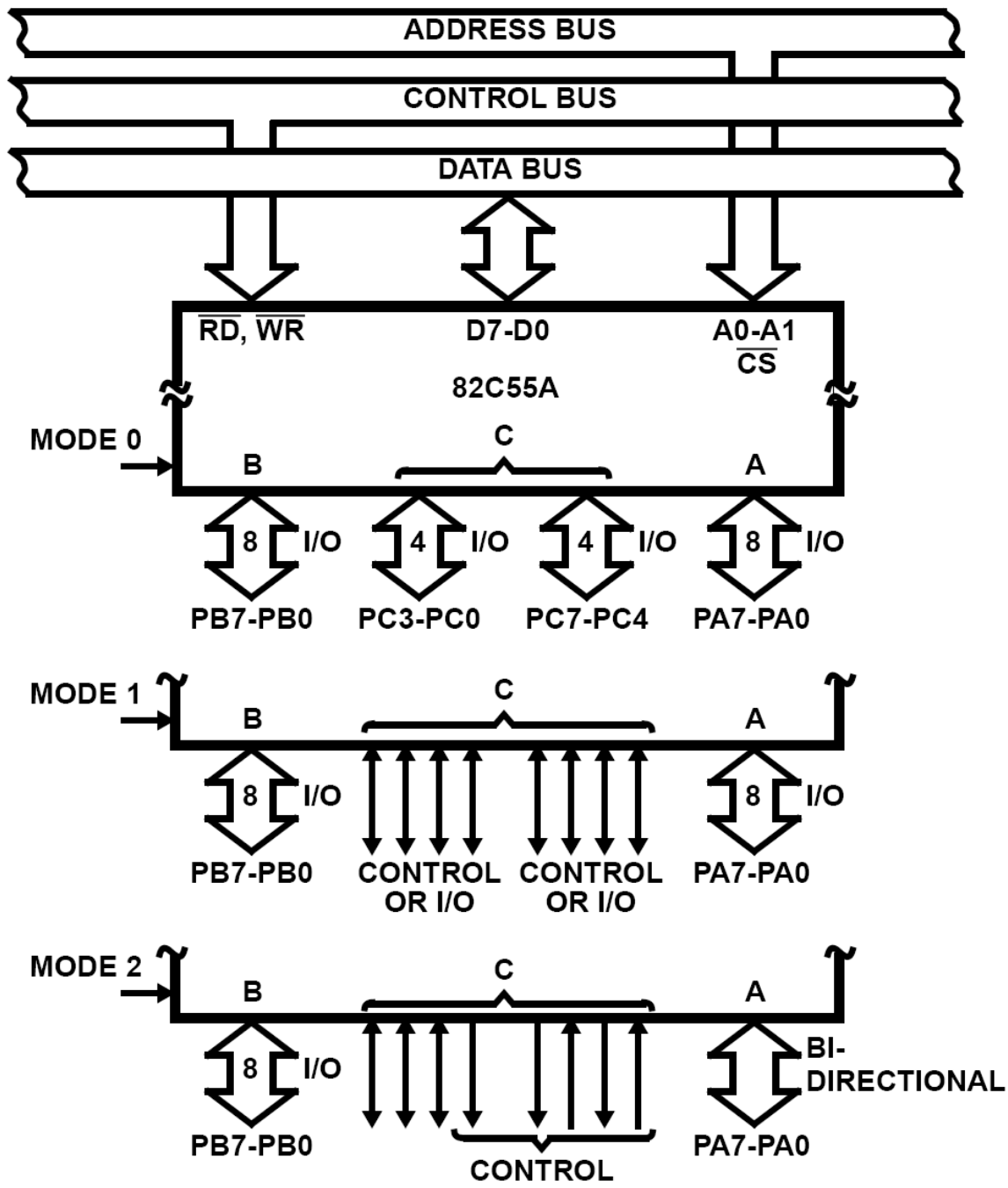
Ghi chú: $PC_H = PC7 \div PC4$, $PC_L = PC3 \div PC0$.

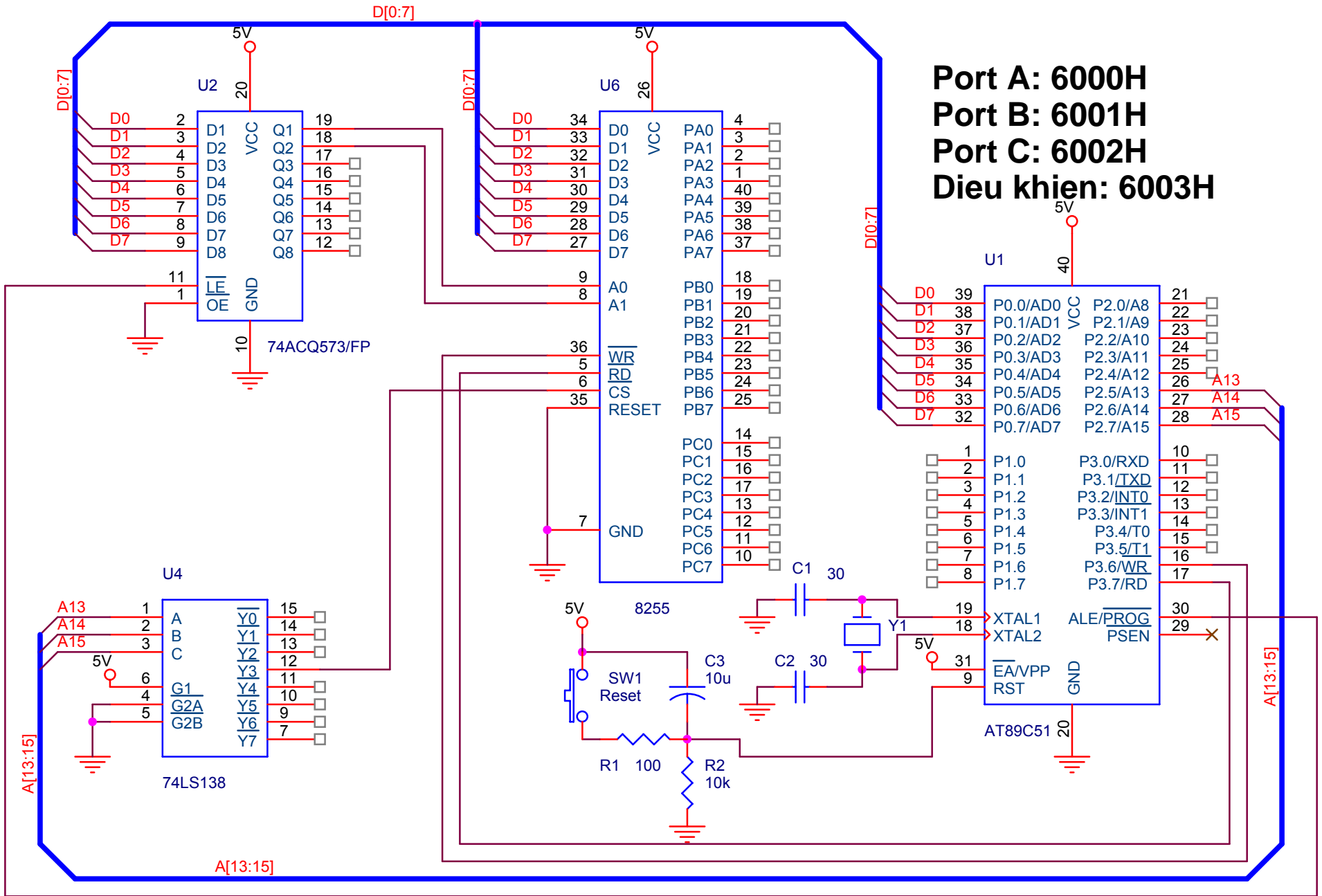
- Hoạt động BSR – Bit set/reset ($D7 = 0$):



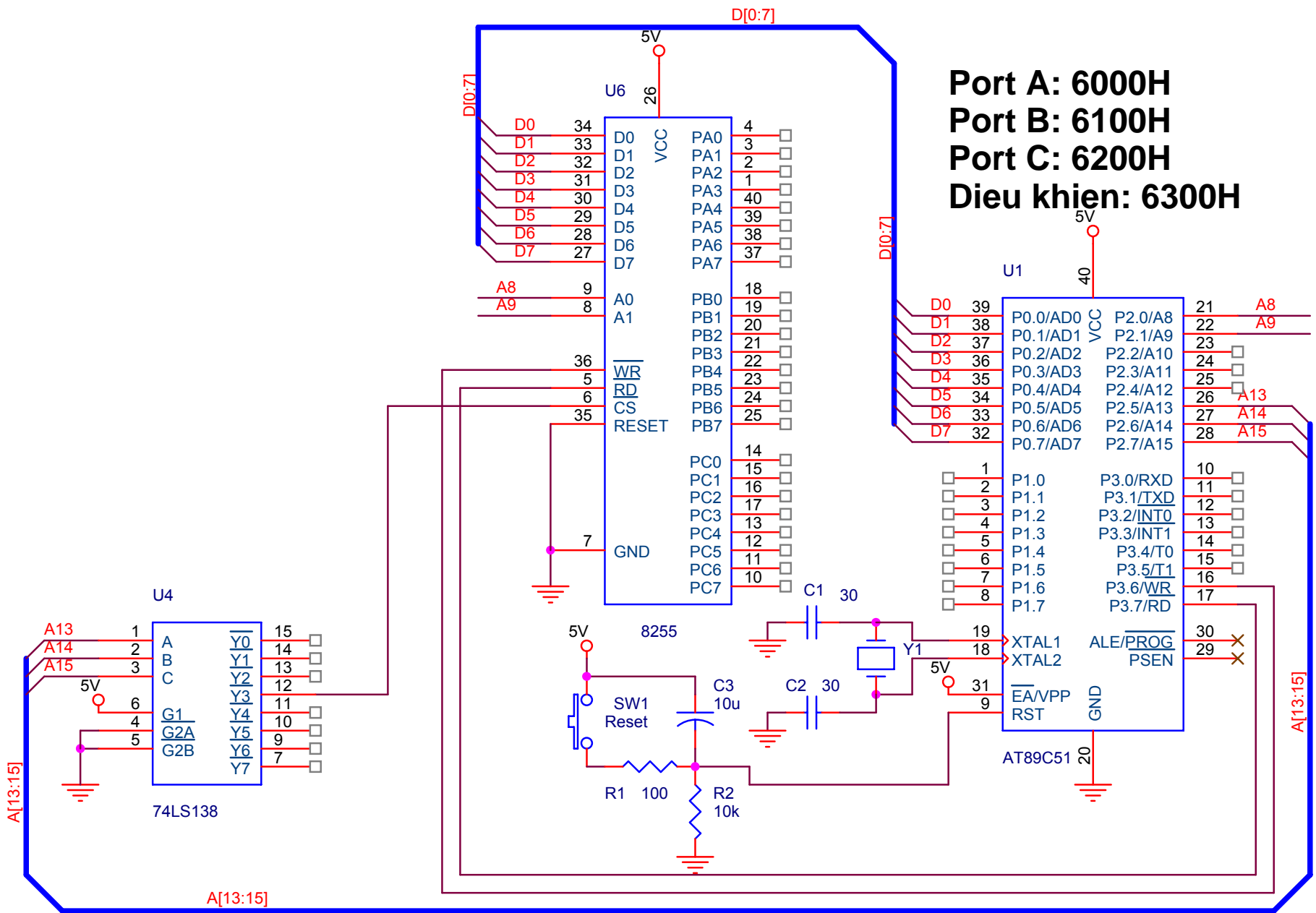
- Hoạt động I/O cơ bản có 3 mode:
 - Mode 0: I/O đơn giản.
 - Mode 1: I/O có bắt tay.
 - Mode 2: bus 2 chiều.

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
OUTPUT OPERATION (WRITE)					
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
DISABLE FUNCTION					
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

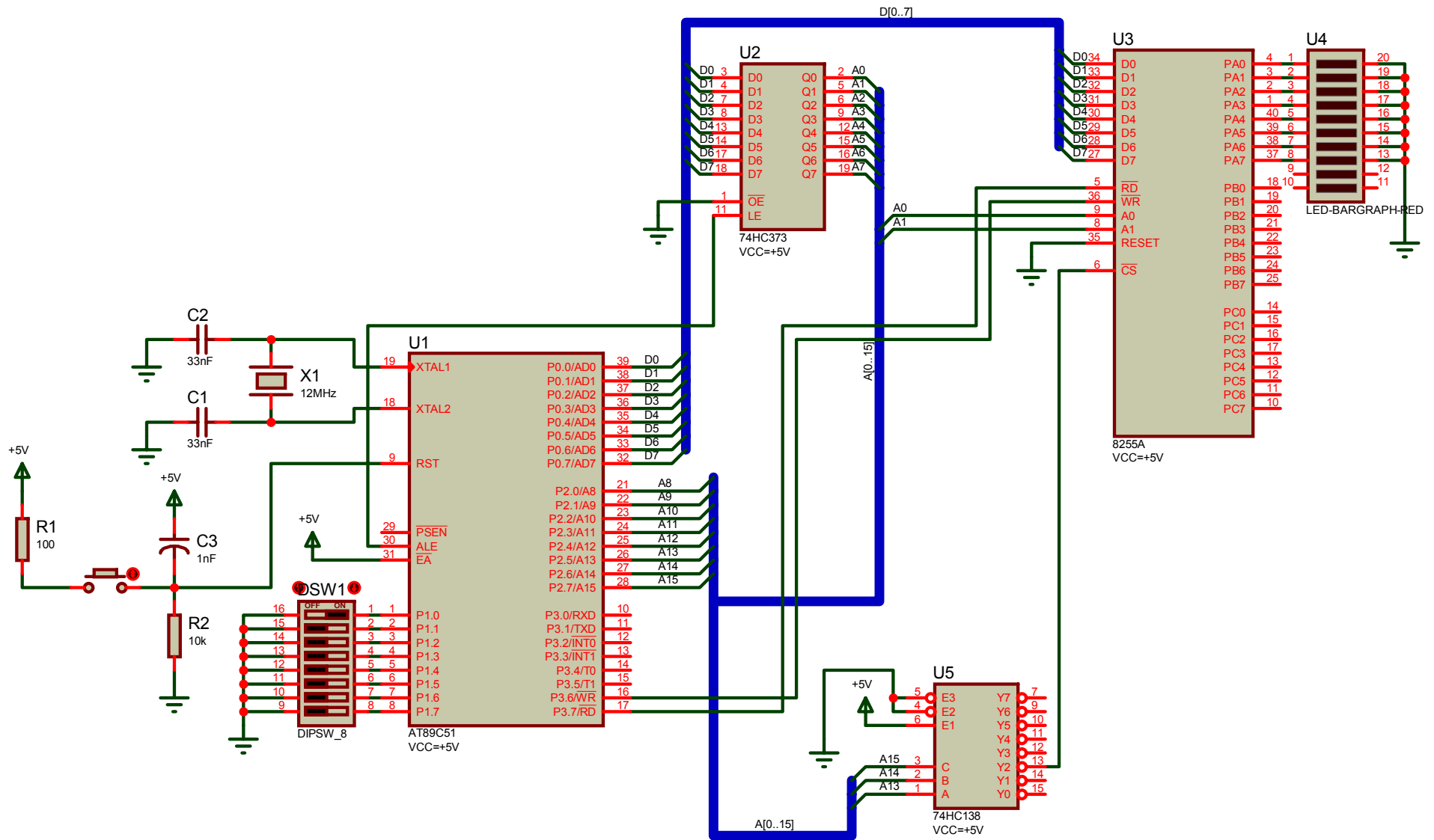




Port A: 6000H
Port B: 6001H
Port C: 6002H
Dieu khien: 6003H



Port A: 6000H
Port B: 6100H
Port C: 6200H
Dieu khien: 6300H



Địa chỉ cơ bản: 4000H, Địa chỉ PortA: 4000H
 PortB: 4001H, PortC: 4002H, Điều khiển: 4003H

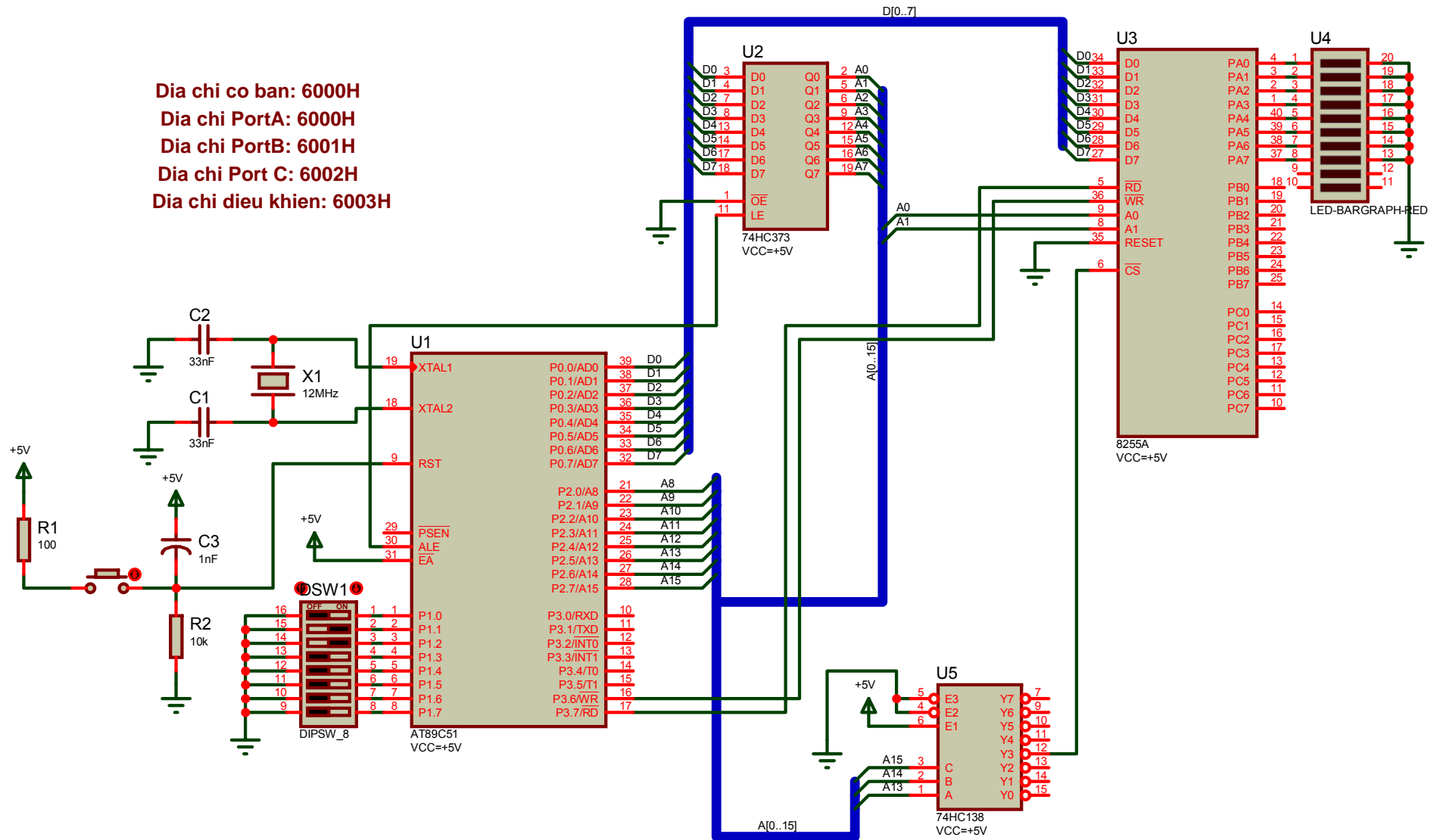
Ví dụ 1

- Cách khai báo: PortA xuất, PortB xuất, PortC xuất.
- Cách khai báo: PortA xuất, PortB nhập, PortC xuất.
- Cách khai báo: PortA nhập, PortB xuất, PortC nhập.

Ví dụ 2:

- Viết chương trình khởi động 8255:
 - PA xuất, PB nhập, PC nhập
 - Liên tục nhập dữ liệu từ P1 của VĐK và xuất ra PA của 8255.

Địa chỉ cơ bản: 6000H
 Địa chỉ PortA: 6000H
 Địa chỉ PortB: 6001H
 Địa chỉ Port C: 6002H
 Địa chỉ điều khiển: 6003H



Địa chỉ cơ bản: 6000H, Địa chỉ PortA: 6000H
 PortB: 6001H, PortC: 6002H, Điều khiển: 6003H

Ví dụ 3

- Cách khai báo: PortA xuất, PortB xuất, PortC xuất.
- Cách khai báo: PortA xuất, PortB nhập, PortC xuất.
- Cách khai báo: PortA nhập, PortB xuất, PortC nhập.

Ví dụ 4:

- Viết chương trình khởi động 8255:
 - PA xuất, PB nhập, PC nhập
 - Liên tục nhập dữ liệu từ P1 của VĐK và xuất ra PA của 8255.

8. Một số ký hiệu trong toán logic

Operator	Description
&	Bitwise AND
	Bitwise OR (inclusive OR)
^	Bitwise XOR (exclusive OR)
<<	Left shift
>>	Right shift
~	One's complement

A	B	A AND B	A OR B	A XOR B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Bài tập 1

- Viết lại chương trình cho VĐK cho các mạch trong chương 2 này.

Bài tập 2

- Viết chương trình cho VĐK thực hiện công việc sau:

Khi $P2.1=0$ và $P2.7 = 0$ thì 8 LED nối Port P1 sáng dần lên.

Khi $P2.1=1$ và $P2.6 = 1$ thì 8 LED nối Port P1 tắt dần xuống.