



**Faculty of Engineering & Technology
Electrical & Computer Engineering Department**

**MICROPROCESSOR-BASED SYSTEMS
ENCS338**

Project Part 2

Ahmad Barhoum

Design in Proteus:

We added the design in figure 1 to project 1 in figure 2

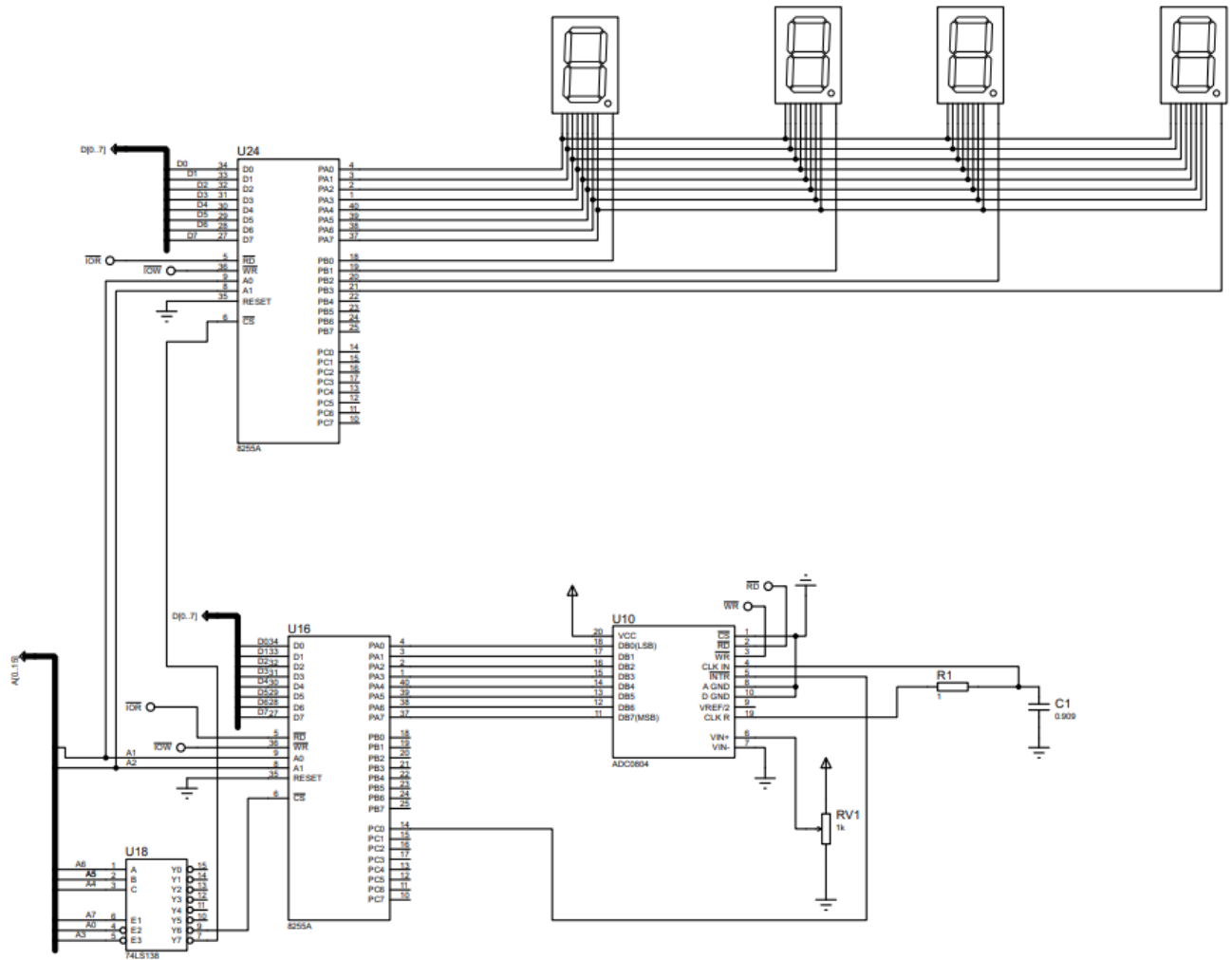


Figure1: project 2 design

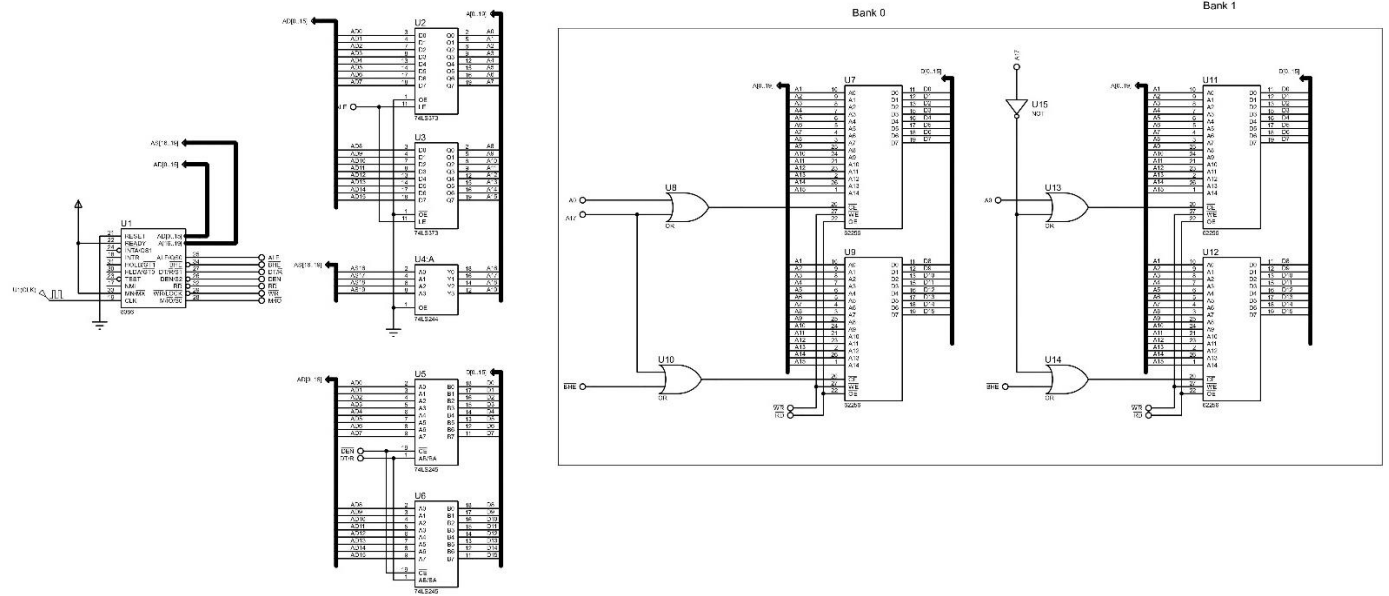


Figure 2: project 1

We can also see a 3 to 8 decoder which was used as chip select for the two PPIs that we used, 001 used to select the PPI connected to the ADC in figure 3, and 111 used to select the PPI connected to the 7-segments display in figure 4.

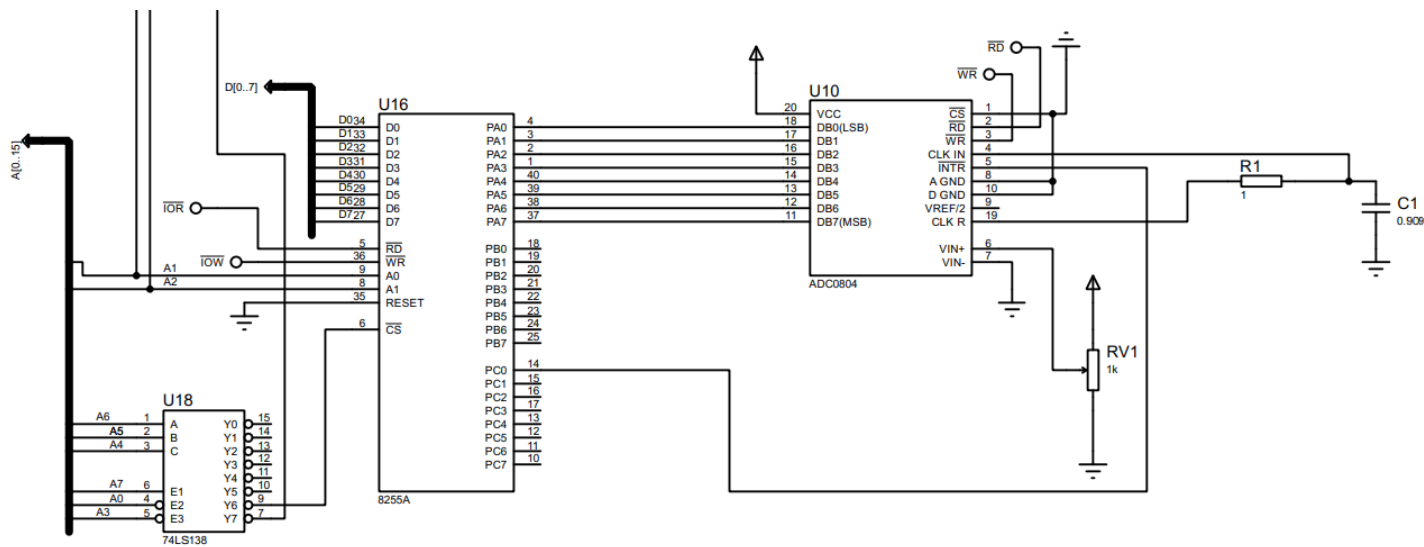


Figure 3: PPI connected to ADC

The 7 segment displays were used to display the average of the data taken every second.

Port A displays the output on all 4 7 segment displays, port b acts as select for which segment to display the output on.

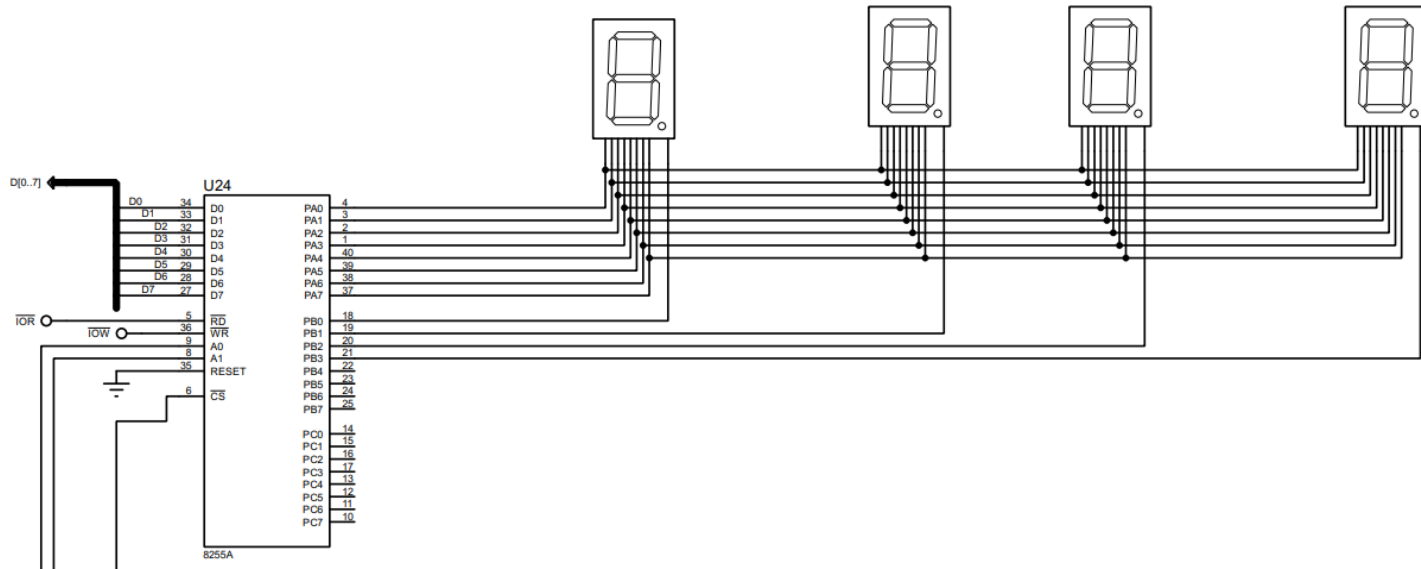


Figure 4: PPI connected to 4 7-segment displays

Code:

First we set the command registers in the PPIs, since we used A1 and A2 as select for the registers/ports inside the PPIs the difference was 2.

For the PPI which was connected to the ADC:

11100xx0 : E0 to E6

11100000 for port A

11100010 for port B

11100100 for port C

11100110 for command register

For the PPI which was connected to the 7 segments display:

11110xx0: F0 to F6

11110000 for port A

11110010 for port B

11110100 for port C

11110110 for command register

```
; setting command registers
```

```
mov dx,0E6h ;11100110 ;setting device 2
```

```
mov al,10110100b ;port A input ;mode 1 ;bc port c instructions
```

```
out dx,al
```

```
mov dx,0F6h ;11110110 ;setting device 1
```

```
mov al,10000000b ;port A and Port b output
```

```
out dx,al
```

Then in a loop that loops 100 times we took the input data from port A from the PPI in figure 3 and calculated the average, then displayed the average in the 7- segment display.

The selection pattern 01110111 in port B, as we can see in figure 4 only the first 4 pins were used, however since the port b rotates non of the pins should be left empty, the same patten on the first 4 pins was repeated for the last 4 pins so the order won't change after rotation.

```
21
22  mov Bx, 100
23  lea si, array
24  readData:
25  cmp Bx, 100
26  jz exit
27  dec bx
28
29  IN al, 0E0 ; input from port A
30
31  mov [si], al
32  inc si
33
34  add sum, al
35  inc num
36  cmp num, 0 ;to avoid divition by 0
37  jz readData
38  mov al, sum
39  xor ah, ah
40  mov bl, num
41  div bl
42
43  mov avg, al
44
45  mov ah, selectionPattern
46  mov dx, 0F2H ; 11110010 ; A2A1 01 for port b
47  mov al, ah
48  out dx, al
49  ROR ah, 1
50  mov selectionPattern, ah
51
52  mov al, avg
53  mov dx, 0F0H ; port a
54  out dx, al
55
56  Call DELAY
57  jmp readData
58
```

Each time DELAY procedure was called so there will be a 1 second delay before the next input is loaded.

```
61  
62  DELAY PROC  
63  MOV CX,1 ;1 second  
64  L1:  
65  LOOP L1  
66  RET  
67  DELAY ENDP  
68
```


Full Code:

```
1  .model small
2  .stack 100
3  .DATA
4  array DB 100 DUP(?)
5  sum DB 0
6  num db 0
7  avg Db 0
8  selectionPattern db 77H ; use this variable as selection pattern for port b
9
10 .code
11 mov ax,@data
12
13 ; setting command registers
14
15 mov dx,0E6h ;11100110 ;setting device 2
16 mov al,10110100b ;port A input ;mode 1 ;bc port c instructions
17 out dx,al
18 mov dx,0F6h ;11110110 ;setting device 1
19 mov al,10000000b ;port A and Port b output
20 out dx,al
21
22 mov Bx, 100
23 lea si, array
24 readData:
25 cmp Bx,100
26 jz exit
27 dec bx
28
29 IN al, 0E0 ; input from port A
30
31 mov [si], al
32 inc si
33
34 add sum, al
35 inc num
36 cmp num, 0 ;to avoid divition by 0
37 jz readData
38 mov al, sum
39 xor ah,ah
```

```

40  mov bl, num
41  div bl
42
43  mov avg, al
44
45  mov ah, selectionPattern
46  mov dx, 0F2H ; 11110010 ; A2A1 01 for port b
47  mov al, ah
48  out dx, al
49  ROR ah,1
50  mov selectionPattern, ah
51
52  mov al, avg
53  mov dx, 0F0H ; port a
54  out dx, al
55
56  Call DELAY
57  jmp readData
58
59  exit:
60  JMP exitt
61
62  DELAY PROC
63  MOV CX,1 ;1 second
64  L1:
65  LOOP L1
66  RET
67  DELAY ENDP
68
69  exitt:
70  END
71

```