

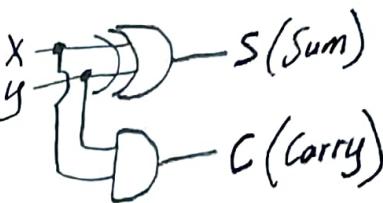
# Bilgisayar Mimarisi

## Half Adder (Yarım Toplayıcı)

x	y	C	S
00	00	0	$C = xy$
01	01	0	$S = x'y + xy' = x \oplus y$
10	01	0	
11	10	1	

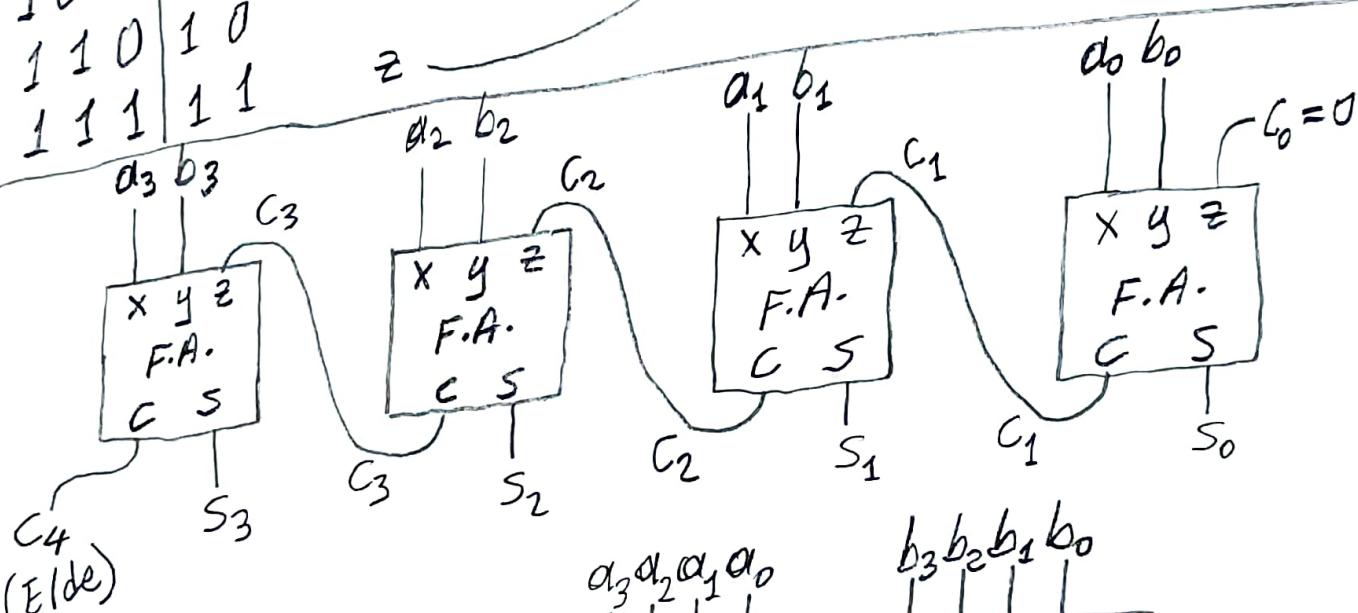
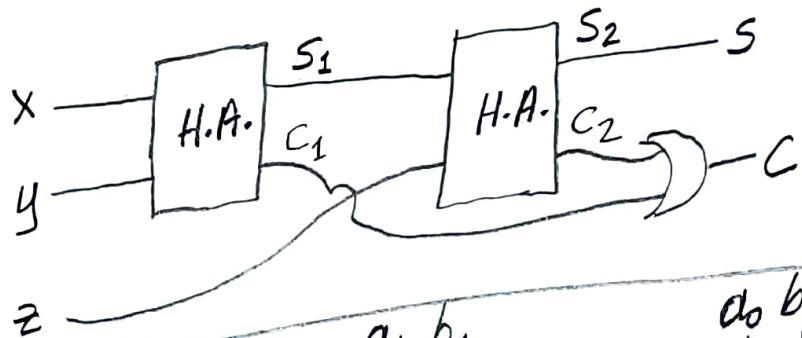
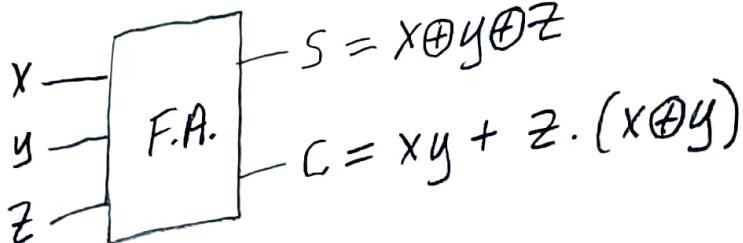
$$S = x'y + xy' = x \oplus y$$

$x+y \rightarrow$  Toplama İşlemi



## Full Adder (Tüm Toplayıcı)

x	y	z	C	S
000	00		0	0
001	01		0	1
010	01		0	1
011	10		0	0
100	01		0	1
101	10		0	0
110	10		0	0
111	11		1	1

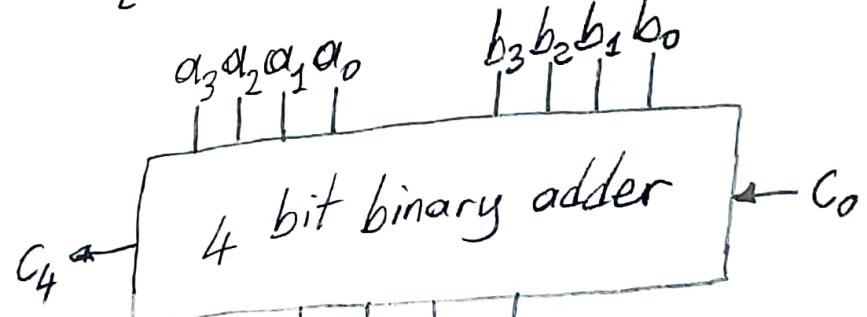


$$A = a_3 a_2 a_1 a_0$$

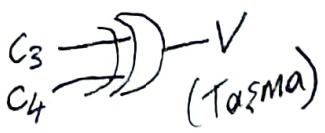
$$B = b_3 b_2 b_1 b_0$$

$$+ \quad \quad \quad$$

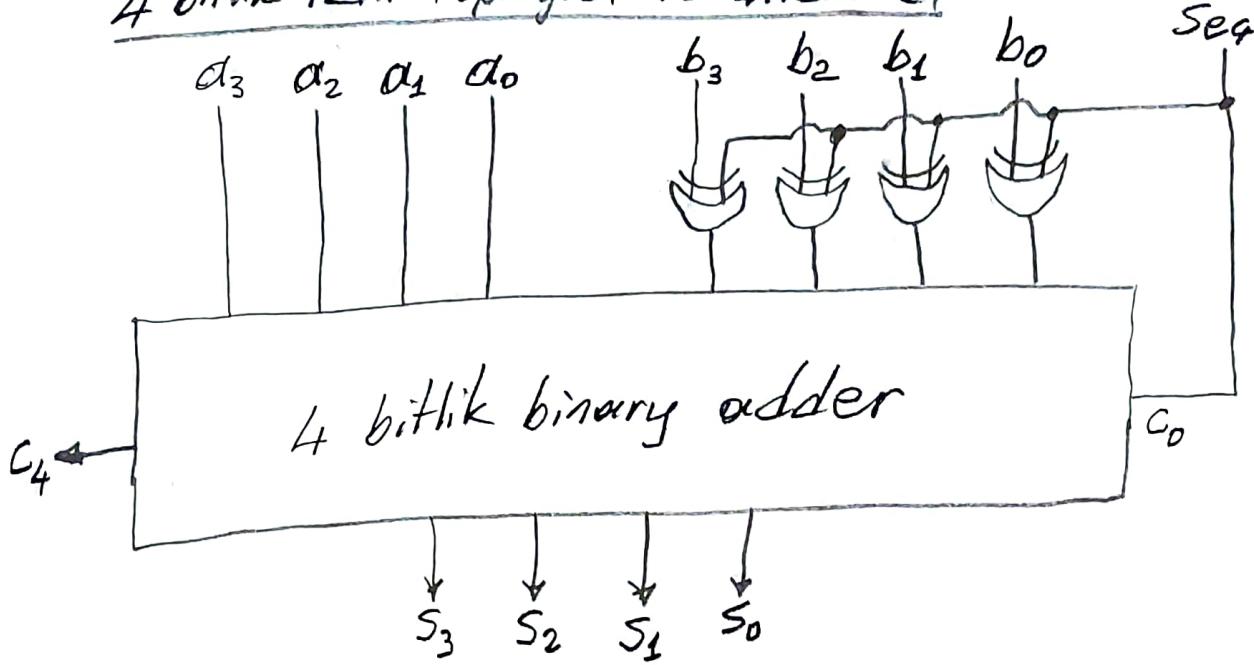
$$C_4 S_3 S_2 S_1 S_0$$



3 tanesi 4 bitlik ikili toplayıcı  
peş peşe bağlanırsa 12 bitlik  
ikili toplayıcı elde edilir.



## 4 bitlik ikili toplayıcı ve çıkartıcı



$Seg = 0$  ise

Exor kapıları  
 $B'$ yi annen geçirir.  
 $C_0 = 0$  olur.

$$S = A + B$$

Toplaman işlemi  
 $C_4$  elde biti

$Seg = 1$  ise

Exor kapıları  $B'$ 'nin birli tümleyenini  
geçirir.  $C_0 = 1$  olur.

$$S = A + \bar{B} + 1 = A - B$$

Çıkarma işlemleri  
 $C_4$  bors biti

$S_3 S_2 S_1 S_0$

$$\begin{array}{r} 0 \\ 0 \\ 0 \\ 0 \end{array} \rightarrow 0$$

$$\begin{array}{r} 0 \\ 1 \\ 1 \\ 1 \end{array} \rightarrow 7$$

$$\begin{array}{r} 1 \\ 0 \\ 0 \\ 0 \end{array} \rightarrow -8$$

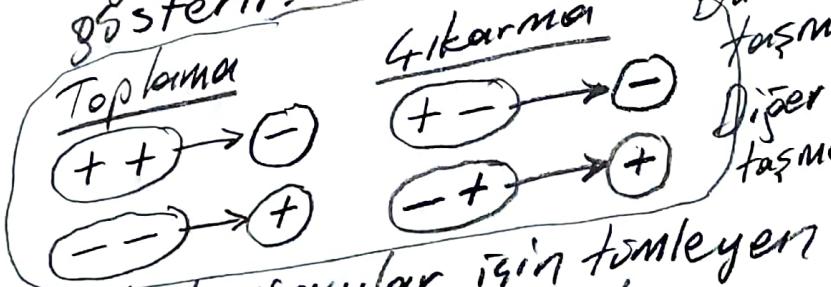
$$\begin{array}{r} 1 \\ 1 \\ 0 \\ 1 \end{array} \rightarrow -3$$

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \end{array} \rightarrow -1$$

İşaret biti

$C_3 \rightarrow \oplus$   $C_4 \rightarrow \oplus$   $\rightarrow V$  (Taşma)

Taşma: işaretli tam sayıların toplanmasından veya çıkartılmasından oluşur. Sonucun ayrılan bit ile ifade edilemeyeceğini gösterir.



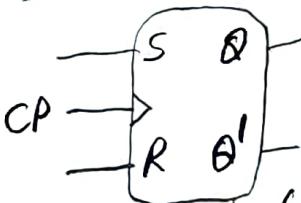
Bu durumlarda taşma olusur.  
Diğer durumlarda taşma olusmaz

Bilgisayarda işaretli tam sayılar için tümleyen aritmetiği (ikili tümleyen) kullanılır.

$$\begin{array}{r} 1101 \\ (-3) \end{array} \xrightarrow{\text{2'lük t.}} \begin{array}{r} 0011 \\ (+3) \end{array} \xrightarrow{\text{2'lük t.}} \begin{array}{r} 1101 \\ (-3) \end{array}$$

# Flip-Flop'lar

## RS flip-flop



$$\begin{aligned} Q_{t+1} &= S + R' Q_t \\ SR &= 0 \text{ olmalı} \\ \text{Karakteristik} \\ \text{Denklem} \end{aligned}$$

Grafik Sembol

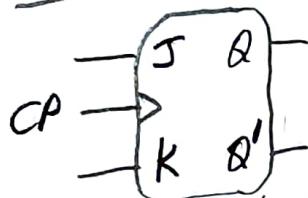
SR	$Q_{t+1}$
00	$Q_t$
01	0
10	1
11	Kullanılmaz

Karakteristik Tablo

$Q_t Q_{t+1}$	SR
00	0X
01	10
10	01
11	X0

Uyarım Tablosu

## JK flip-flop



$$\begin{aligned} Q_{t+1} &= J Q_t' + K' Q_t \\ \text{Karakteristik} \\ \text{Denklem} \end{aligned}$$

Grafik Sembol

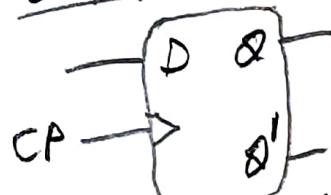
JK	$Q_{t+1}$
00	$Q_t$
01	0
10	1
11	$Q_t'$

Karakteristik Tablo

$Q_t Q_{t+1}$	JK
00	0X
01	1X
10	X1
11	X0

Uyarım Tablosu

## D flip-flop



$$\begin{aligned} Q_{t+1} &= D \\ \text{Karakteristik} \\ \text{Denklem} \end{aligned}$$

Grafik Sembol

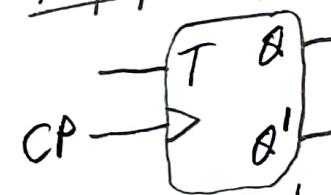
D	$Q_{t+1}$
0	0
1	1

Karakteristik Tablo

$Q_t Q_{t+1}$	D
00	0
01	1
10	0
11	1

Uyarım Tablosu

## T flip-flop



$$\begin{aligned} Q_{t+1} &= T Q_t' + T' Q_t \\ \text{Karakteristik} \\ \text{Denklem} \end{aligned}$$

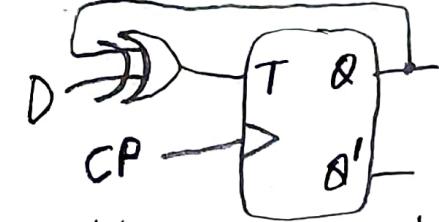
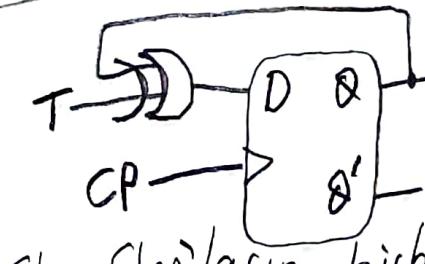
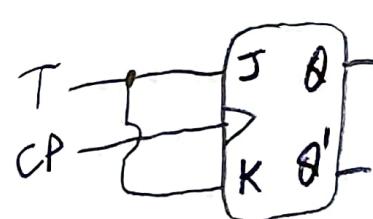
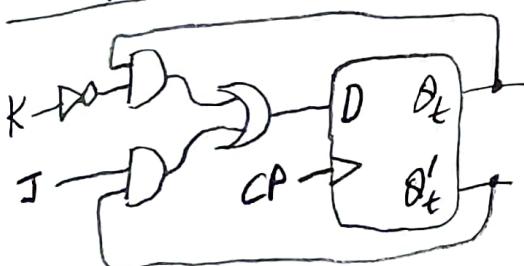
Grafik Sembol

T	$Q_{t+1}$
0	$Q_t$
1	$Q_t'$

Karakteristik Tablo

$Q_t Q_{t+1}$	T
00	0
01	1
10	1
11	0

Uyarım Tablosu



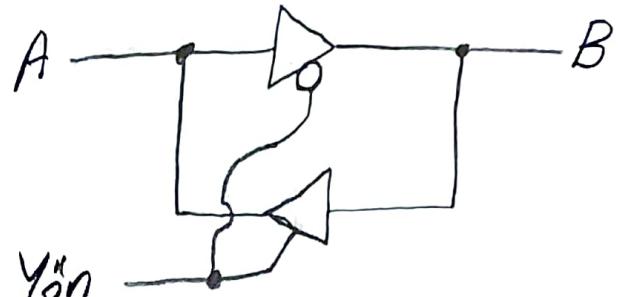
Flip-Flop'ların birbirine dönüştürülmesi

## Üç Durumlu Eleman

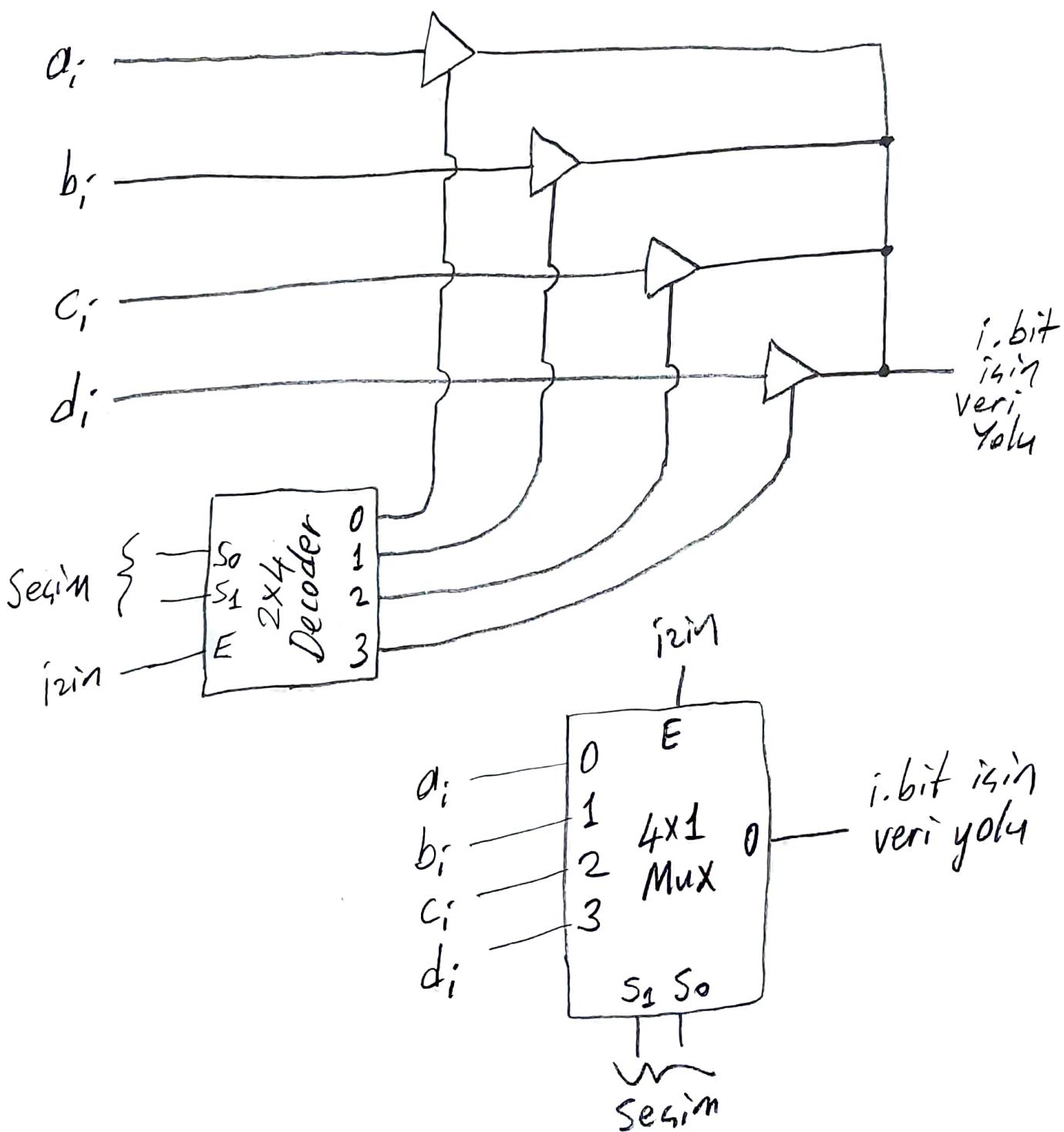
(4)



Kontrol	Gerekli
0	Yüksek Impedans
1	Giriş



$Yon = 0$  ise  $B = A$  Sağda  
 $Yon = 1$  ise  $A = B$  Sola

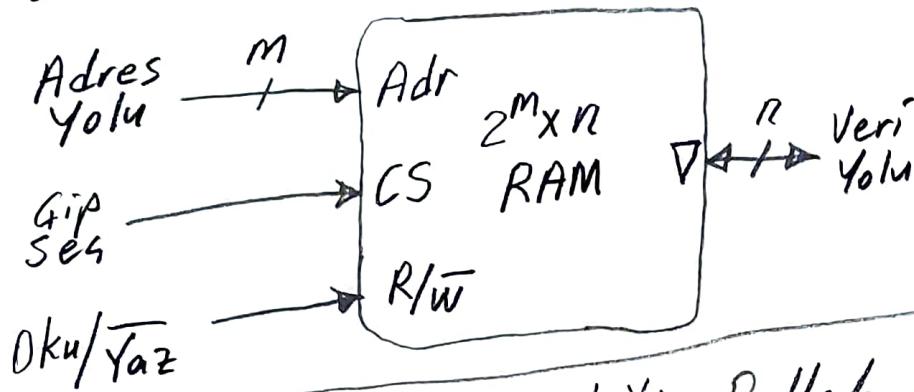


## Rasgele Erişimli Bellek (RAM)

RAM - Random Access Memory

SRAM - Statik RAM

DRAM - Dinamik RAM



CS	R/W	İşlem
0	X	Ysk. Emp.
1	0	Yaz
1	1	Oku

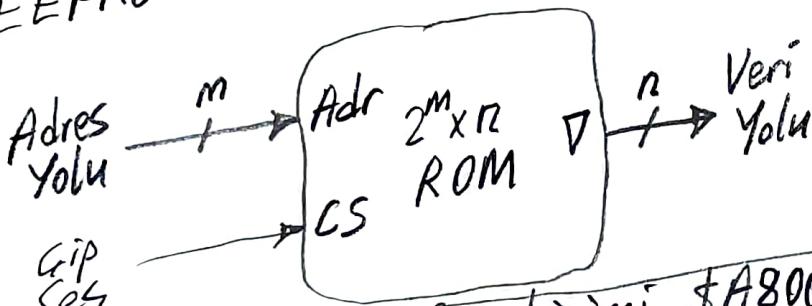
## Sadece Okunabilir Bellek (ROM)

ROM - Read Only Memory

PROM - Programlanabilir ROM

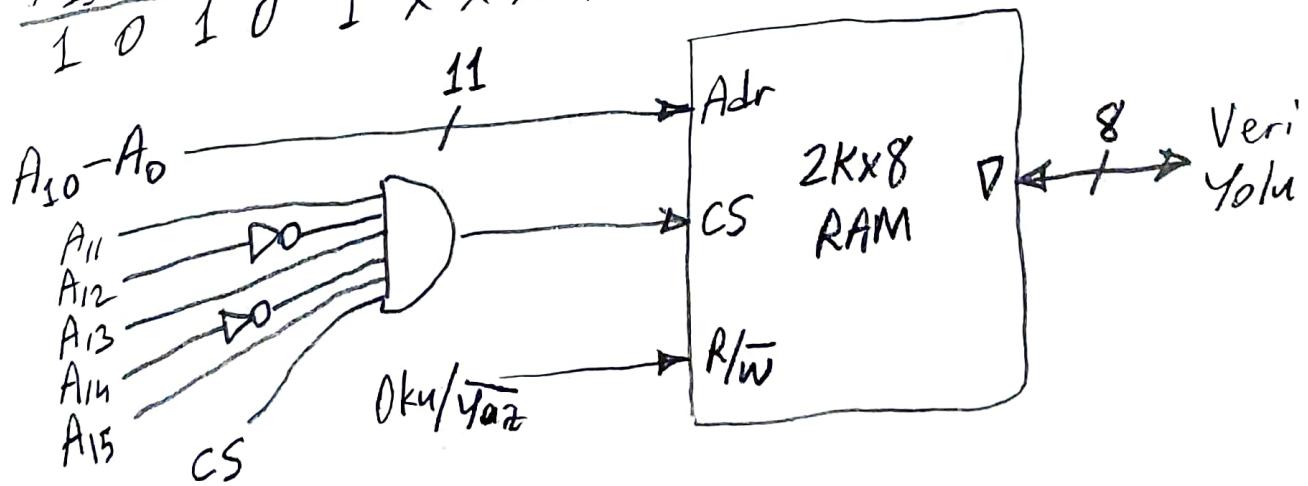
EPROM - Silinip Programlanabilir ROM

EEPROM - Elektriksel olarak Silinip Programlanabilir ROM



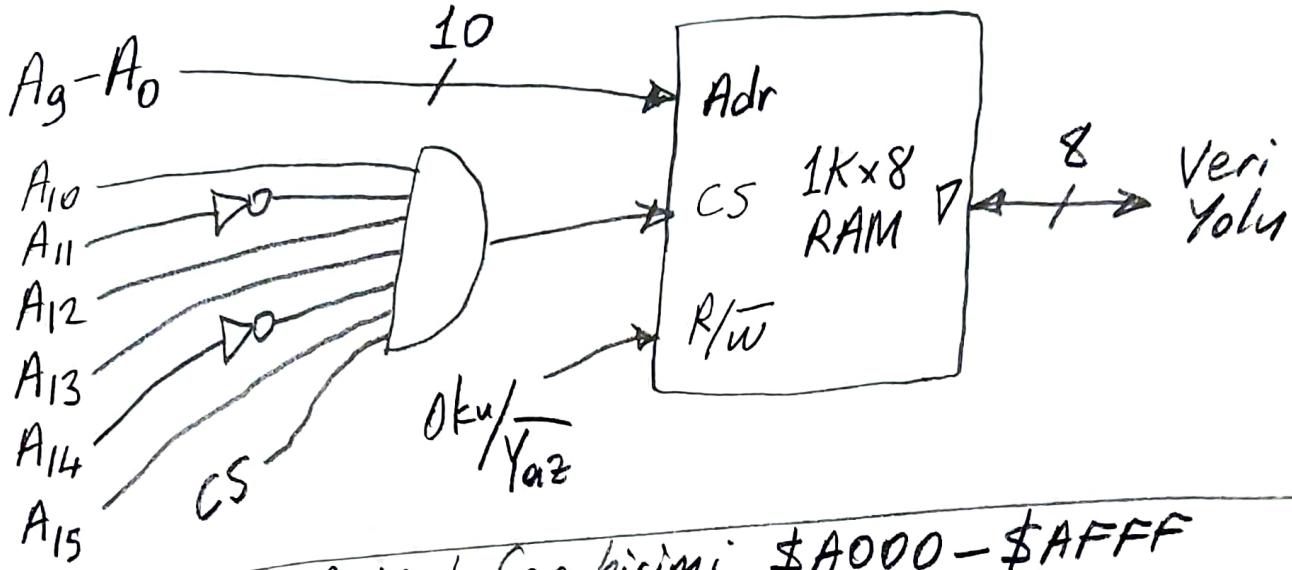
CS	İşlem
0	Yüksek Empedans
1	Oku

2Kx8 bitlik RAM hafıza birimi \$A800-\$AFFF adresleri arasında  
bağlanıyor. Hafıza biriminin kapı bağlantılarıyla beraber tasarıla.  
~~A<sub>15</sub> A<sub>14</sub> A<sub>13</sub> A<sub>12</sub> A<sub>11</sub> A<sub>10</sub> A<sub>9</sub> A<sub>8</sub> A<sub>7</sub> A<sub>6</sub> A<sub>5</sub> A<sub>4</sub> A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>~~



$1K \times 8$  bitlik RAM hafıza birimi  $\$B400 - \$B7FF$  adresleri arasında bağlanıyor. Hafıza biriminin kapı bağlantıları ile beraber tasarılmıştır.

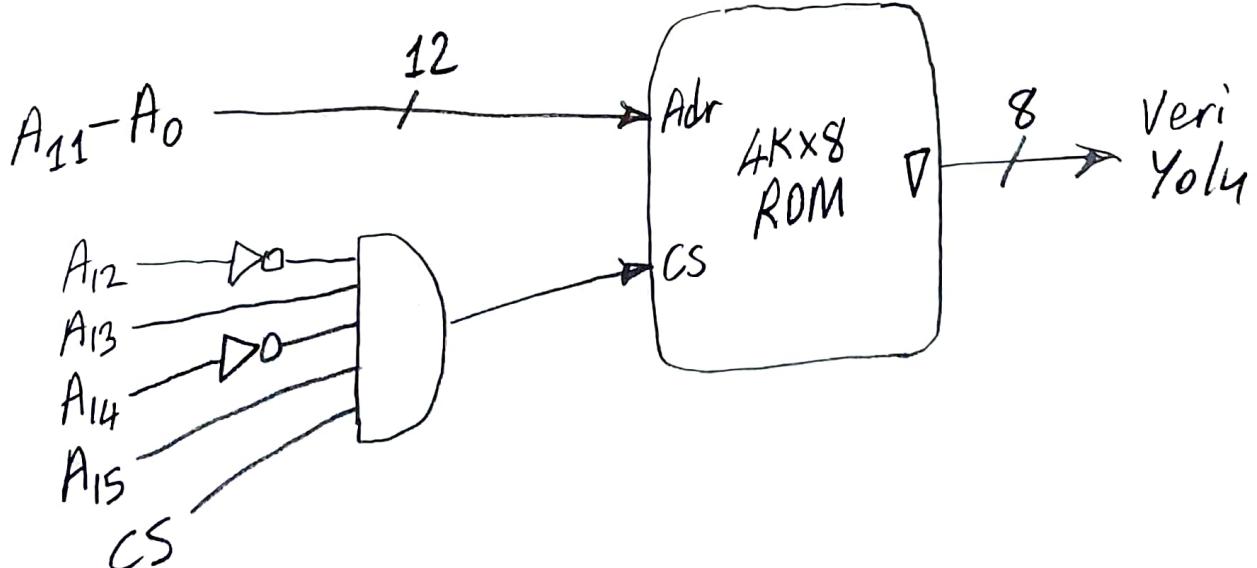
$A_{15} A_{14} A_{13} A_{12}$	$A_{11} A_{10} A_9 A_8$	$A_7 A_6 A_5 A_4$	$A_3 A_2 A_1 A_0$
1 0 1 1	0 1 X X	X X X X	X X X X



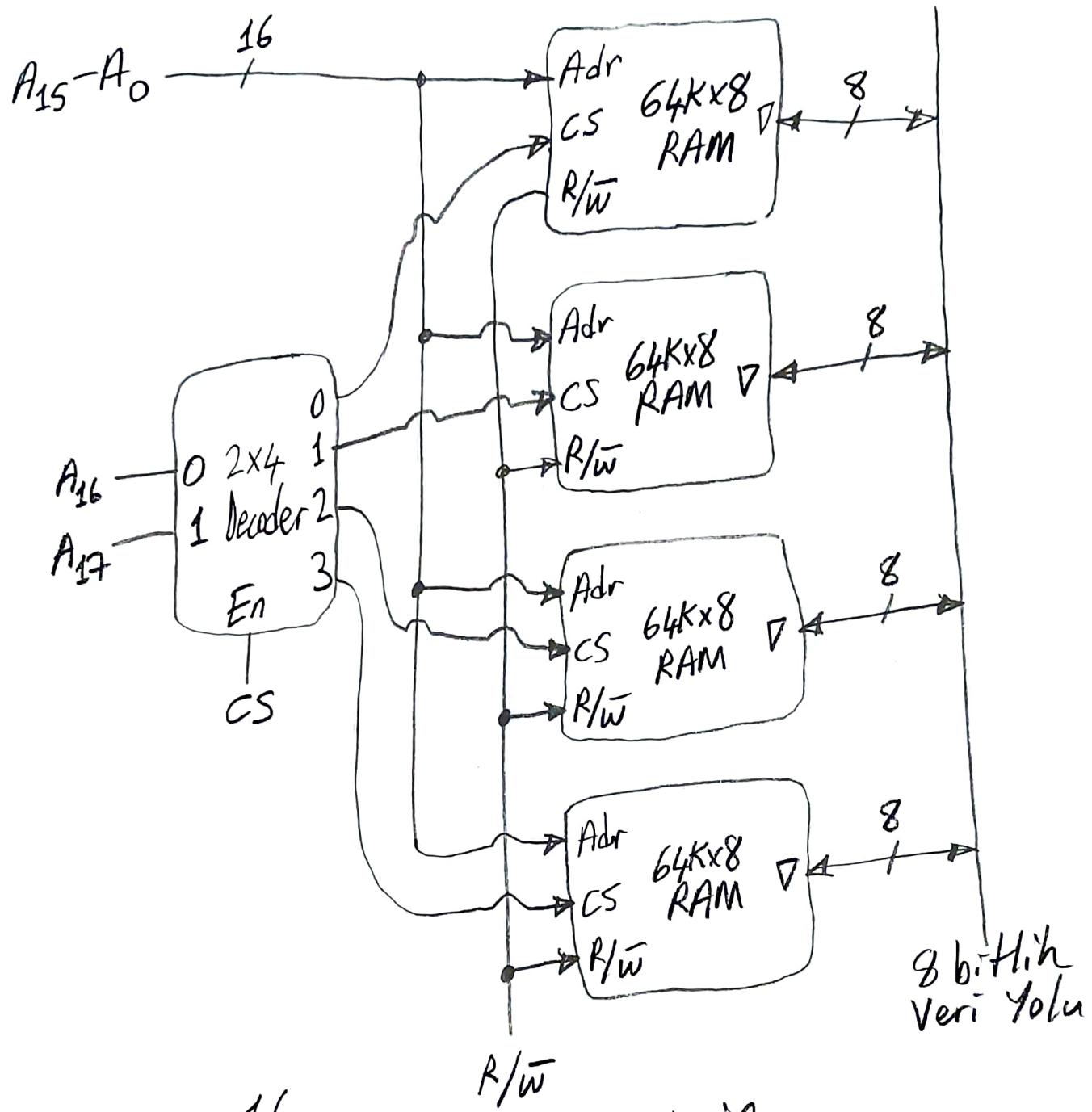
$4K \times 8$  bitlik ROM hafıza birimi  $\$A000 - \$AFFF$  adresleri arasında bağlanıyor. Hafıza biriminin kapı bağlantıları ile beraber tasarılmıştır.

$A_{15} A_{14} A_{13} A_{12}$	$A_{11} A_{10} A_9 A_8$	$A_7 A_6 A_5 A_4$	$A_3 A_2 A_1 A_0$
1 0 1 0	X X X X	X X X X	X X X X

$4K = 4096 = 2^{12}$



4 tane  $64K \times 8$  bitlik RAM, 1 tane  $2 \times 4$  Decoder  
kullanarak  $256K \times 8$  bitlik RAM tasarla.

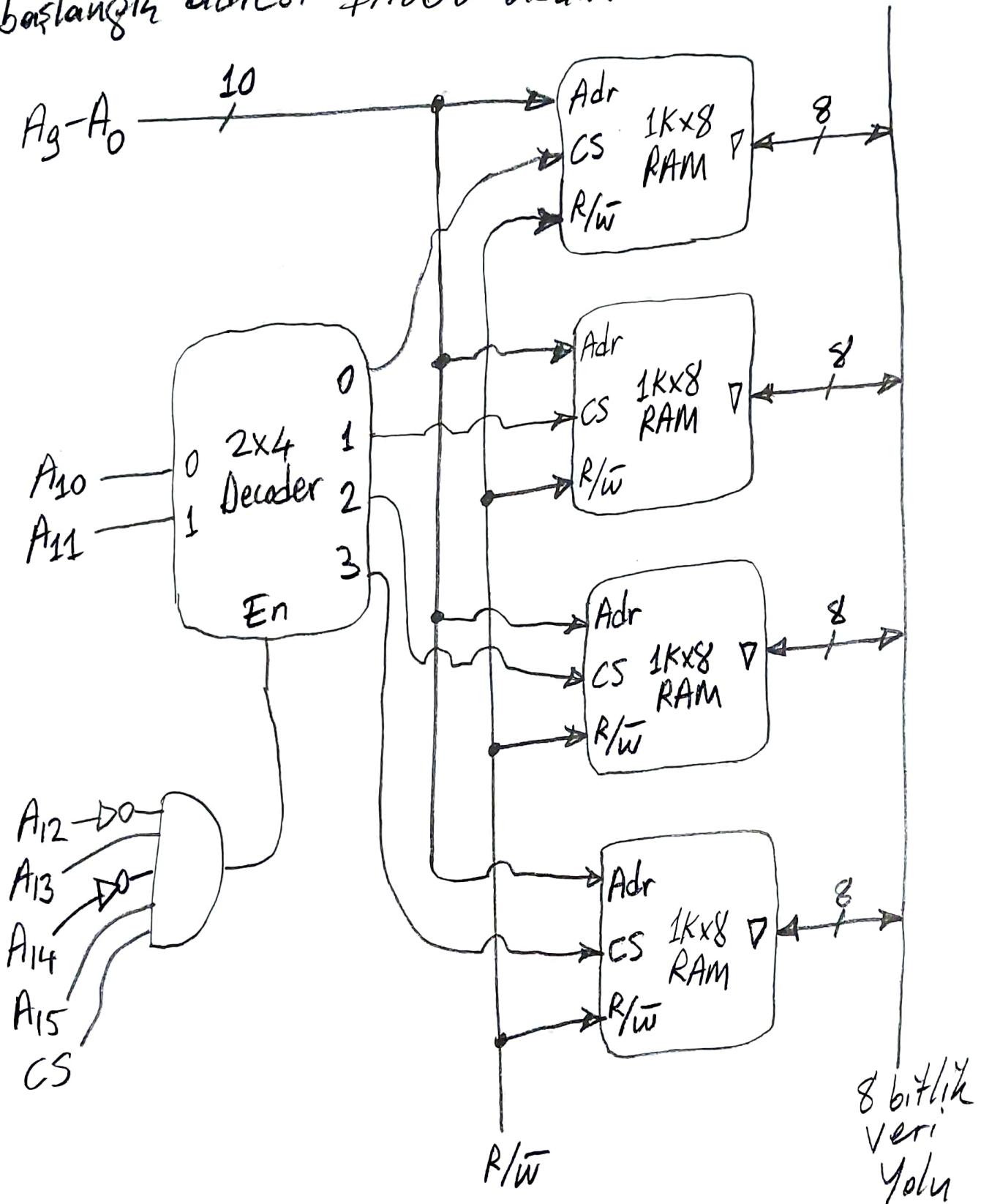


$$64K = 2^{16}$$

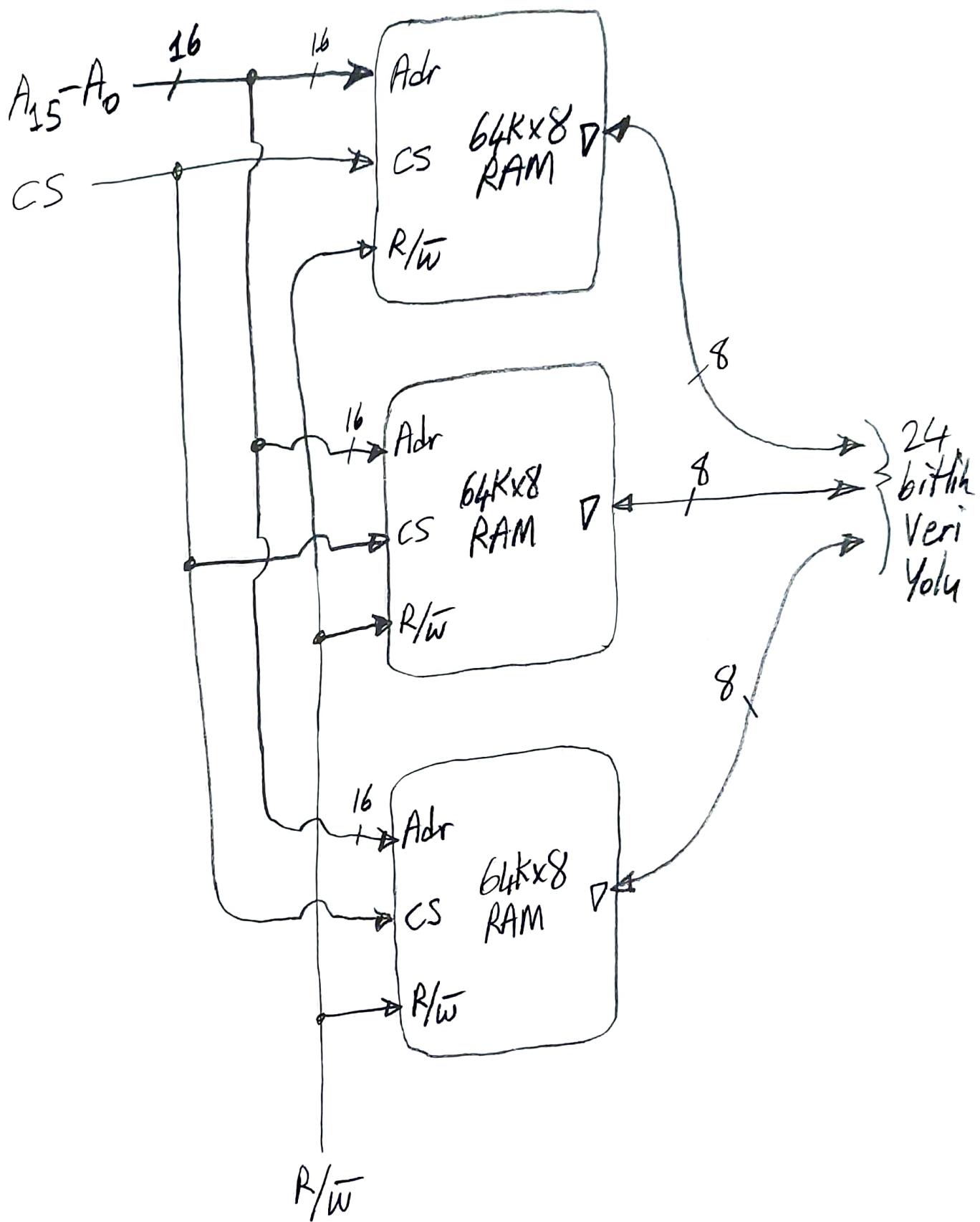
$$256K = 2^{18}$$

Decoder sıklıkları RAM'lerin  
CS'lerine girecek.  
Decoder'in Enable'i  
ortak CS olacak.

4 tane  $1K \times 8$  bitlik RAM, 1 tane  $2 \times 4$  Decoder ve  
 en az kapı elemanı kullanarak 1 tane  $4K \times 8$  bitlik  
 RAM yani hafıza birimi tasarıla. Hafıza biriminin  
 başlangıç adresi  $\$A000$  olsun. (8)

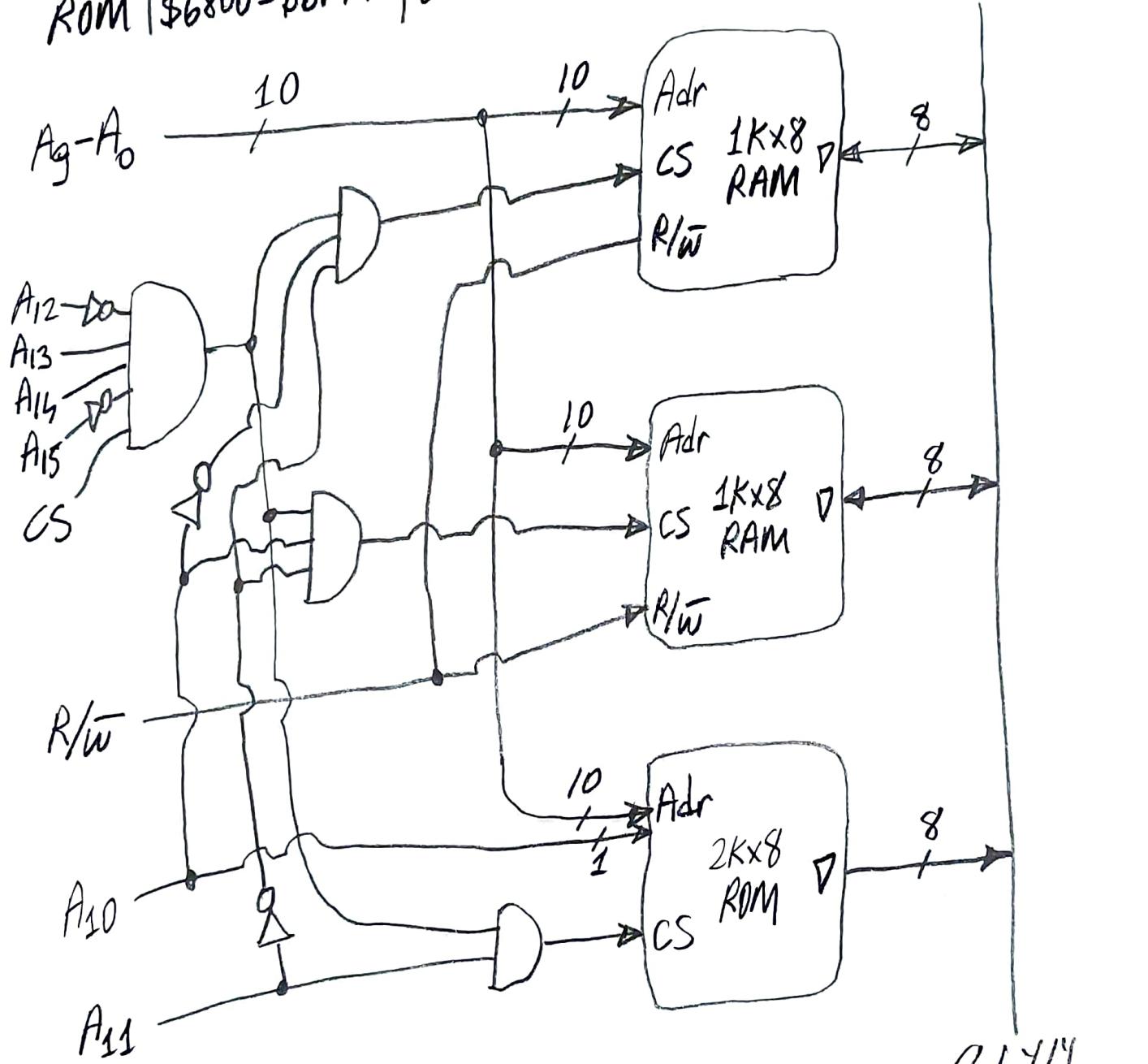


3 tane  $64K \times 8$  bitlik RAM kullanarak  
1 tane  $64K \times 24$  bitlik RAM tasarıla.



16 bitlik adres yolu olan bir hafıza biriminin \$6000-\$6FFF adres bölgüsüne 2 tane 1Kx8 bitlik RAM, 1 tane 2Kx8 bitlik ROM bağlanıyor.

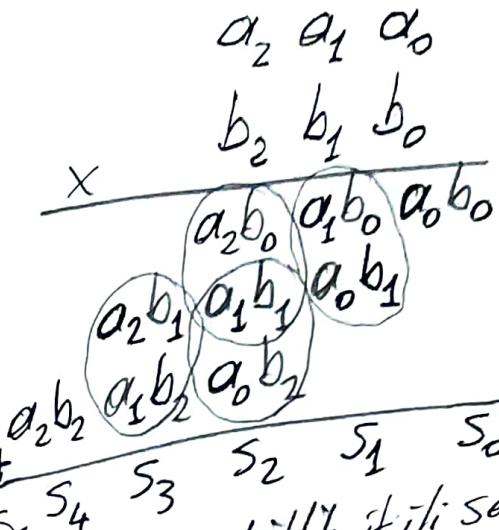
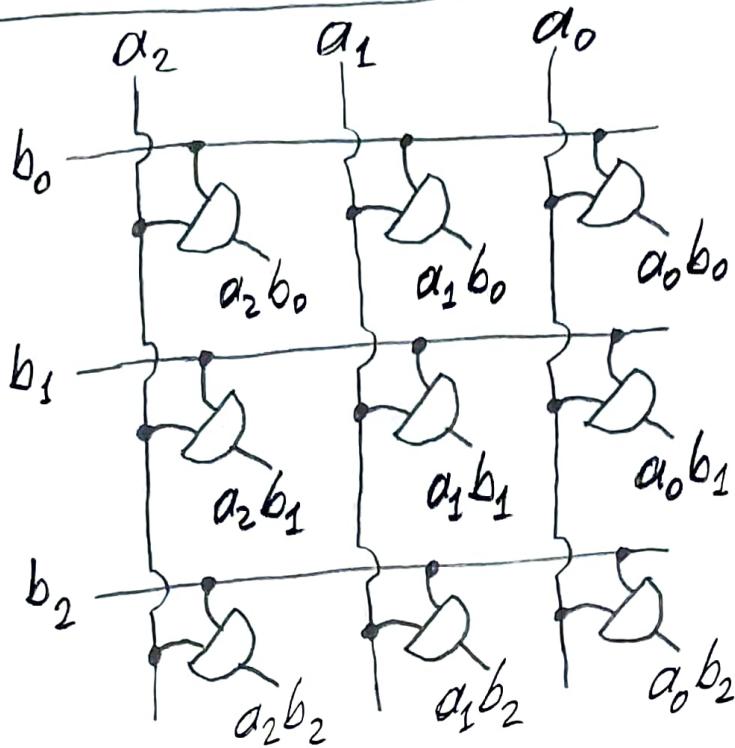
B.legen	Adres	A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub> A <sub>11</sub> A <sub>10</sub> A <sub>9</sub> A <sub>8</sub> A <sub>7</sub> A <sub>6</sub> A <sub>5</sub> A <sub>4</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>
RAM1	\$6000-\$63FF	0 1 1 0 0 0 X X XXX XXX
RAM2	\$6400-\$67FF	0 1 1 0 0 1 X X XXX XXX
ROM	\$6800-\$6FFF	0 1 1 0 1 X X X XXX XXX



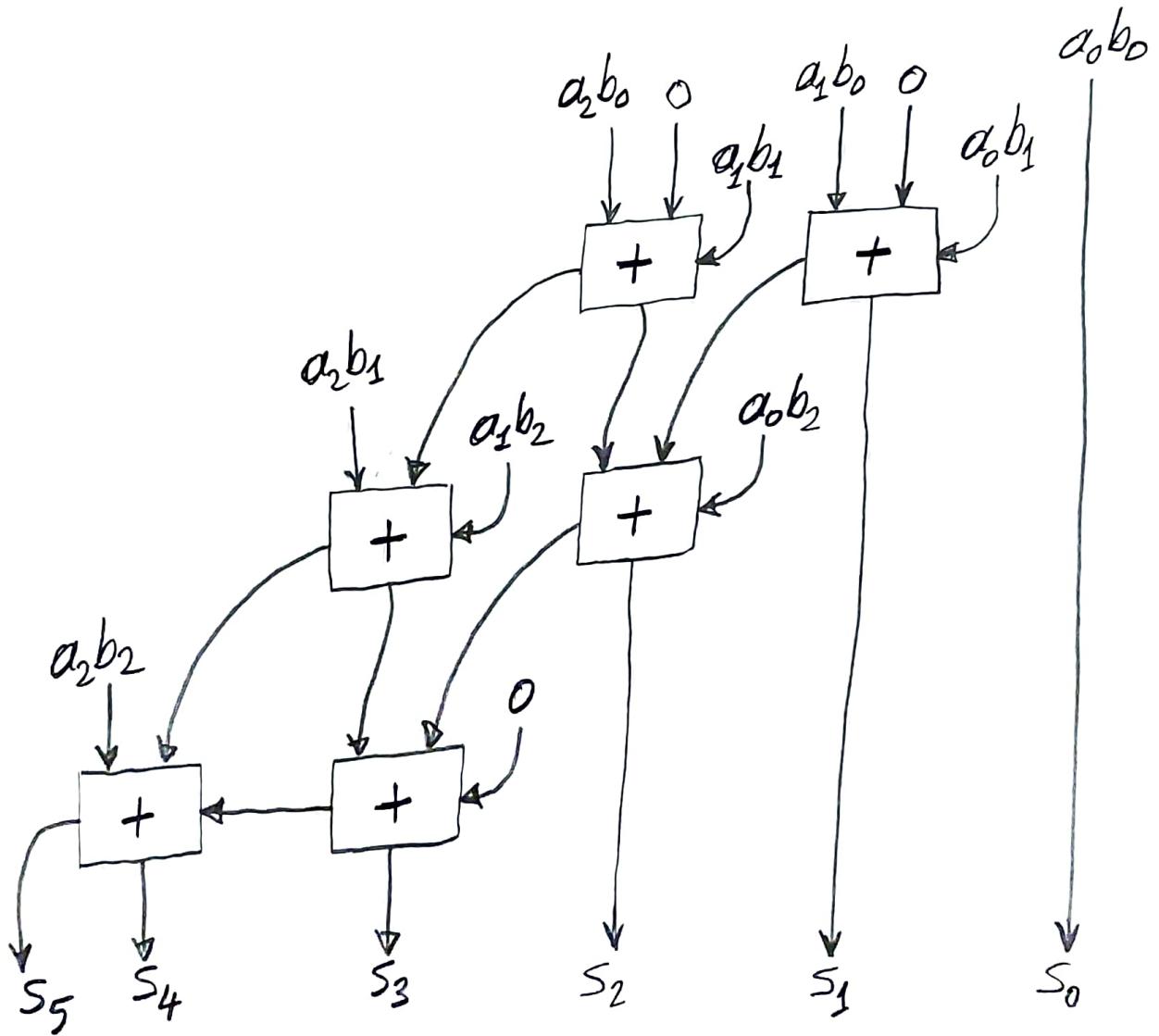
$$\begin{aligned} 1K &= 2^{10} \\ 2K &= 2^{11} \end{aligned}$$

8 bitlik  
Veri  
Yolu

# İkili Tabanda Çarpma



2 tane 3 bitlik ikili sayı  
çarpılınca en fazla 6 bitlik  
ikili sayı elde edilir.  
 $2^3 \times 2^3 = 2^6 = 64$   
0 ile 63 arası



iki tane 4 bitlik ikili toplayıcı ve en sağ kapi elementini  
 kullanarak  $4 \times 3$ 'lik ikili sarmaşık devresini gerçekleştirebiliriz.  
 $A = a_3 a_2 a_1 a_0$  yani 4 bit,  $B = b_3 b_2 b_1$  yani 3 bit olsun.

Sarmaşık sonucu  
 en fazla  $4+3=7$   
 bit olur.

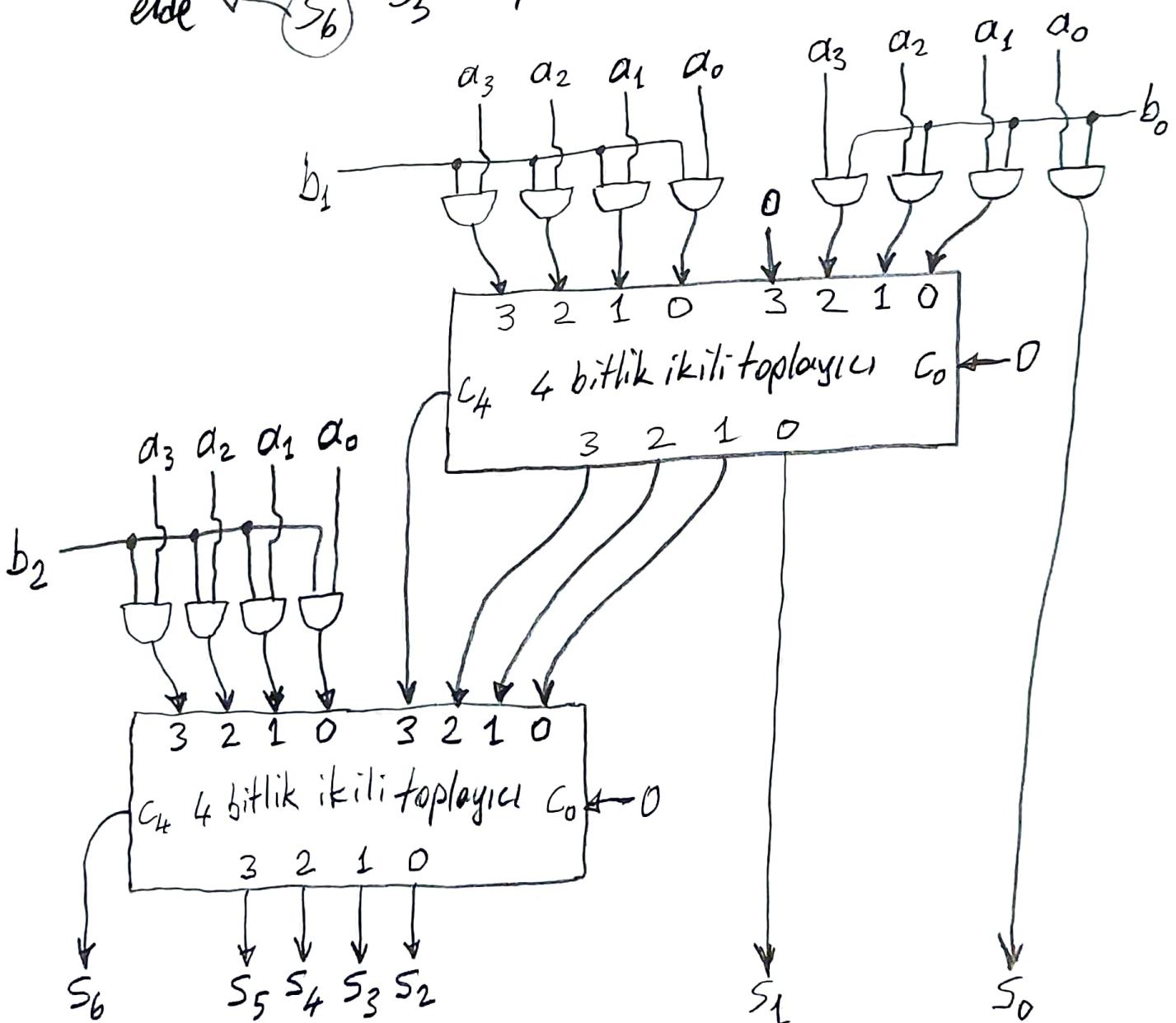
$$\begin{array}{r}
 A = \begin{matrix} a_3 & a_2 & a_1 & a_0 \end{matrix} \\
 B = \begin{matrix} b_2 & b_1 & b_0 \end{matrix} \\
 \hline
 \end{array}$$

$\times$

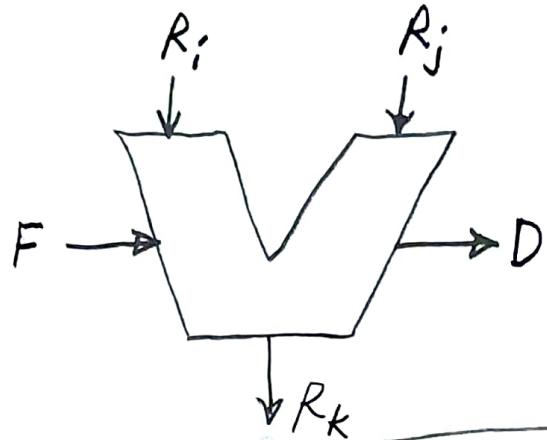
$a_3 b_0$	$a_2 b_0$	$a_1 b_0$	$a_0 b_0$	$a_3 b_1$	$a_2 b_1$	$a_1 b_1$	$a_0 b_1$	$a_3 b_2$
$a_3 b_1$	$a_2 b_1$	$a_1 b_1$	$a_0 b_1$	$a_3 b_2$	$a_2 b_2$	$a_1 b_2$	$a_0 b_2$	$a_3 b_3$

$\overset{+}{\hline}$

$S_6 \quad S_5 \quad S_4 \quad S_3 \quad S_2 \quad S_1 \quad S_0$



# Aritmetik Mantık Birimi (ALU - Arithmetic Logic Unit) 13

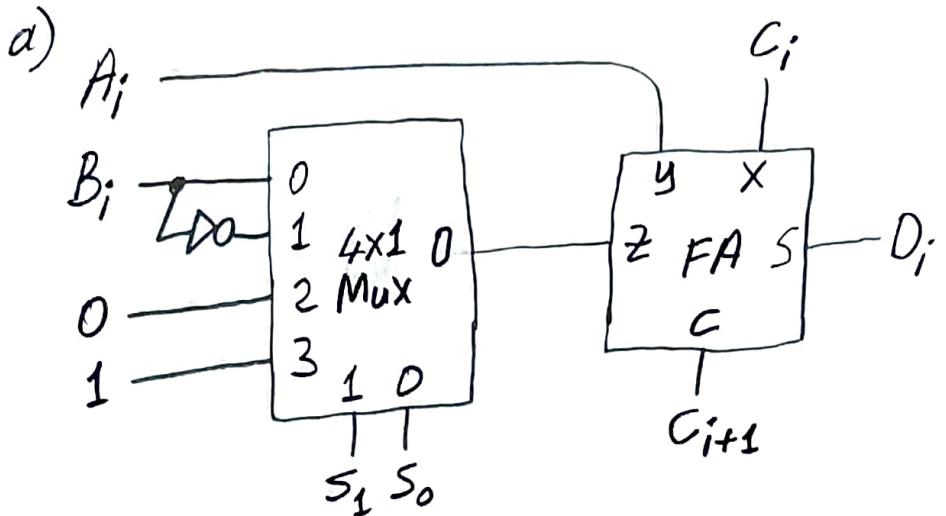


$R_i$  ve  $R_j$  herhangi iki yazacan gelen girdiler,  $R_k$  herhangi bir yazaca giden sıktılar,  $F$  denetim biriminden (Control Unit) gelen girdiler,  $D$  ise durum bildiren sıktılar.

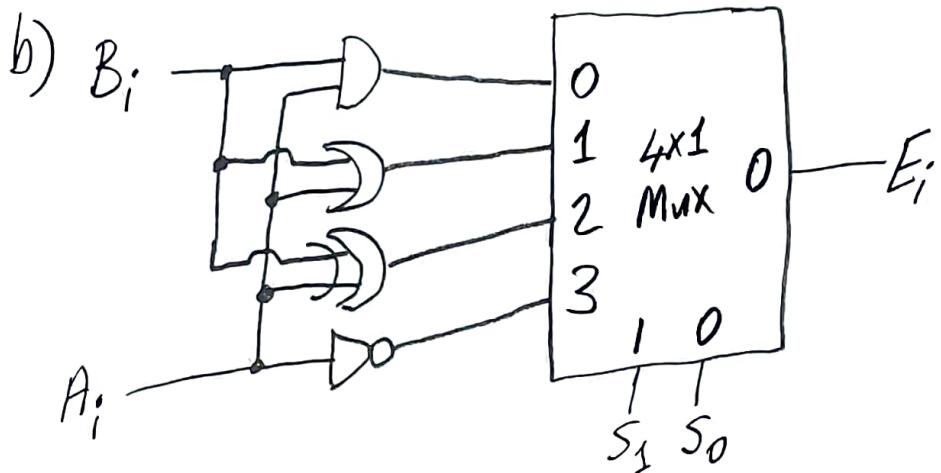
$S_3\ S_2\ S_1\ S_0\ C_i$	İşlem
0 0 0 0 0	$F \leftarrow A + B$
0 0 0 0 1	$F \leftarrow A + B + 1$
0 0 0 1 0	$F \leftarrow A + \bar{B}$
0 0 0 1 1	$F \leftarrow A + \bar{B} + 1$
0 0 1 0 0	$F \leftarrow A$
0 0 1 0 1	$F \leftarrow A + 1$
0 0 1 1 0	$F \leftarrow A - 1$
0 0 1 1 1	$F \leftarrow A$
0 1 0 0 X	$F \leftarrow A \wedge B$
0 1 0 1 X	$F \leftarrow A \vee B$
0 1 1 0 X	$F \leftarrow A \oplus B$
0 1 1 1 X	$F \leftarrow \bar{A}$
1 0 X X X	$F \leftarrow \text{Shr } A$
1 1 X X X	$F \leftarrow \text{Shl } A$

Fonksiyon tablosu  
yanda verilen  
Aritmetik Mantık  
Birimini (ALU)  
devresinin bir bitlik  
kismini aşağıdaki  
aşamaları dikkate  
alarak tasarlayınız.

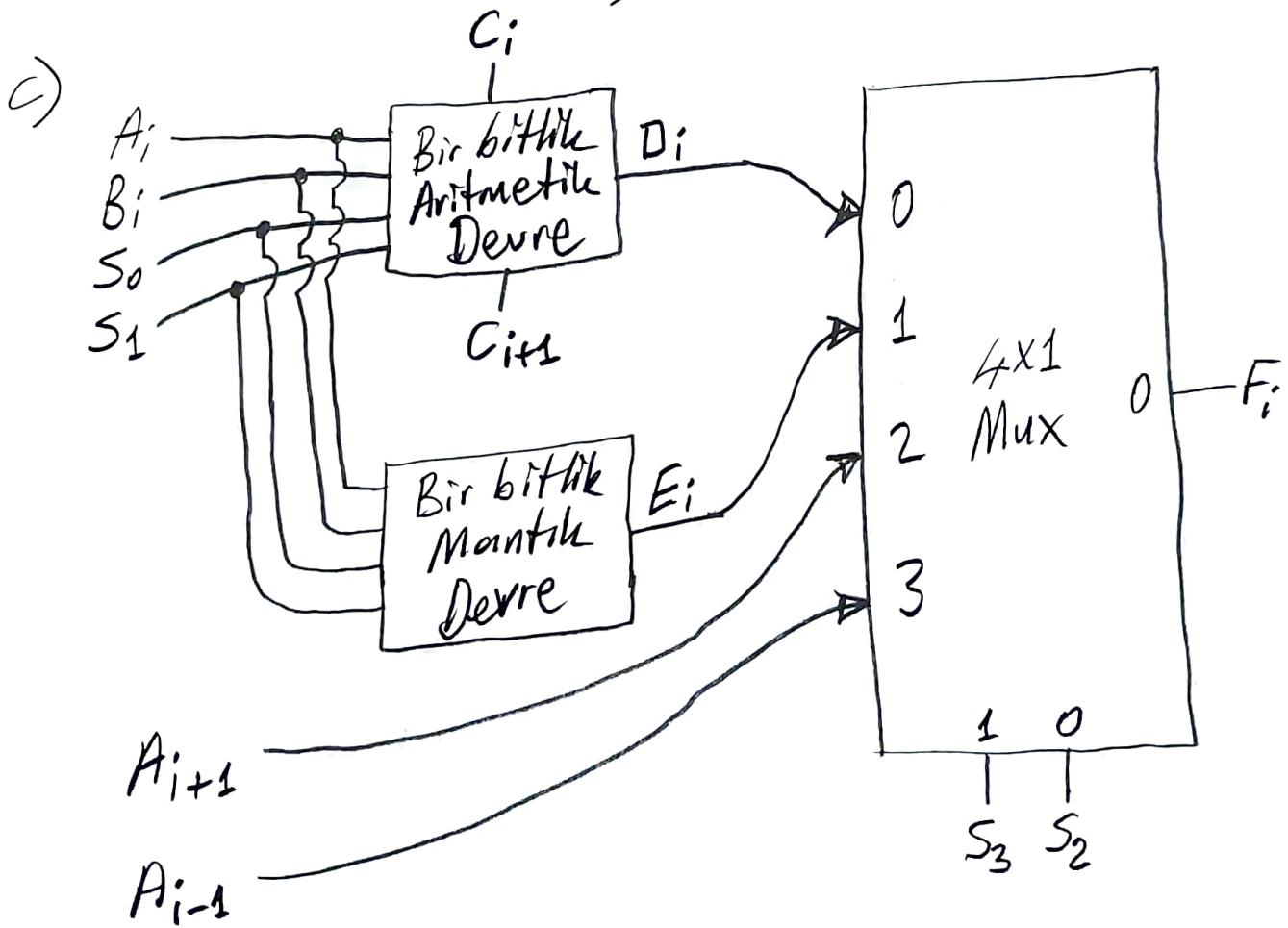
- Bir bitlik aritmetik devre aşaması
- Bir bitlik mantık devre aşaması
- Bir bitlik ALU aşaması

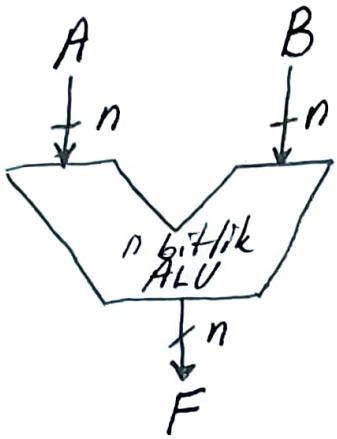


Bir bitlik  
aritmetik  
devre  
organizasyonu



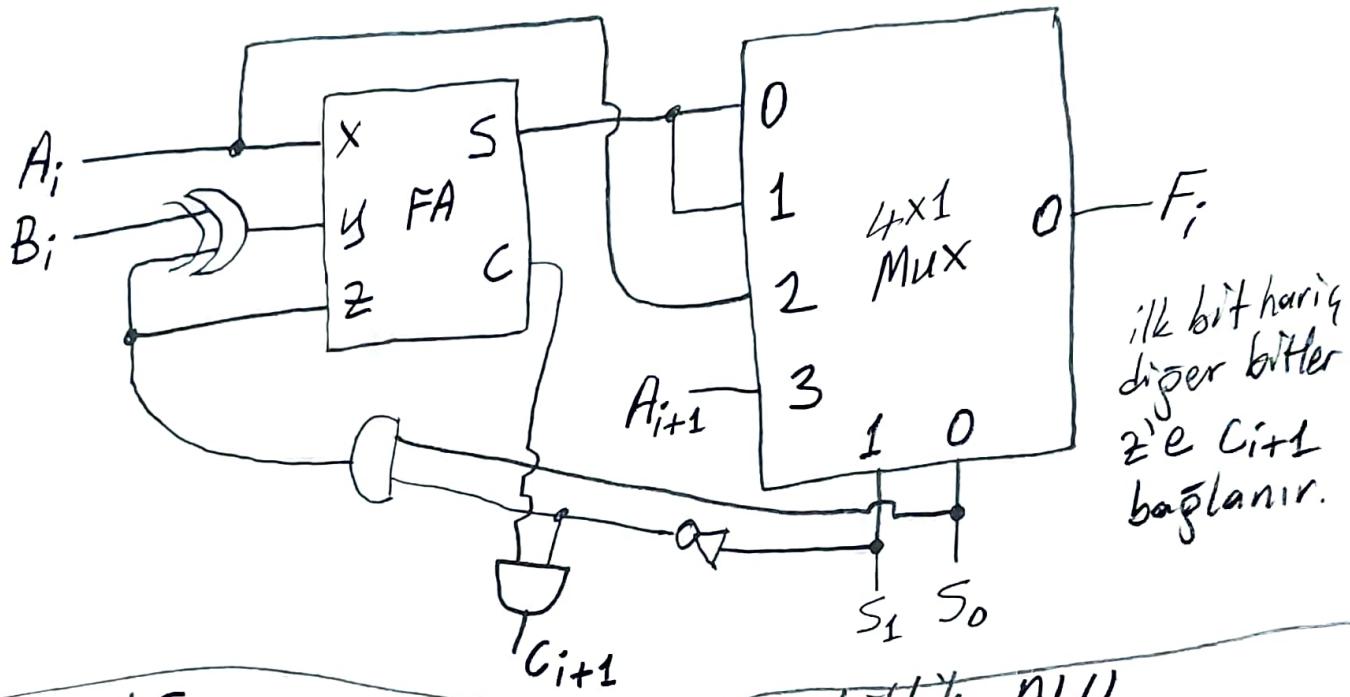
Bir bitlik  
mantik  
devre  
organizasyonu





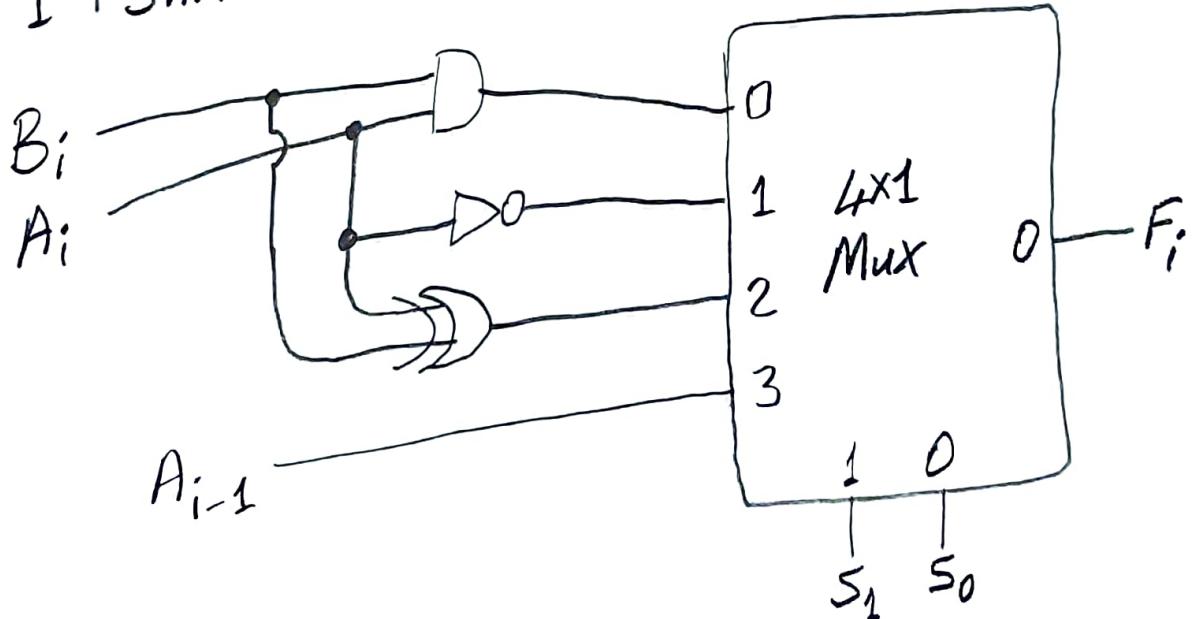
$S_1\ S_0$	F
00	$A+B$
01	$A-B$
10	A
11	ShrlA

Yanda verilen n bitlik ALU devresinin bir bitlik kısmını  $4 \times 1$  Mux, FA ve en az kapı elemanı kullanarak tasarlara. (15)



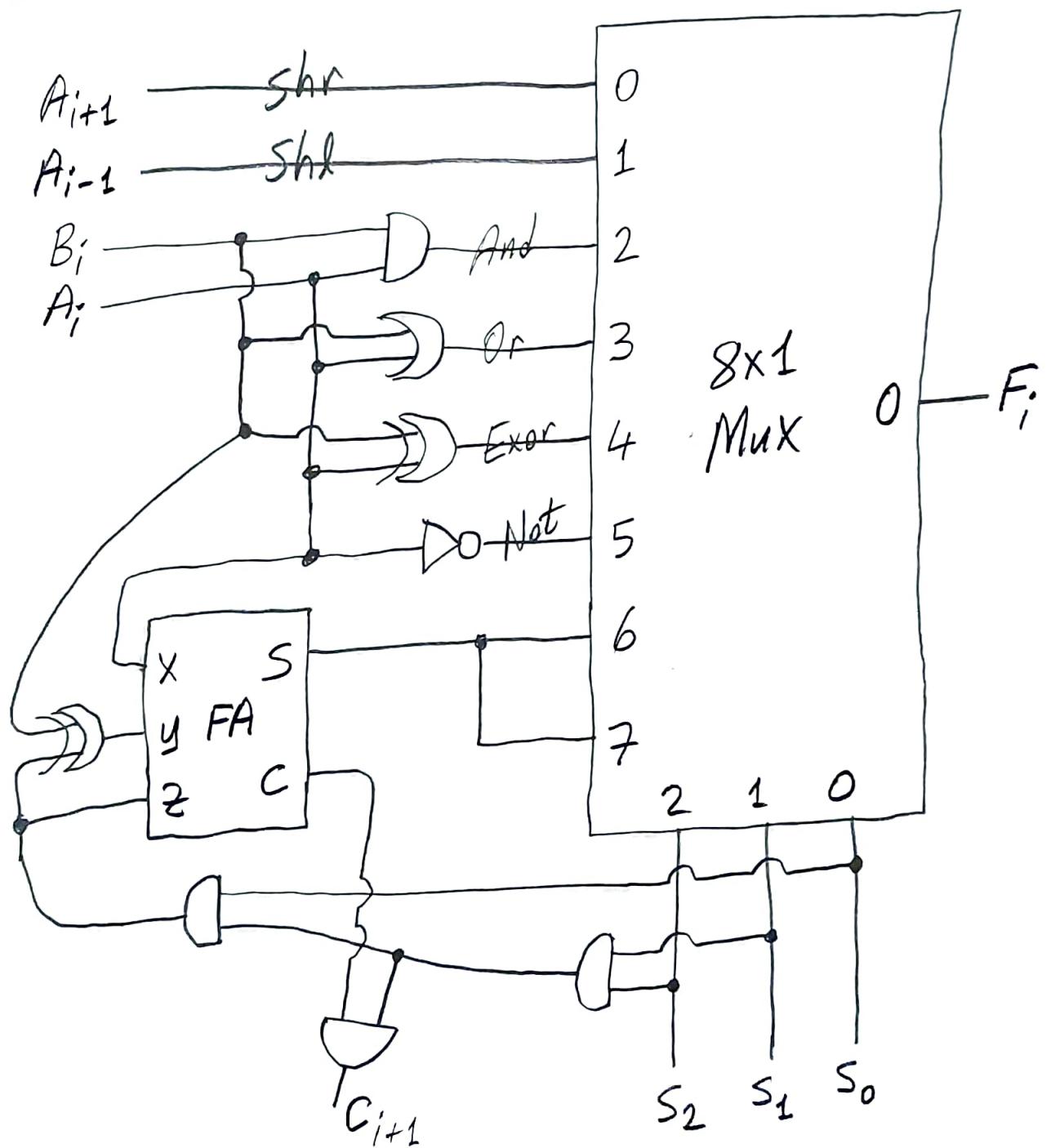
$S_1\ S_0$	F
00	$A \wedge B$
01	$\bar{A}$
10	$A \oplus B$
11	ShrlA

Yanda verilen n bitlik ALU devresinin bir bitlik kısmını  $4 \times 1$  Mux ve en az kapı elemanı kullanarak tasarlara.

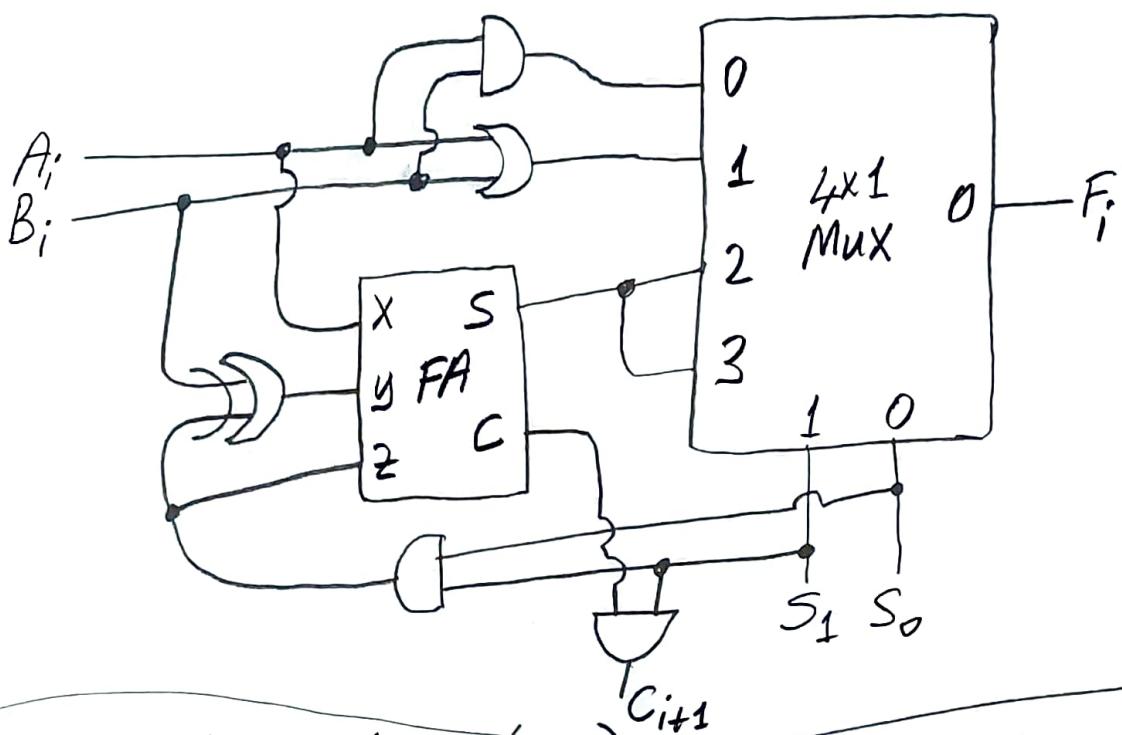


$S_2 S_1 S_0$	F
0 0 0	ShrA
0 0 1	ShlA
0 1 0	A $\wedge$ B
0 1 1	A $\vee$ B
1 0 0	A $\oplus$ B
1 0 1	$\bar{A}$
1 1 0	A + B
1 1 1	A - B

Dogruluk tablosu yandaki verilen n bitlik ALU devresinin bir bitlik kismini  $8 \times 1$  Multiplexer, Full Adder (Tam Toplayıcı) ve en az kapi elemeleri kullanarak tasarla.



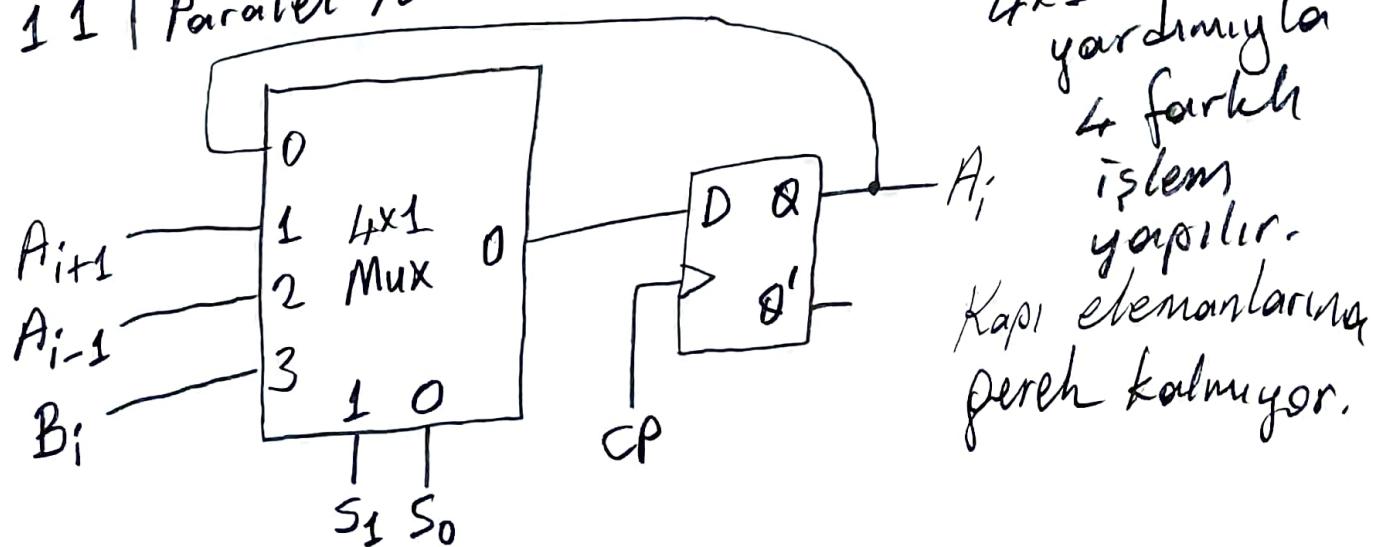
$S_1 S_0$	F	Dogruluk tablosu yandır verilen n bitlik ALU devresinin bir bitlik kismini 4x1 Mux, FA ve en az kapi elementi kullanarak tasarlar.
00	$A \oplus B$	
01	$A \ominus B$	
10	$A + B$	
11	$A - B$	



### Yazaglar (Registers)

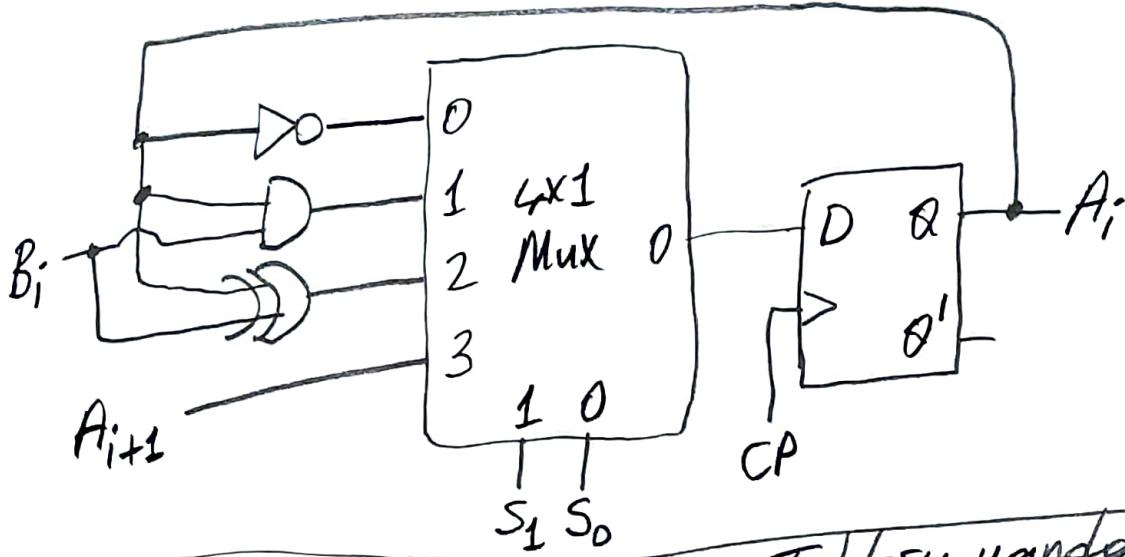
$S_1 S_0$	İşlem
00	Degişiklik Yok
01	Sağ Kaydır
10	Sola Kaydır
11	Parallel Yükle

Fonksiyon tablosu yandır verilen  
paralel yüklemeli çift yönlü  
kaydırma yazacın bir bitlik  
kismini 4x1 Mux ve 0 flip-flop  
kullanarak tasarlar.

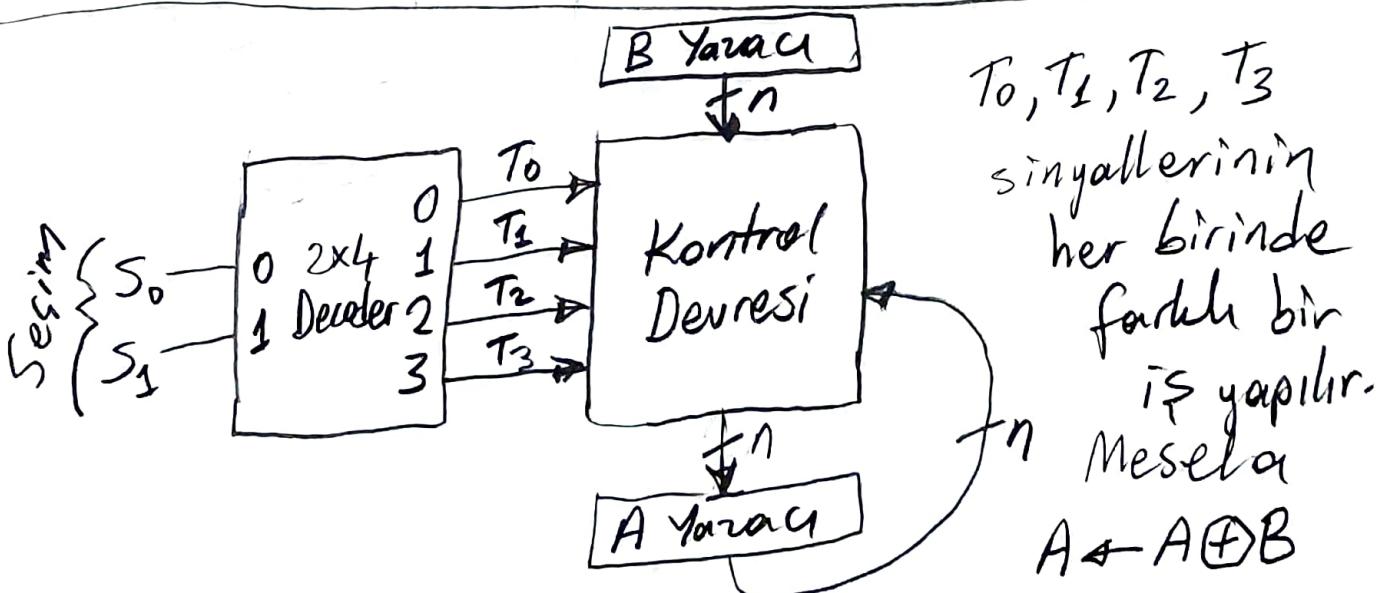
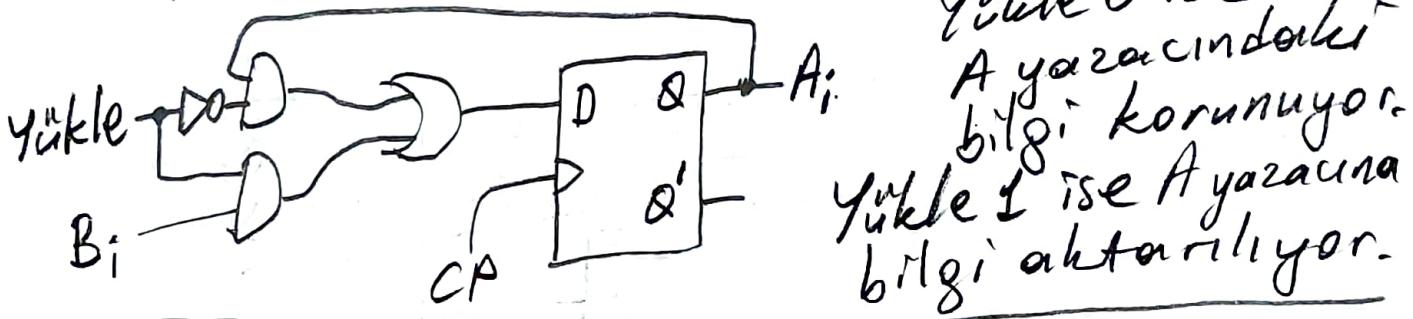


4x1 Mux  
yardımıyla  
4 farklı  
işlem  
yapılır.  
Kapi elementlarına  
dereh kalmıyor.

$S_1 S_0$	İşlem	Fonksiyon tablosu yanda verilen n bitlik A yazarının bir bitlik kismini $4 \times 1$ Mux, D flip-flop ve kapı elemanı kullanarak tasarla.	(18)
00	$A \leftarrow \bar{A}$		
01	$A \leftarrow A \wedge B$		
10	$A \leftarrow A \oplus B$		
11	$A \leftarrow \text{Shift}$		



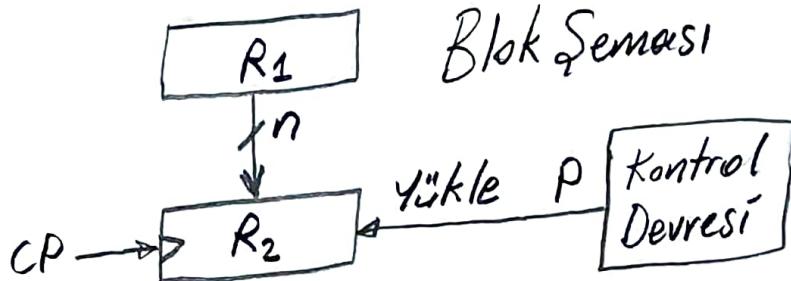
<u>Yükle</u>	<u>İşlem</u>	Fonksiyon Tablosu yanda verilen yazarın bir bitlik kismini D flip-flop ve en az kapı elemanı kullanarak tasarla.
0	Bilgiyi Kora	
1	Yeni Bilgi Yükle	



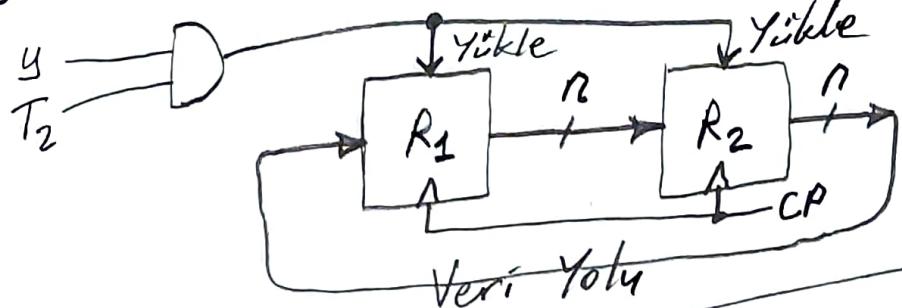
# Yazılım Aktarımı

P :  $R_2 \leftarrow R_1$

(19)

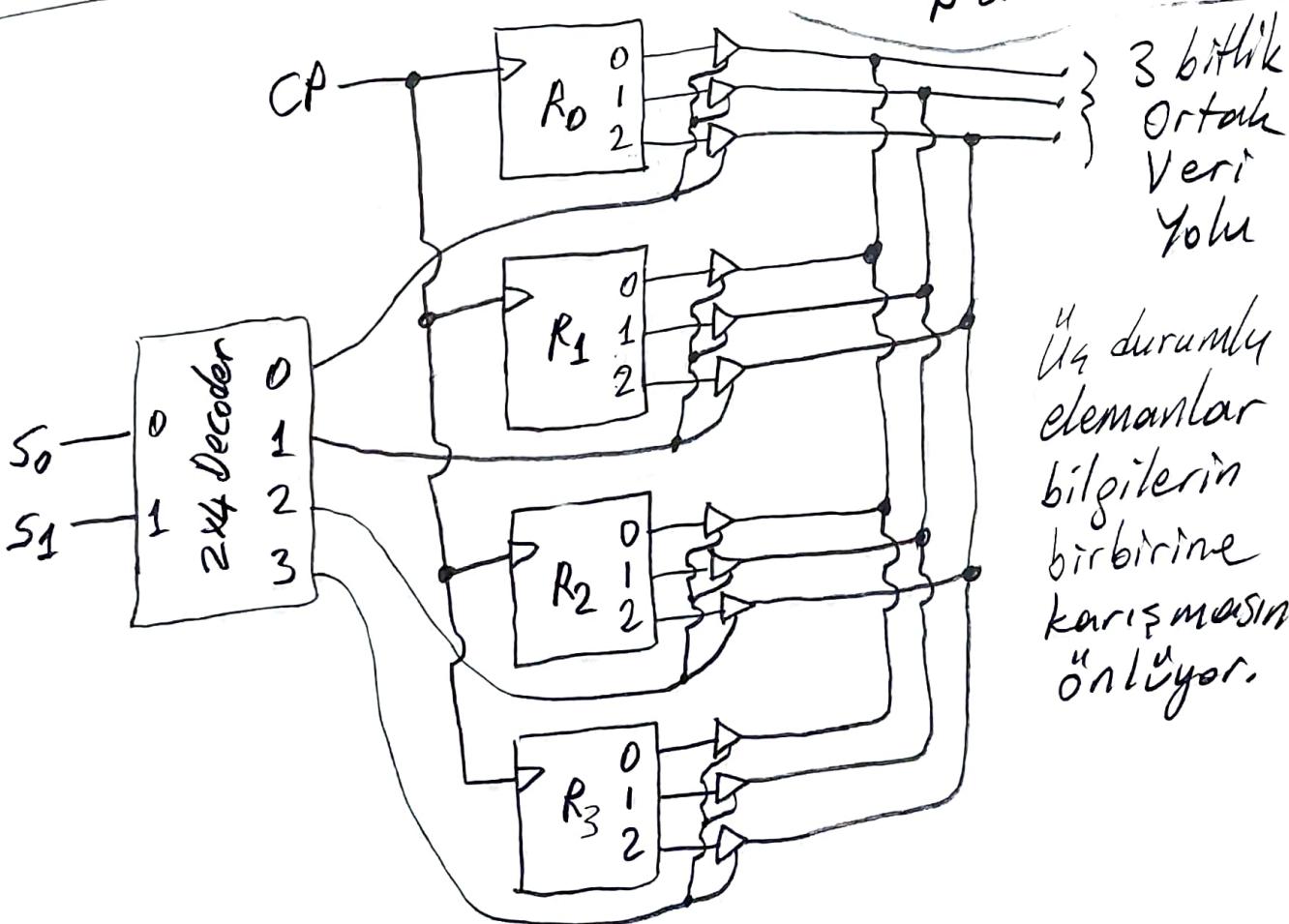
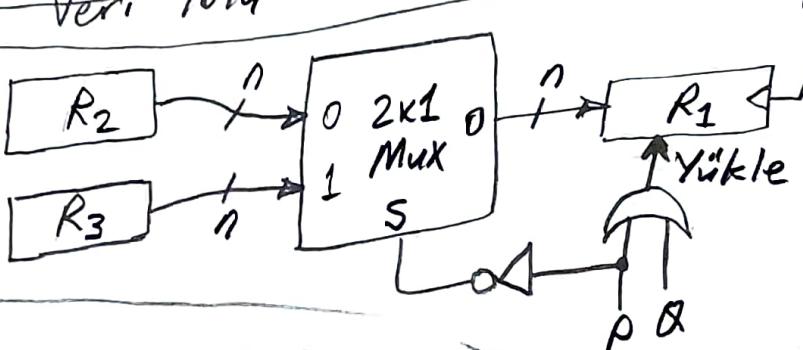


$yT_2 : R_2 \leftarrow R_1, R_1 \leftarrow R_2$  (Exchange - Yer Değiştir)



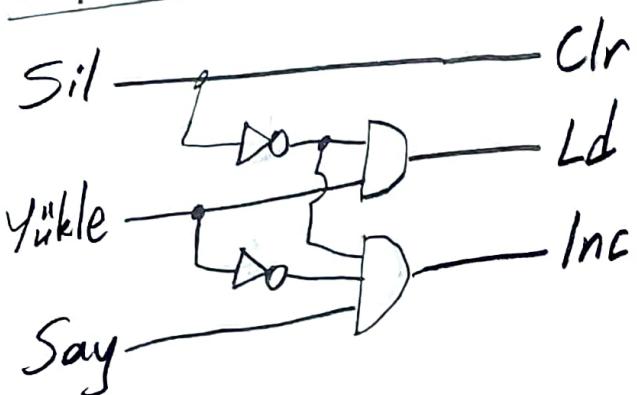
$y=1$  ve  $T_2=1$   
iken CP o'dan  
i'e geseince  
 $R_2 \leftarrow R_1$  olur.

P :  $R_1 \leftarrow R_2$   
P'Q :  $R_1 \leftarrow R_3$

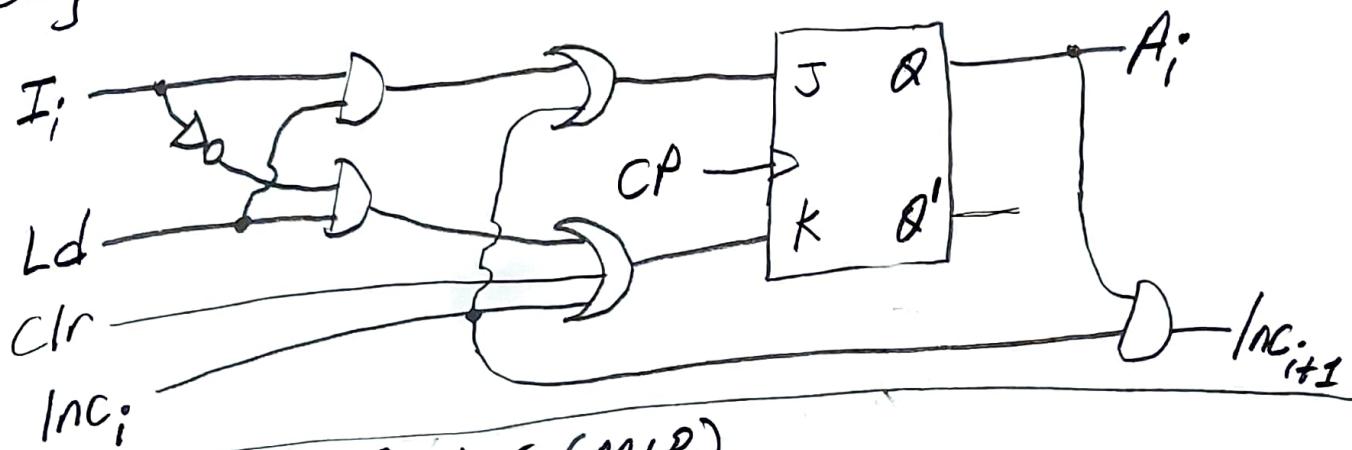


Sil	Yükle	Sayı	İşlem
0	0	0	Koru
0	0	1	Sayı
0	1	X	Yükle
1	X	X	Sil

Fonksiyon tablosu yanda verilen birlik yazacın ortak olan kontrol kısmını ve bir bitlik kısmını JK flip-flop kullanarak tasarıla.

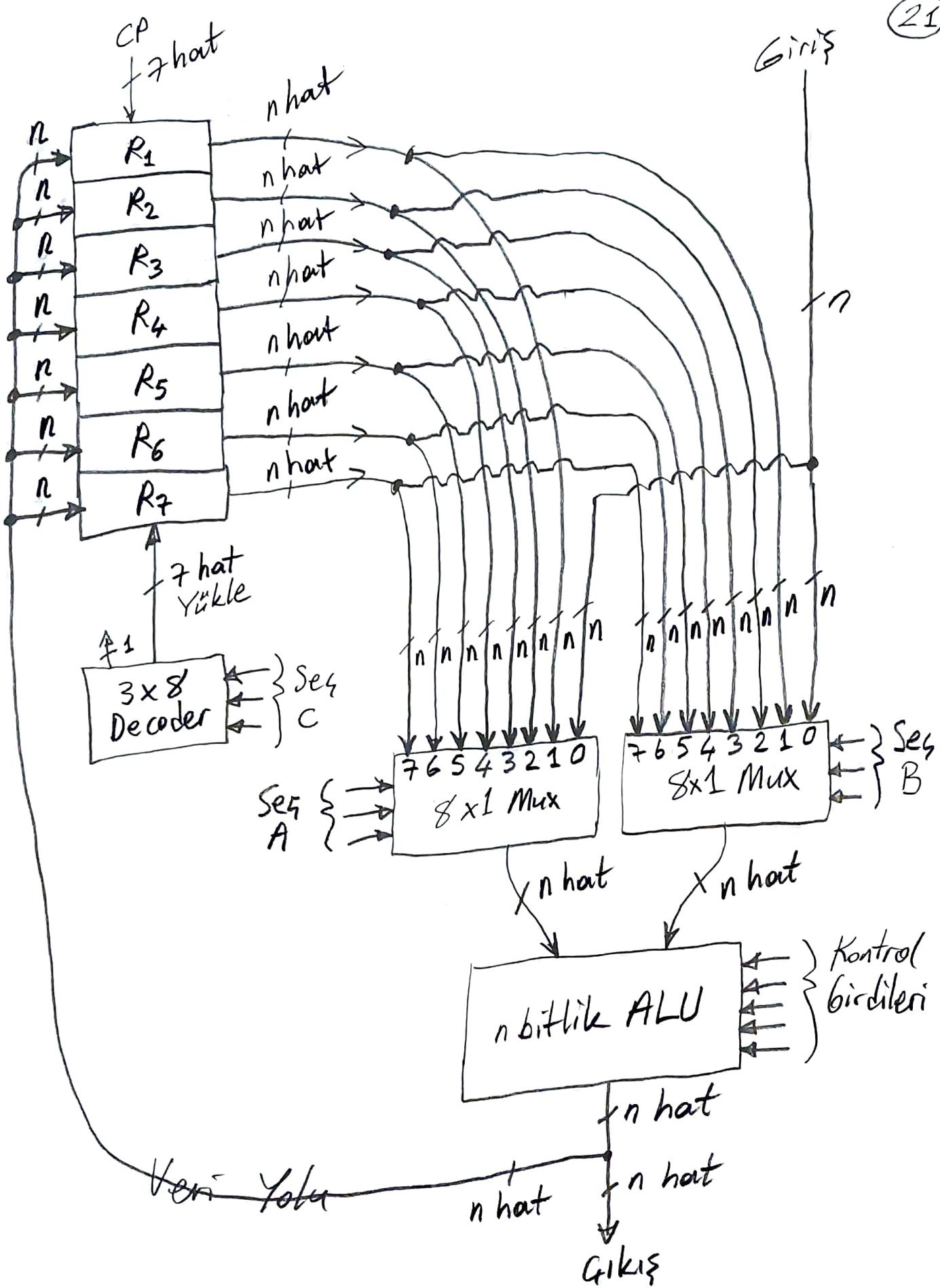


JK	$Q_{t+1}$
00	$Q_t$
01	0
10	1
11	$Q'_t$

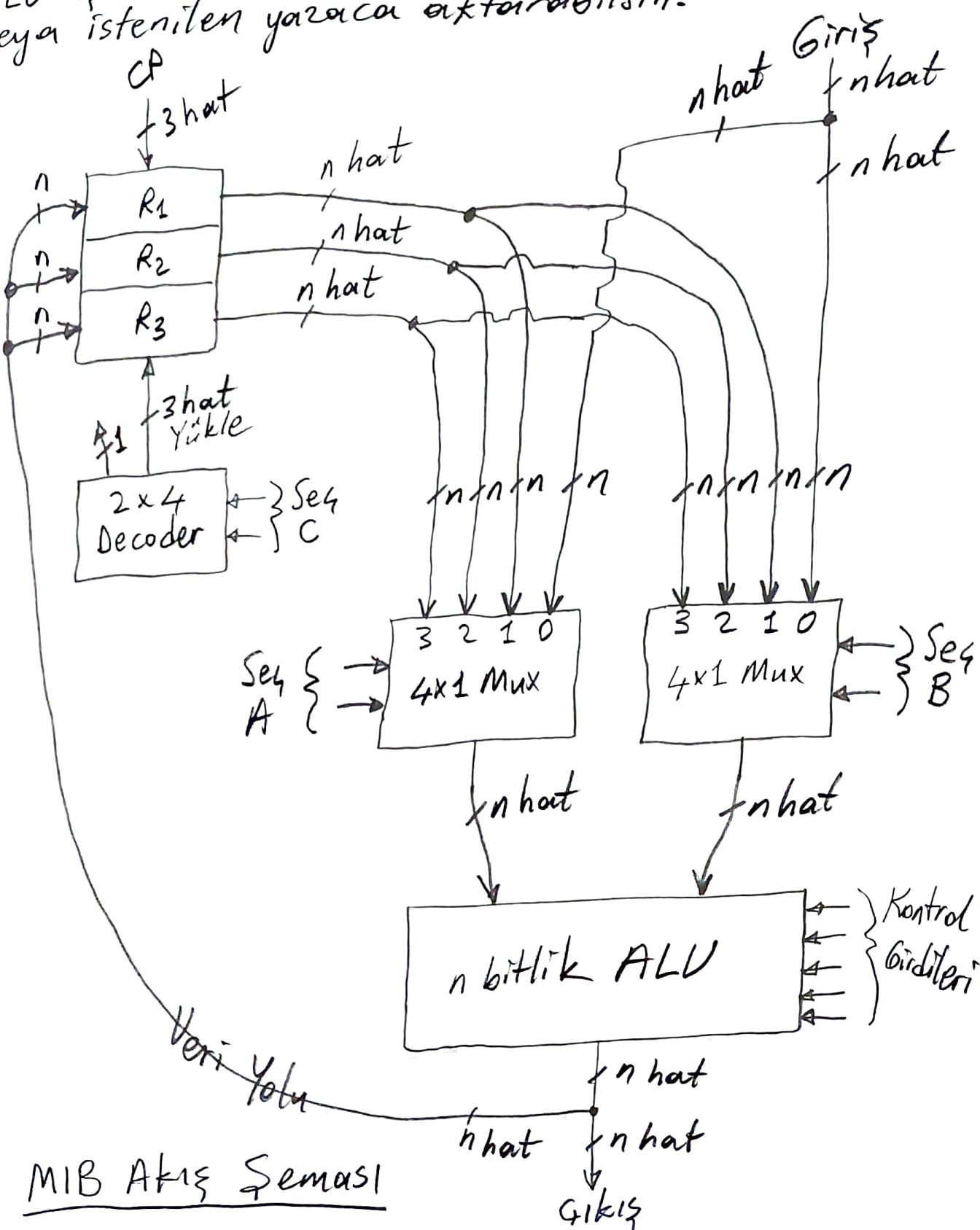


### Merkezi İşlem Birimi (MIB) Central Processing Unit (CPU)

- Aşağıdaki soruları cevaplayın CPU devresinin blok diyagramını çiziniz.
- \* CPU devresi 7 yazar, 1 ALU, 3 Seçiciden oluşsun.
  - \* ALU devresi 5 tane kontrol girişine sahip olsun.
  - \* ALU devresi 5 tane kontrol girişlerinin hangi işlemleri yaptığı önemsiz.
  - \* ALU devresi SegA ve SegB veri seçicileri yardımıyla hem girişten hem de yazarlardan veri alabilse.
  - \* Sonuç SegC veri seçicisi yardımıyla çıkışın veya istenilen yaza aktarılabilse.



Aşağıdakî şartları sağlayan Merkezi İşlem Biriminin (MIB) blok şemasını çiziniz. MIB içinde 1 tane ALU, 3 tane yazar ve 3 tane seçici bulunsun. ALU 5 kontrol girişine sahip olsun. ALU, A ve B seçicileri yardımıyla hem girişten hem de yazarlardan girdilerini alabilse sin. ALU işlem sonucunu C seçici yardımıyla sıkıştır veya istenilen yazaçor aktarabilse sin.



$$T_0 : A \leftarrow \bar{A}$$

$$T_1 : A \leftarrow B$$

$$T_2 : A \leftarrow A \oplus B$$

$$T_3 : A \leftarrow A \wedge B$$

Yanda verilen işlemleri yapan devrenin (23) bir bitlik kısmını JK flip-flop ve en az kapı elemanı kullanarak tasarlaymentiz.

JK |  $\theta_{t+1}$

$\theta_t \theta_{t+1}$  | JK

00 |  $\theta_t$

01 | 0

10 | 1

11 |  $\theta'_t$

Karakteristik  
Tablo

00 | 0X

01 | 1X

10 | X1

11 | X0

Üyelik  
Tablosu

$$\underline{T_0 : A \leftarrow \bar{A}}$$

$$J_i = 1, K_i = 1$$

$$\underline{T_2 : A \leftarrow A \oplus B}$$

$$A_i B_i | A_i | J_i K_i$$

$$00 | 0 | 0X$$

$$01 | 1 | 1X$$

$$10 | 1 | X0$$

$$11 | 0 | X1$$

$$J_i = B_i, K_i = B'_i$$

$$\underline{T_1 : A \leftarrow B}$$

$$J_i = B_i, K_i = B'_i$$

$$\underline{T_3 : A \leftarrow A \wedge B}$$

$$A_i B_i | A_i | J_i K_i$$

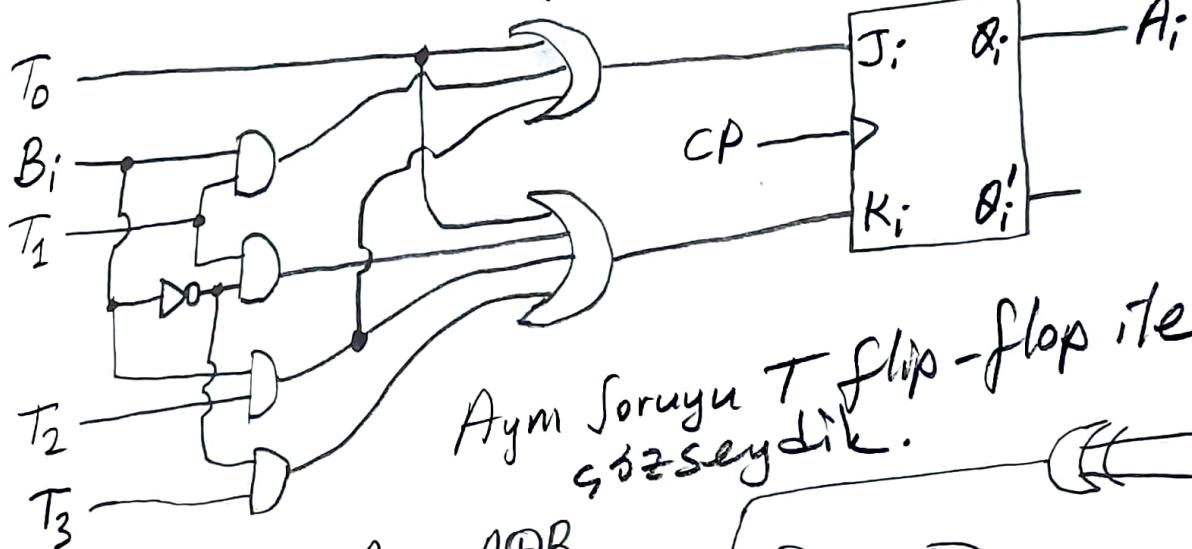
$$00 | 0 | 0X$$

$$01 | 0 | 0X$$

$$10 | 0 | X1$$

$$11 | 1 | X0$$

$$J_i = 0, K_i = B'_i$$



Aynı soruyu T flip-flop ile  
çözüseydik.

$$\underline{T_0 : A \leftarrow \bar{A}}$$

$$T_i = 1$$

$$\underline{T_1 : A \leftarrow B}$$

$$\underline{A_i B_i | A_i | T_i}$$

$$00 | 0 | 0$$

$$01 | 1 | 1$$

$$10 | 0 | 1$$

$$11 | 1 | 0$$

$$T_i = A_i \oplus B_i;$$

$$\underline{T_2 : A \leftarrow A \oplus B}$$

$$\underline{A_i B_i | A_i | T_i}$$

$$00 | 0 | 0$$

$$01 | 1 | 1$$

$$10 | 1 | 0$$

$$11 | 0 | 1$$

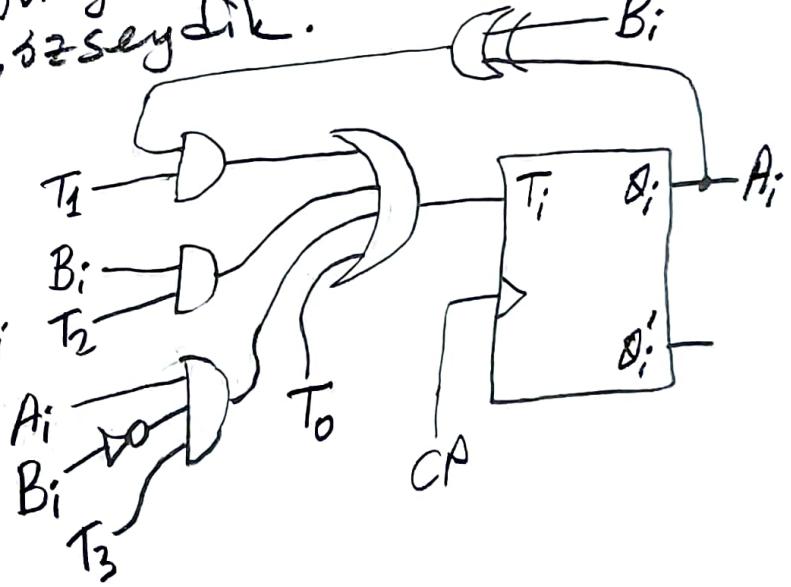
$$T_i = A_i \oplus B_i;$$

$$00 | 0 | 0$$

$$01 | 0 | 0$$

$$10 | 0 | 1$$

$$11 | 1 | 0$$



$$00 | 0 | 0$$

$$01 | 0 | 0$$

$$10 | 0 | 1$$

$$11 | 1 | 0$$

$T_0: A \leftarrow 0$	Yanda verilen işlemleri yapan devrenin (24) bir bittik kismını JK flip-flop ve en az kapi elemanı kullanarak tasarlayın.		
$T_1: A \leftarrow \bar{A}$			
$T_2: A \leftarrow AVB$			
$T_3: A \leftarrow A \oplus B$			
$JK$	$\delta_{t+1}$	$JK$	$T_0: A \leftarrow 0$
$00$	$00$	$0X$	$J_i = 0, K_i = 1$
$01$	$01$	$1X$	$T_1: A \leftarrow \bar{A}$
$10$	$10$	$X1$	$J_i = 1, K_i = 1$
$11$	$11$	$X0$	$T_2: A \leftarrow AVB$
$11$	$\delta'_t$		$A_i; B_i   A_i   J_i; K_i$
$Uyarım$			$00   0   0X   J_i = B_i$
Karakteristik Tablo			$01   1   1X   K_i = 0$
$T_3: A \leftarrow A \oplus B$			$10   1   X0  $
$A_i; B_i   A_i   J_i; K_i$			$11   1   X0  $
$00   0   0X  $			$J_i   \delta_i   A_i$
$01   1   1X  $			$K_i   \delta'_i   A'_i$
$10   1   X0  $			
$11   0   X1  $			
$J_i = B_i, K_i = B_i$			
$T_0: A \leftarrow 0$	$T_1: A \leftarrow \bar{A}$	$T_2: A \leftarrow AVB$	$T_3: A \leftarrow A \oplus B$
$A_i; B_i   A_i   T_i$	$A_i; B_i   A_i   T_i$	$A_i; B_i   A_i   T_i$	$A_i; B_i   A_i   T_i$
$00   0   0$	$00   1   1$	$00   0   0$	$00   0   0$
$01   0   0$	$01   1   1$	$01   1   1$	$01   1   1$
$10   0   1$	$10   0   1$	$10   1   0$	$10   1   0$
$11   0   1$	$11   0   1$	$11   1   0$	$11   0   1$
$T_i = A_i$	$T_i = 1$	$T_i = A'_i; B_i$	$T_i = B_i$
$B_i$			
$T_0$			
$T_1$			
$T_2$			
$T_3$			
$CP$			
$A_i$			
$A'_i$			