

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه شهید باهنر کرمان

پروژه درس ریزپردازنده و زبان اسمبلی

طراحی اسانسور

استاد : سرکارخانم دکتر قزوینی

اعضای گروه:

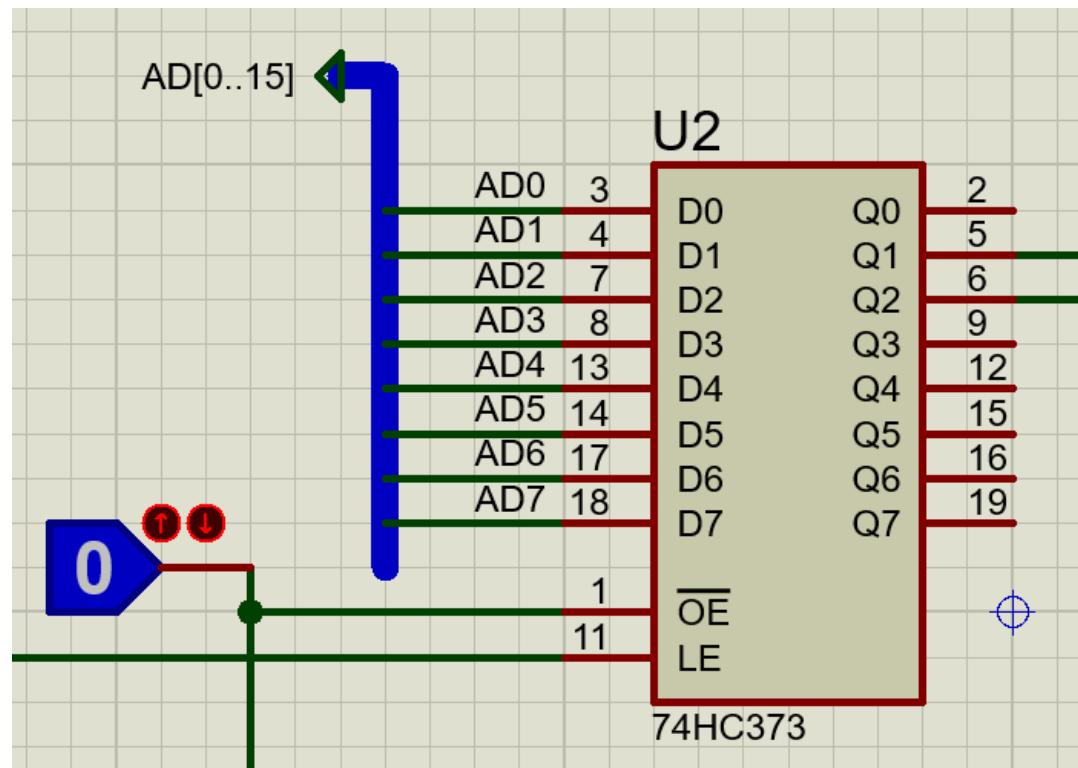
مهدی برهانی نژاد

بابک فتحی

اسماعیل باقری

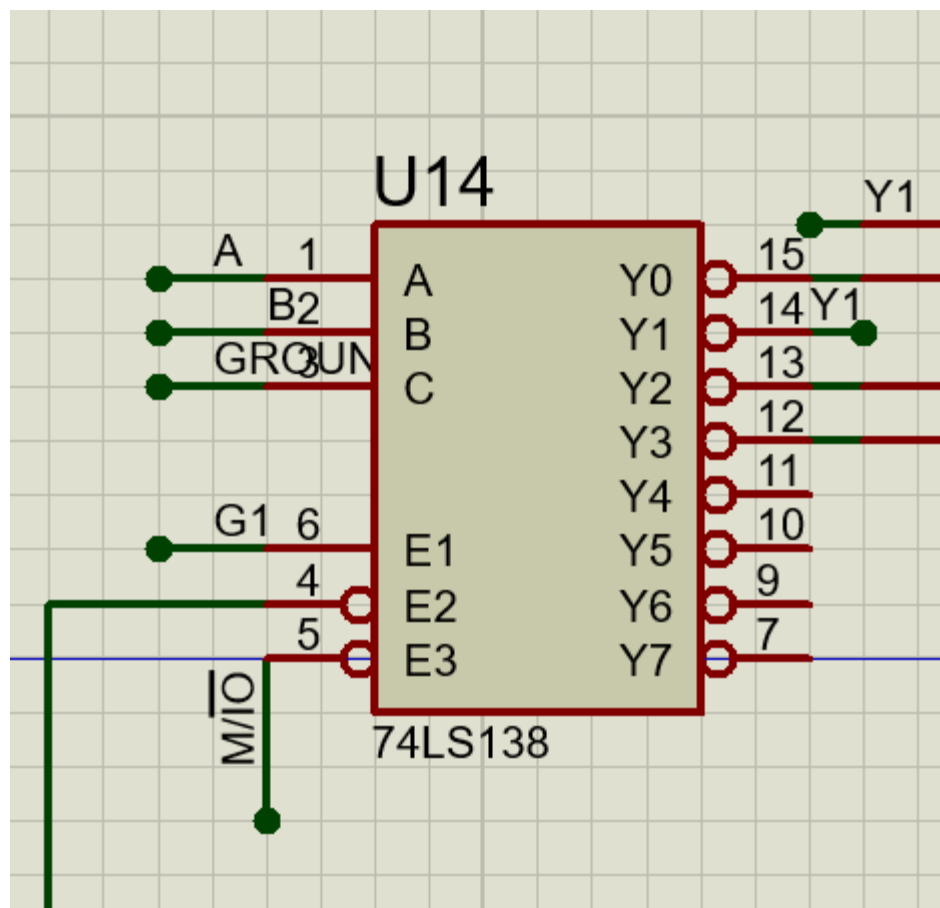
74H373(LATCH):

این لچ دارای دو پایه output enable و latch enable که پایه output enable به دلیل active low بودن همیشه به زمین متصل است و latch enable زمانی فعال میشود که ادرسی در دیتاباس ما قرار بگیرد



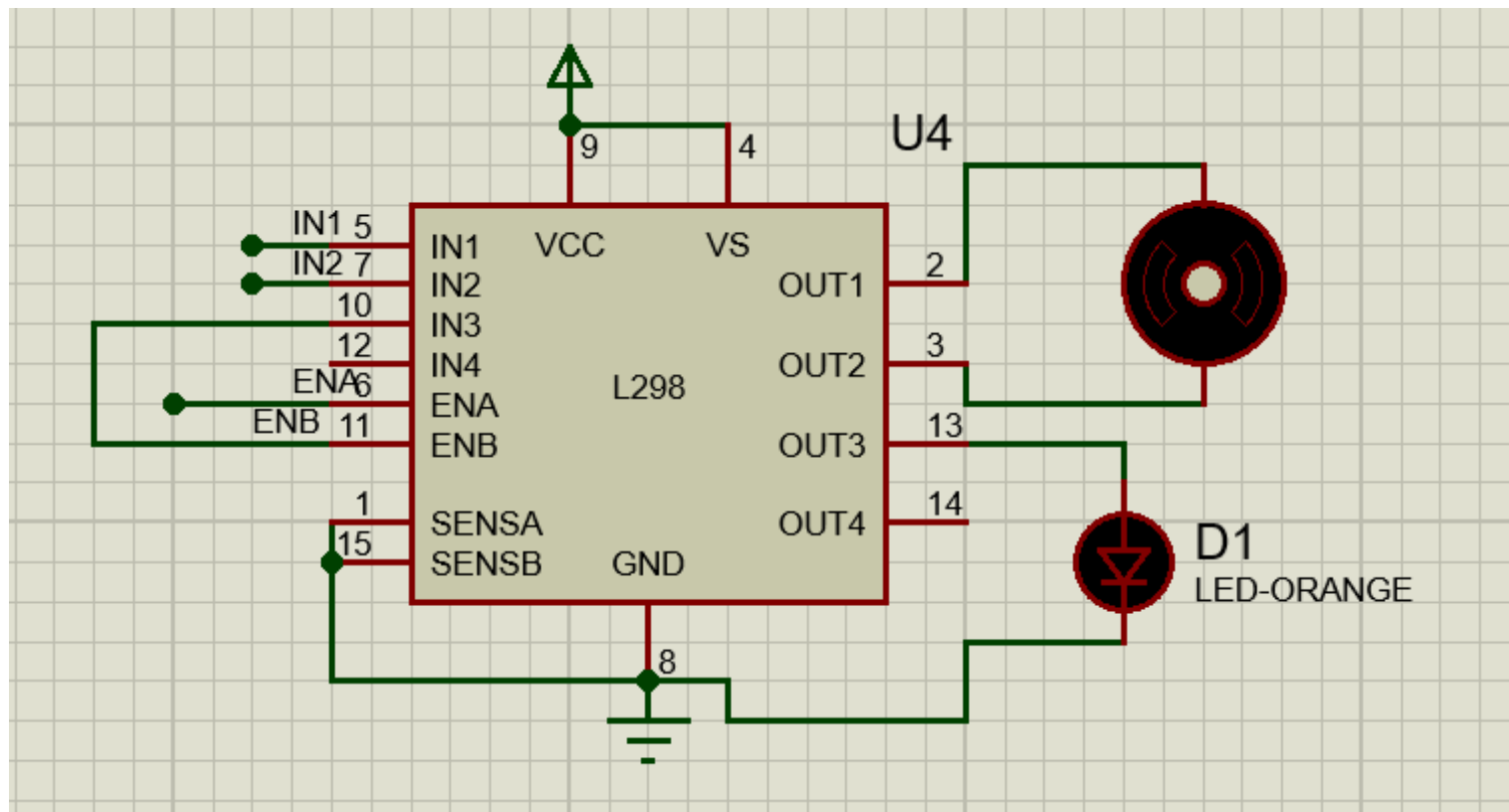
74LS138(DECODER):

در این دیکودر خطوط (A,B,C) خطوط انتخاب ما هستند که مشخص میکنند کدام I/O باید شروع به فعالیت کند و علاوه براین در این دیکودر سه خط enable وجود دارد که مهم ترین آن E2 است زیرا باعث میشود که این دیکودر فقط از طریق ادرس های خاصی شروع به فعالیت کند



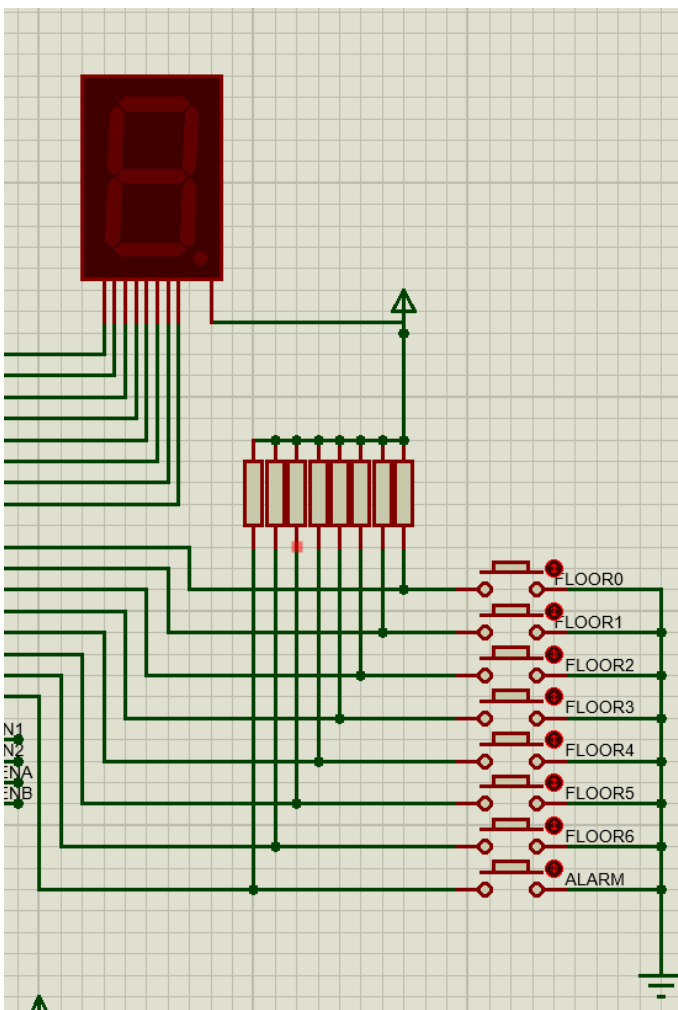
L298:

این قطعه مسئول فعال سازی موتور ما در هنگام بالا یا پایین رفتن و فعال سازی اژیر هشدار در اثر فشردن شدن دکمه الارم اسانسور ما است



7seg:

این قطعه به صورت کاتد مشترک است که در صورت صفر شدن یکی از خطوط آن خط مربوط به آن روشن میشود



BUTTON:

طراحی به صورت pull up که در این طراحی با فشردن دکمه مقدار خروجی آن از 1 به 0 تغییر میکند

وضعیت پایه های 8255 در شکل زیر نشان داده شده است. در این تراشه خطوط D0 تا D7 مسئول مشخص کردن حالت سه درگاه I/O (A,B,C) است.

D7 (پر ارزش ترین بیت): برای اینکه مشخص کنیم سیستم ما به صورت یک ورودی و خروجی ساده کار میکند آن را برابر 1 میگذاریم.

D6_D5: تعیین مود کاری A از سه مود وجود که ما باید آن را به عنوان یک ورودی خروجی ساده در نظر بگیریم بنابراین هر دو آنها برابر 0 میشوند.

D4: تعیین استفاده A به عنوان ورودی (1) یا خروجی (0).

D3_D0: PORTC میتواند هم نقش ورودی را داشته باشد و هم نقش خروجی پس پایه های آن را به دو بخش بیت های کم ارزش (PC0-3) و پر ارزش (PC4-7) تقسیم میکنیم که پایه D4 مسئول تعیین حالت چهار بیت پر ارزش است و D0 مشخص کننده حالت چهار بیت کم ارزش است.

D2: تایین مود کاری PORTB که در اینجا از 0 استفاده شده زیرا حالت 1 برای HANDSHACKING استفاده میشود که در این پروژه کارایی ندارد.

D1: تایین ورودی یا خروجی بودن PORTB.

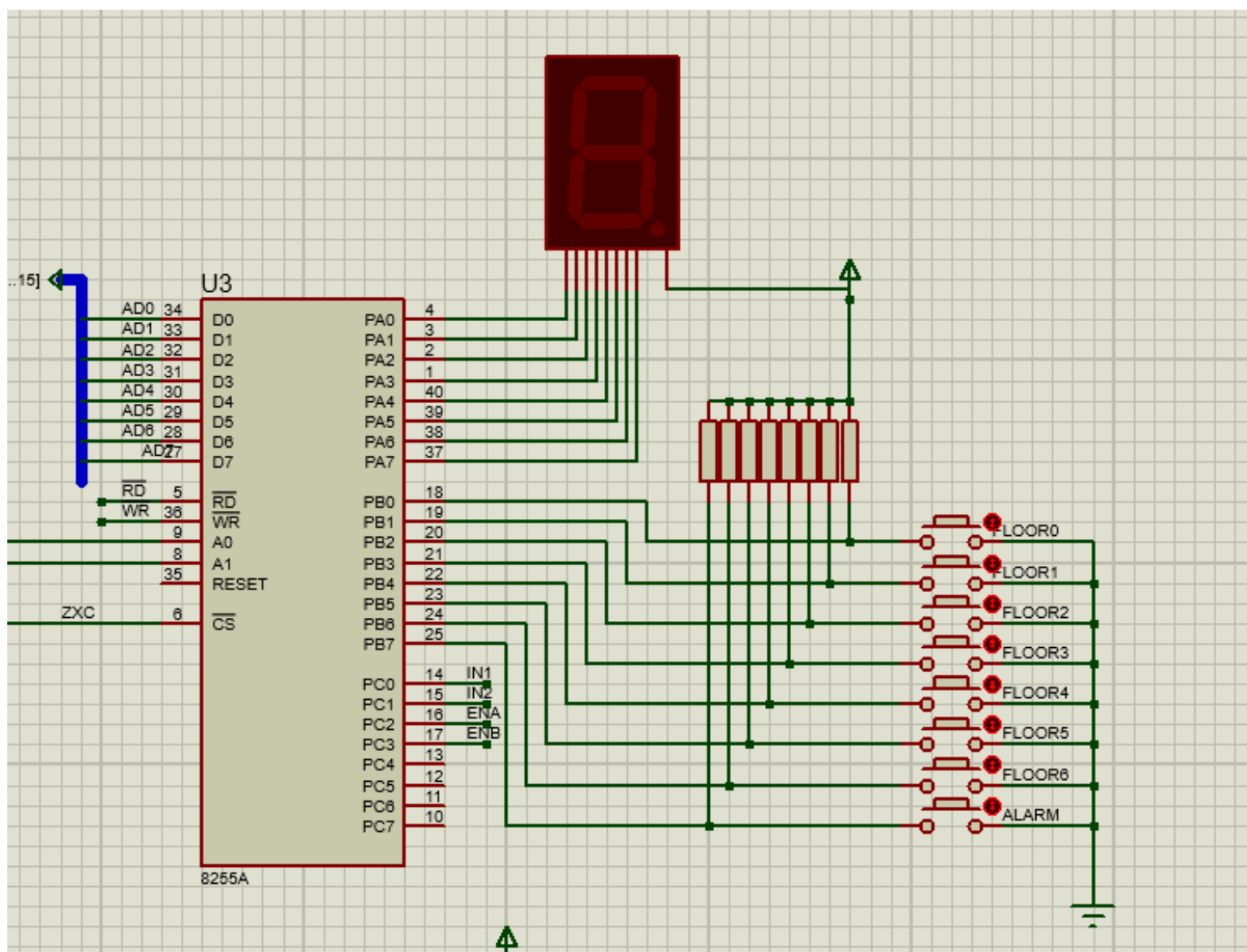
A0_A1: این 2 پایه برای فعال سازی پورت های موجود در 8255 استفاده میشوند که دارای 4 حالت است.

00- فعال سازی PORTA

01- فعال سازی PORTB

10- فعال سازی PORTC

11- فعال سازی پایه ی کنترل



این بخش طراحی شده تا در صورت قرار گرفتن دیتا روی دیتا باس ادرس valid را به درستی بردارد به همین منظور از یک fully decoding system استفاده شده است. ما باید طبق مراحل که در کلاس یاد گرفتیم مراحل دیکود کردن ادرس ها طی کنیم.

1- در این سیستم ادرس های معتبر برابرند با (00,02,04,06) که در صورت بست دادن متوجه میشویم که هشت بیت پر ارزش ان برابر صفر است که در تمام آنها را به اولین لچ خود وصل میکنم و سپس با گرفتن یک nand از خروجی های آن یک منطقی ایجاد میکنیم که این یک منطقی یکی از پای های enable دیکودر ما را تشکیل میدهد.

2- در هشت بیت کم ارزش ما پایه های 7-D3 و D0 هم صفر تولین میکند که از انها به دلیل active low بودن پایه enable از انها or گرفته میشود.

پایه enable دیگر به پایه M/IO پردازنده 8086 متصل میشود.

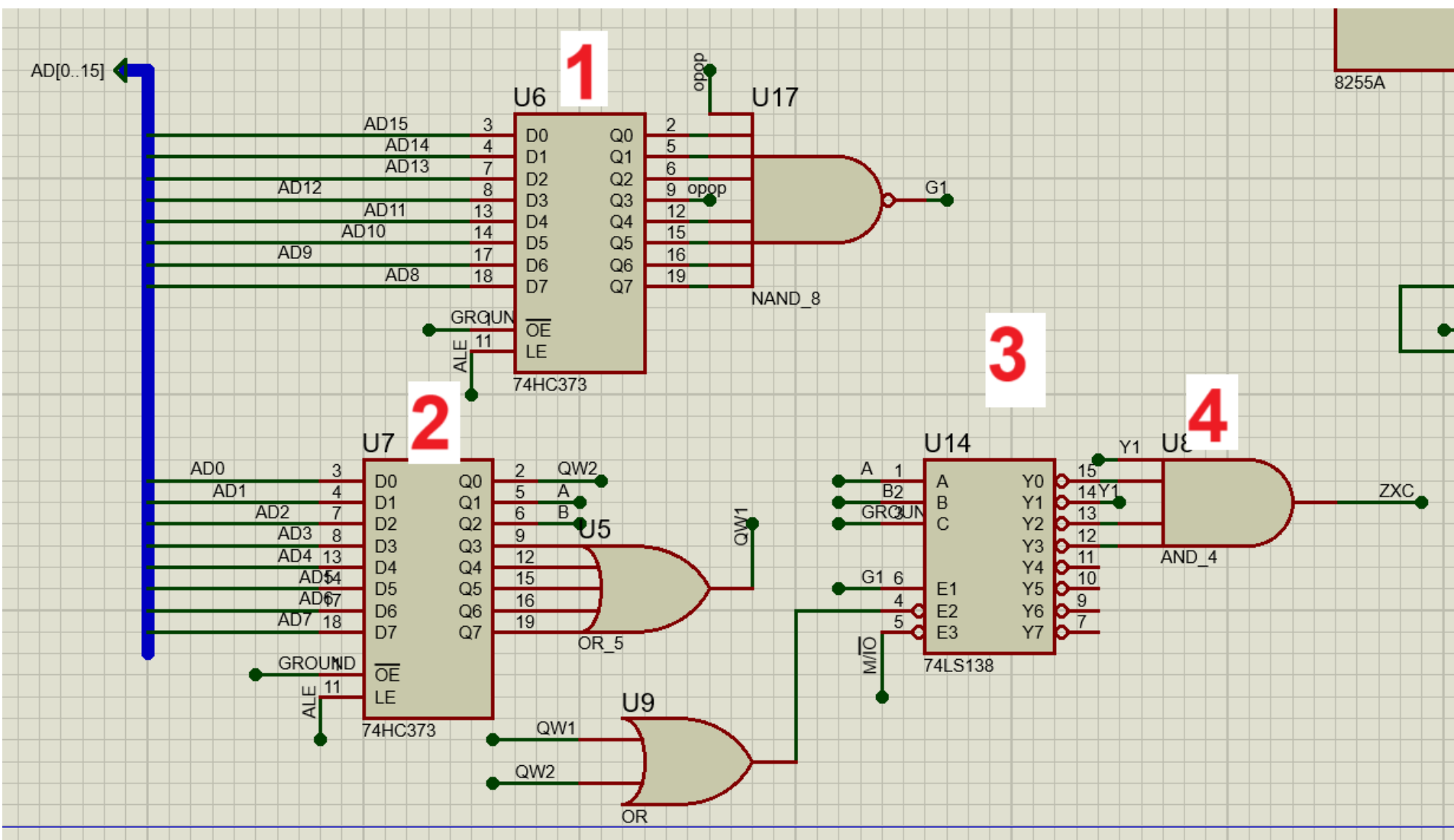
3- پایه های D2,D1 با متصل شدن به خطوط A,B دیکودر ادرس های valid ما را تشکیل میدهند و خط C باقی مانده که وظیفه فعال سازی و غیر فعال سازی چهار بیت کم ارزش و پر ارزش را بر عهده دارد و از ان جایی که ما فقط ۴ ادرس داریم پایه C را به زمین متصل میکنیم تا در صورت قرار گرفتن یکی از چهار ادرس معتبر یکی از ۴ خطوط بالا فعال شود

4- در صورت فعال شدن یکی از ادرس های valid یکی از خروجی های decoder تبدیل به صفر میشود که از nand موجود در چهار بیت اول استفاده میشود تا در صورت فعال سازی cs active low را فعال کند و به 8255 بفهماند ی ادرس در دیتا باس قرار گرفته .

پس از فعال شدن پایه cs با ادرس 06H پایه کنترل ما فعال میشود و مشخص میکند هر کدام از پورت های (A,B,C) باید به صورت خروجی استفاده شوند یا ورودی که از طریق خطوط D که تمامی انها توضیح داده شده این کار را انجام میدهد خطوط کنترلی پروژه اسانسور

D7,D6,D5,D4,D3,D2,D1,D0

10000010



کد های 8086

DATA SEGMENT

PORTA EQU 00H

PORTB EQU 02H

PORTC EQU 04H

PORT_CON EQU 06H

feli DW 0h

maghsad DW 0h

Table7Seg DB 0c0h, 0f9h, 0a4h, 0b0h, 099h, 092h, 082h, 088h

;HERE WE DEFINE LOOKUP TABLE FOR DISPLAY NUMBERS IN 7SEGMENT

DATA ENDS

CODE SEGMENT

START ASSUME DS:DATA ;

;USE DS AS DEFAULT SEGMENT;

MOV DX, PORT_CON

MOV AL, 082h

;ACTIVATING CONTROL PORT FOR 8255

OUT DX, AL

MOV AL, 0c0h

MOV DX, PORTA

;SHOWING ZERO IN 7SEGMENT;

OUT DX, AL


```
click:                                ; IN THIS FUNCTION COMPARE THE BUTTON INPUT
MOV DX, PORTB                        ; IN ORDER TO UNDERSTAND WHICH BUTTON IS USED
IN AL, DX
```

```
CMP AL, 11111110B
JZ F0
CMP AL, 11111101B
```

```
JZ F1
CMP AL, 11111011B
JZ F2
CMP AL, 11110111B
JZ F3
CMP AL, 11101111B
JZ F4
CMP AL, 11011111B
JZ F5
CMP AL, 10111111B
JZ F6
CMP AL, 01111111B
JZ ALARM
JMP click
```

```
F0:                                ; COMPARING THE MAGSHAD AND FELI STATE TO UNDERSTAND
MOV SI, OFFSET maghsad             ; THAT ELEVATOR MUST GO DOWN OR UP;
MOV BX, 0                          ; OTHER LEVEL ARE NOT WRITTEN TO COMPACT THE CODE SIZE
```

```
MOV [SI], BX
MOV SI, OFFSET feli
MOV CX, [SI]
CMP BX, CX
JG L6
CMP BX, CX
JL L8
```

```
MOV AL, 11000000B
MOV DX, PORTA
OUT DX, AL
JMP click
```

ALARM:

```
    mov dx,portc
    mov al,00001000B    ;TURN ON THE LED;
    out dx,al
    MOV AL, 088h    ; HERE WE PUSH CX AND MAKE A INTERNAL DELAY FUNCTION
    PUSH CX
    MOV CX, 05FH
    MOV DX, PORTA
    out dx,ax
D2:
    nop
    LOOP D2
                                ;A LOOP TO CONSUME SPEND SOME TIME IN ALARM MODE;

    POP CX
```

```
    MOV SI, OFFSET feli
    MOV aX, [SI]
    MOV SI, OFFSET Table7Seg
    ADD SI, ax            ;SHOWING FELI STATE IN 7SEGMENT AND START THE
    mov ax,[si]          ;AGAIN;
    OUT DX, Ax
        mov dx,portc
    mov al,00000000B    ;WITH THIS INSTRUCTION WE DISABLE MOTOR ELEVATOR
                                ;AND ALARA LED THAT CONNECTED TO PORT_C
    out dx,al
    JMP click
```

```
    JMP START
L6:                                ;THIS FUNCTION IS FOR UNDRESTDANDING HOW MUCH WE MUST
        mov dx,portc    ;GO DOWN BY USING SUB METHOD
    mov al,00000110B    ; HERE WE TURN ON THE ELEVATOR MOTOR AND ELEVATOR
                                ;STARTS GOING DOWN
    out dx,al
    MOV SI, OFFSET maghsad
    MOV BX, [SI]
    MOV SI, OFFSET feli
    MOV CX, [SI]
    SUB BX, CX
    XCHG BX, CX
```

L7: ;GOING DOWN STEP BY STEP USING A LOOP;

```
CALL DELAY ;WE WILL DISCUSS ABOUT THIS FUNCTION LATER
INC BX
MOV SI, OFFSET Table7Seg
ADD SI, bx
MOV AL, [SI]
MOV DX, PORTA
OUT DX, AL ; WE DISPLAY CURRENT FLOOR IN 7SEGMENT
MOV SI, OFFSET feli ; HERE WE UPDATE VALUE OF FELI
MOV aX, [SI]
inc ax
mov [si],ax
LOOP L7
    mov dx,portc
mov al,00000000B ; WE ARRIVED TO THE DESIRED FLOOR AND DISABLE MOTOR
out dx,al
JMP click
```

L8: ;HOW MUCH WE MUST GO UP BY COMPARING MAGSHA AND FELI STATE

```
    mov dx,portc ; USING ADD METHOD;
mov al,00000101B ;HERE WE TURN ON THE ELEVATOR MOTOR AND ELEVATOR
                    ; STARTS GOING UP

out dx,al
MOV SI, OFFSET maghsad
MOV BX, [SI] ;HERE WE LOAD MAGHSAD IN BX REGISTOR
MOV SI, OFFSET feli
MOV CX, [SI] ;HERE WE LOAD FELI IN CX REGISTOR
SUB cx,bx ; WE CALCULATE DIFFERENCES BETWEEN FLOORS AND SAVE VALUVE IN CX FOR
          ;USING IT IN LOOP

MOV SI, OFFSET feli ;HERE WE LOAD FELI IN BX REGISTOR AGAIN FOR USING IN L9 LABLE
MOV bx, [SI]
```

L9: ; GOING UP STEP BY STEP USING A LOOP;

```
call delay
MOV SI, OFFSET feli
MOV aX, [SI] ; WE HAVE CURRENT FLOOR IN AX
dec ax ;WE REDUSE THE CURRENT FLOOR
mov [si],ax ; WE UPDATE VALUE OF FELI
MOV SI, OFFSET Table7Seg
ADD SI, ax

MOV AL, [SI]
MOV DX, PORTA ;WE DISPLAY THE CURRENT FLOOR IN 7SEGMENT
OUT DX, AL
LOOP L9 ;THIS LOOP RUNS AS MUCH AS THE DIFFERENCES BETWEEN THE FLOORS
    mov dx,portc
mov al,00000000B ;WE DISABLE MOTOR
out dx,al
JMP click
```

;EXPLAIN OF DELAY FUNCTIONS

DELAY PROC NEAR

PUSH CX

MOV CX, 01FH ;HERE WE USE A SPECIFIED NUMBER TO MAKE A DELAY

D1: ;HERE WE READ PORT_B EVERY MOMENTS AND IF THE BUTTON WAS THE ALARM WE STOP
THE ELEVATOR

MOV DX, PORTB in al,dx

cmp al,07fh

jz ALARM

LOOP D1

POP CX

RET ;WE MUST RETURN FROM THIS SUB FUNCTION

DELAY ENDP

CODE ENDS END OF THE PROJECT;

END START



The End