

Vending Machine

SAYISAL DEVRE ELEMANLARI

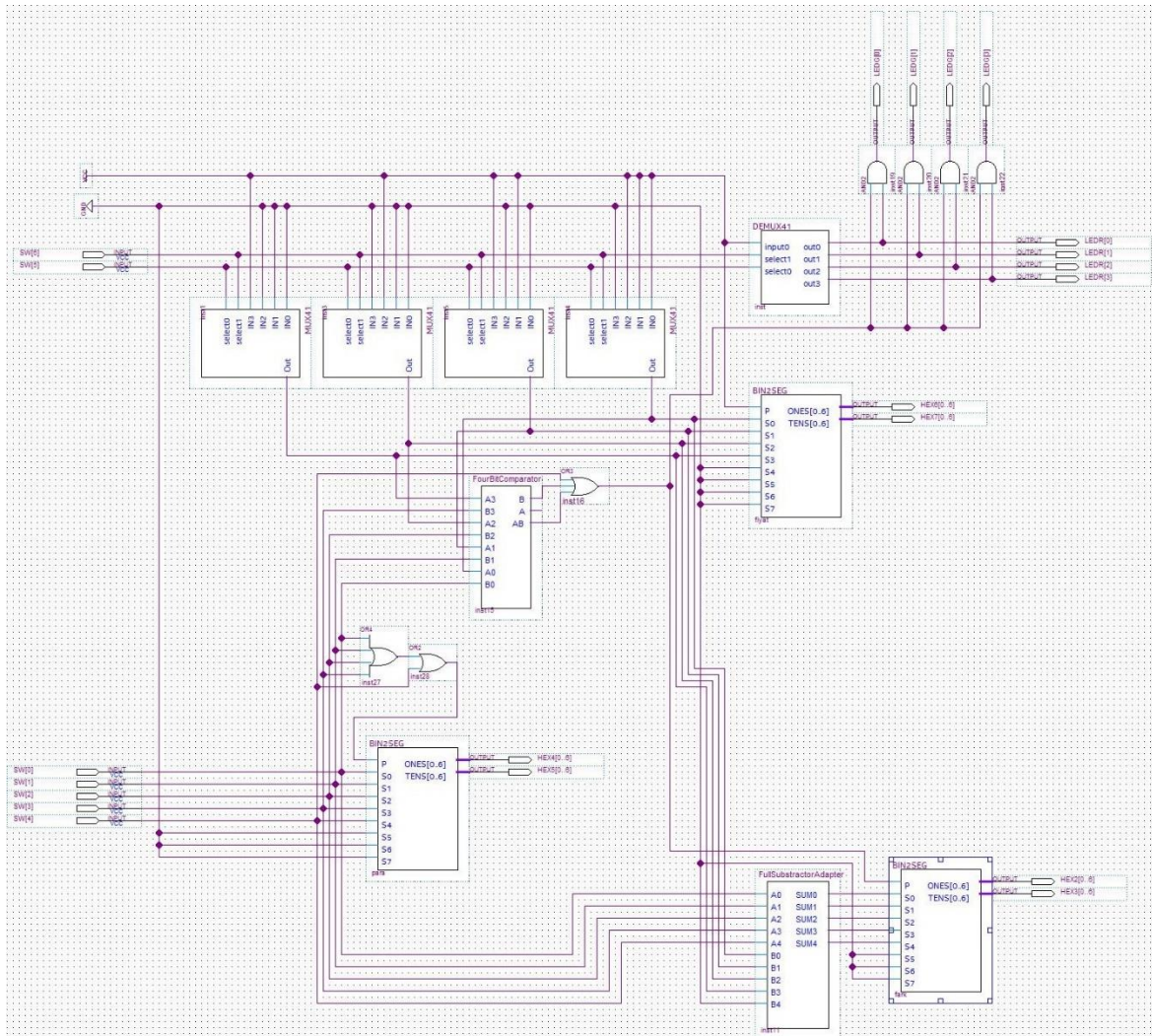
Ahmet Bersan Şahin - Samera Sahar - Kseniia Kholodkovskaia - Asil Aygün
 13061087 - 15060925 - 15060919 - 15060017

İçindekiler

• Sistem	2
○ Devre Şeması.....	2
○ Modüller.....	3
○ Giriş ve Çıkışlar.....	3
○ Çalışma Prensibi.....	3
• Modüller.....	4
○ MUX41.....	4
○ DEMUX41.....	4
○ BIN2BCD.....	5
○ BCD2SEG.....	7
○ BIN2SEG.	8
○ FourBitComparator.....	9
○ Para Giriş Kontrolcüsü.....	10
○ FullSubtractor.....	10
○ FullSubtractorAdaptor.....	11

Sistem

Devre Şeması



Modüller

- MUX41
- DEMUX41
- FourBitComparator
- FullSubtractorAdapter
 - FullSubtractor
- BIN2SEG
 - BIN2BCD
 - Add3
 - BCD2SEG

Girişler ve Çıkışlar (I/O)

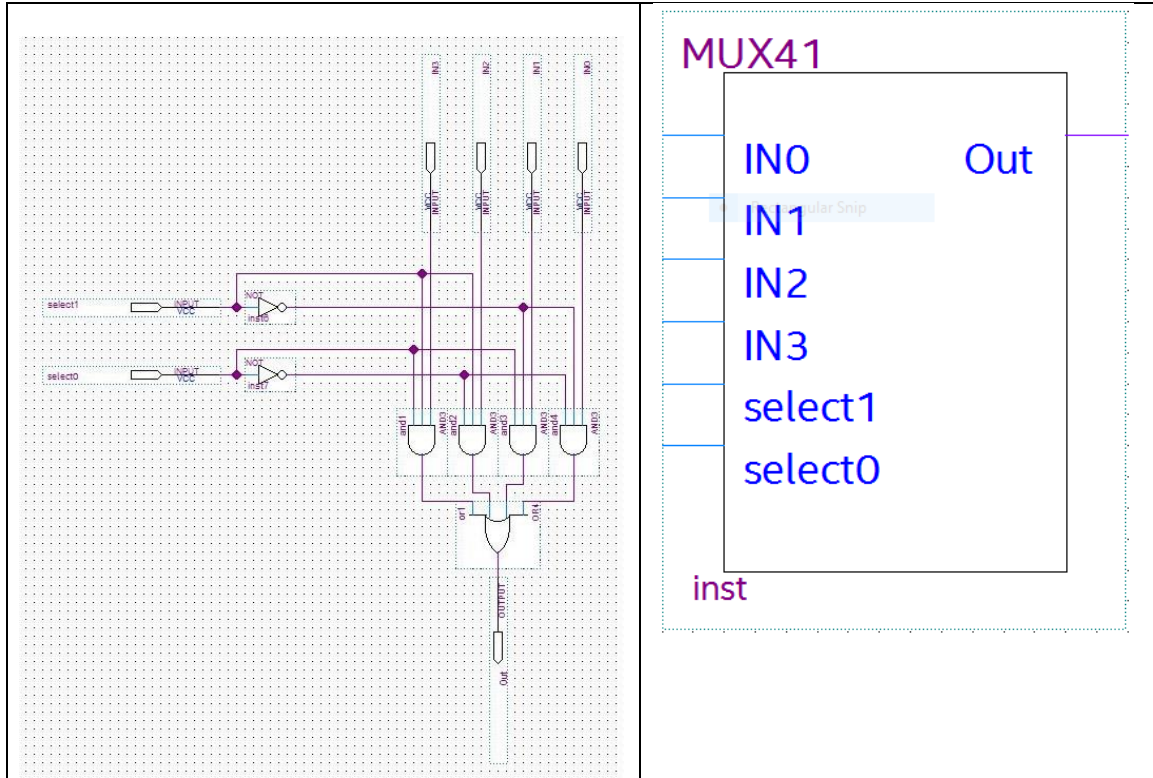
- SW[4..0] : 5 bitlik para girdisi
- SW[6], SW[7] : 2 bitlik ürün seçme girdisi
- LEDR[3..0] : Seçiler ürünü gösteren kırmızı LEDler
- LEDG[3..0] : Seçiler ürünün alınabileceğini gösteren yeşil ledler
- HEX2, HEX3 : Para üstünü gösteren Double-7-Segment-Display
- HEX4, HEX5 : Girilen parayı gösteren Double-7-Segment-Display
- HEX6, HEX7 : Seçilen ürünün fiyatının gösteren Double-7-Segment-Display

Çalışma Prensipleri

SW[5,6] ile 4 adet üründen biri seçilir, Seçilen ürünün fiyatı HEX6,7 7-Segment-Displaylerde gözükür ve o ürüne ait kırmızı LED yanar. SW[4..0] ile 5 bitlik para girişi sağlanır. Herhangi bir para girdisi algılandığı an girilen parayı gösteren HEX4,5 7-Segment-Displayler yanar. Girilen para ürünün fiyatından büyük olduğu sürece para üstünü göstermekten sorumlu HEX2,3 7-Segment-Displayleri yanar.

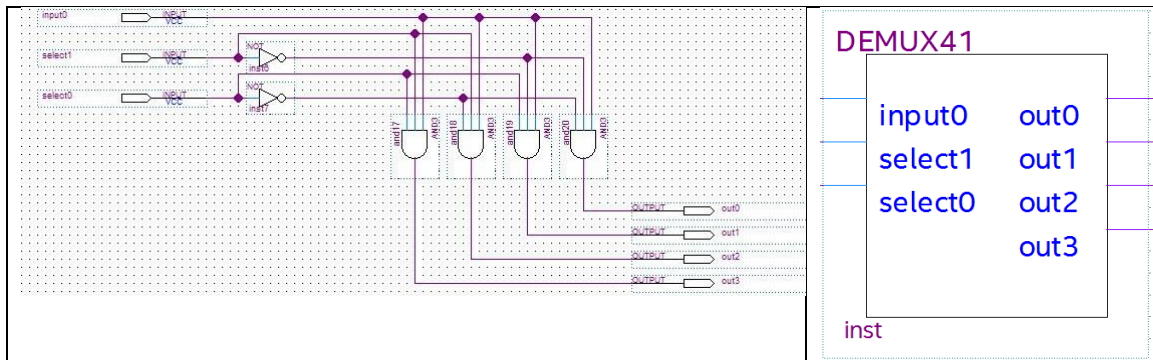
Modüller

MUX414x1 MUX devresidir. Block Diagram ile tasarlandı.



DEMUX41

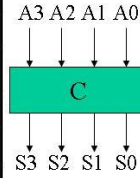
1X4 lük DEMUX devresidir. Block Diagram ile tasarlandı



BIN2BCD

8 bitlik girdiyi 4 bitlik birler, onlar ve yüzler basamağına ayırır. İçinde basamak kaydırıcı “add3” modülü kullanılmakta. “add3” ve “BIN2BCD” modülleri Verilog HDL ile tasarlandı.

Operation	Tens	Units	Binary
HEX			E
Start			1 1 1 0
Shift 1		1	1 1 0
Shift 2		1 1	1 0
Shift 3		1 1 1	0
Add 3		1 0 1 0	0
Shift 4	1	0 1 0 0	
BCD	1	4	



A3	A2	A1	A0	S3	S2	S1	S0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

```

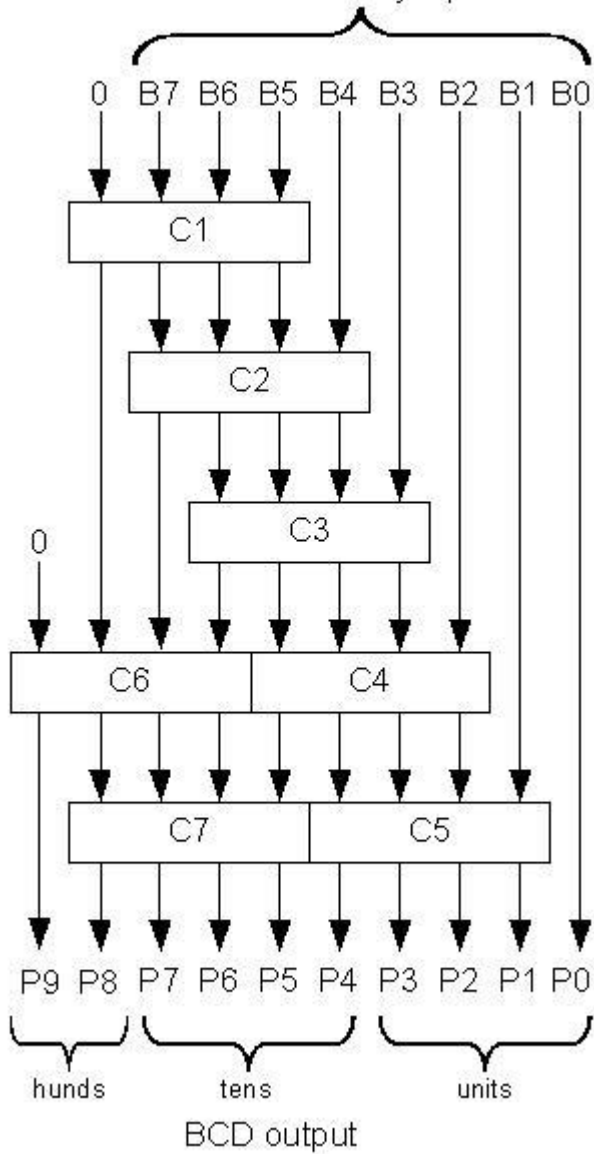
module add3(in,out);

input [3:0] in;
output [3:0] out;
reg [3:0] out;

always @ (in)
    case (in)
        4'b0000: out <= 4'b0000;
        4'b0001: out <= 4'b0001;
        4'b0010: out <= 4'b0010;
        4'b0011: out <= 4'b0011;
        4'b0100: out <= 4'b0100;
        4'b0101: out <= 4'b1000;
        4'b0110: out <= 4'b1001;
        4'b0111: out <= 4'b1010;
        4'b1000: out <= 4'b1011;
        4'b1001: out <= 4'b1100;
        default: out <= 4'b0000;
    endcase
endmodule

```

8-bit binary input



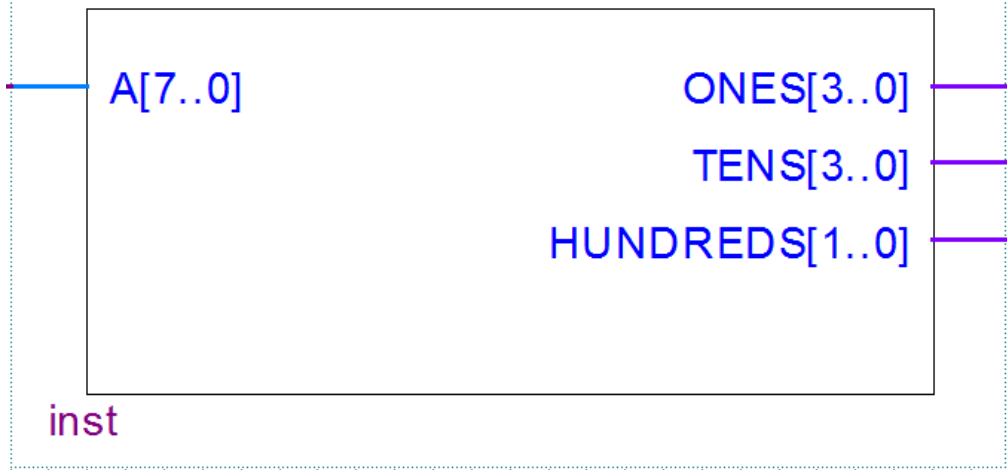
```
module BIN2BCD(A,ONES,TENS,HUNDREDS);
```

```
input [7:0] A;
output [3:0] ONES, TENS;
output [1:0] HUNDREDS;
wire [3:0] c1,c2,c3,c4,c5,c6,c7;
wire [3:0] d1,d2,d3,d4,d5,d6,d7;
```

```
assign d1 = {1'b0,A[7:5]};
assign d2 = {c1[2:0],A[4]};
assign d3 = {c2[2:0],A[3]};
assign d4 = {c3[2:0],A[2]};
assign d5 = {c4[2:0],A[1]};
assign d6 = {1'b0,c1[3],c2[3],c3[3]};
assign d7 = {c6[2:0],c4[3]};
add3 m1(d1,c1);
add3 m2(d2,c2);
add3 m3(d3,c3);
add3 m4(d4,c4);
add3 m5(d5,c5);
add3 m6(d6,c6);
add3 m7(d7,c7);
assign ONES = {c5[2:0],A[0]};
assign TENS = {c7[2:0],c5[3]};
assign HUNDREDS = {c6[3],c7[3]};
```

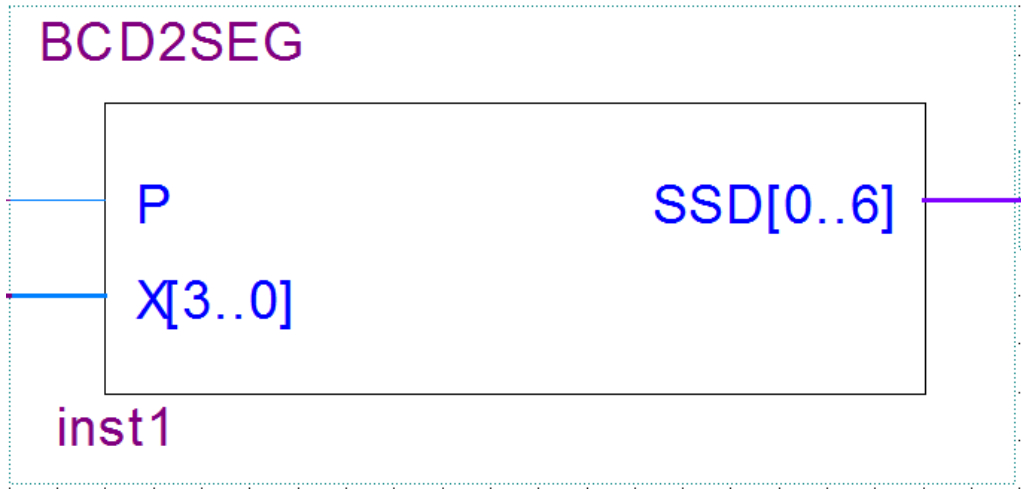
```
endmodule
```

BIN2BCD



BCD2SEG

“BIN2BCD” modülünden çıkan 4 bitlik basamakları 7-Segment-Display ile gösterilmesi için 7 çıkışa dönüştüren decoder. “P” girişi sayesinde modül ihtiyaca göre açılıp kapanabiliyor. Kapalı olduğu sürece module bağlı HEX çalışmıyor. Modül Verilog HDL ile tasarlandı.



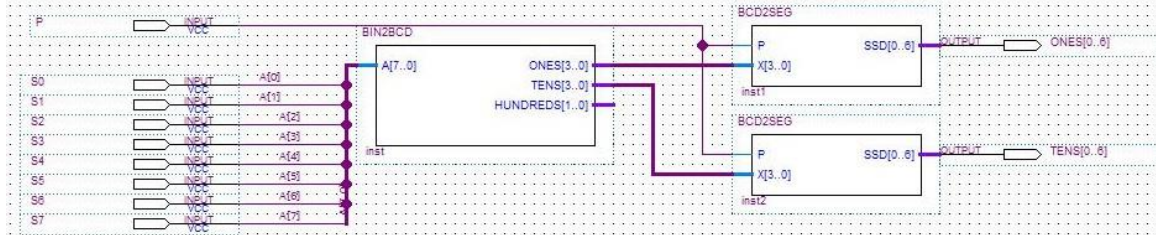
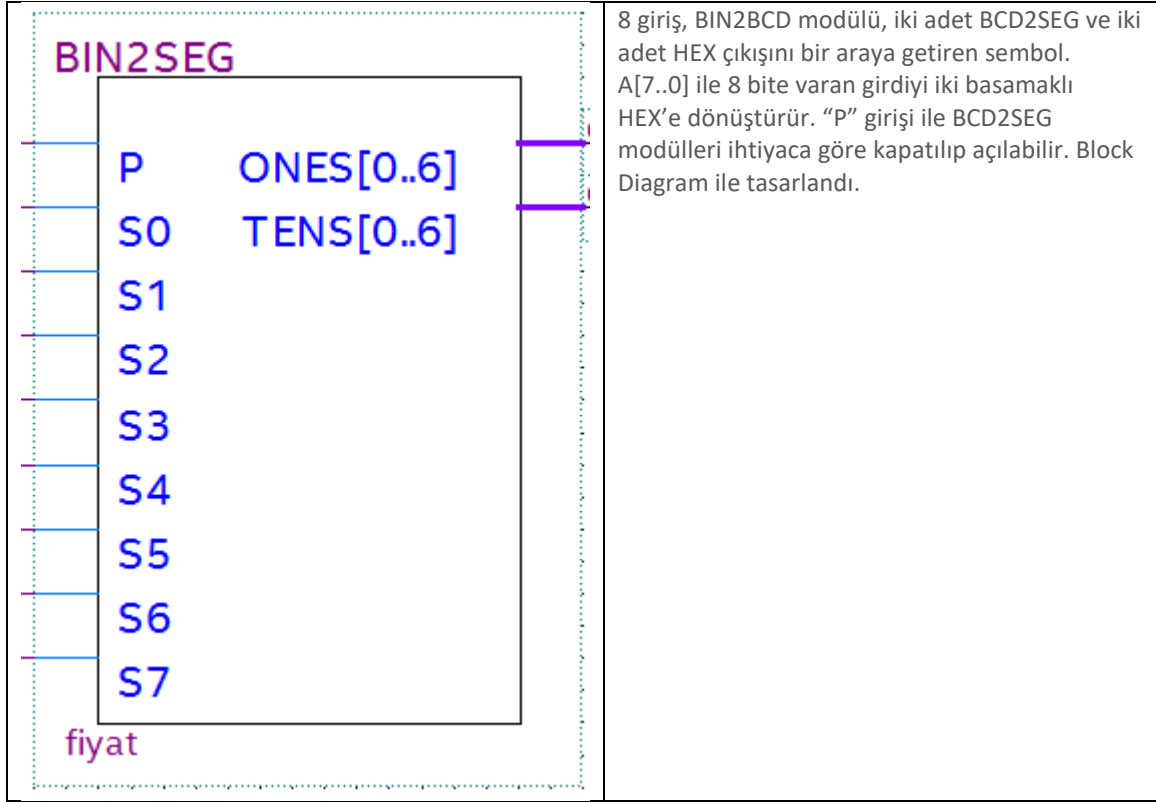
```
module BCD2SEG (P, X, SSD);

    input [3:0] X;
    input P;
    output [0:6] SSD;

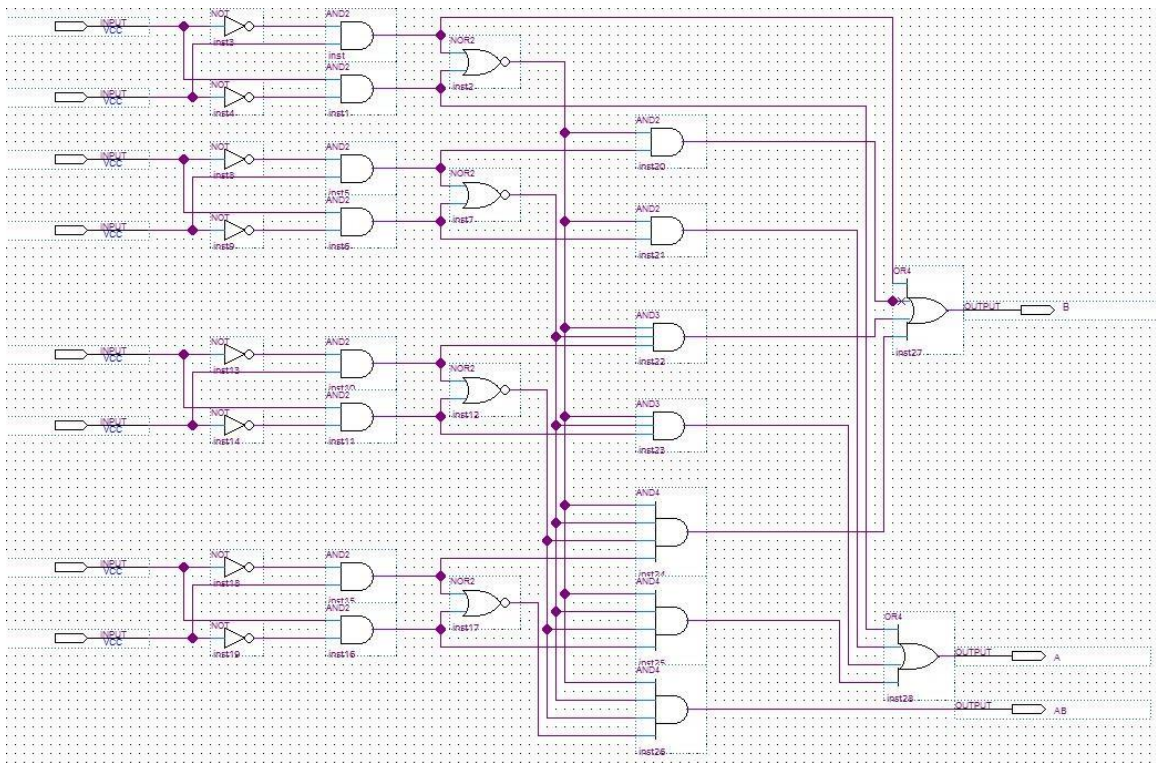
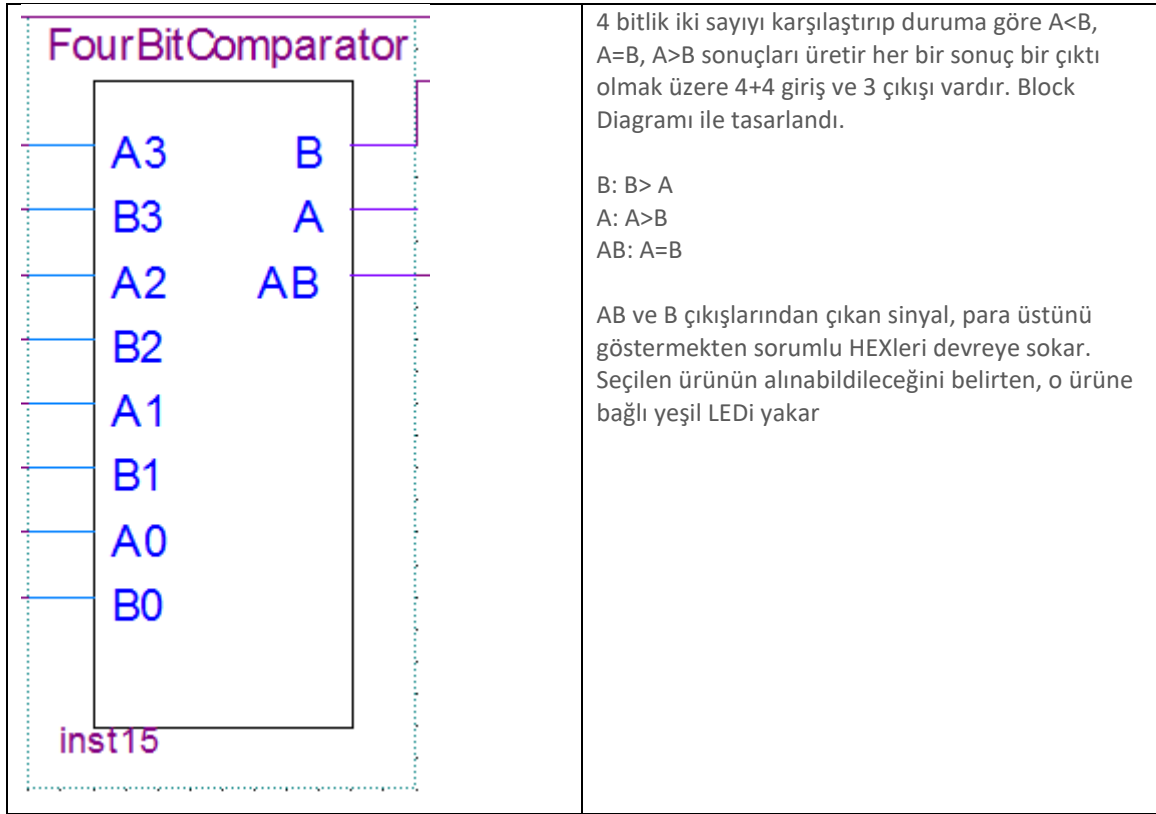
    assign SSD[0] = (((~X[3] & ~X[2] & ~X[1] & X[0]) | (~X[3] & X[2] & ~X[1] & ~X[0])) | ~P;
    assign SSD[1] = (((~X[3] & X[2] & ~X[1] & X[0]) | (~X[3] & X[2] & X[1] & ~X[0])) | ~P;
    assign SSD[2] = ((~X[3] & ~X[2] & X[1] & ~X[0])) | ~P;
    assign SSD[3] = (((~X[3] & ~X[2] & ~X[1] & X[0]) | (~X[3] & X[2] & ~X[1] & ~X[0]) | (~X[3] & X[2] & X[1] & X[0]) | (X[3] & ~X[2] & ~X[1] & X[0])) | ~P;
    assign SSD[4] = (~(X[2] & ~X[0]) | (X[1] & ~X[0])) | ~P;
    assign SSD[5] = (((~X[3] & ~X[2] & ~X[1] & X[0]) | (~X[3] & ~X[2] & X[1] & ~X[0]) | (~X[3] & ~X[2] & X[1] & X[0]) | (~X[3] & X[2] & X[1] & X[0])) | ~P;
    assign SSD[6] = (((~X[3] & ~X[2] & ~X[1] & X[0]) | (~X[3] & ~X[2] & ~X[1] & ~X[0]) | (~X[3] & X[2] & X[1] & X[0])) | ~P;

endmodule
```

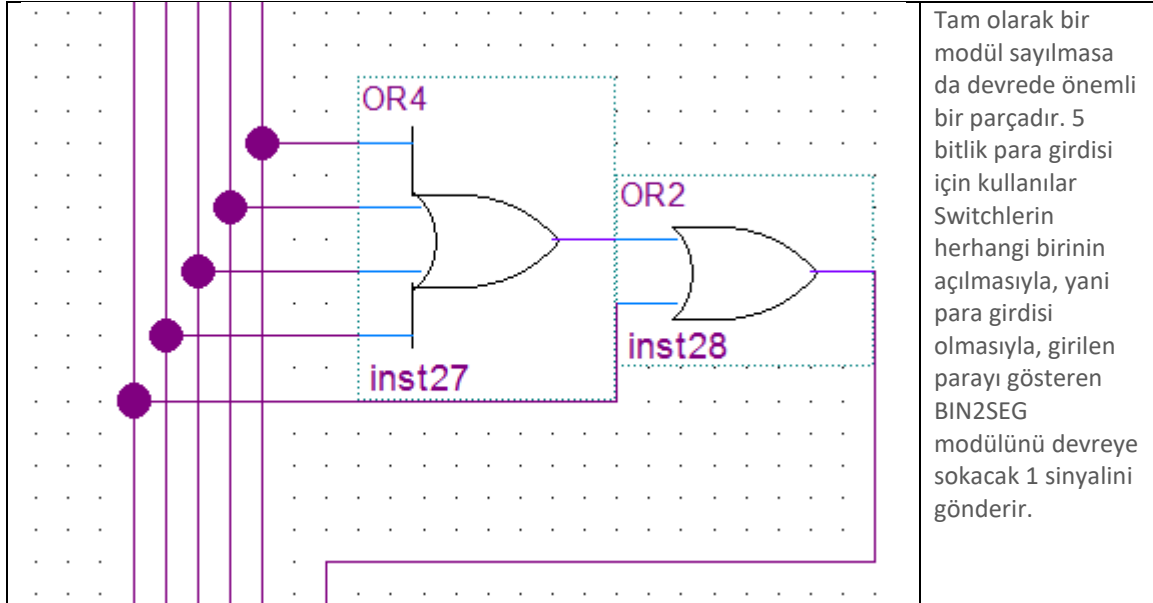

BIN2SEG



FourBitComparator



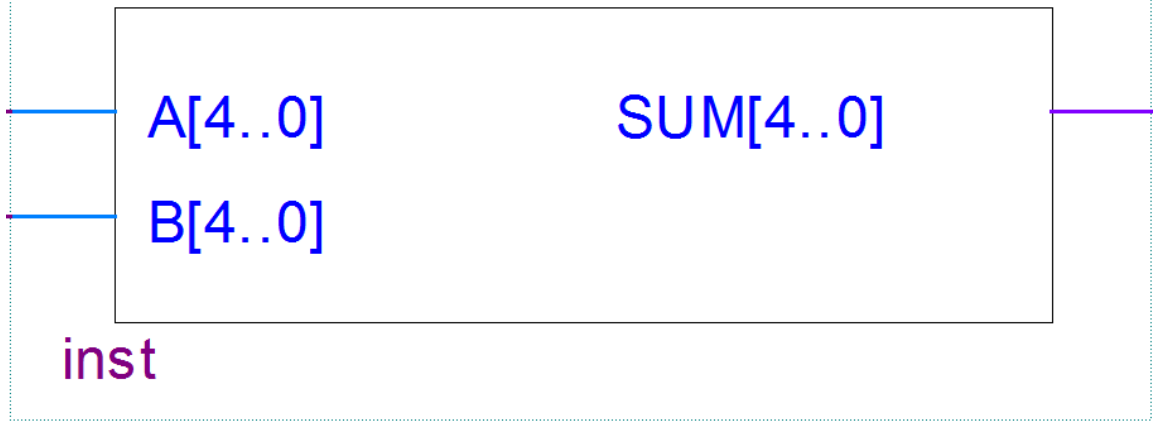
Para Giriş Kontrolcüsü



FullSubtractor

5 bitlik iki sayıyı birbirinden çıkarır ve 5 bitlik çıktı üretir. $A > B$ durumunda $A - B$ için doğru sonuç verir fakat $A < B$ durumunda iken $A - B$ işlemi doğru çıktı vermez ancak module bağlı HEXler sadece $A > B$ durumunda çalışacağı için $A < B$ durumu önemli değildir. Verilog HDL ile tasarlandı	<pre> module FullSubtractor(A, B, SUM); input [4:0] A; input [4:0] B; output [4:0] SUM; assign SUM = A - B; endmodule </pre>
---	--

FullSubtractor



FullSubtractorAdapter

