

**GEBZE**  
**TEKNİK ÜNİVERSİTESİ**

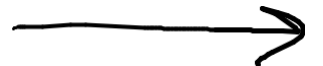


ELM234 - LOJİK DEVRELER  
VE  
TASARIM  
ÖDEV 1

Bayındır  
200102002087



Elektronik Mühendisliği'



# Problem 1. Alıştırımlar [4 + 4 + 4 = 12 puan]

A. Aşağıda 10'luk sistemde verilen sayıların **8-bitlik** ve **11-bitlik** 2's complement şekliyle gösteriniz.

a. 49

b. -104

c. 127

d. -17

B. Aşağıda 2'lik sistemde verilen işlemleri hem **işaretsiz (unsigned)** varsayarak, hem de **2's complement** varsayarak yapınız. Sonuçları ondalık sayı olarak ifade ediniz.

a. 11011 - 11110

b. 01100 + 10111

c. 10110 - 00111

d. 01011 + 01111

C. Aşağıda verilen boolean denklemini sadece NAND kapıları kullanarak yazın.

$$X = \overline{AB} + AC + B(\overline{A} + C)$$

A şıkkı;

$$49_{10} = 2^5 + 2^4 + 2^0 = 00110001 \text{ (Binary)}$$

$$\begin{array}{r} 1100110 \\ + \quad \quad 1 \\ \hline 11001111 \rightarrow \text{Two's complement} \end{array}$$

$$-104_{10} \Rightarrow 104_{10} = 2^6 + 2^5 + 2^3 = 01101000 \text{ (Binary)}$$

$$\begin{array}{r} 10010111 \\ + \quad \quad 1 \\ \hline 10010111 \rightarrow \text{Two's complement} \end{array}$$

$-104_{10} = 10010111 \rightarrow \text{Binary}$   
 $01101000 \rightarrow \text{Complement}$

$$127_{10} = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 01111111$$
$$\begin{array}{r} 01111111 \\ + \quad \quad 1 \\ \hline 10000000 \text{ (Complement)} \end{array}$$

$$-17_{10} \Rightarrow 17_{10} = 2^4 + 2^0 = 00010001 \text{ (Binary)}$$

$$\begin{array}{r} 11101110 \\ + \quad \quad 1 \\ \hline 11101111 \text{ (Complement)} \end{array}$$

$$-17 = 11101111 \text{ (Binary)} \rightarrow 00010000 + 1 = 00010001$$

Formal Sayı ↓	8-Bit Binary	Two's Complement (8-Bit)	11-Bit Binary	Two's Complement (11-Bit)
49	0011 0001	1100 1111	000 0011 0001	111 1100 1111
-104	1001 1000	0110 1000	111 1001 1000	000 0110 1000
127	0111 1111	1000 0001	000 0111 1111	111 1000 0001
-17	1110 1111	0001 0001	111 1110 1111	000 0001 0001

11 Bit için de  
benzer işlemler  
yapıldı.

### B Sıkkları

$$a) \underbrace{11011}_{27_{10}} - \underbrace{11110}_{30_{10}} = -3_{10} \text{ (unsigned)}$$

$$\underbrace{11011}_{-5_{10}} - \underbrace{11110}_{-2_{10}} = -2_{10} \text{ (Two's complement)}$$

$$b) \underbrace{01100}_{12_{10}} + \underbrace{10111}_{23_{10}} = 35_{10} \text{ (unsigned)}$$

$$\underbrace{01100}_{-4_{10}} + \underbrace{10111}_{-9_{10}} = -13_{10} \text{ (Two's complement)}$$

$$c) \frac{10110}{22_{10}} - \frac{00111}{7_{10}} = 15_{10} \text{ (unsigned)}$$

$$\frac{10110}{-10_{10}} - \frac{00111}{-1_{10}} = -9_{10} \text{ (two's complement)}$$

$$d) \frac{01011}{11_{10}} + \frac{01111}{15_{10}} = 26_{10} \text{ (unsigned)}$$

$$\frac{01011}{-5_{10}} + \frac{01111}{-1_{10}} = -6_{10} \text{ (two's complement)}$$

$$X = \overline{A\bar{B}} + A\bar{C} + B(\bar{A} + C)$$

$$(\overline{A\bar{B}})(\overline{A\bar{C}}) + B(\bar{A} + C)$$

$$(\bar{A} + B)(\bar{A} + C) + B(\bar{A} + C)$$

$$\bar{A}\bar{A} + \bar{A}\bar{C} + B\bar{A} + B\bar{C} + B\bar{A} + BC$$

$$\underbrace{B\bar{C} + BC}_{B(\bar{C} + C)}$$

$$\underbrace{1}_{1}$$

$$\bar{A} + \bar{A}\bar{C} + B\bar{A} + B$$

$$\bar{A}(\underbrace{1+B}_1) + \bar{A}\bar{C} + B$$

$$\bar{A} + \bar{A}\bar{C} + B$$

$$\bar{A}(\underbrace{1+\bar{C}}_1) + B$$

$\bar{A} + B \rightarrow$  Sadece NAND istendiği için;

$$\overline{\bar{A} + B}$$

$$\downarrow$$

$$\overline{AB}$$

## Problem 2. Devre tasarımı [2 + 4 + 4 + 10 = 20 puan]

Yeni yapılan GTÜ konferans salonunun 3 adet girişi ve bu girişlerin yanında salondaki tavan lambalarını kontrol eden 3 adet anahtar bulunmaktadır. Her bir anahtarın lambaları kapatıp açabildiği bir lojik devre tasarlayın. (Her bir anahtar, lambalar açık ise kapatacak, kapalı ise açacak)

- Doğruluk tablosunu oluşturun. (Truth table)
- Boolean teoremleri kullanarak sadeleştirin.
- 74xx serisi IC'ler kullanarak devre şemasını çizin. Bütün bağlantıları ( $V_{dd}$ , gnd dahil) net bir şekilde gösterin. Ders kitabının Appendix A3'deki IC'leri referans alın.
- Devrenizi HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin. (Bunun için paylaşılan Modelsim youtube videosunu izlemeniz gerekir.)

A Sıkkı;

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

B Sıkkı;

$$\rightarrow \bar{A}\bar{B}C$$
$$+$$

$$\rightarrow \bar{A}B\bar{C}$$

$$\rightarrow A\bar{B}\bar{C}$$

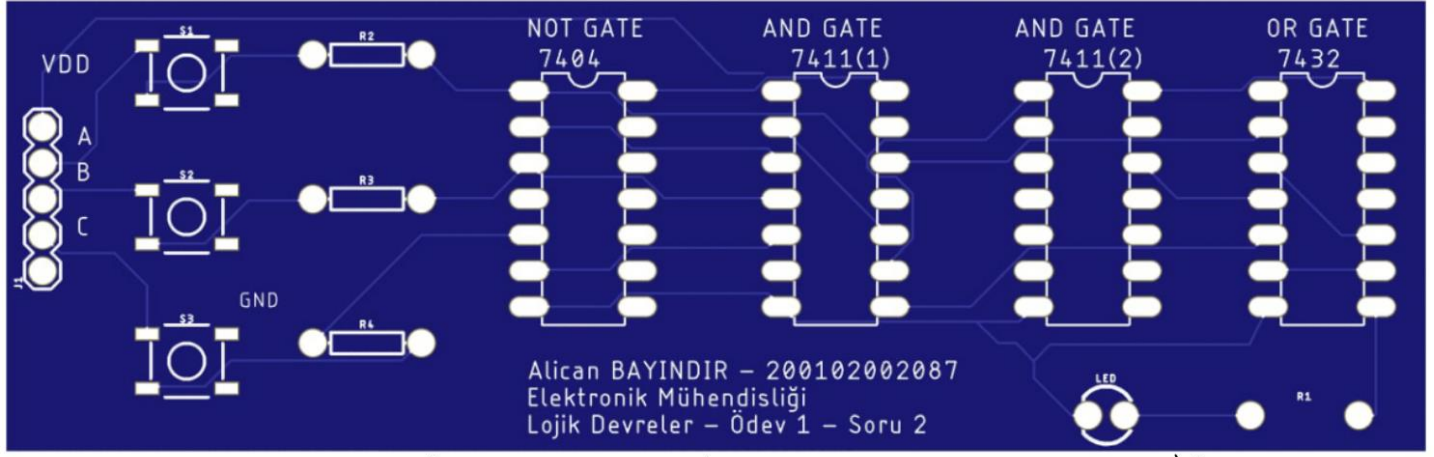
+

$$\rightarrow ABC$$

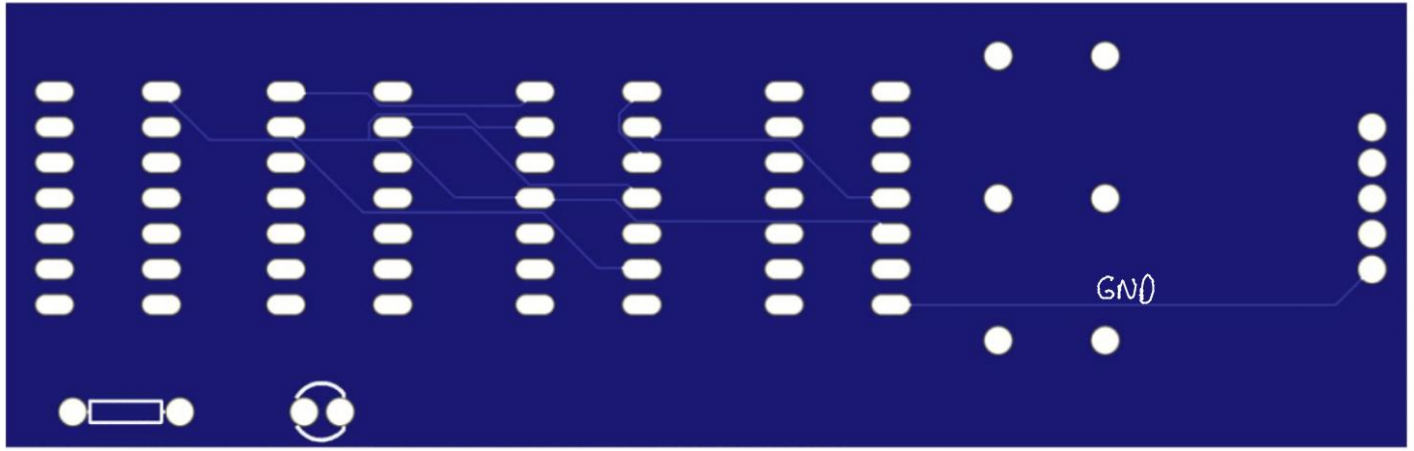
AB \ C	0	1
00	0	1
01	1	0
10	1	0
11	0	1

Doğru olduğundan emin olsam  
da K-Map yapıp kontrol etmek  
istedim.

$$\bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$$



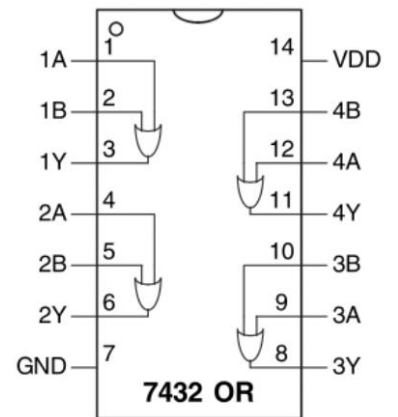
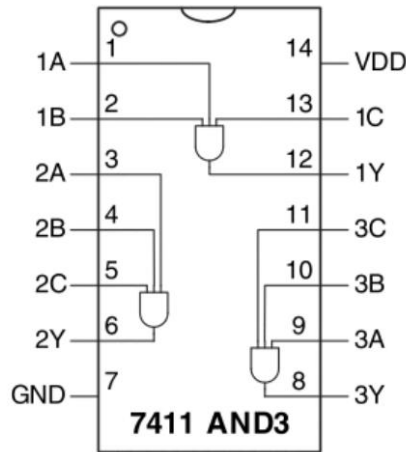
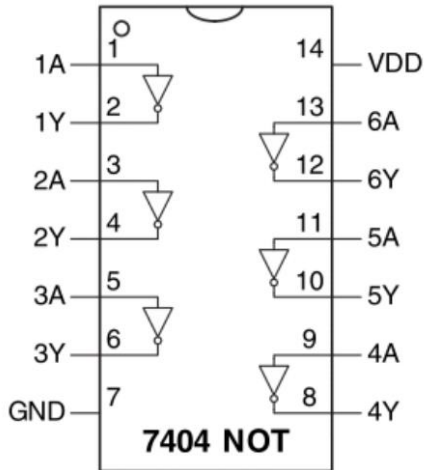
Tasarlanan Kartın Üstten Görünümü

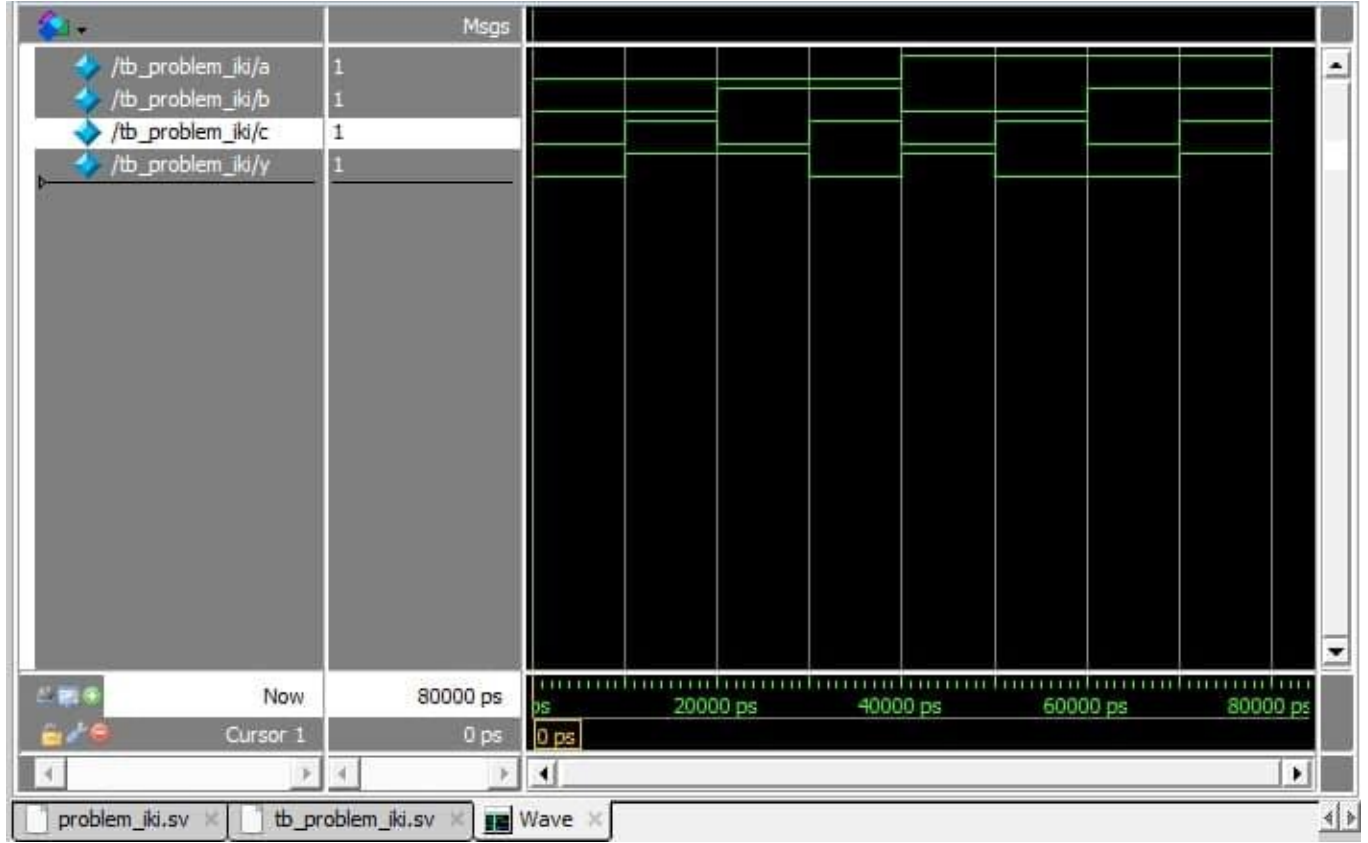


Kartın Alt Katman görüntüsü

Alt katmanda sadece GND ve bazı IC bağlantıları yapılmıştır.

Kullanılan 74xx serisi IC'ler





Şekil 1 Soru 2 Modelsim simülasyon sonucu.



```
module problem_iki (  
    input  logic a, b, c,  
    output logic y  
);  
  
assign y = (~a & ~b & c) || (~a & ~c & b) || (~b & ~c & a) || (a & b & c);  
  
endmodule
```

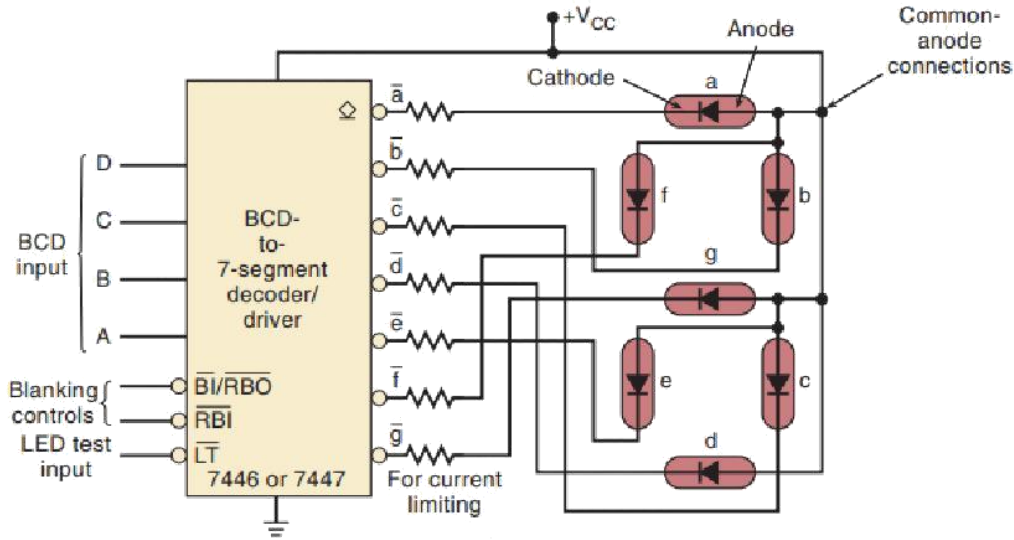
```
/* tb_problem_iki  
 *  
 * soru iki devre tasarimi sorusu  
 */  
  
`timescale 1ns/1ps  
  
module tb_problem_iki ();  
  
    logic a, b, c;  
    logic y;  
  
    problem_iki dut0(a, b, c, y);  
  
    initial begin  
        a = 0; b = 0; c = 0; #10;  
        c = 1; #10;  
        b = 1; c = 0; #10;  
        c = 1; #10;  
        a = 1; b = 0; c = 0; #10;  
        c = 1; #10;  
        b = 1; c = 0; #10;  
        c = 1; #10;  
  
        $stop;  
    end  
endmodule
```



### Problem 3. Decoder tasarımı [4 + 4 + 4 + 18 = 30 puan]

Elimizdeki bir 7-Segment Şekil 1 de verildiği gibi BCD-to-7SD decoder devresine bağlanmıştır. Sizden 7-Segment de sadece 3-8 arası sayıları arasını gösterecek şekilde BCD girişlerini {3,4,5,6,7,8} rakamları ile eşleştiren bir decoder devresi tasarlamanız istenmektedir. Örnek olarak:

- BCD girişlerinden 4 sayısı verildiği zaman (DCBA → 0100), 7-Segment 4 sayısını gösterecektir (bcfg segmentleri).
- BCD girişlerinden 8 sayısı verildiği zaman (DCBA → 1000), 7-Segment 8 sayısını gösterecektir (abcdefg segmentleri).
- BCD girişlerinden 0-2 arası sayıları veya 9 ve üzeri sayılar verildiği zaman, 7-Segmentin ne gösterdiğinin önemi yok (don't care)
- Decoder çıkışlarının **active-low** olduğunu dikkate almayı unutmayın.



Şekil 1

- Her 6 farklı durum için (3,4,5,6,7,8 çıkışları) doğruluk tablosu oluşturun.
- İstediğiniz herhangi iki durumu **boolean teoremleri kullanarak** en sade haline getirin.
- Geri kalan durumları **K-map kullanarak** sadeleştirin (don't care lardan yararlanmayı unutmayın)
- Devrenizi HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de simüle edin.

3.-)

	DCBA	a	b	c	d	e	f	g
0	0000	X	X	X	X	X	X	X
1	0001	X	X	X	X	X	X	X
2	0010	X	X	X	X	X	X	X
3	0011	0	0	0	0	1	1	0
4	0100	1	0	0	1	1	0	0
5	0101	0	1	0	0	1	0	0
6	0110	0	1	0	0	0	0	0
7	0111	0	0	0	1	1	1	1
8	1000	0	0	0	0	0	0	0
9	1001	X	X	X	X	X	X	X
A	1010	X	X	X	X	X	X	X
B	1011	X	X	X	X	X	X	X
C	1100	X	X	X	X	X	X	X
D	1101	X	X	X	X	X	X	X
E	1110	X	X	X	X	X	X	X
F	1111	X	X	X	X	X	X	X

3.B) Durum olarak a ve b'yi seçtim.

$$a \rightarrow \bar{D}\bar{C}\bar{B}\bar{A}$$

$$b \rightarrow \bar{D}\bar{C}\bar{B}A + \bar{D}\bar{C}B\bar{A}$$

3c) C için;

BA \ DC	00	01	11	10
00	X	0	X	0
01	X	0	X	X
11	0	0	X	X
10	X	0	X	X

d için;

BA \ DC	00	01	11	10
00	X	1	X	0
01	X	0	X	X
11	0	1	X	X
10	X	0	X	X

$$\bar{D}C\bar{B}\bar{A} + \bar{D}CBA$$

e için;

BA \ DC	00	01	11	10
00	X	1	X	0
01	X	1	X	X
11	1	1	X	X
10	X	0	X	X

$$\bar{D}\bar{B} + \bar{D}A$$

f için;

BA \ DC	0	01	11	10
00	X	0	X	0
01	X	0	X	X
11	1	1	X	X
10	X	0	X	X

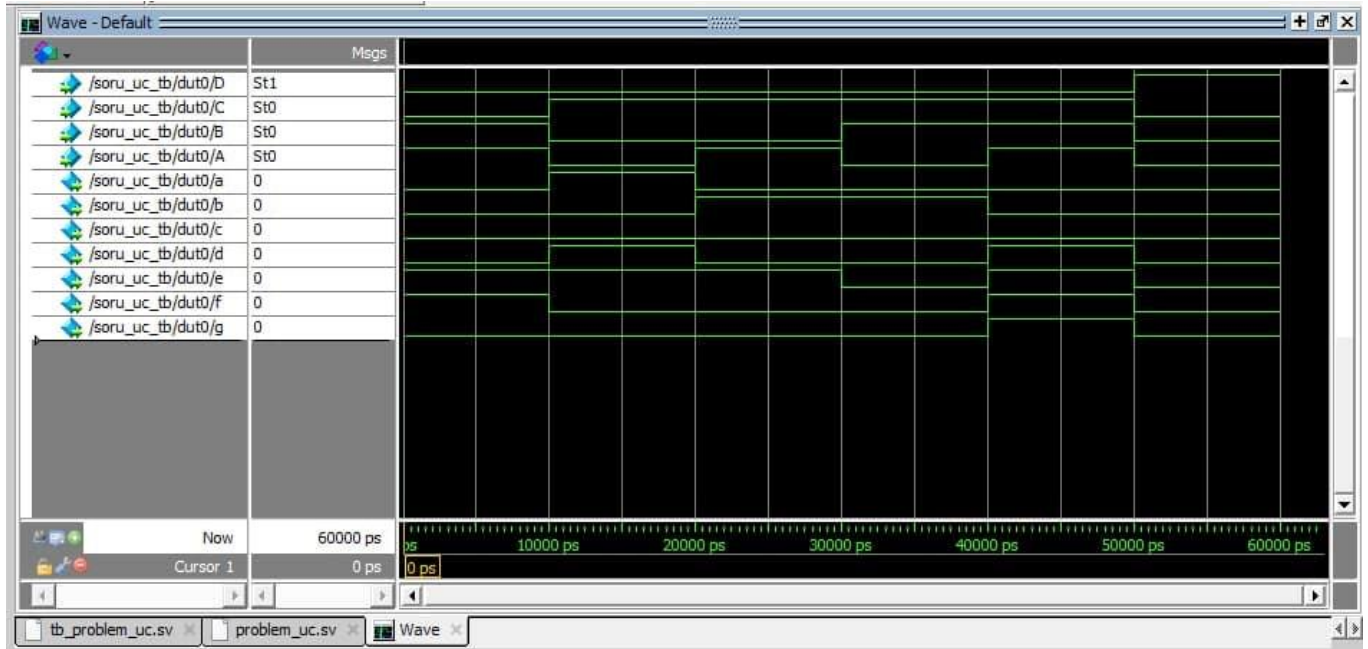
$$BA$$

g için;

BA / DC

	00	01	11	10
00	X	0	X	0
01	X	0	X	X
11	0	1	X	X
10	X	0	X	X

$\overline{DCBA}$



Şekil 2 Soru 3 Decoder tasarım sorusu Modelsim simülasyon sonucu

```
//odev_bir_soru_uc
```

```
module soru_uc(  
    input logic D,C,B,A,  
    output logic a,b,c,d,e,f,g  
);  
assign a=~D & C & ~B & ~A;  
assign b=(~D & C & ~B & A)|(~D & C & B & ~A);  
assign c=0;  
assign d=(~D & C & ~B & ~A)|(~D & C & B & A);  
assign e=(~D & ~B)| A;  
assign f=B & A;  
assign g=~D & C & B & A;  
endmodule
```

```
//odev_bir_soru_uc_tb
```

```
`timescale 1ns/1ps  
module soru_uc_tb();  
    logic D,C,B,A;  
    logic a,b,c,d,e,f,g;  
    soru_uc dut0(D,C,B,A,a,b,c,d,e,f,g);  
    initial begin  
        D=0; C=0; B=1; A=1; #10;  
        C=1; B=0; A=0; #10;  
        A=1; #10;  
        B=1; A=0; #10;  
        A=1; #10;  
        D=1; C=0; B=0; A=0; #10;  
        $stop;  
    end  
endmodule
```

Dijital Sistemlerde hata tespiti için kullanılan Parity biti hakkında bilgi veriniz.

A. İki türünü açıklayınız

B. ASCII ile kodlanan **COVID-19** kelimesini msb'ye parity biti eklenmiş şekilde ve 8-bitlik gruplar halinde iki tür parity için de ayrı ayrı gösteriniz.

## Parity Bits (Eşlik Bitleri)

Eşlik bitleri gönderilen bir mesajın doğru olup olmadığını kontrol etmek için kullanılan en basit kontrol yöntemlerinden biridir. Gönderilen mesajdaki "1"lerin tek veya çift olma durumuna göre işlem yapılır.

### Even parity (Çiftlik)

Gönderilen mesajdaki 1'lerin toplamının çift olması durumu

### Odd parity (Teklik)

Gönderilen mesajdaki 1'lerin toplamının tek olması durumu

Mesaj gönderilirken mesaj üzerinde bazı yanlışlıklar olabilir. Bunu önlemek için en sağa 1 bit ekleyip mesajı korunmaya almış oluruz,

## B Şekli

C	O	V	I	D
01000011	01001111	01010110	01001001	01000100

-	1	9
0010101	00110001	00111001



Even Parity

C	O	V	I	D
010000111	010011111	010101100	010010011	010001000

-	1	9
00101011	001100011	001110010

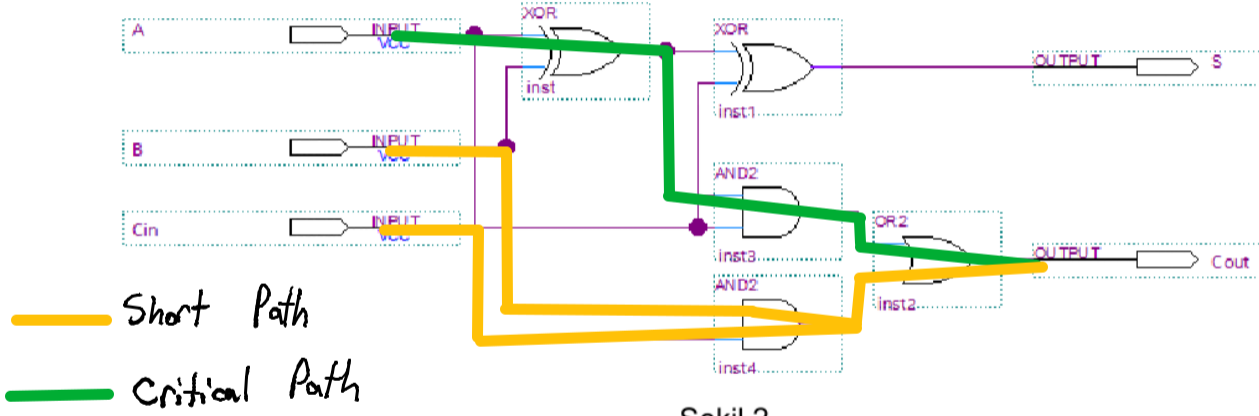
Odd Parity

C	O	V	I	D
010000110	010011110	010101101	010010010	010001001

-	1	9
00101010	001100010	001110011

## Problem 5. Gecikme [8 puan]

Şekil 2 de verilen devrenin **critical ve short path** lerini Tablo 1 deki gecikme değerlerini kullanarak hesaplayın.



### Short Path

Short path hesaplamasında  $t_{cd}$  kullanılır,

$$30 + 25 = 55$$

### Critical Path

Critical path hesaplamasında  $t_{pd}$  kullanılır,

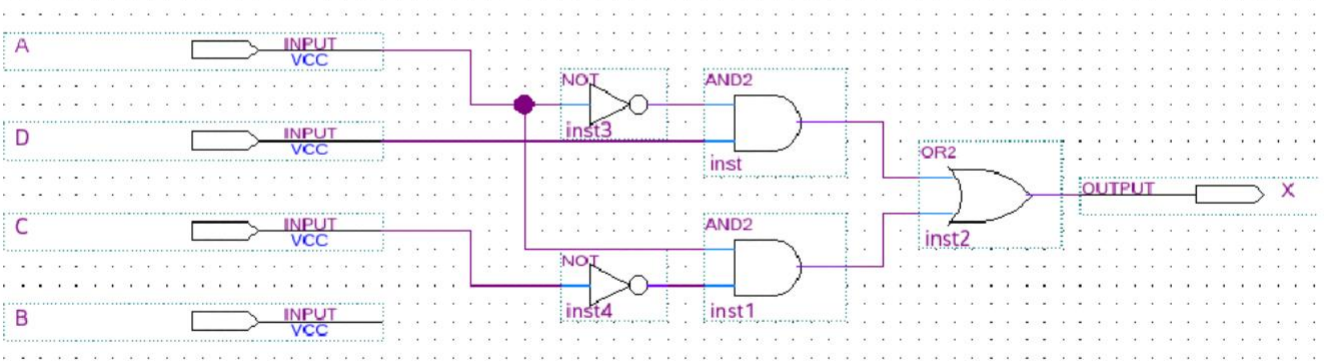
$$80 + 55 + 65 = 200$$

Kapı	$t_{pd}$ (ps)	$t_{cd}$ (ps)
NOT	25	20
2-giriş AND	55	30
2-giriş OR	65	25
2-giriş XOR	80	65

Tablo 1

## Problem 6. Glitch [8 + 12 = 20 puan]

- A. Şekil 3 de en sade şekli verilen 4 girişli devrenin hangi koşulda glitch oluşabileceği veya hiçbir şekilde oluşmayacağı hakkında yorum yapın. Varsa bu glitchi önleyebilecek bir devre ekleyin. Tablo 1 deki değerleri baz alın.
- B. Devrenizi HDL kullanarak tasarlayın, bir testbench devresi oluşturun ve Modelsim de verilen gecikme değerlerini ekleyerek simüle edin.



Şekil 3

Bu devre  $\bar{A}D + A\bar{C}$  olarak ifade edilebilir.

Kapı	$t_{pd}$ (ps)	$t_{cd}$ (ps)
NOT	25	20
2-giriş AND	55	30
2-giriş OR	65	25
2-giriş XOR	80	65

Tablo 1

AD \ C	00	01	11	10
0	0	1	1	1
1	0	1	0	0

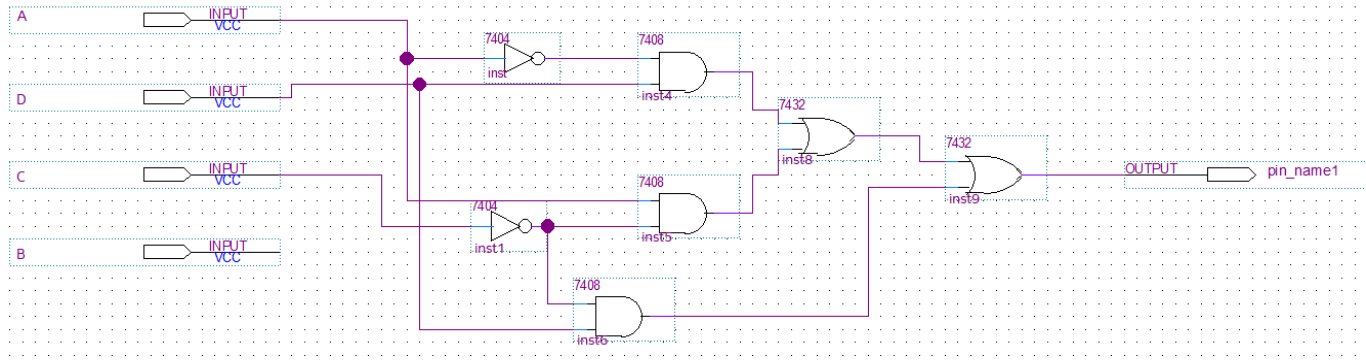
Yandaki tabloda  
Mor ve yeşil bubble  
lar devreye işlenmiş,  
Turuncu bubble'ı,

devreye eklersek devredeki Glitch çözülmüş olacaktır.  
Çakışan bubble'lardan dolayı burada glitch düşüyor,

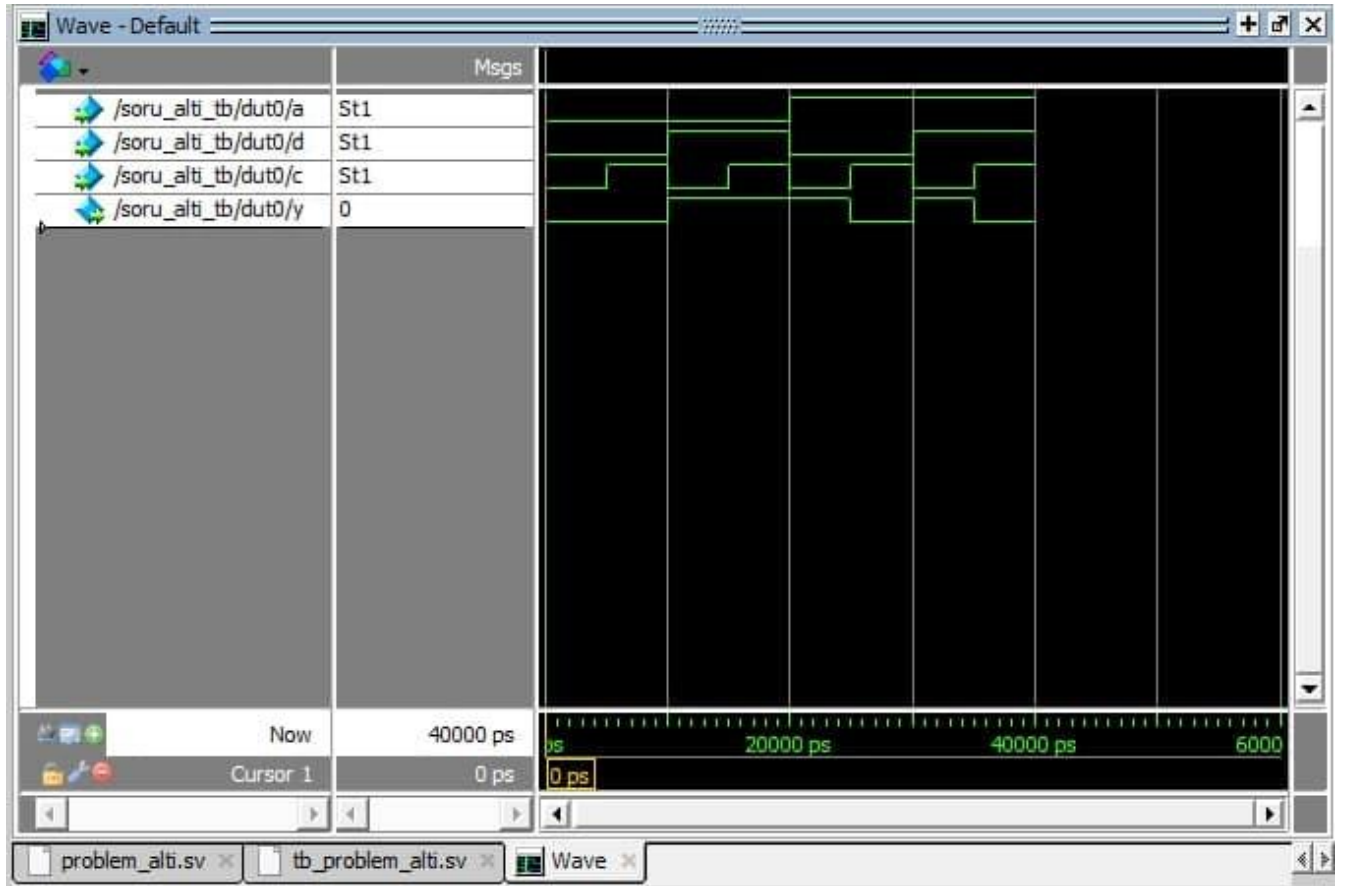
Glitch'i önlemek için devre denkleminin;

$$\bar{A}D + A\bar{C} + D\bar{C}$$

şeklinde olmalıdır. Yeni devre şöyle görünecektir.



Şekil 3 Glitch önleyecek devre.



Şekil 4 Soru 6 Modelsim simülasyon sonucu.

```
/*  
    problem_alti  
*/  
  
module soru_alti (  
    input  logic a, d, c,  
    output logic y  
);  
assign y = (~a & d) |(a & ~c)| (d & ~c);  
endmodule
```

```
/*  
    odev_bir_soru_alti  
*/  
  
`timescale 1ns/1ps  
module soru_alti_tb ();  
    logic a, d, c;  
    logic y;  
    soru_alti dut0(a, d, c, y);  
    initial begin  
        a = 0; d = 0; c = 0;      #10;  
        c = 1;                    #10;  
        d = 1; c = 0;             #10;  
        c = 1;                    #10;  
        a = 1; d = 0; c = 0;      #10;  
        c = 1;                    #10;  
        d = 1; c = 0;             #10;  
        c = 1;                    #10;  
        $stop;  
    end  
endmodule
```