

Bilgisayar Mimarisi

1

Sayıların Bilgisayarda Gösterimi

① İşaretsiz Tamsayılar

n bitlik işaretsiz tamsayı için $\min = 0$, $\max = 2^n - 1$ olur.

unsigned char
8 bit (1 bayt)

$$\min = 0$$

$$\max = 2^8 - 1 = 255$$

unsigned short
16 bit (2 bayt)

$$\min = 0$$

$$\max = 2^{16} - 1 = 65,535$$

unsigned int
32 bit (4 bayt)

$$\min = 0$$

$$\max = 2^{32} - 1 = 4,294,967,295$$

unsigned long \rightarrow 64 bit (8 bayt)

$$\min = 0$$

$$\max = 2^{64} - 1 = \text{çok büyük}$$

$$\begin{array}{r} 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \\ 1101 \quad 1001 \end{array} \rightarrow 217$$

$$\begin{array}{r} 0110 \quad 1110 \end{array} \rightarrow 110$$

$$\begin{array}{r} + \\ 0100 \quad 0111 \end{array}$$

elde oluştu

$$\begin{array}{r} 71 \\ \hline \text{yanlış sonuç} \end{array}$$

Elde
iki işaretsiz tamsayı toplandığında elde oluşuyorsa anlamı sonuç n bite sığmıyor demektir. (n+1) bit gerekiyor.

$$\begin{array}{r} 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \\ 0110 \quad 1110 \end{array} \rightarrow 110$$

$$\begin{array}{r} 1101 \quad 1001 \end{array} \rightarrow 217$$

$$\begin{array}{r} - \\ 1001 \quad 0101 \end{array}$$

borç oluştu

$$\begin{array}{r} 149 \\ \hline \text{yanlış sonuç} \end{array}$$

Borç
işaretsiz bir tamsayıdan işaretsiz bir tamsayı çıkartıldığında borç oluşuyorsa anlamı çıkan çıkartılandan büyüktür.

② İşaretlili Tamsayılar

n bitlik işaretlili tamsayı için $\min = -2^{n-1}$, $\max = 2^{n-1} - 1$ olur.

char \rightarrow 8 bit (1 bayt)

$$\min = -2^7 = -128$$

$$\max = 2^7 - 1 = 127$$

short \rightarrow 16 bit (2 bayt)

$$\min = -2^{15} = -32,768$$

$$\max = 2^{15} - 1 = 32,767$$

long
64 bit (8 bayt)

$$\min = -2^{63}$$

$$\max = 2^{63} - 1$$

int \rightarrow 32 bit (4 bayt)

$$\min = -2^{31}$$

$$= -2,147,483,648$$

$$\max = 2^{31} - 1$$

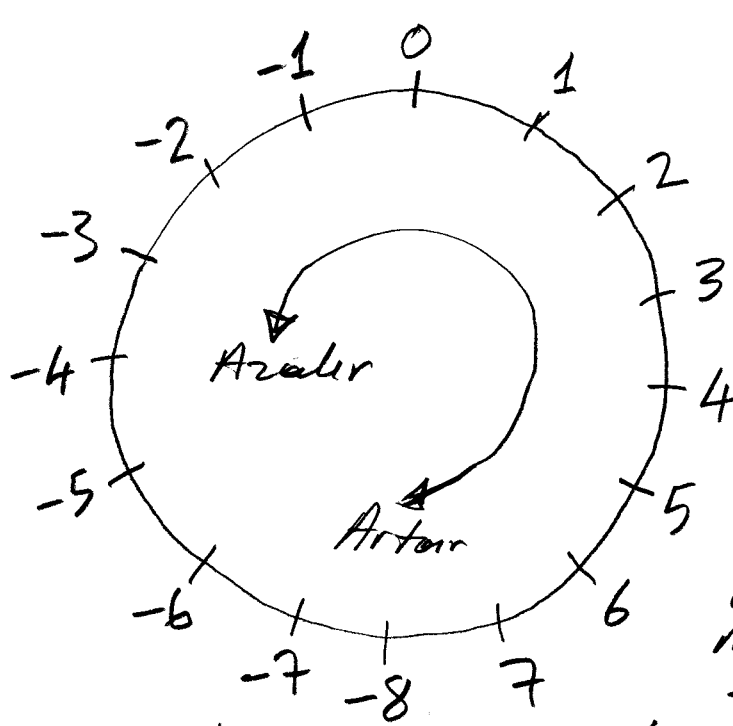
$$= 2,147,483,647$$

Bilgisayarda işaretlili tamsayılar için tümleyen aritmetisi kullanılır.

min işaretlili tamsayının tümleyeni max işaretlili tamsayıdır.

max işaretlili tamsayının tümleyeni min işaretlili tamsayıdır.

0000	→ 0
0001	→ 1
0010	→ 2
0011	→ 3
0100	→ 4
0101	→ 5
0110	→ 6
0111	→ 7
1000	→ -8
1001	→ -7
1010	→ -6
1011	→ -5
1100	→ -4
1101	→ -3
1110	→ -2
1111	→ -1



Max işaretli
tamsayı 1
artırılırsa
min işaretli
tamsayı olur.

Min işaretli
tamsayı 1
azaltılırsa
max işaretli
tamsayı olur.

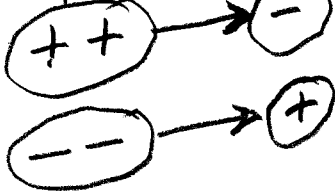
En soldaki bit işaret bitidir.

0 ise tamsayı pozitifdir
1 ise tamsayı negatiftir.

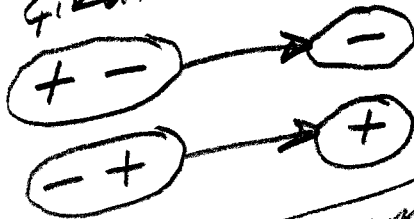
0101 $\xrightarrow{\text{tümleyen}}$ 1010 $\xrightarrow{\text{1 ekle}}$ 1011
(+5) (-6) (-5)

Taşma
işaretti tamsayıların toplanmasından veya
çıkartılmasından oluşur. Sonunun ayrılan bit
ile ifade edilemeyeceğini gösterir.

Toplama



Çıkarma



Bu durumlarda
taşma
oluşur.

Diger durumlarda taşma oluşmaz.

0 için toplama
 $S = A + B$

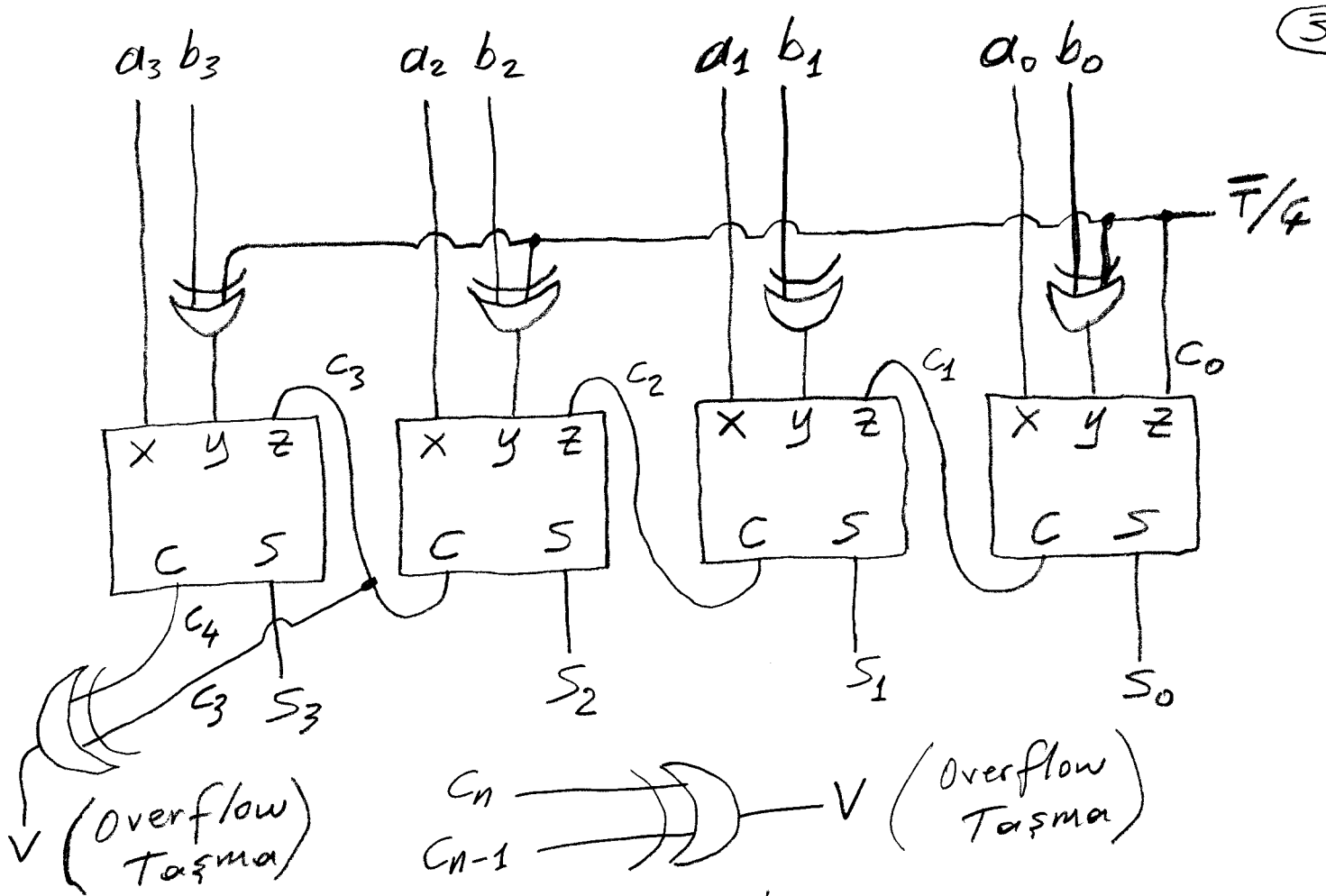
$\overline{T}/4$

1 için çıkarma
 $S = A + \overline{B} + 1$
 $= A - B$

$$A = a_3 a_2 a_1 a_0, n=4$$

$$A = a_0 + 2a_1 + 4a_2 - 8a_3$$

$$\text{işaretti Tamsayı} = -2^{n-1} a_{n-1} + \sum_{k=0}^{n-2} 2^k \cdot a_k$$



$$\begin{aligned} V &= C_3 \oplus C_4 \\ &= 1 \oplus 0 \\ &= 1 \text{ Torsion Var} \end{aligned}$$

7'den büyük
7 sıklık
 $V = C_3 \oplus C_4$
 $= 1 \oplus 0$
 $= 1$ Tamsama
var

8 bitlik işaretli tamsayı, 16 bitlik işaretli tamsayıya göre
 gösterilirse $0000\ 0101\ \underline{1011}$

8 bitlik isaretni tamam
 10101 1011 → 0000 0000 0101 1011
 5 B 0 0 5 B
 0x5B → 0x005B = +91

$$\begin{array}{ccccccc} \overline{1010} & \overline{0101} & \xrightarrow{0x5B} & \overline{1111} & \overline{1111} & \overline{1010} & \overline{0101} \\ A & 5 & & F & F & A & 5 \end{array}$$

$$0x A5 \xrightarrow{A} 0xFFA5 = -91$$

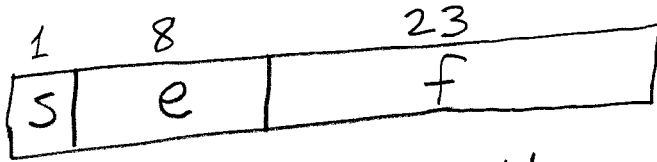
③ Reel Sayılar

④

$$572\ 480\ 000 = 5.7248 \times 10^8$$

$$0.000\ 057\ 248 = 5.7248 \times 10^{-5}$$

float 32 bit (4 bayt)



$$N = (-1)^s \times (1.f) \times 2^{e-127}$$

$$e = 255 \begin{cases} \rightarrow f = 0 \text{ ise } N = \mp \infty \\ \rightarrow f \neq 0 \text{ ise } \%, \frac{\infty}{\infty}, 0 \times \infty \text{ Nan (Sayı değil)} \end{cases}$$

$$e = 0 \begin{cases} \rightarrow f = 0 \text{ ise } N = \mp 0 \\ \rightarrow f \neq 0 \text{ ise Normalize olmayan sayı} \end{cases}$$

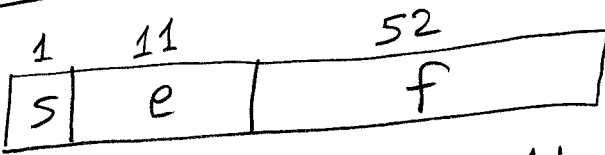
$$s = 0, e = 1, f = 0 \text{ ise } N = +2^{-126} \text{ Sıfıra yakın en küçük pozitif sayı}$$

$$s = 0, e = 254, f = (\underbrace{11 \dots 1}_{23 \text{ tane}})_2 \text{ ise}$$

$$N = + (1.11 \dots 1)_2 \times 2^{127} = (2 - 2^{-23}) \times 2^{127}$$

$$= +2^{128} \times (1 - 2^{-24}) \text{ En büyük pozitif sayı}$$

double 64 bit (8 bayt)



$$N = (-1)^s \times (1.f) \times 2^{e-1023}$$

$$e = 2047 \begin{cases} \rightarrow f = 0 \text{ ise } N = \mp \infty \\ \rightarrow f \neq 0 \text{ ise } \%, \frac{\infty}{\infty}, 0 \times \infty \text{ Nan (Sayı değil)} \end{cases}$$

$$e = 0 \begin{cases} \rightarrow f = 0 \text{ ise } N = \mp 0 \\ \rightarrow f \neq 0 \text{ ise Normalize olmayan sayı} \end{cases}$$

$$s = 0, e = 1, f = 0 \text{ ise } N = +2^{-1022} \text{ Sıfıra yakın en küçük pozitif sayı}$$

$$s = 0, e = 2046, f = (\underbrace{11 \dots 1}_{52 \text{ tane}})_2 \text{ ise}$$

$$N = + (1.11 \dots 1)_2 \times 2^{1023} = (2 - 2^{-52}) \times 2^{1023}$$

$$= +2^{1024} (1 - 2^{-53}) \text{ En büyük pozitif sayı}$$

$(89.6875)_{10}$ reel sayısını hem float hem de double olarak gösterelim. (5)

89	
44	1
22	0
11	0
5	1
2	1
1	0
0	1

	0.6875
1	.375
0	.75
1	.5
1	.0

$$(89.6875)_{10} = (1011001.1011)_2$$

$$= 1.0110011011 \times 2^6$$

float için

$$s=0, e=6+127=133 = (\overbrace{10000101}^{8 \text{ tane}})_2$$

$$f = (\underbrace{01100110110 \dots 0}_{23 \text{ tane}})_2$$

double için

$$s=0, e=6+1023=1029$$

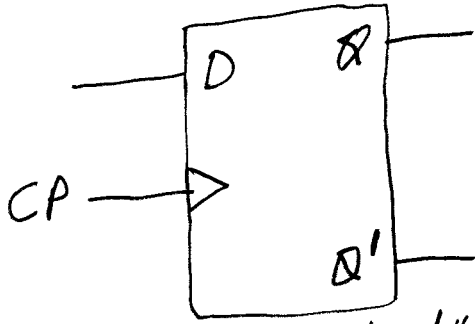
$$= (\underbrace{10000000101}_{11 \text{ tane}})_2$$

$$f = (\underbrace{01100110110 \dots 0}_{52 \text{ tane}})_2$$

Flip-Flop'lar (Yaz-Bozlar)

6

D flip-flop



Grafik Sembolü

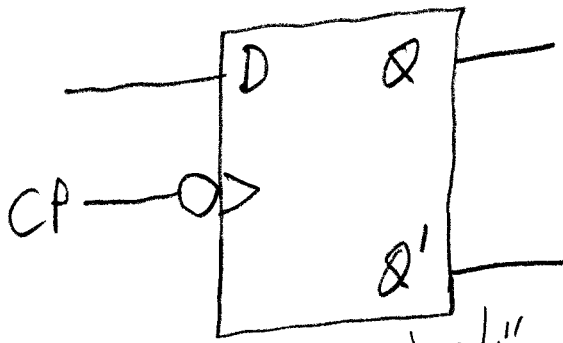
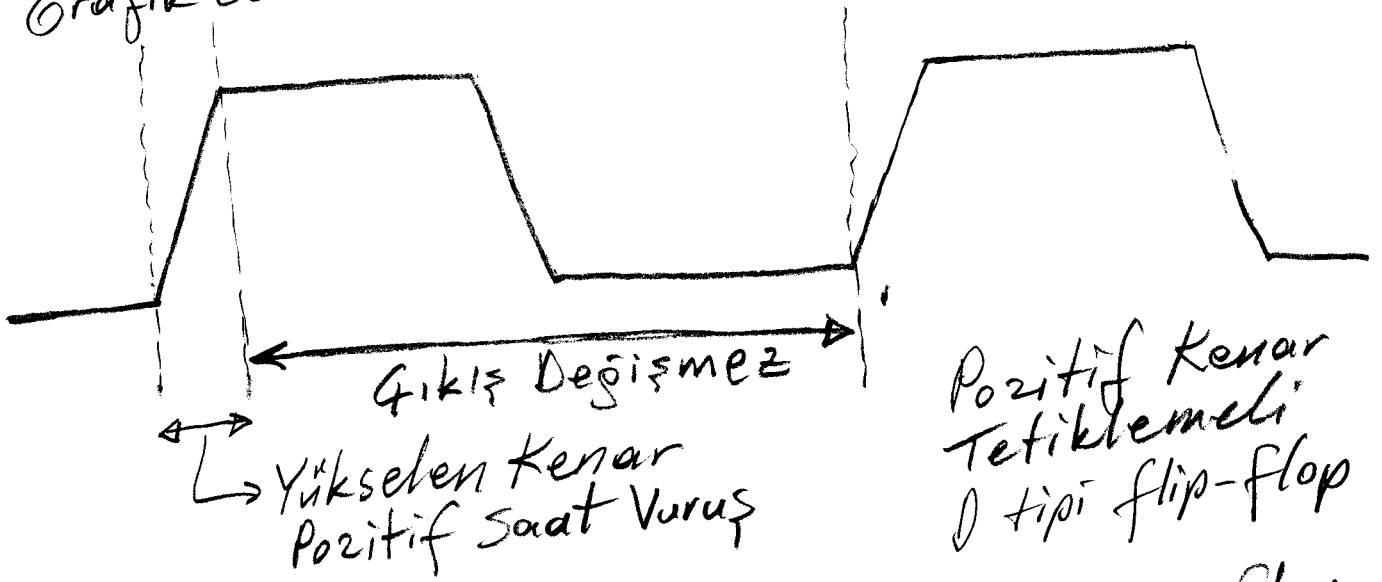
$$Q_{t+1} = D$$

D	Q_{t+1}
0	0
1	1

Karakteristik Tablosu

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

Yarım Tablosu

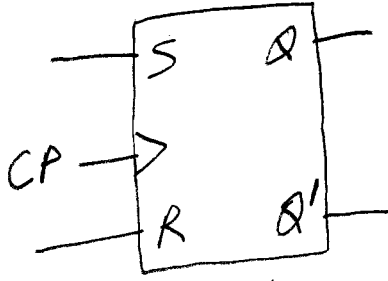


Grafik Sembolü

Negatif Kenar Tetiklemeli D tipi flip-flop Aynı işi yapar. Sadece negatif saat vuruşunda durum değiştirir.



RS flip-flop



S	R	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Kullanılmaz

Karakteristik Tablosu

Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

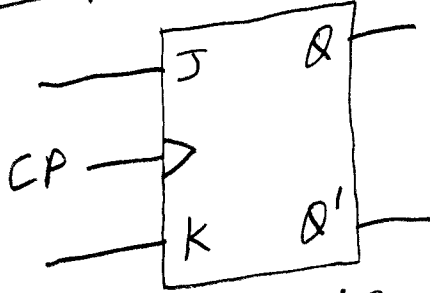
Uyarım Tablosu

Q_t	Q_{t+1}	0	1
0	0	X	1
0	1	0	1
1	0	1	0
1	1	X	0

$$Q_{t+1} = S + R'Q_t$$

SR = 0 olmalı

JK flip-flop



J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q_t'

Karakteristik Tablosu

Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

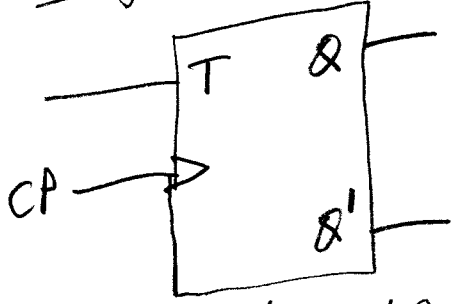
Uyarım Tablosu

Q_t	Q_{t+1}	0	1
0	0	X	1
0	1	1	X
1	0	X	1
1	1	1	X

$$Q_{t+1} = JQ_t' + K'Q_t$$

Karakteristik Denklem

T flip-flop



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

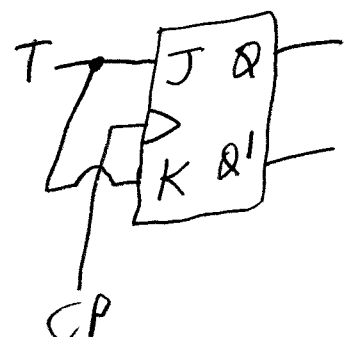
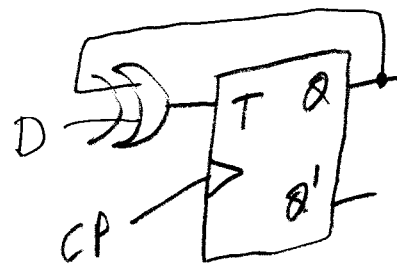
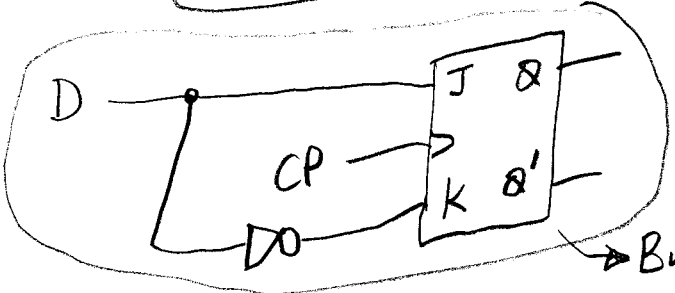
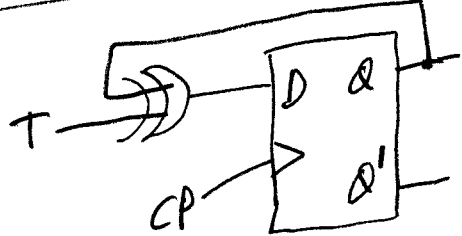
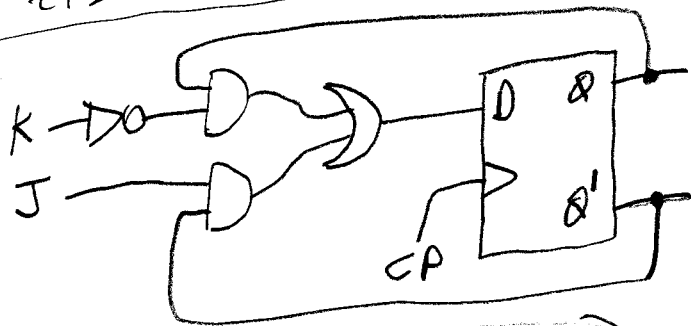
Karakteristik Tablo

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Uyarım Tablosu

Q_t	Q_{t+1}	0	1
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	1

$$Q_{t+1} = TQ_t' + T'Q_t \rightarrow \text{Karakteristik Denklemi}$$



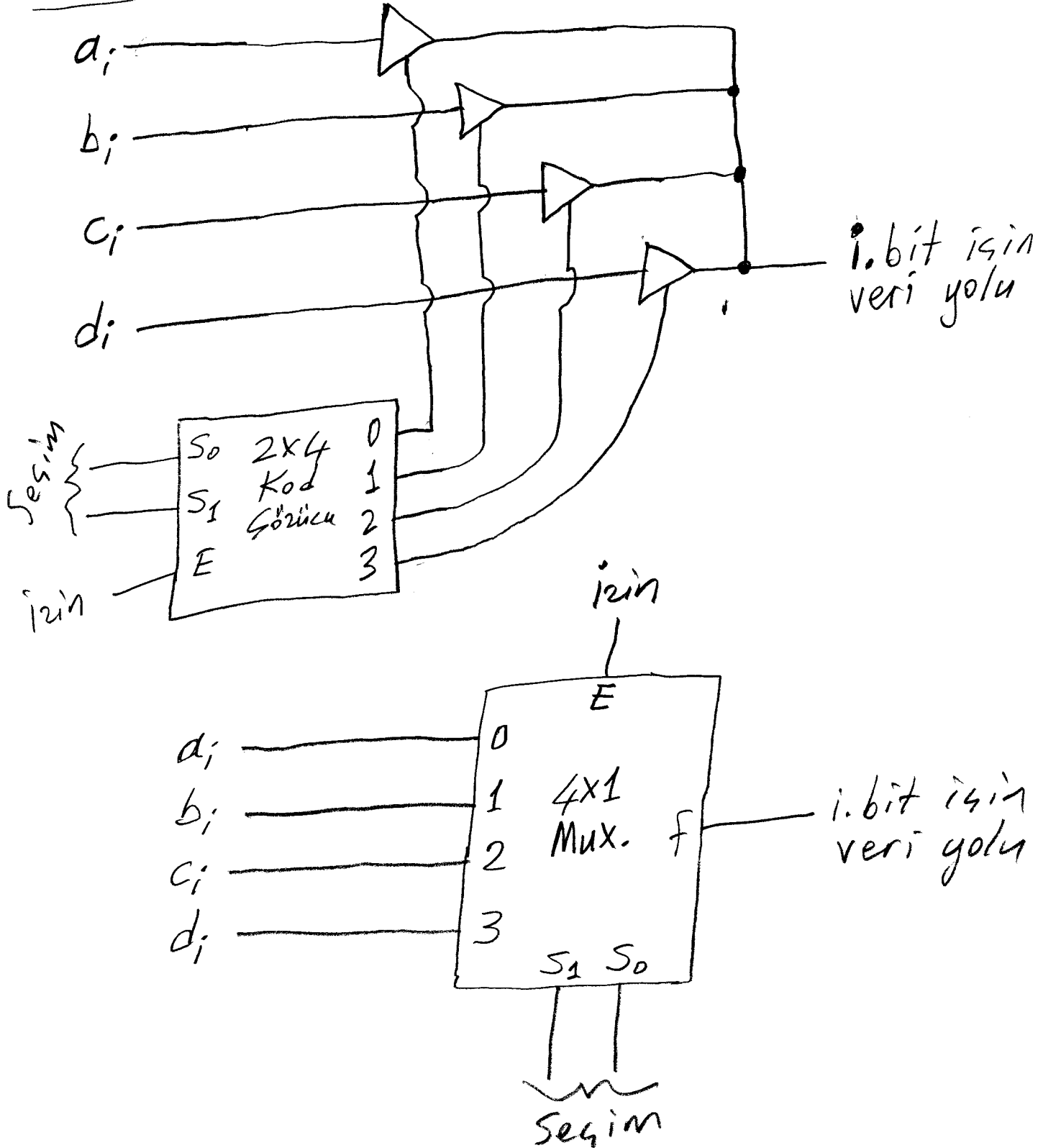
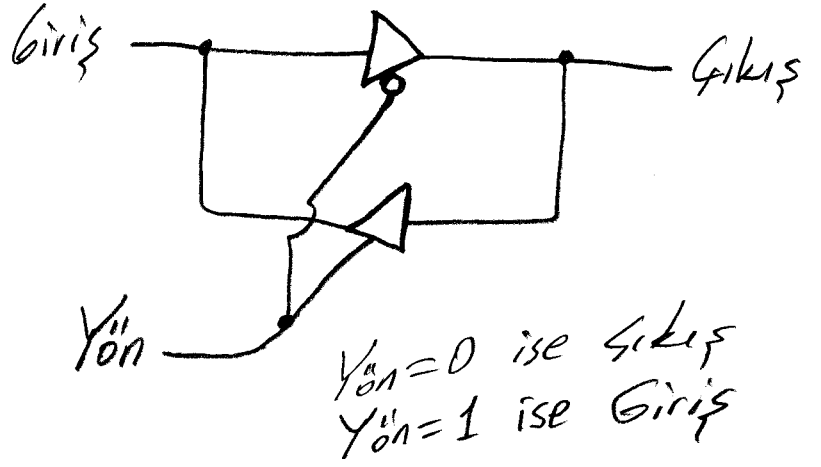
Burada JK flip-flop yerine RS flip-flop kullanılabilir.

"Üç Durumlu Eleman"

(8)



Kontrol	Çıkış
0	Yüksek Empedans
1	Giriş



Mikroişlemci Komutları

9

① Veri Transfer Komutları

Verilerin ana bellek ile kaydediciler arasında içeriği değişmeden transferini sağlar

Ld (Load - Yükle) Push (Push - Yığma koy)
St (Store - Depola) Pop (Pop - Yığından al)
Mov (Move - Taşı) In (Input - Giriş)
Exc (Exchange - Yer değiştir) Out (Output - Çıkış)

② Veri İşleme Komutları

Verilerin içeriğini değiştiren her türlü işlemi kapsar.

a) Aritmetik Komutlar (Add, Sub, Mul, Div, Inc, Neg)
b) Lojik Komutlar (And, Or, Com, Xor)
c) Kaydırma Komutları (Shr, Shl, Asr, Asl, Ror, Rol)

③ Program Kontrol Komutları

Br (Branch - Dallar) Call (Call - Çağır)
Jmp (Jump - Atla) Ret (Return - Dön)

RTI (Return from Interrupt)
RTS (Return from Subroutine)

Subroutine (Altprogram)

Belli bir hesapsal işi yapmaya yarayan program parçasıdır.

$M[SP] \leftarrow PC$
 $SP \leftarrow SP + 1$
 $PC \leftarrow \text{Altprogramın Adresi}$

Call
Altprograma
Gidiş

$SP \leftarrow SP - 1$
 $PC \leftarrow M[SP]$

Return
Altprogramdan
Dönüş

PC : Program Counter

SP : Stack Pointer

PSW : Processor Status Word

Interrupt (Kesinti)

İsten veya dıştan gelen bir istek sonucu, normal program akışının diğer bir servis rutine aktarılmasıdır. Program, servis rutin bittikten sonra normal rutine geri döner.

Kesinti Rutinine Gidiş

- * Koşturulan Komutu Bitir.
- * PC ve PSW'yi yığına aktar.
- * Kesinti rutinine git.

Kesinti Rutininden Dönüş

- * RTI komutunu koştur.
- * PC ve PSW'yi yığından al.
- * Ana programa kaldığı yerden devam et.

Kesinti ile Altprogram arasındaki farklar

- * Kesinti isten veya dıştan bir işaretle başlar
- * Altprogram çağırma komutuyla başlar.
- * Kesinti rutin adresleri daha önceden belirlenmiş adreslerdir.
- * Altprogram adreslerini programcı belirler.
- * Kesinti PC değerini ve CPU'nun tüm durumunu gösteren bilgileri saklar
- * Altprogram sadece PC değerini saklar.

Mimari

CISC (Complex Instruction Set Computer)
Kompleks Komut Kümeli Bilgisayar Mimarişi

RISC (Reduced Instruction Set Computer)
Azaltılmış Komut Kümeli Bilgisayar Mimarişi

CISC Mimarinin Temel Özellikleri

(11)

- * Çok sayıda komut (100 ile 250 arası)
- * Az sıklıkla kullanılan özel amaçlı komutlar
- * Çok çeşitli adresleme modları (5 ile 20 arası)
- * Değişken uzunluklu komut formatları
- * Ana bellek referanslı komut formatları

RISC Mimarinin Temel Özellikleri

- * Daha az sayıda komut
- * Daha az adresleme modu
- * İşlemlerin kaydedicilerde yapılması
Tek kaydedici varsa buna işlemci yazıcı adı verilir.
- * Ana bellek erişiminin Load ve Store komutları ile gerçekleştirilmesi
- * Sabit uzunluklu, kolay işlenebilen komut formatları
- * Tek farzda komut koşturulması kavramı
- * Yazılımla kontrol yerine donanımla kontrol.

$$F \leftarrow (A+B) * (C+D)$$

CISC Mimariyle

Mov R1, A ($R1 \leftarrow M[A]$)
Add R1, B ($R1 \leftarrow R1 + M[B]$)
Mov R2, C ($R2 \leftarrow M[C]$)
Add R2, D ($R2 \leftarrow R2 + M[D]$)
Mul R1, R2 ($R1 \leftarrow R1 * R2$)
Mov F, R1 ($M[F] \leftarrow R1$)

RISC Mimariyle

Load R1, A ($R1 \leftarrow M[A]$)
Load R2, B ($R2 \leftarrow M[B]$)
Add R1, R2 ($R1 \leftarrow R1 + R2$)
Load R2, C ($R2 \leftarrow M[C]$)
Load R3, D ($R3 \leftarrow M[D]$)
Add R2, R3 ($R2 \leftarrow R2 + R3$)
Mul R1, R2 ($R1 \leftarrow R1 * R2$)
Store F, R1 ($M[F] \leftarrow R1$)

$$F \leftarrow (A+B)/C - D * E \text{ veriliyor.}$$

(12)

Sıfır Adres
Buyruklu Komut
Kümesi olan
bir dil ile

Push A
Push B
Add
Push C
Div
Push D
Push E
Mul
Sub
Pop F

Bir Adres
Buyruklu Komut
Kümesi olan
bir dil ile

Load D ($Acc \leftarrow M[0]$)
Mul E ($Acc \leftarrow Acc * M[E]$)
Store F ($M[F] \leftarrow Acc$)
Load A ($Acc \leftarrow M[A]$)
Add B ($Acc \leftarrow Acc + M[B]$)
Div C ($Acc \leftarrow Acc / M[C]$)
Sub F ($Acc \leftarrow Acc - M[F]$)
Store F ($M[F] \leftarrow Acc$)

1 tane Register (Kaydedici)
vardır. Accumulator (Birikeç)
kısaca Acc yani işlemci yazıcı diye
adlandırılır.

iki adres buyruklu komut kümesi olan bir dille

(CISC Mimari)

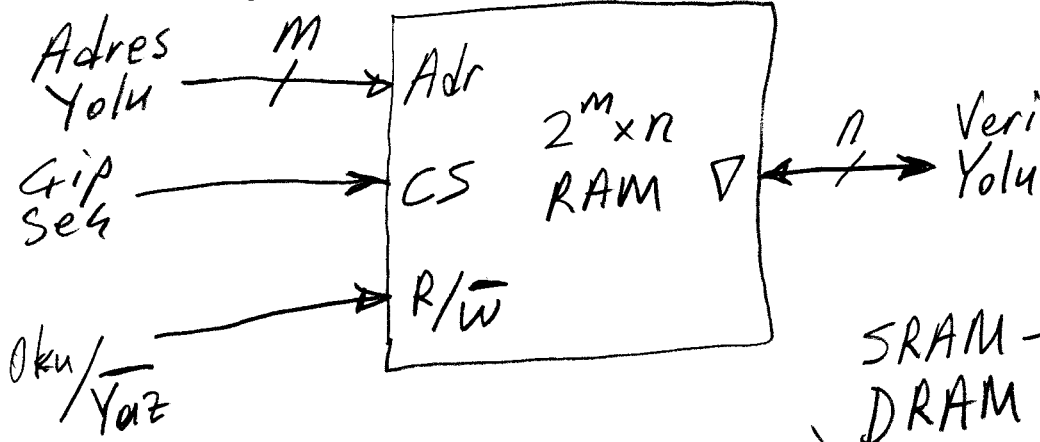
Mov R1, A ($R1 \leftarrow M[A]$)
Add R1, B ($R1 \leftarrow R1 + M[B]$)
Div R1, C ($R1 \leftarrow R1 / M[C]$)
Mov R2, D ($R2 \leftarrow M[D]$)
Mul R2, E ($R2 \leftarrow R2 * M[E]$)
Sub R1, R2 ($R1 \leftarrow R1 - R2$)
Mov F, R1 ($M[F] \leftarrow R1$)

RISC Mimari

Load R1, A ($R1 \leftarrow M[A]$)
Load R2, B ($R2 \leftarrow M[B]$)
Add R1, R2 ($R1 \leftarrow R1 + R2$)
Load R2, C ($R2 \leftarrow M[C]$)
Div R1, R2 ($R1 \leftarrow R1 / R2$)
Load R2, D ($R2 \leftarrow M[D]$)
Load R3, E ($R3 \leftarrow M[E]$)
Mul R2, R3 ($R2 \leftarrow R2 * R3$)
Sub R1, R2 ($R1 \leftarrow R1 - R2$)
Store F, R1 ($M[F] \leftarrow R1$)

RAM (Random Access Memory)

Rastgele Erişimli Bellek

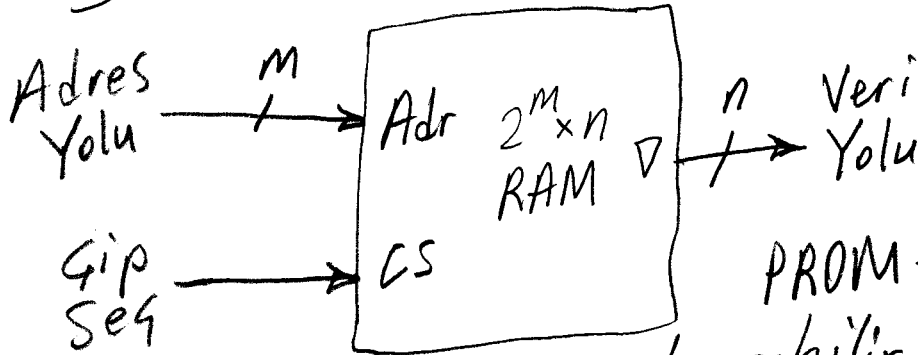


CS	R/ $\overline{\text{W}}$	İşlem
0	X	Yük. Emp
1	0	Yaz
1	1	Oku

SRAM - Statik RAM
DRAM - Dinamik RAM

ROM (Read Only Memory)

Sadece Okunabilir Bellek



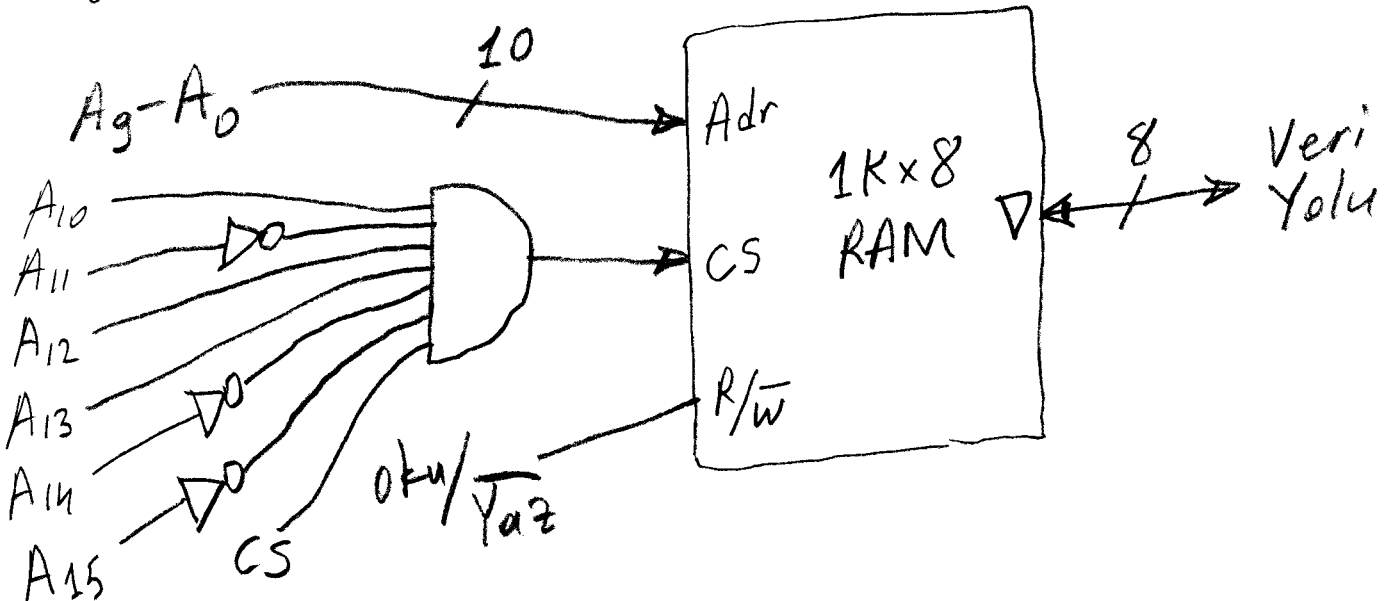
CS	İşlem
0	Yüksek Empedans
1	Oku

PROM - Programlanabilir ROM

EPROM - Silinip Programlanabilir ROM
EEPROM - Elektriksel olarak Silinip Programlanabilir ROM

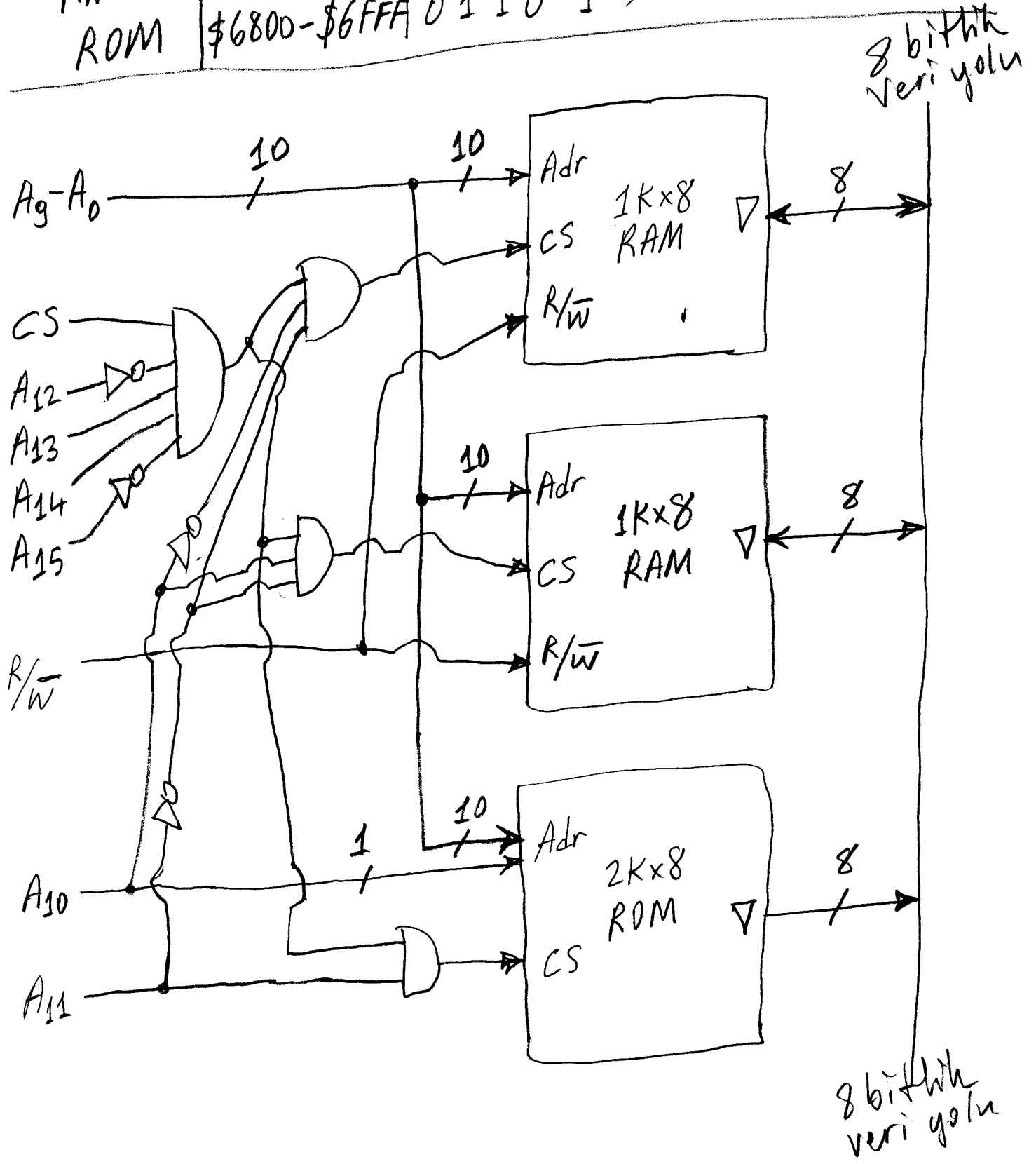
1Kx8 bitlik RAM, Adresi = \$3400 - \$37FF

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



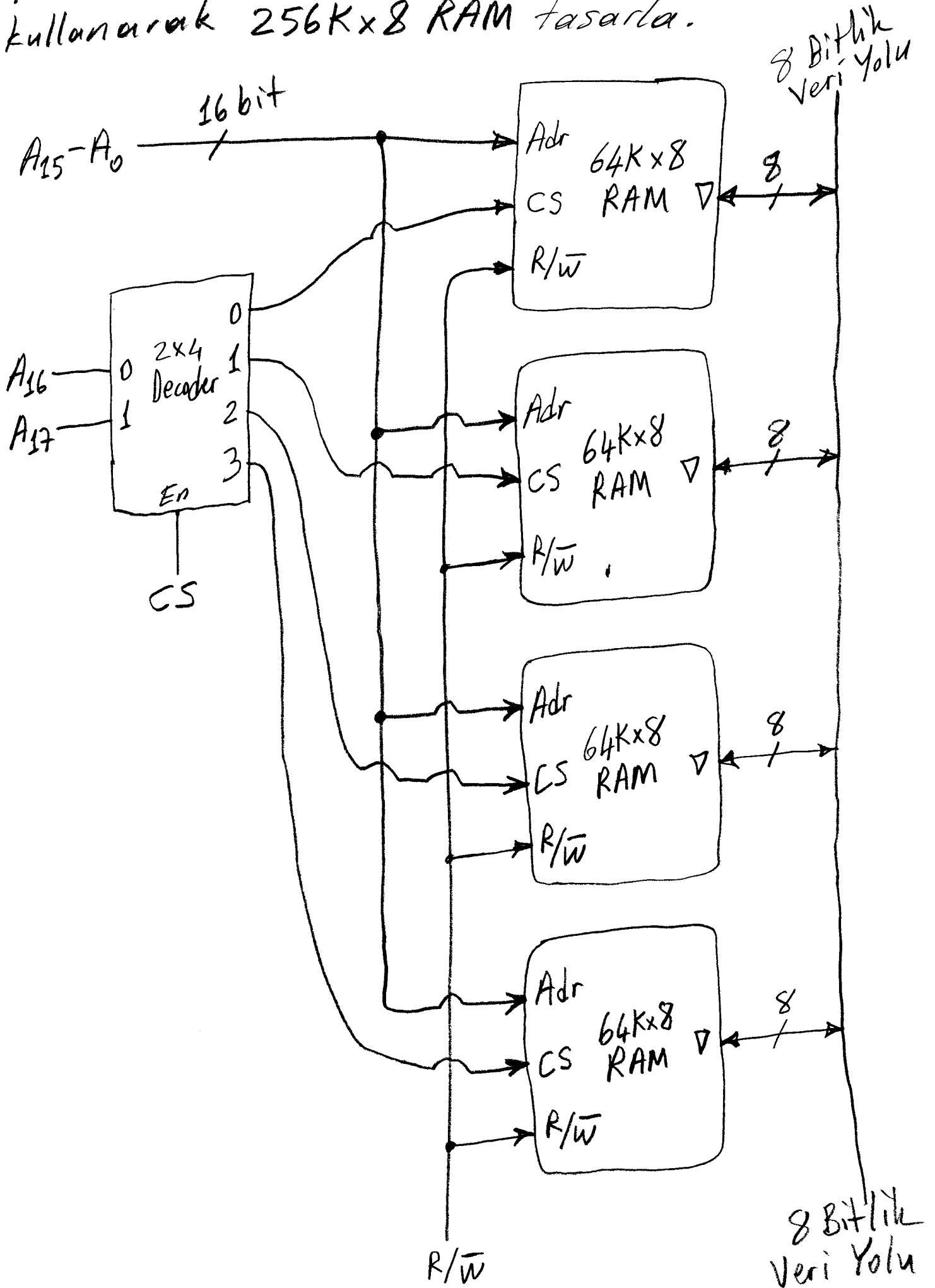
16 bitlik adres yolu olan bir hafıza biriminin
 $\$6000 - \$6FFF$ adres bölgesine 2 tane
 1Kx8 bitlik RAM, 1 tane 2Kx8 bitlik ROM
 yerleştir.

Bileşen	Onaltılı Adres	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$
RAM 1	$\$6000 - \$63FF$	0 1 1 0	0 0	x x x x	x x x x
RAM 2	$\$6400 - \$67FF$	0 1 1 0	0 1	x x x x	x x x x
ROM	$\$6800 - \$6FFF$	0 1 1 0	1	x x x x	x x x x



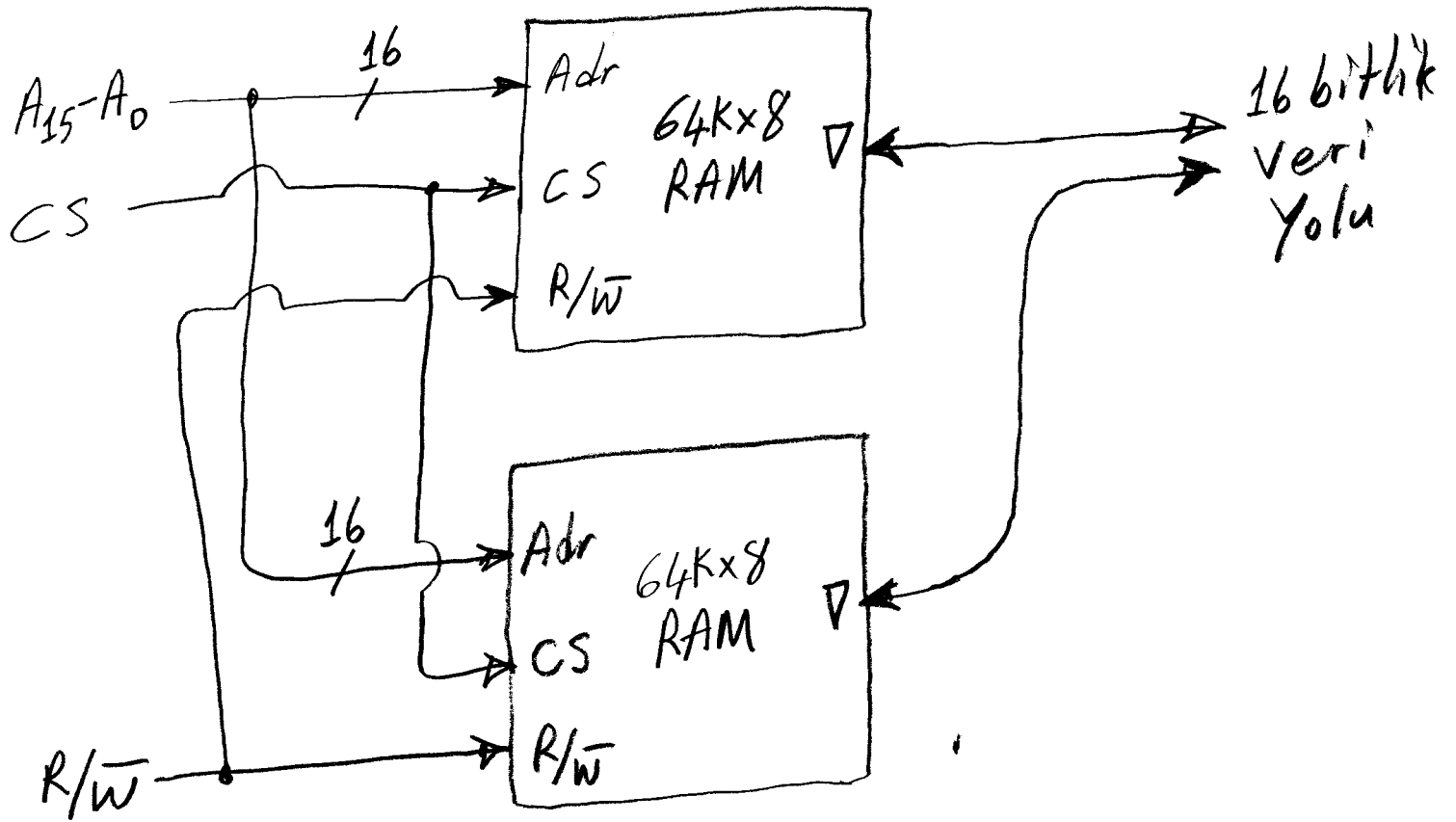
4 tane $64K \times 8$ RAM, 1 tane 2×4 Decoder kullanarak $256K \times 8$ RAM tasarla.

15

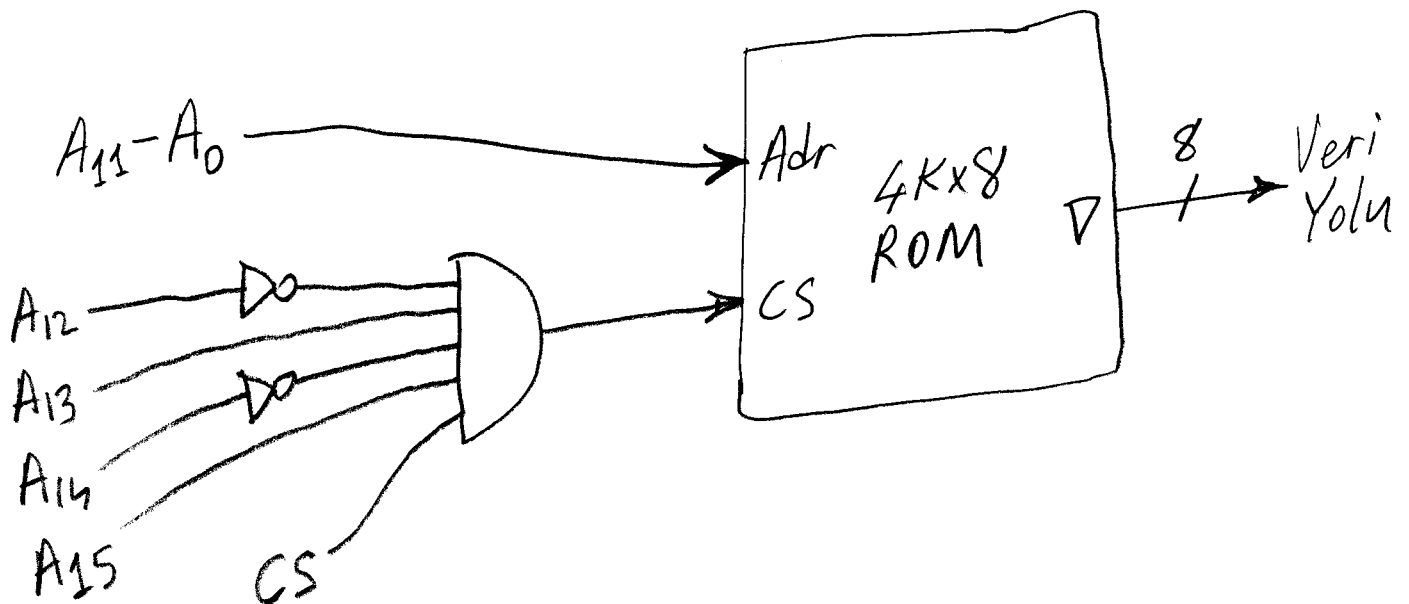


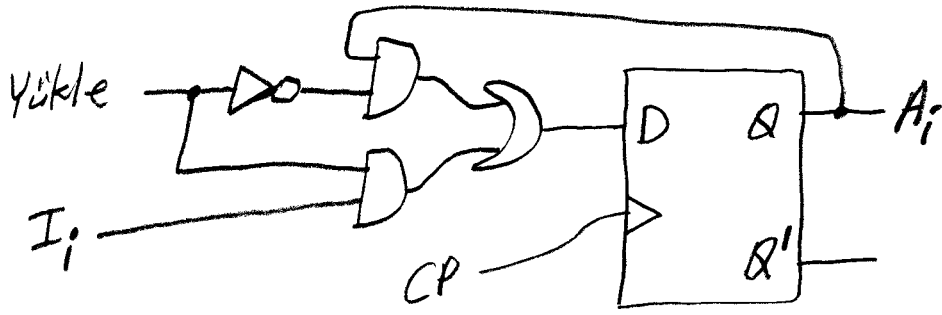
2 tane $64K \times 8$ bitlik RAM kullanarak
1 tane $64K \times 16$ bitlik RAM tasarlan

(16)

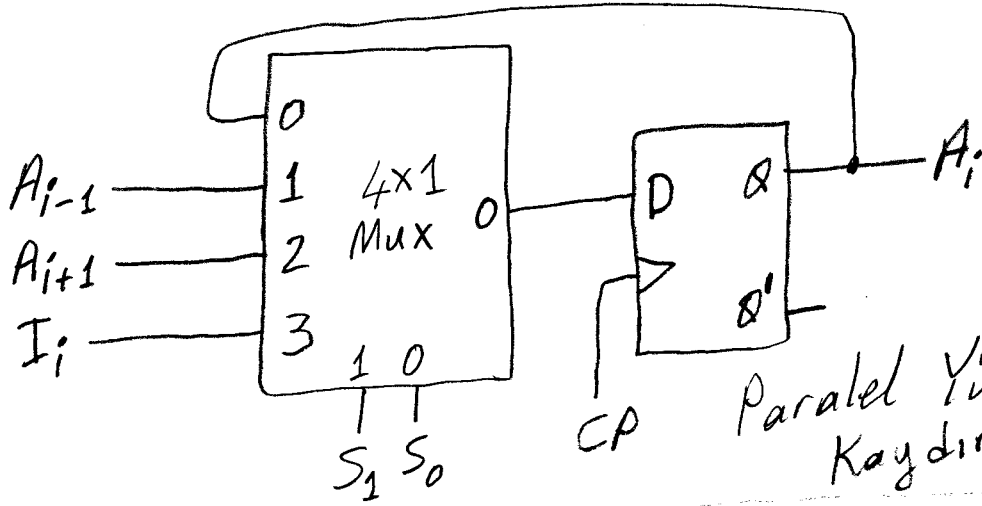


$4K \times 8$ bitlik ROM, Adresi = $\$A000 - \$FFFF$
 $A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$
 $1 \ 0 \ 1 \ 0 \ X \ X \ X \ X \ X \ X \ X \ X \ X \ X \ X \ X$





Yükle	İşlem
0	Koru
1	Yükle



S ₁ S ₀	İşlem
00	Koru
01	Saga Kaydır
10	Sola Kaydır
11	Yükle

Paralel Yükleme Gifit Yönlü Kaydırma Yazacı

- $T_0: A \leftarrow 0$
- $T_1: A \leftarrow \bar{A}$
- $T_2: A \leftarrow A \vee B$
- $T_3: A \leftarrow A \oplus B$

Yanda verilen işlemleri yapan devrenin bir bitlik kısmını JK flip-flop ve en az kapı elemanı kullanarak tasarlayınız.

JK	Q_{t+1}	$Q_t Q_{t+1}$	JK
00	Q_t	00	0X
01	0	01	1X
10	1	10	X1
11	Q_t'	11	X0

Karakteristik Tablosu

Uyarım Tablosu

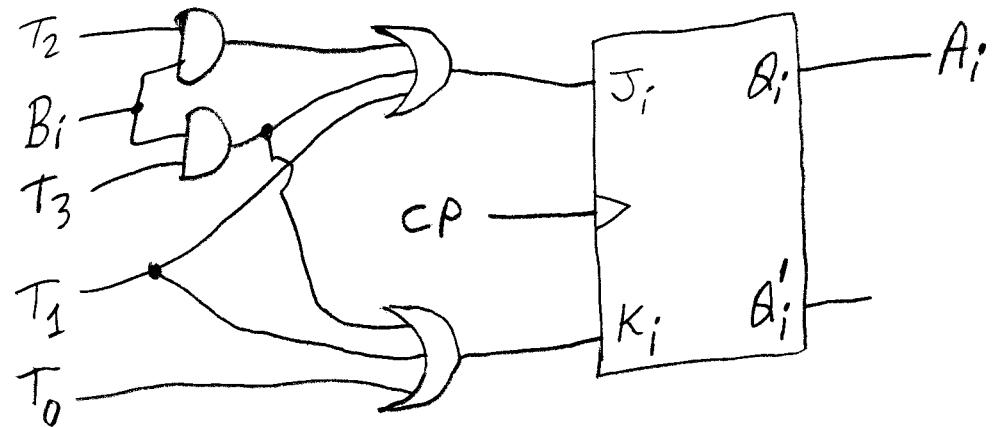
$T_2: A \leftarrow A \vee B$		
$A_i B_i$	A_i	$J_i K_i$
00	0	0X
01	1	1X
10	1	X0
11	1	X0

$J_i = B_i, K_i = 0$

$T_3: A \leftarrow A \oplus B$		
$A_i B_i$	A_i	$J_i K_i$
00	0	0X
01	1	1X
10	1	X0
11	0	X1

$J_i = B_i, K_i = B_i$

- $T_0: A \leftarrow 0$
 $J_i = 0, K_i = 1$
- $T_1: A \leftarrow \bar{A}$
 $J_i = 1, K_i = 1$

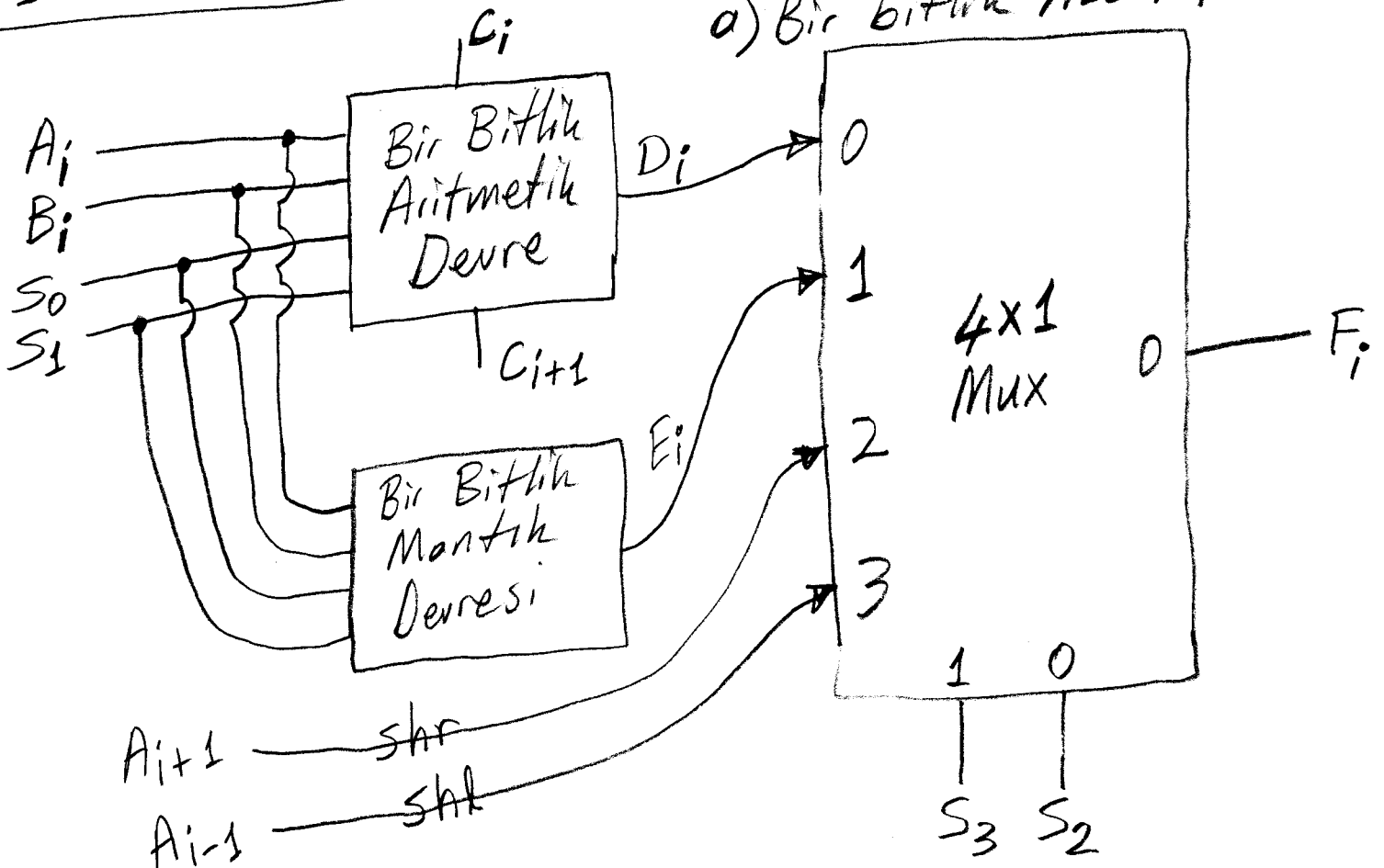


$S_3 S_2 S_1 S_0 C_{gir}$	İşlem
0 0 0 0 0	$F \leftarrow A+B$
0 0 0 0 1	$F \leftarrow A+B+1$
0 0 0 1 0	$F \leftarrow A+\bar{B}$
0 0 0 1 1	$F \leftarrow A+\bar{B}+1$
0 0 1 0 0	$F \leftarrow A$
0 0 1 0 1	$F \leftarrow A+1$
0 0 1 1 0	$F \leftarrow A-1$
0 0 1 1 1	$F \leftarrow A$
0 1 0 0 X	$F \leftarrow A \wedge B$
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	$F \leftarrow shr A$
1 1 X X X	$F \leftarrow shl A$

Fonksiyon tablası
yanda verilen
Aritmetik Mantık
Birimi (ALU)
devresinin bir bitlik
kısımını aşağıdaki
aşamaları dikkate
alarak tasarlayınız.

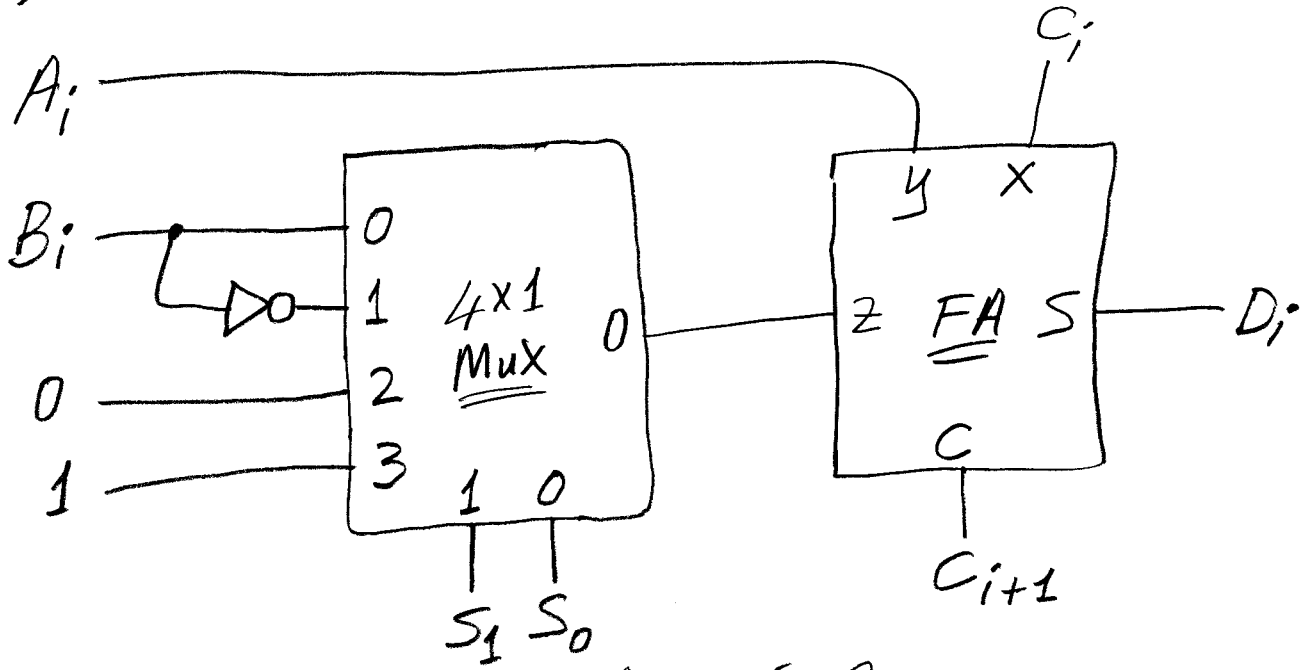
- Bir bitlik ALU aşaması
- Bir bitlik Aritmetik Devre Aşaması
- Bir bitlik Mantık Devre Aşaması

a) Bir bitlik ALU Aşaması

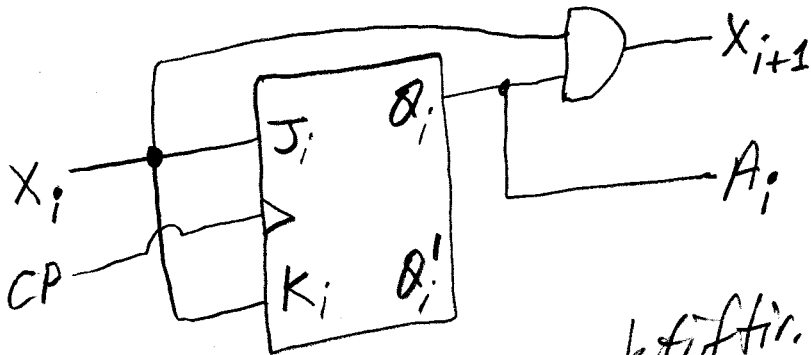
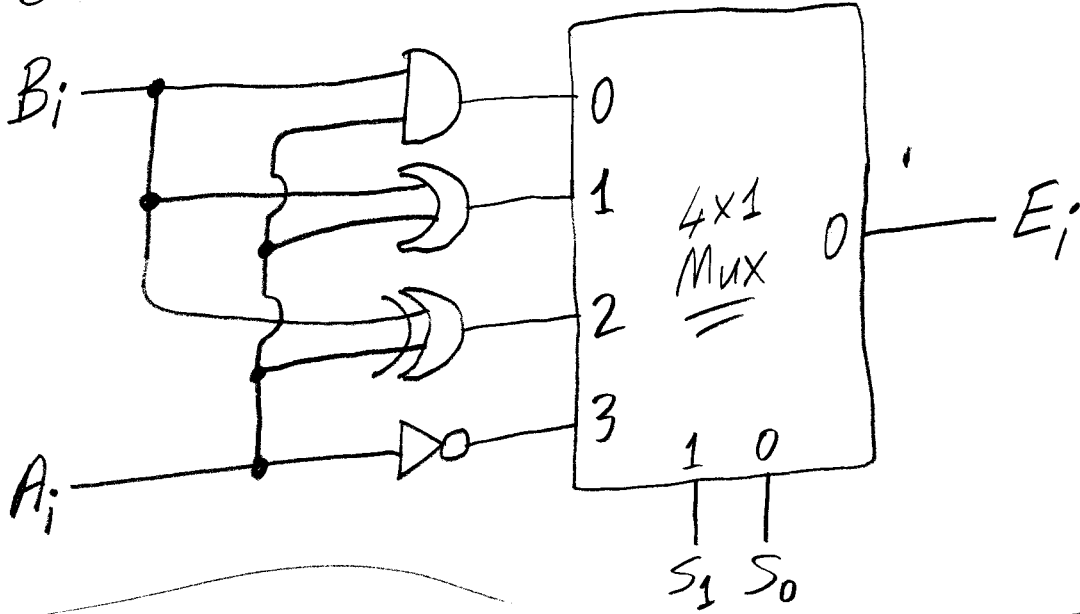


b) Bir bitlik Aritmetik Devre Aşaması

(20)



c) Bir bitlik Mantık Devresi Aşaması

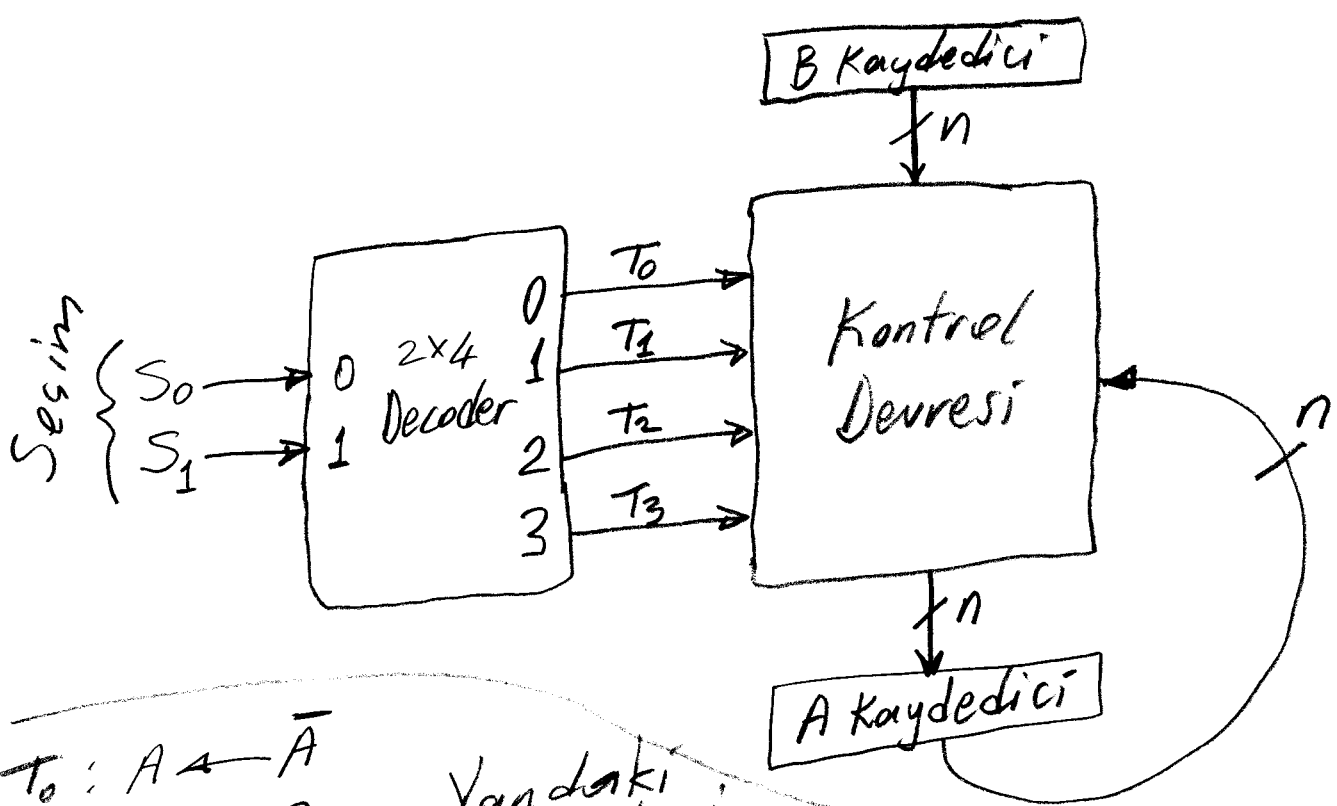


$x_0 = 1$ verilirse sayma aktiftir.

$A = A_{n-1} \dots A_2 A_1 A_0$
n bitlik sayıcı

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	1	1	1

başarıldı



$$\begin{aligned}
 T_0: A &\leftarrow \bar{A} \\
 T_1: A &\leftarrow B \\
 T_2: A &\leftarrow A \oplus B \\
 T_3: A &\leftarrow A \wedge B
 \end{aligned}$$

Yandaki işlemleri yapan devrenin tek bitlik kısmını T flip-flop kullanarak tasarla ve en az kapı elemanı kullanarak

$$T_0: A \leftarrow \bar{A} \text{ için } T_i = 1$$

$$T_1: A \leftarrow B \text{ için}$$

$A_i B_i$	A_i	T_i
00	0	0
01	1	1
10	0	1
11	1	0

$$T_i = A_i \oplus B_i$$

$$T_2: A \leftarrow A \oplus B \text{ için}$$

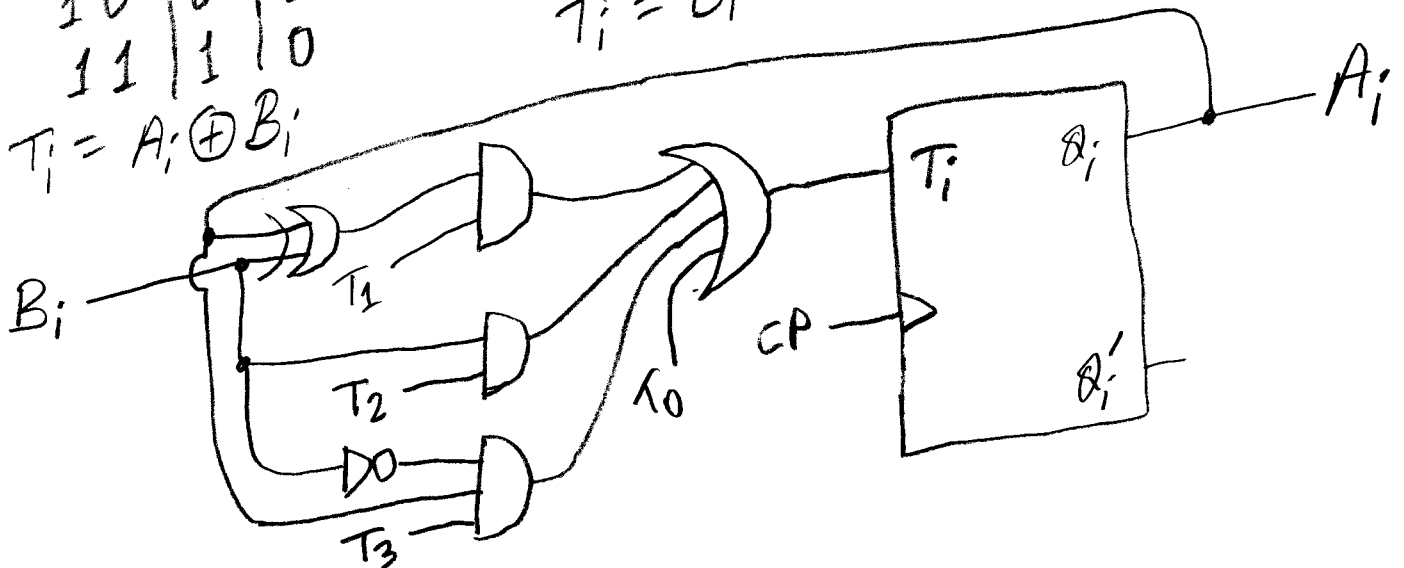
$A_i B_i$	A_i	T_i
00	0	0
01	1	1
10	1	0
11	0	1

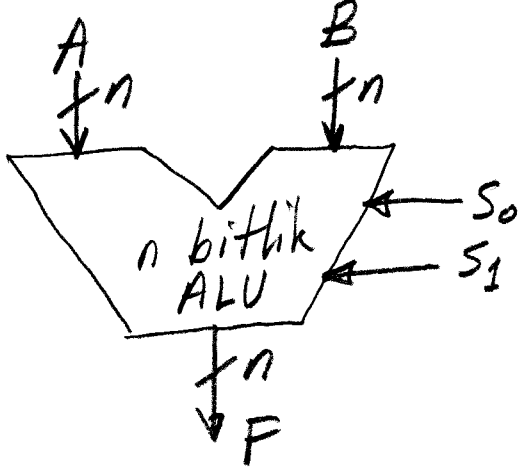
$$T_i = B_i$$

$$T_3: A \leftarrow A \wedge B \text{ için}$$

$A_i B_i$	A_i	T_i
00	0	0
01	0	0
10	0	1
11	1	0

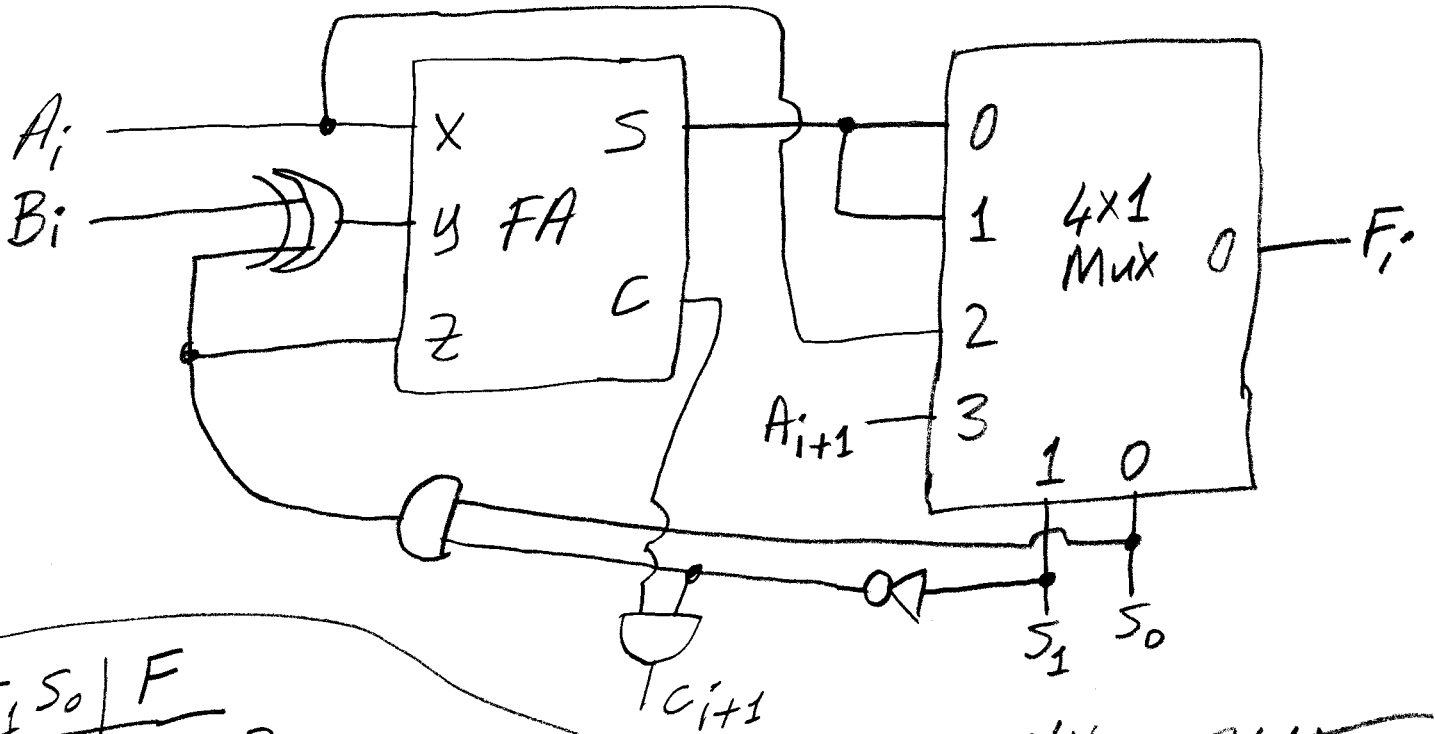
$$T_i = A_i B_i$$





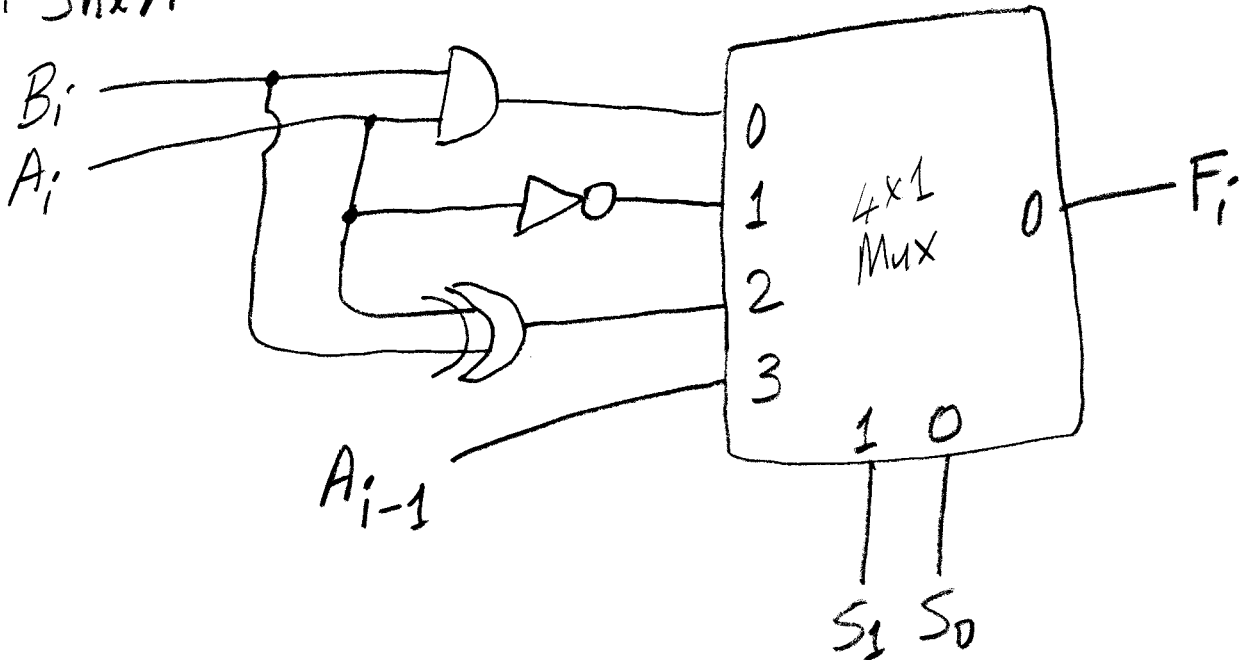
$S_1 S_0$	F
0 0	$A+B$
0 1	$A-B$
1 0	A
1 1	$shr A$

Yanda verilen
n bitlik ALU
devresinin
bir bitlik
kismını tasarla



$S_1 S_0$	F
0 0	$A \wedge B$
0 1	\bar{A}
1 0	$A \oplus B$
1 1	$shl A$

Yanda verilen n bitlik ALU
devresinin bir bitlik kismını
tasarla.



CPU (Central Processing Unit)

(23)

CPU : Merkezi İşlemci Birimi

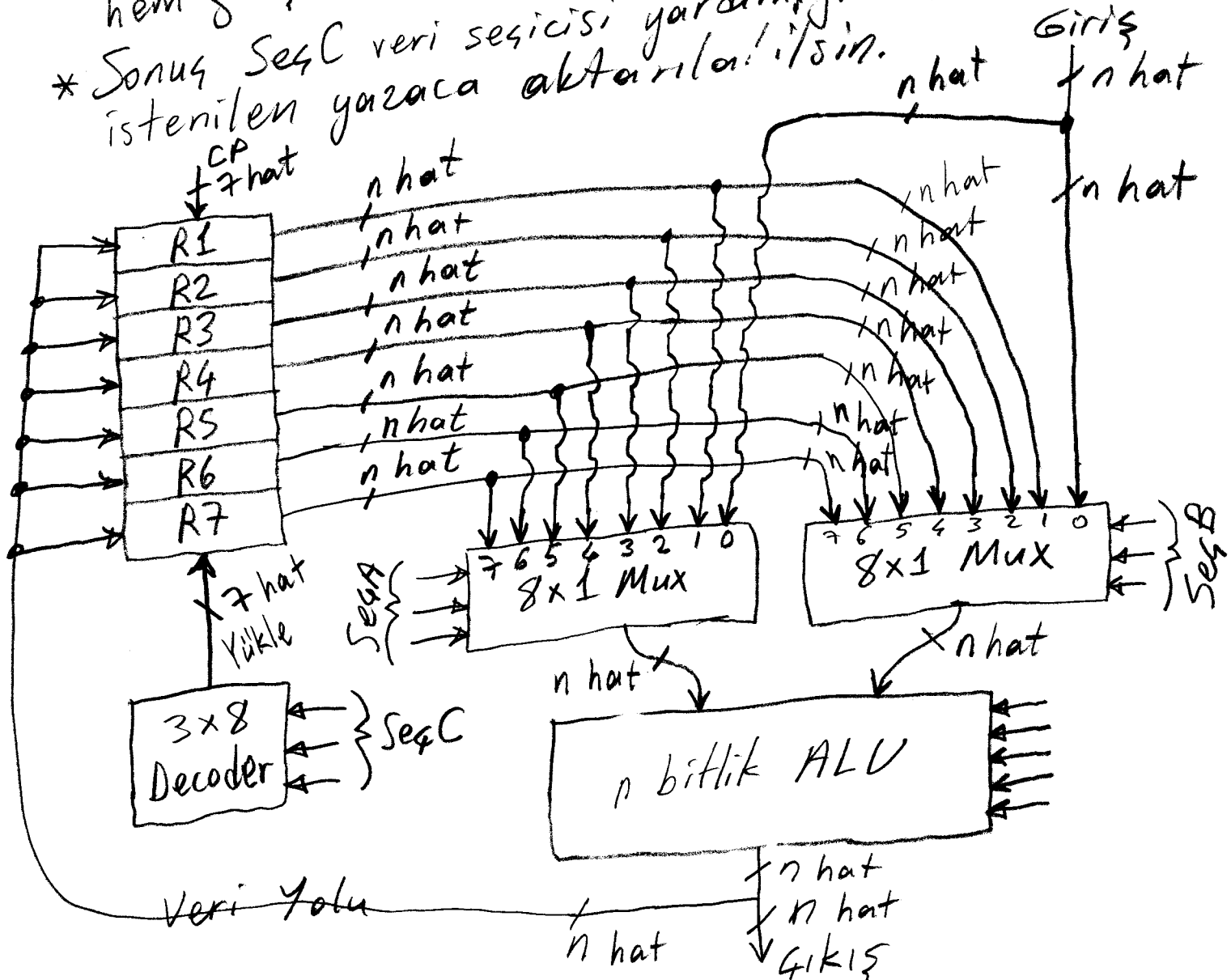
ALU : Aritmetik Mantık Birimi

Register : Yazas veya Kaydedici

Data Selector : Veri Seçici

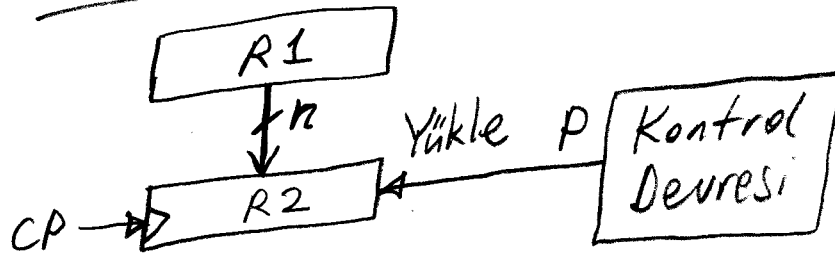
Aşağıdaki şartları sağlayan CPU devresinin blok diyagramını çiziniz.

- * CPU devresi 7 yazas, 1 ALU, 3 Seçiciden oluşsun.
- * ALU devresi 5 tane kontrol girişine sahip olsun.
- * Kontrol girişlerinin hangi işlemleri yaptığı önemsiz.
- * ALU devresi SeçA ve SeçB veri seçicileri yardımıyla hem girişten hem de yazaslardan veri alabilsin.
- * Sonuç SeçC veri seçicisi yardımıyla sıkısa veya istenilen yazaca aktarılabilir.



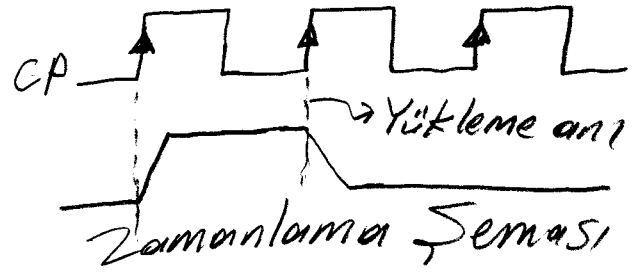
Yazma Aktarımı

(24)



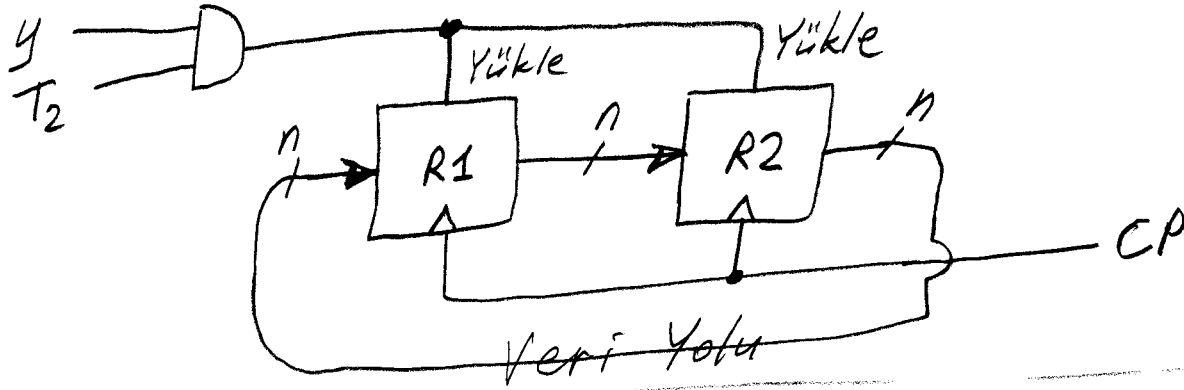
Blok Şeması

$$P: R2 \leftarrow R1$$



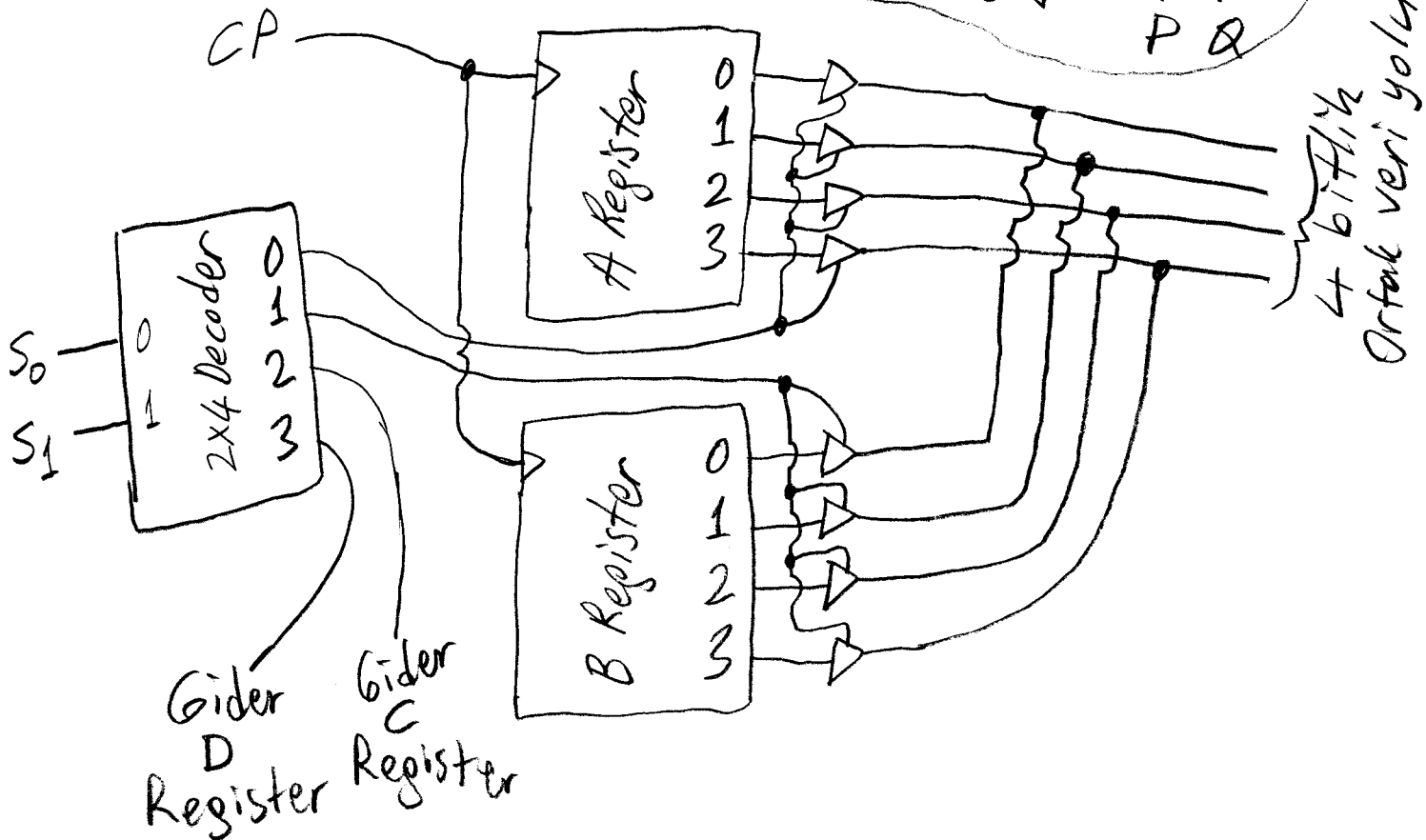
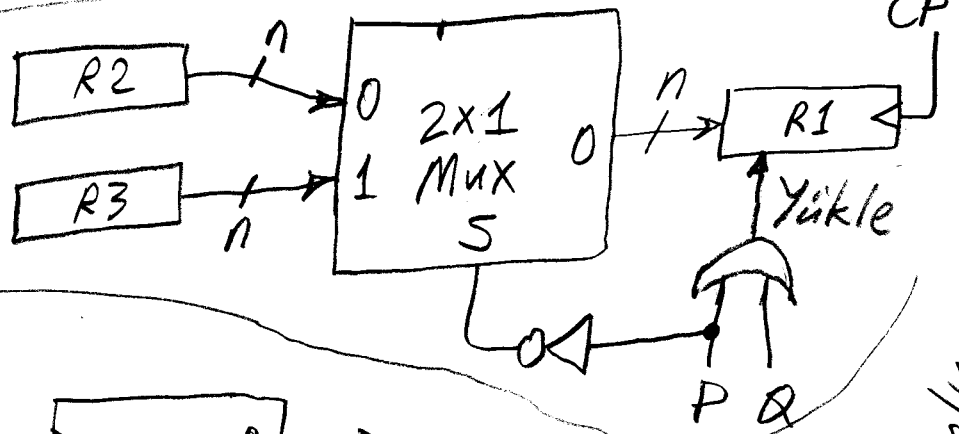
Zamanlama Şeması

$$yT_2: R2 \leftarrow R1, R1 \leftarrow R2 \text{ (Exchange)}$$

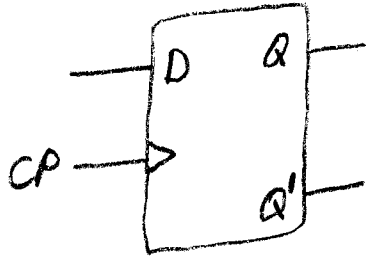


$$P: R1 \leftarrow R2$$

$$P'Q: R1 \leftarrow R3$$



D flip-flop



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

D	Q_{t+1}
0	0
1	1

Karakteristik tablosu

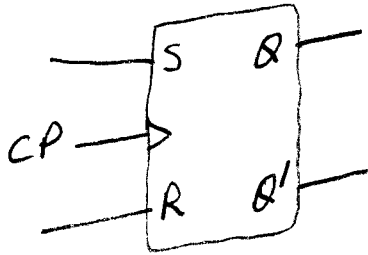
Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Uyarım tablosu

Q_t	Q_{t+1}	0	1
0	0	0	1
1	0	0	1

Uyarım tablosu

RS flip-flop



$$Q_{t+1} = S + R'Q_t$$

SR = 0 olmalı

S	R	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Kullanılmaz

Karakteristik tablosu

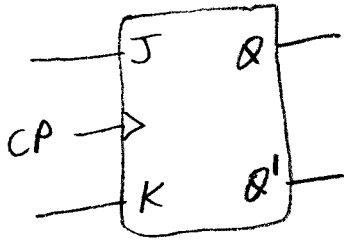
Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Uyarım tablosu

Q_t	Q_{t+1}	0	1
0	0, X	1, 0	
1	0, 1	X, 0	

Uyarım tablosu

JK flip-flop



$$Q_{t+1} = JQ_t' + K'Q_t$$

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q_t'

Karakteristik tablosu

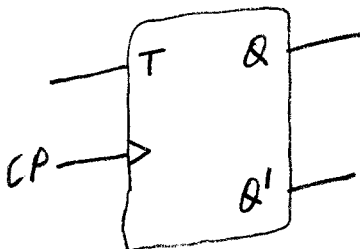
Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Uyarım tablosu

Q_t	Q_{t+1}	0	1
0	0, X	1, X	
1	X, 1	X, 0	

Uyarım tablosu

T flip-flop



$$Q_{t+1} = TQ_t' + T'Q_t$$

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

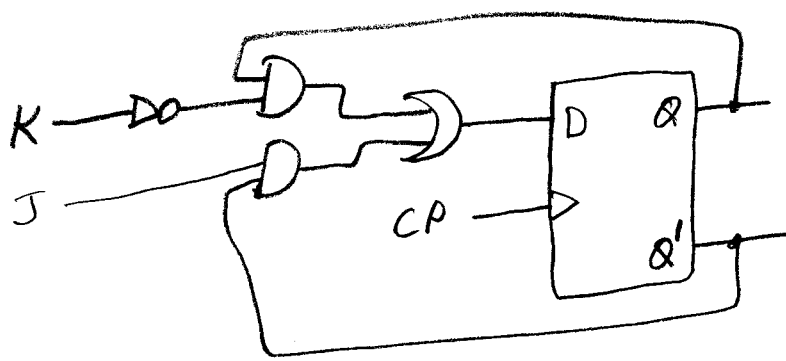
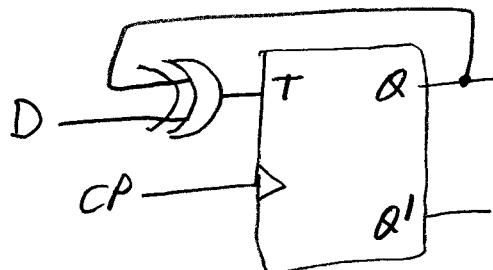
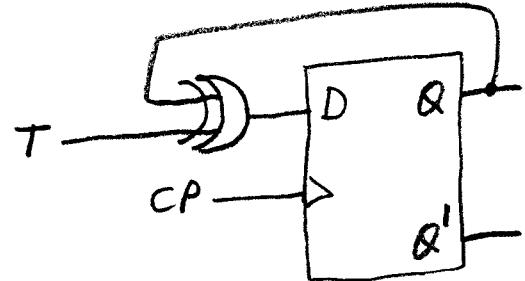
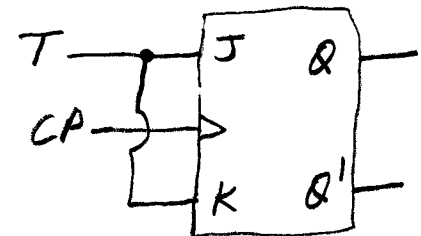
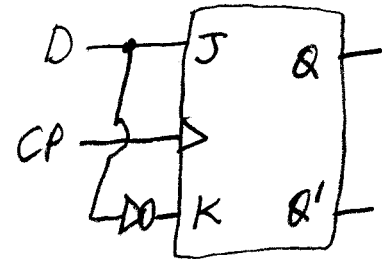
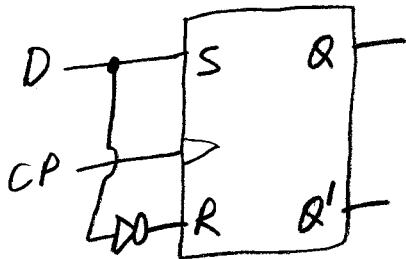
Karakteristik tablosu

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	1	1
1	0	0

Uyarım tablosu

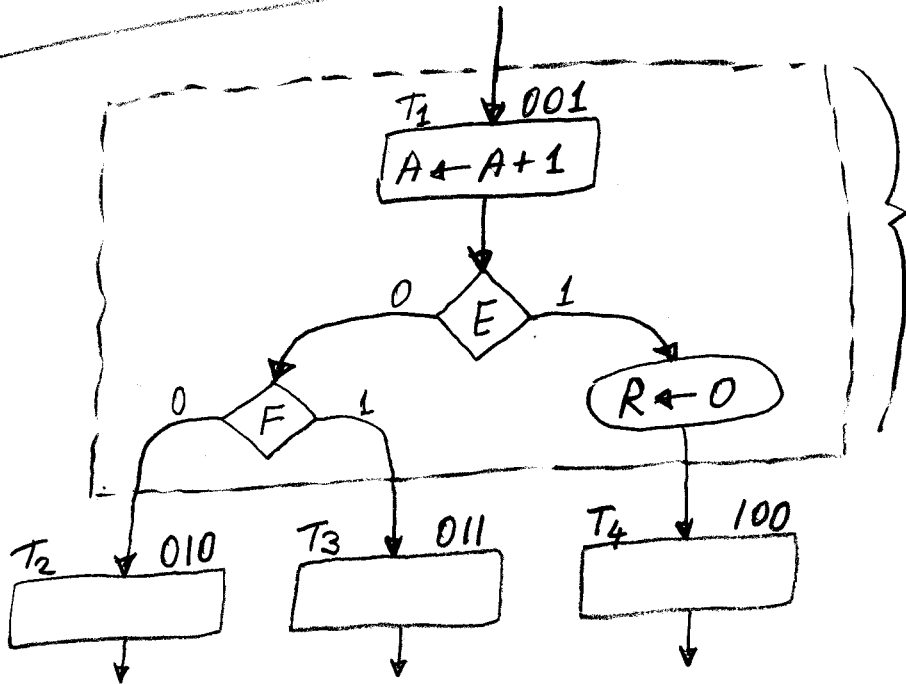
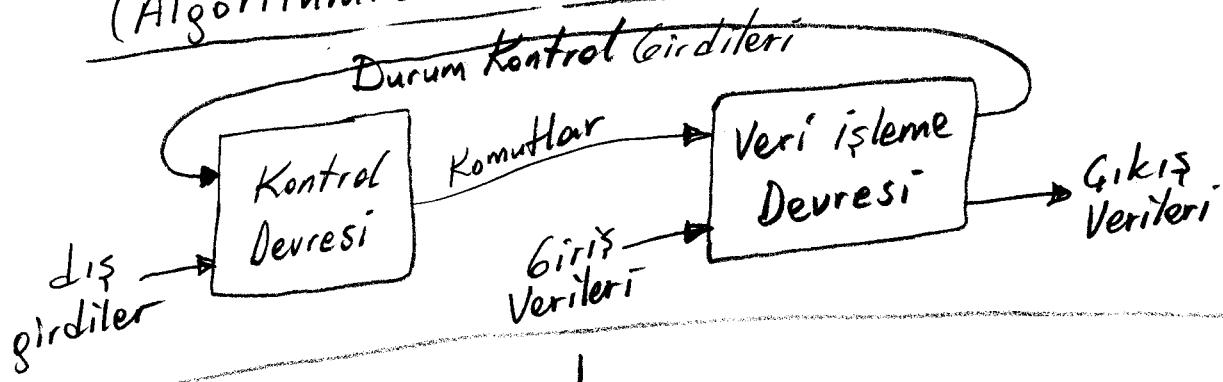
Q_t	Q_{t+1}	0	1
0	0	1	
1	1	0	

Uyarım tablosu

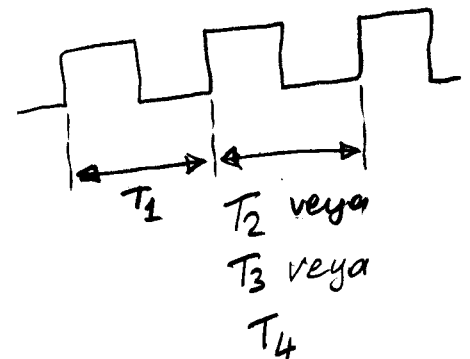


Ayrıca, JK flip-flop J, K ikisi birden 1 olmadığı sürece RS flip-flop gibi davranır.

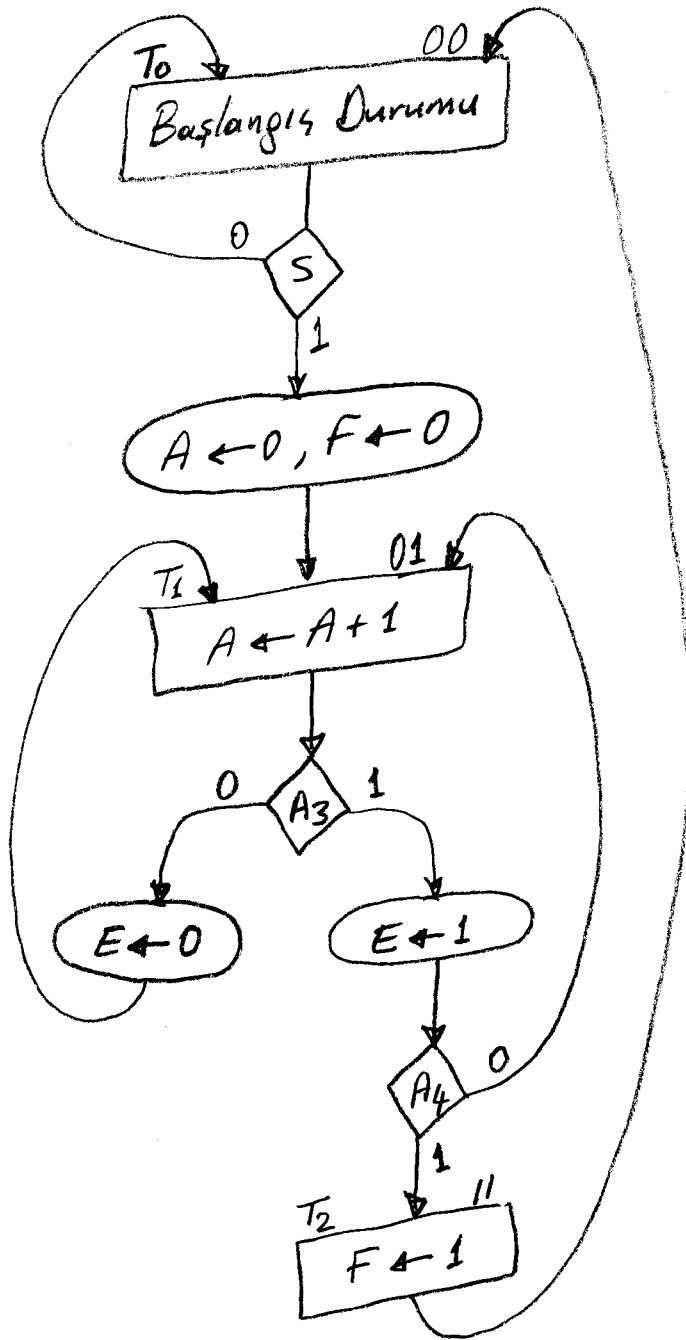
Algoritmik Durum Makinaları (Algorithmic State Machines - ASM)



Her CP için 1 blok icra edilir



Ör Bir tane 4 bitlik sayıcı (A = A₃A₂A₁A₀) ve iki tane flip-floplu (E ve F) bir sayısal sistem düşünelim. Sistem S başlama sinyaliyle A sayıcısını ve F flip-flopunu resetleyerek çalışmaya başlar. Sayma işlemi durana kadar her CP için A sayıcısı 1 artar. A₃=0 ise E=0 olur ve sayma işlemine devam eder. A₃=1 ise E=1 olur ve A₄ bitinin durumuna göre yani A₄=0 ise sayma işlemine devam, değilse diğer CP için F=1 olur ve sayma işlemi durur.



ASM Planı

S T₀ : A ← 0, F ← 0

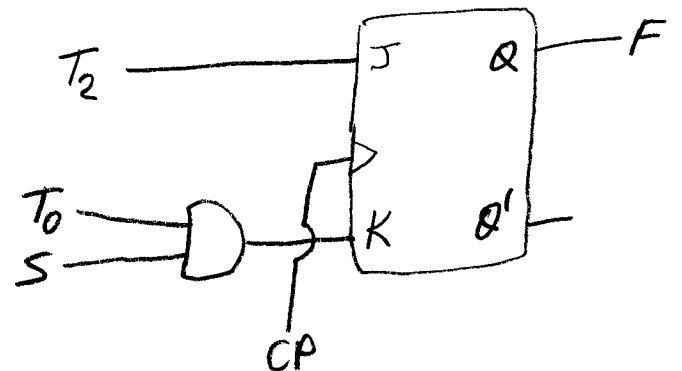
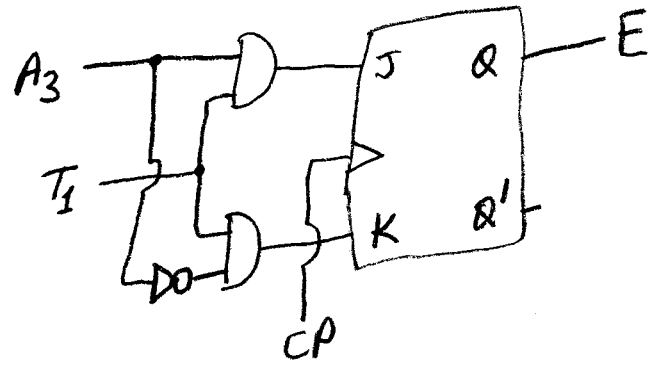
T₁ : A ← A + 1

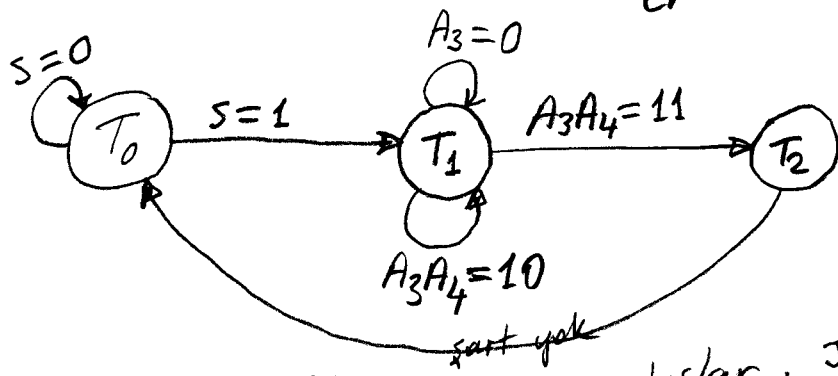
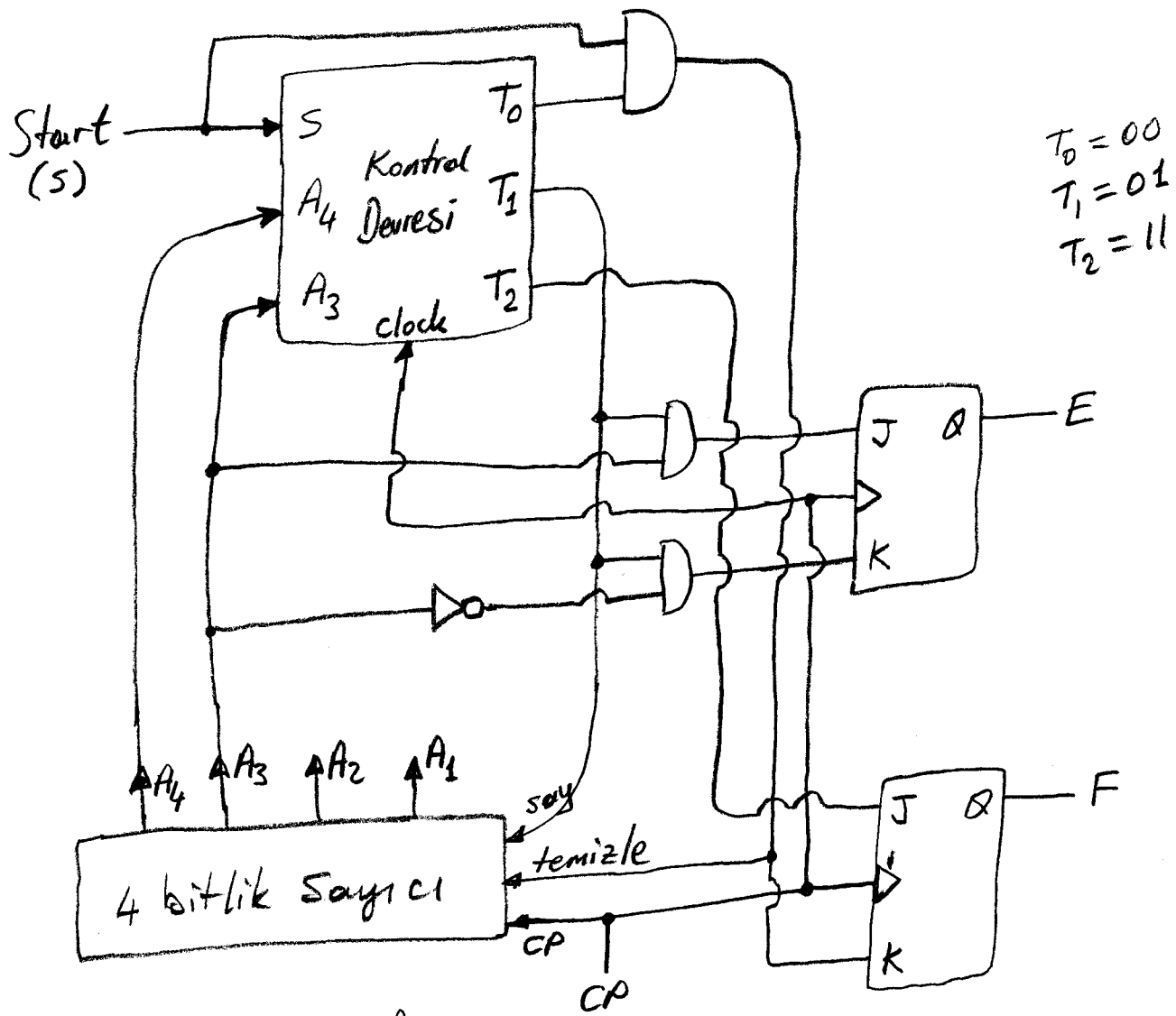
A₃T₁ : E ← 0

A₃T₁ : E ← 1

T₂ : F ← 1

JK	Q _{t+1}
00	Q _t
01	0
10	1
11	Q' _t



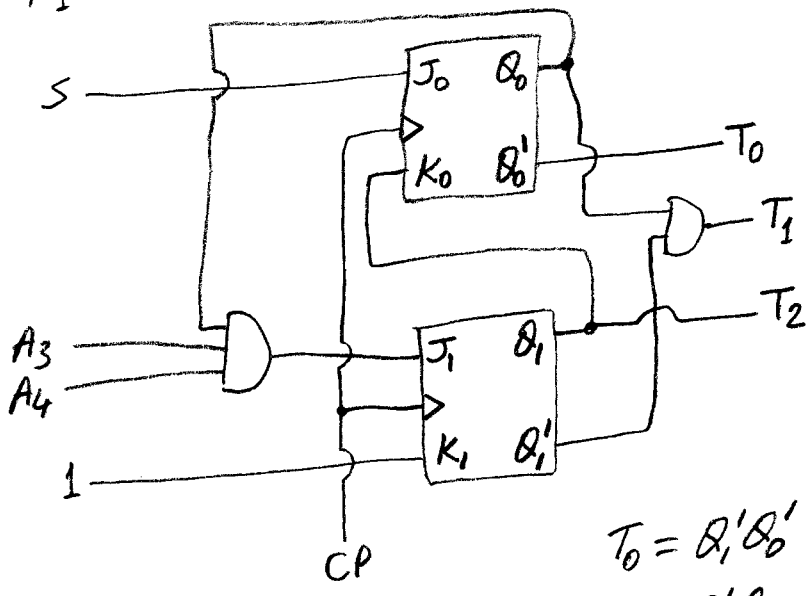


Kontrol devresinin durum diyagramı

PS sembol	PS B_1, B_0	PS Girişler S, A_3, A_4	NS B_1, B_0	4 çıkışlar T_0, T_1, T_2	JK ff ile J, K, J_0, K_0	D ff ile D, D_0
T_0	00	0 X X	00	1 0 0	0 X 0 X	00
T_0	00	1 X X	01	1 0 0	0 X 1 X	01
T_1	01	X 0 X	01	0 1 0	0 X X 0	01
T_1	01	X 1 0	01	0 1 0	0 X X 0	01
T_1	01	X 1 1	11	0 1 0	1 X X 0	11
T_2	11	X X X	00	0 0 1	X 1 X 1	00

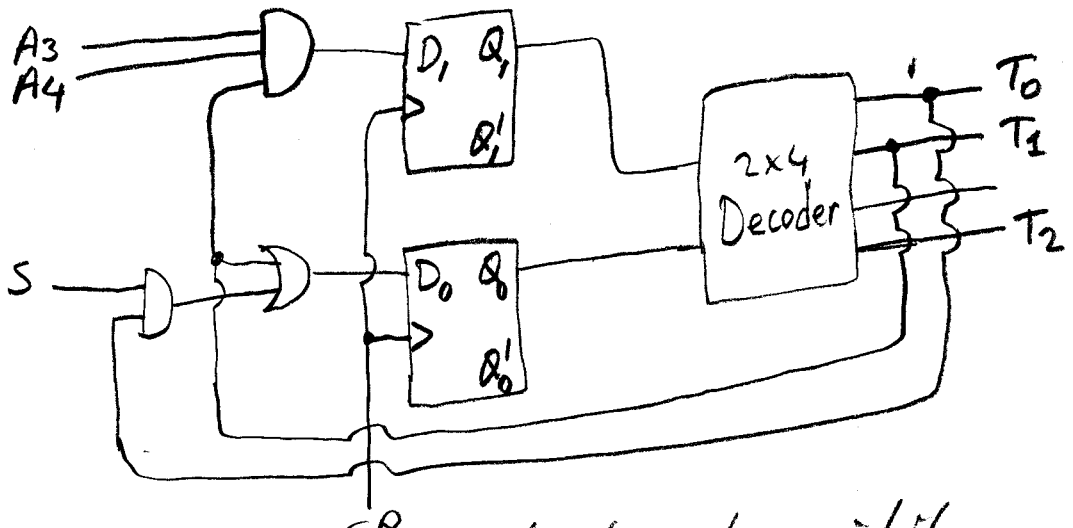
B_1, B_0	JK
00	0X
01	1X
10	X1
11	X0

a) $J_1 = Q_0 A_3 A_4$ $J_0 = S$ $T_0 = Q_0'$, $T_1 = Q_1' Q_0$, $T_2 = Q_1$
 $K_1 = 1$ $K_0 = Q_1$

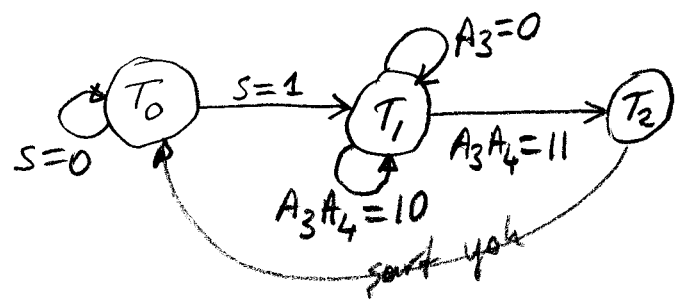


JK flip-flop ile kontrol devresinin tasarımı

b) $D_1 = Q_1' Q_0 A_3 A_4$ $T_0 = Q_1' Q_0'$
 $D_2 = S Q_1' Q_0' + Q_1' Q_0$ $T_1 = Q_1' Q_0$
 $T_2 = Q_1 Q_0$ $D_1 = A_3 A_4 T_1$
 $D_2 = S T_0 + T_1$



c) Üç tane D flip-flop ile tasarlanabilir

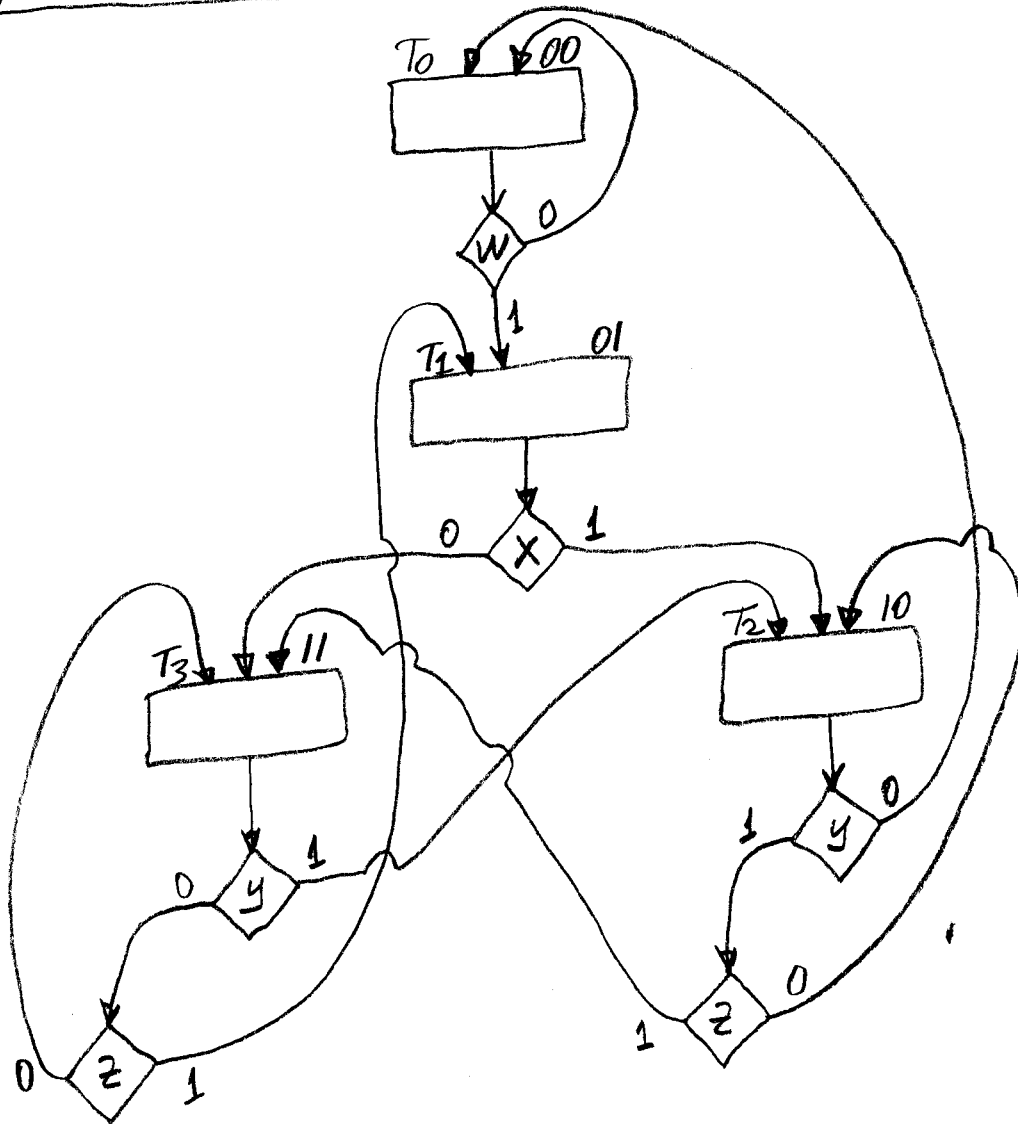


Kontrol devresinin durum diyagramını kullanırık-

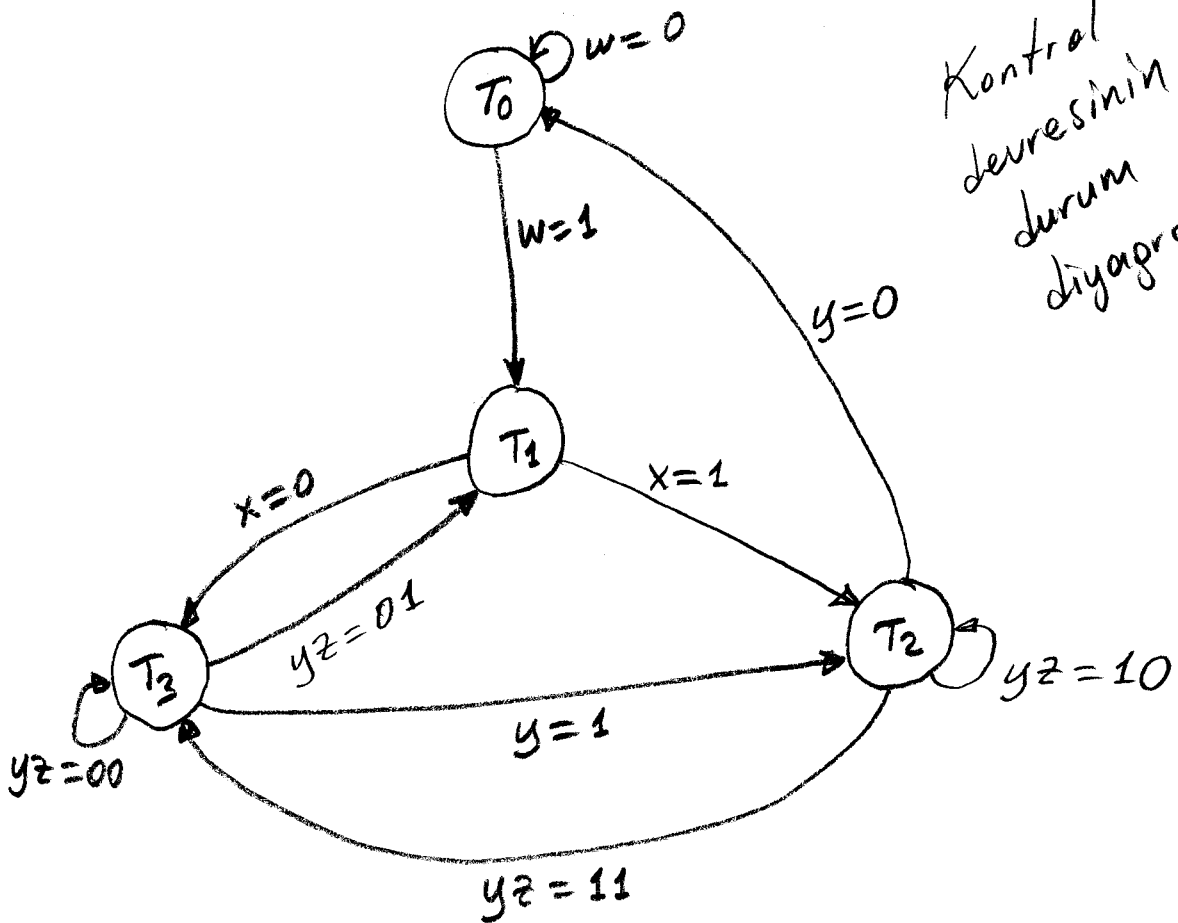
$D_0 = S' T_0 + T_2$ $D_3 = A_3 A_4 T_1$
 $D_1 = S T_0 + A_3' T_1 + A_3 A_4' T_1$ $T_0 = Q_0$
 $= S T_0 + (A_3 A_4)' T_1$ $T_1 = Q_1$
 $T_2 = Q_2$

Kontrol Devresinin multiplexer'lar ile tasarlanması

6



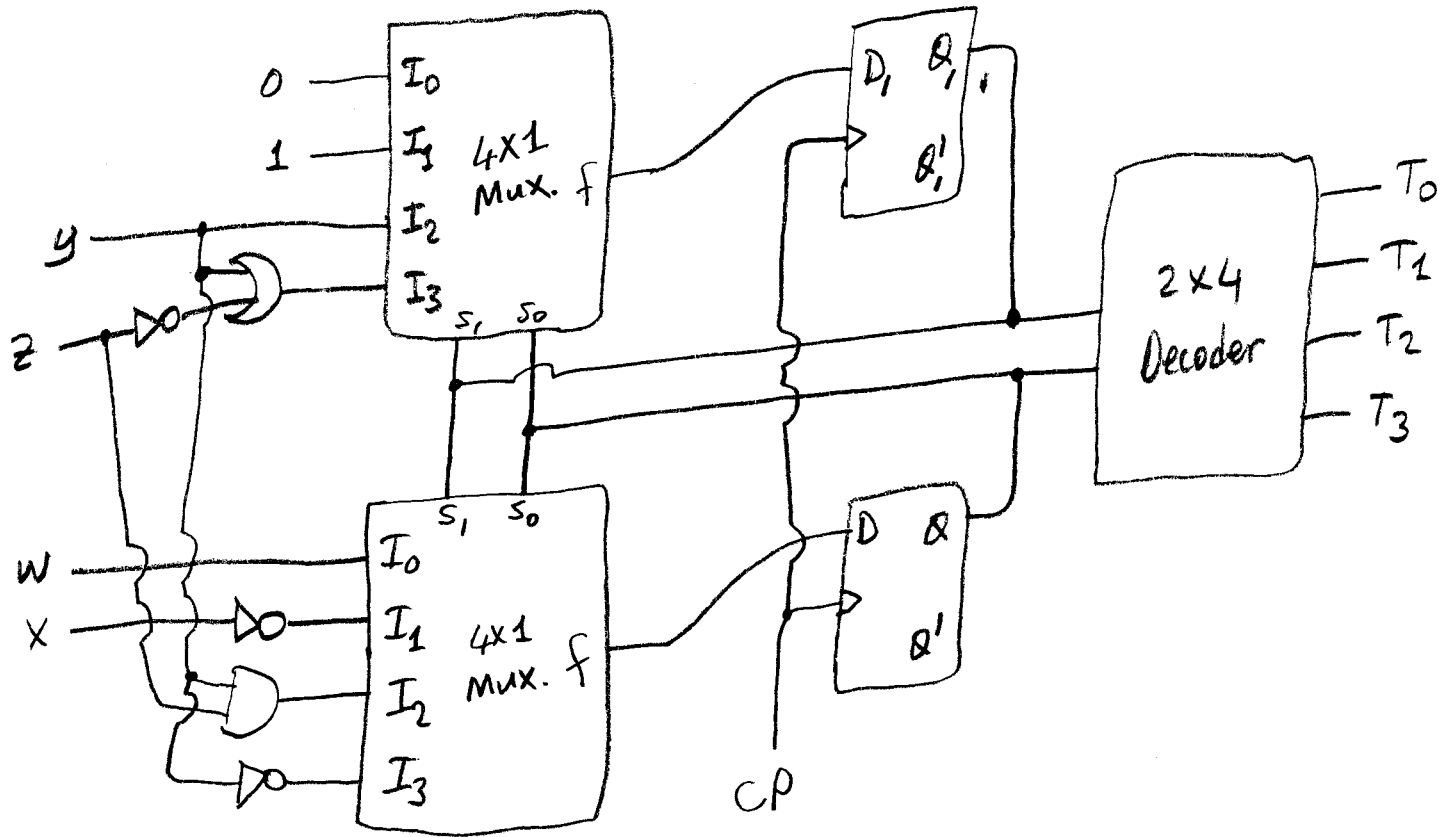
Yanda verilen
ASM planını
kullanarak
devrenin
kontrol kısmını
multiplexer
kullanarak
tasarla



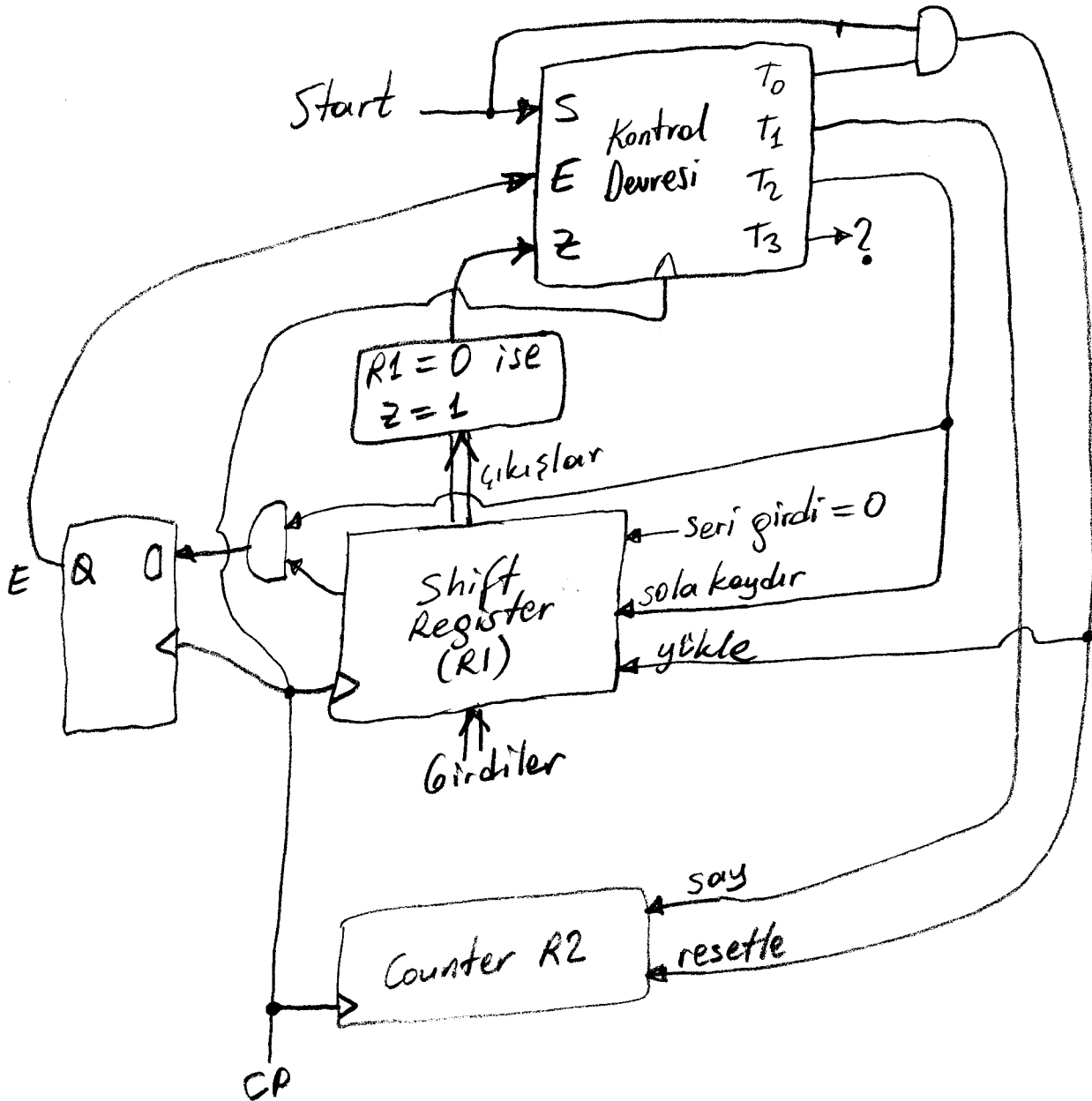
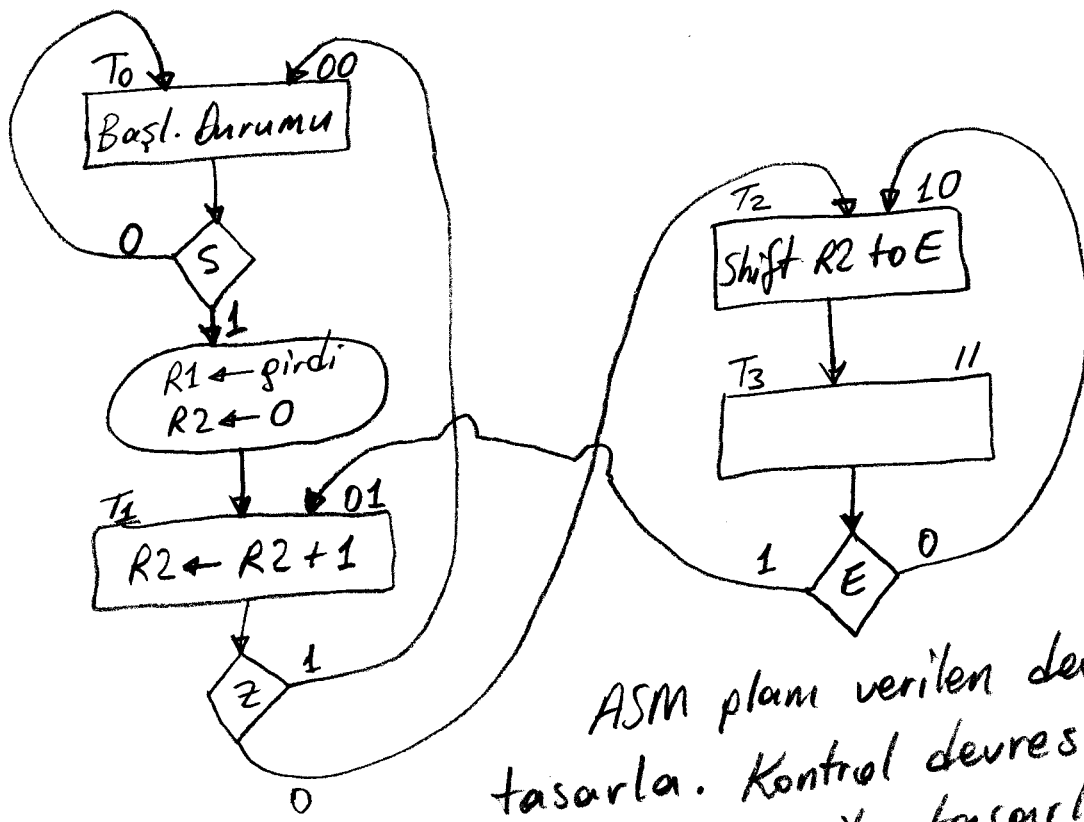
Kontrol
devresinin
durum
diyagramı

PS θ, θ_0	NS θ, θ_0	Girdi Şartları	Mux Girişleri	
			MUX1	MUX2
00	00	w'	0	w
00	01	w		
01	10	x	1	x'
01	11	x'		
10	00	y'	$yz' + yz$ $= y$	yz
10	10	yz'		
10	11	yz		
11	01	$y'z$	$y + y'z'$ $= y + z'$	$y'z + y'z'$ $= y'$
11	10	y		
11	11	$y'z'$		

$$\begin{aligned}
 T_0 &= 00 \\
 T_1 &= 01 \\
 T_2 &= 10 \\
 T_3 &= 11
 \end{aligned}$$

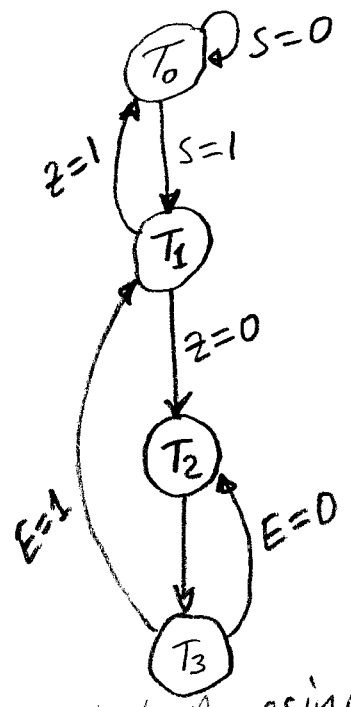


Kontrol Devresi

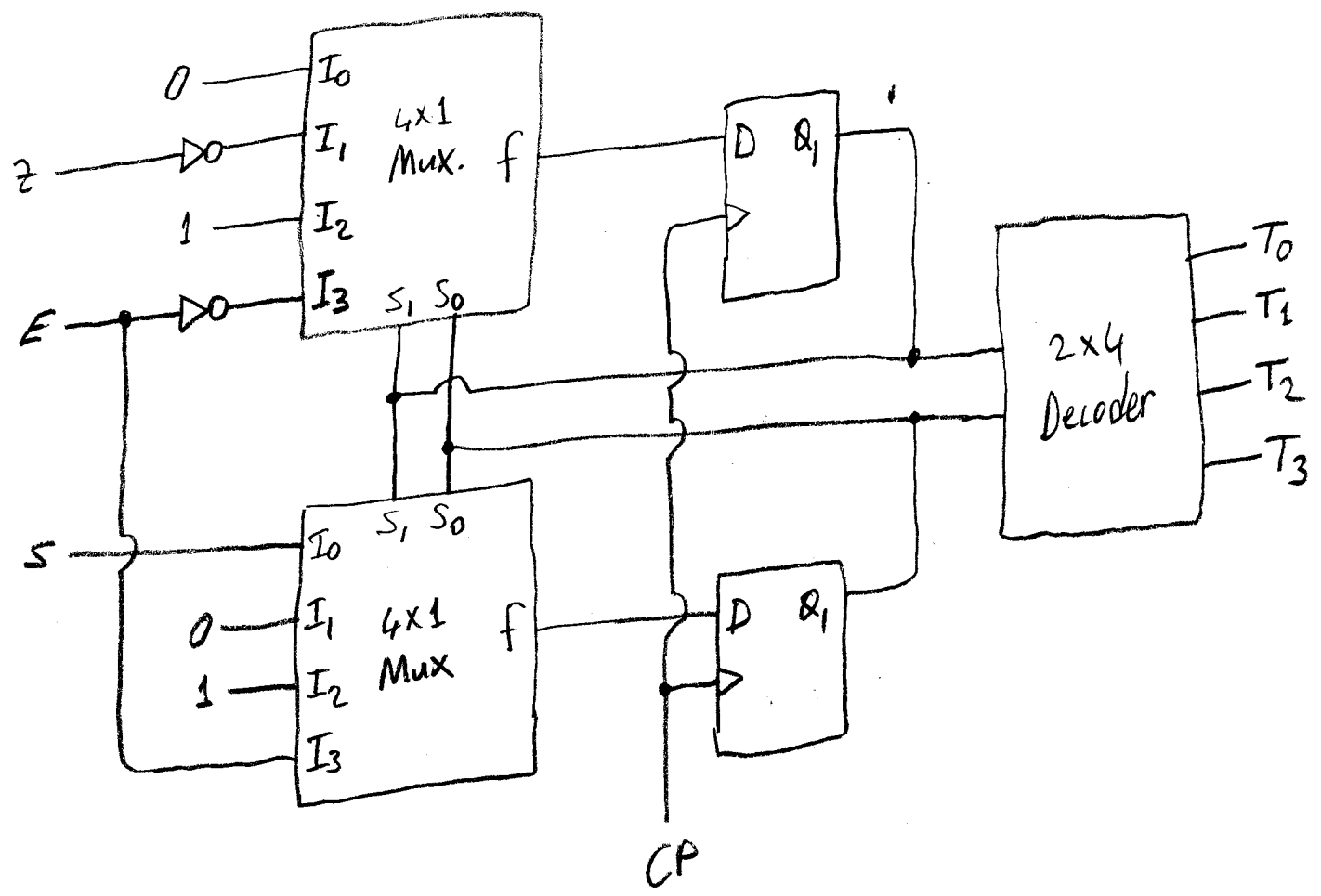


PS Q_1, Q_0	NS Q_1, Q_0	Girdi şartları	Mux. Girişleri	
			Mux1	Mux2
00	00	S'	0	S
00	01	S		
01	00	Z	Z'	0
01	10	Z'		
10	11	1	1	1
11	01	E	E'	E
11	10	E'		

Durum Tablosu

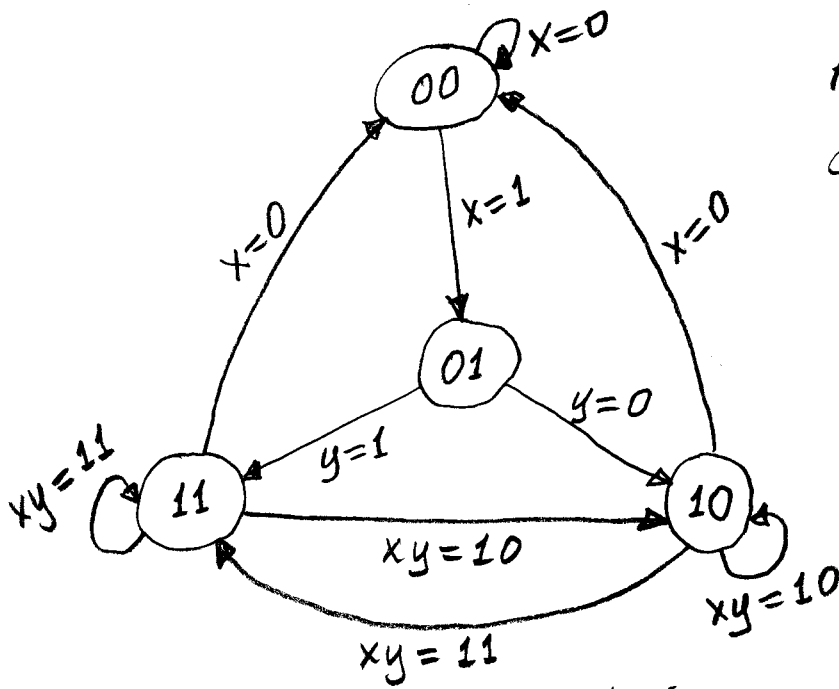


Kontrol Devresinin Durum Diyagramı



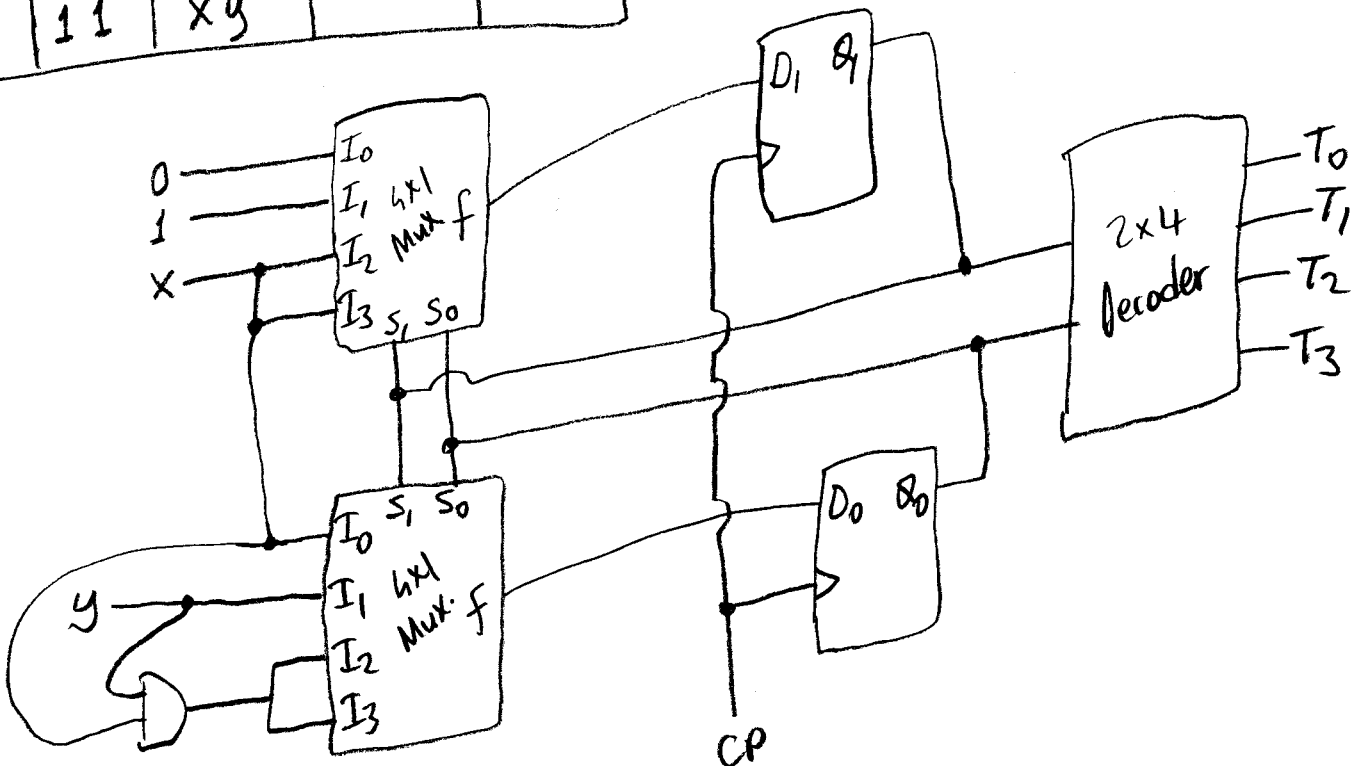
Kontrol Devresi

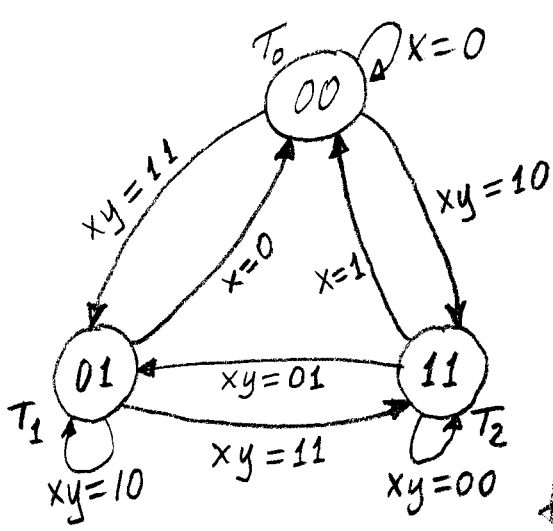
Herhangi bir ASM
devresinin kontrol kısmının
durum diyagramı
yadaki gibidir.
Kontrol kısmını
multiplexer ile
tasarla.



$D_1 D_0$	$D_1 D_0$	Girdi Şartları	Mux. Girişleri Mux1	Mux2
00	00	x'	0	X
00	01	X		
01	10	y'	1	y
01	11	y		
10	00	x'		
10	10	xy'	X	xy
10	11	xy		
11	00	x'		
11	10	xy'	X	xy
11	11	xy		

Durum Tablosu





Bir ASM lojik devresinin kontrol kısmının durum diyagramı yukarıdaki gibidir.

Kontrol devresini

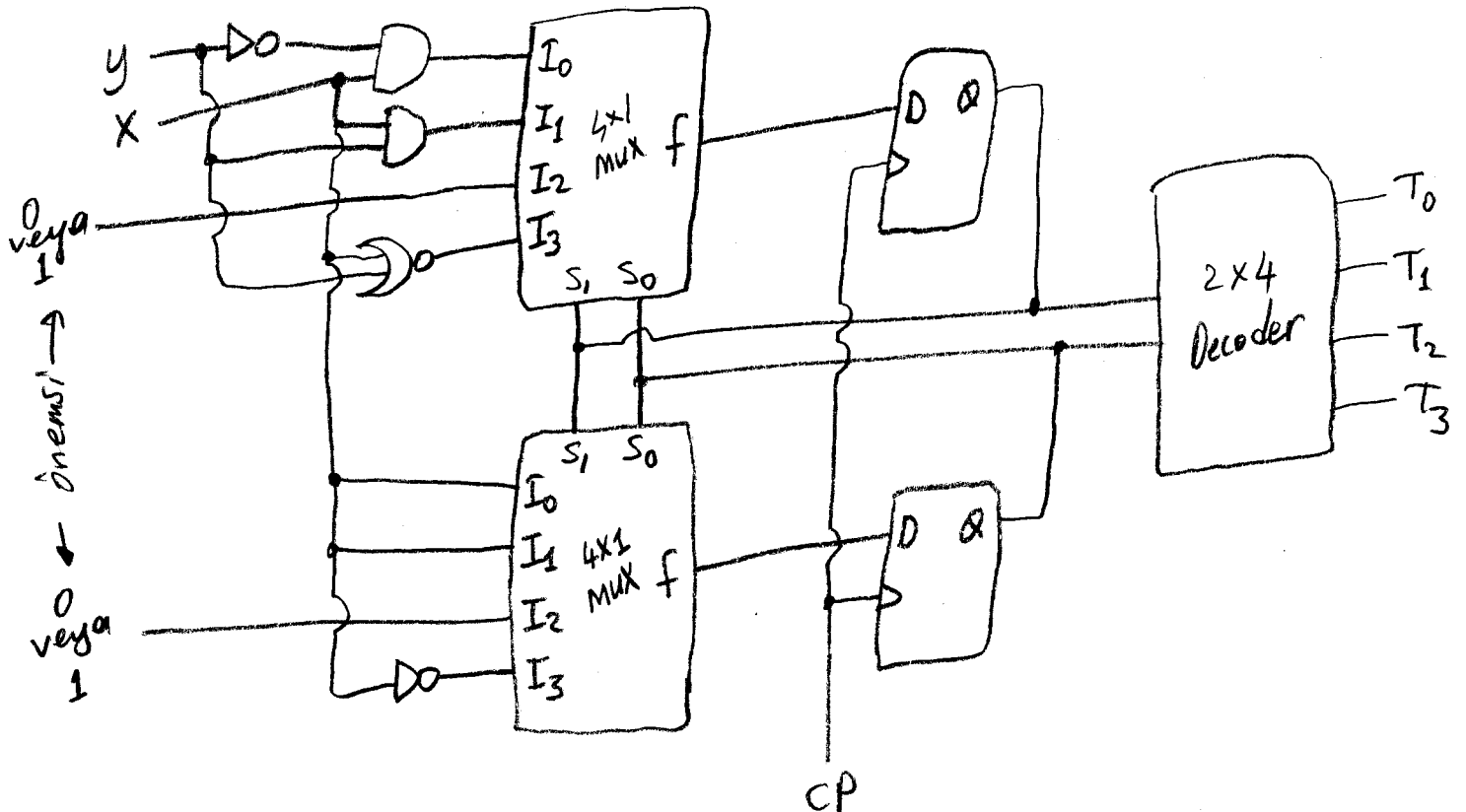
- multiplexer'lar ve D ff'larla
- RS flip-flop'lar ile

tasarlayınız.

a)

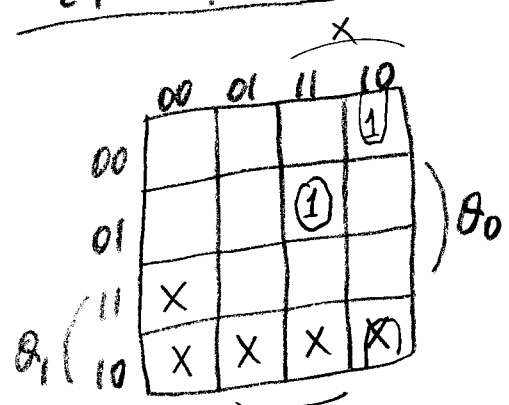
Q_1, Q_0	Q_1, Q_0	Girdi Sartları	mux girişleri mux 1	mux 2
00	00	x'		
00	01	xy	xy'	x
00	11	xy'		
01	00	x'		
01	01	xy'	xy	x
01	11	xy		
11	00	x		
11	01	$x'y$	$(x+y)'$	x'
11	11	$x'y'$		

Durum Tablosu

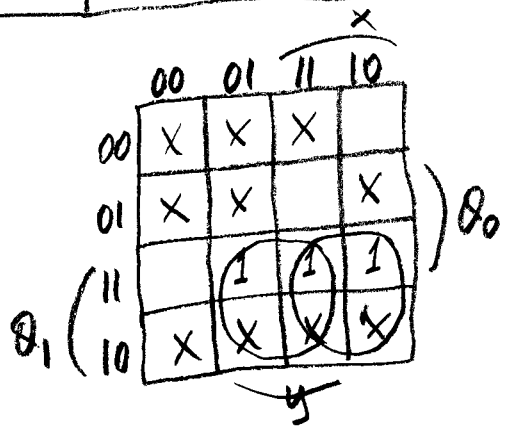


PS	PS $Q_1 Q_0$	birdiler $x y$	NS $Q_1 Q_0$	$T_0 T_1 T_2$	S, R_1	$S_0 R_0$
T_0	00	0X	00	1 0 0	0 X	0 X
T_0	00	11	01	1 0 0	0 X	1 0
T_0	00	10	11	1 0 0	1 0	1 0
T_1	01	0X	00	0 1 0	0 X	0 1
T_1	01	10	01	0 1 0	0 X	X 0
T_1	01	11	11	0 1 0	1 0	X 0
T_2	11	1X	00	0 0 1	0 1	0 1
T_2	11	01	01	0 0 1	0 1	X 0
T_2	11	00	11	0 0 1	X 0	X 0

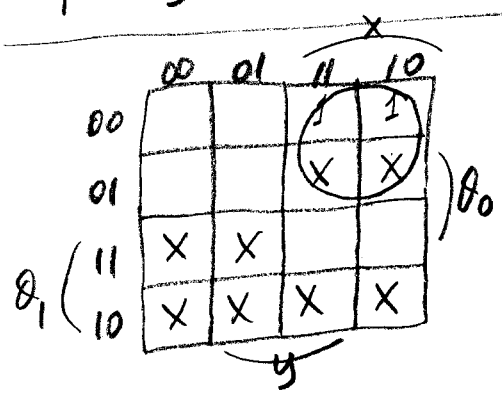
$Q_t Q_{t+1}$	SR
00	0X
01	10
10	01
11	X0



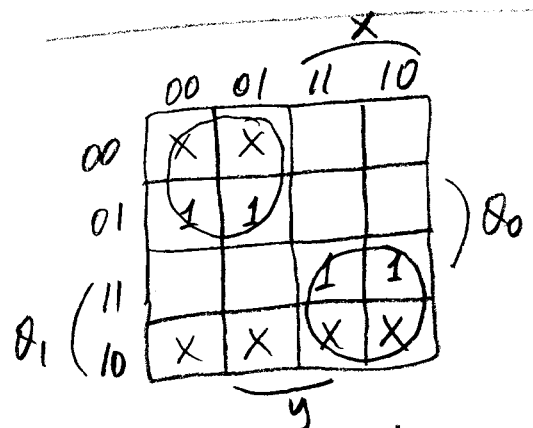
$$S_1 = xyQ_1'Q_0 + xy'Q_0'$$



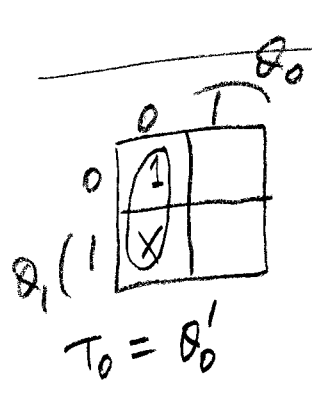
$$R_1 = xQ_1 + yQ_1 = (x+y)Q_1$$



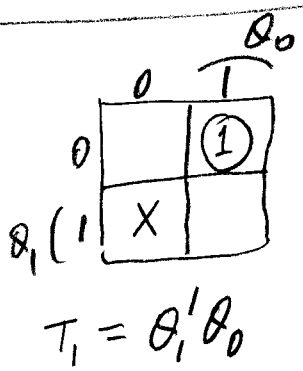
$$S_0 = xQ_1'$$



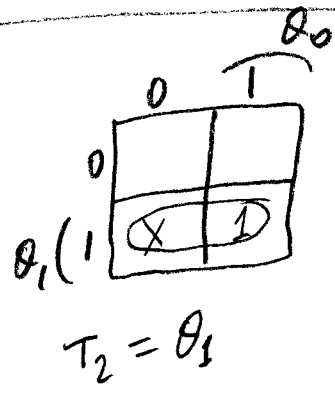
$$R_0 = xQ_1 + x'Q_1' = x \odot Q_1$$



$$T_0 = Q_0'$$



$$T_1 = Q_1'Q_0$$



$$T_2 = Q_1$$

sonuçların
göre devreyi
tasarla

T.C.
DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MİMARİSİ DERSİ ARA SINAVI

27.03.2018, Salı
Saat 15³⁰

Not: Toplam Süre 45 dakikadır. Her bir soru 35 puandır.
Sadece 3 soru çözülecektir. Sorular öğrencide kalabilir.

Soru 1

2Kx8 bitlik RAM hafıza birimi **\$A800 – \$AFFF** adresleri arasına bağlanıyor. Hafıza birimini kapı bağlantıları ile beraber tasarlayınız.

Soru 2

Altprograma gidiş yani **Call** komutu için gerekli mikro işlemleri yazınız. Altprogramdan dönüş yani **Return** komutu için gerekli mikro işlemleri yazınız.

Soru 3

Aşağıdaki şartları sağlayan merkezi işlemci biriminin (MİB) blok şemasını çiziniz. MİB içinde 1 tane ALU, 3 tane yazaç ve 3 tane seçici bulunsun. ALU 5 kontrol girişine sahip olsun. ALU, A ve B seçicileri yardımıyla hem girişten hem de yazaçlardan girdilerini alabilsin. ALU işlem sonucunu C seçicisi yardımıyla çıkışa veya istenilen yazaca aktarabilsin.

Soru 4

$F \leftarrow A * B - C / (D + E)$ fonksiyonu ile verilen işlemi yapan programı

- Sıfır adres buyruklu komut kümesi olan bir dil ile yazınız.
- Bir adres buyruklu komut kümesi olan bir dil ile yazınız.
- İki adres buyruklu komut kümesi olan CISC mimari kullanan bir dil ile yazınız.
- İki adres buyruklu komut kümesi olan RISC mimari kullanan bir dil ile yazınız.

Soru 5

S_2	S_1	S_0	F
0	0	0	$Shr A$
0	0	1	$Shl A$
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	$A - B$

Doğruluk tablosu yanda verilen n bitlik ALU devresinin bir bitlik kısmını 8x1 Mux ve en az kapı elemanı kullanarak tasarlayınız.

T.C.
DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MİMARİSİ DERSİ ARA SINAV MAZERETİ

02.05.2018, Çarşamba
Saat 15³⁰

Not: Toplam Süre 45 dakikadır. Her bir soru 35 puandır.
Sadece 3 soru çözülecektir. Sorular öğrencide kalabilir.

Soru 1

S_1	S_0	Yazac İşlemi
0	0	Değişiklik yok
0	1	Sağa Kaydırma
1	0	Sola Kaydırma
1	1	Paralel Yükleme

Fonksiyon tablosu yanda verilen paralel yüklemeli çift yönlü kaydırma yazacının bir bitlik kısmını tasarlayınız. Tasarımda 4*1 Mux ve D flip-flop kullanınız.

Soru 2

Sil	Yükle	Say	İşlem
0	0	0	Koru
0	0	1	Say
0	1	X	Yükle
1	X	X	Sil

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

Soru 3

$F = (A + B) / C - D * E$ fonksiyonu ile verilen işlemi yapan programı

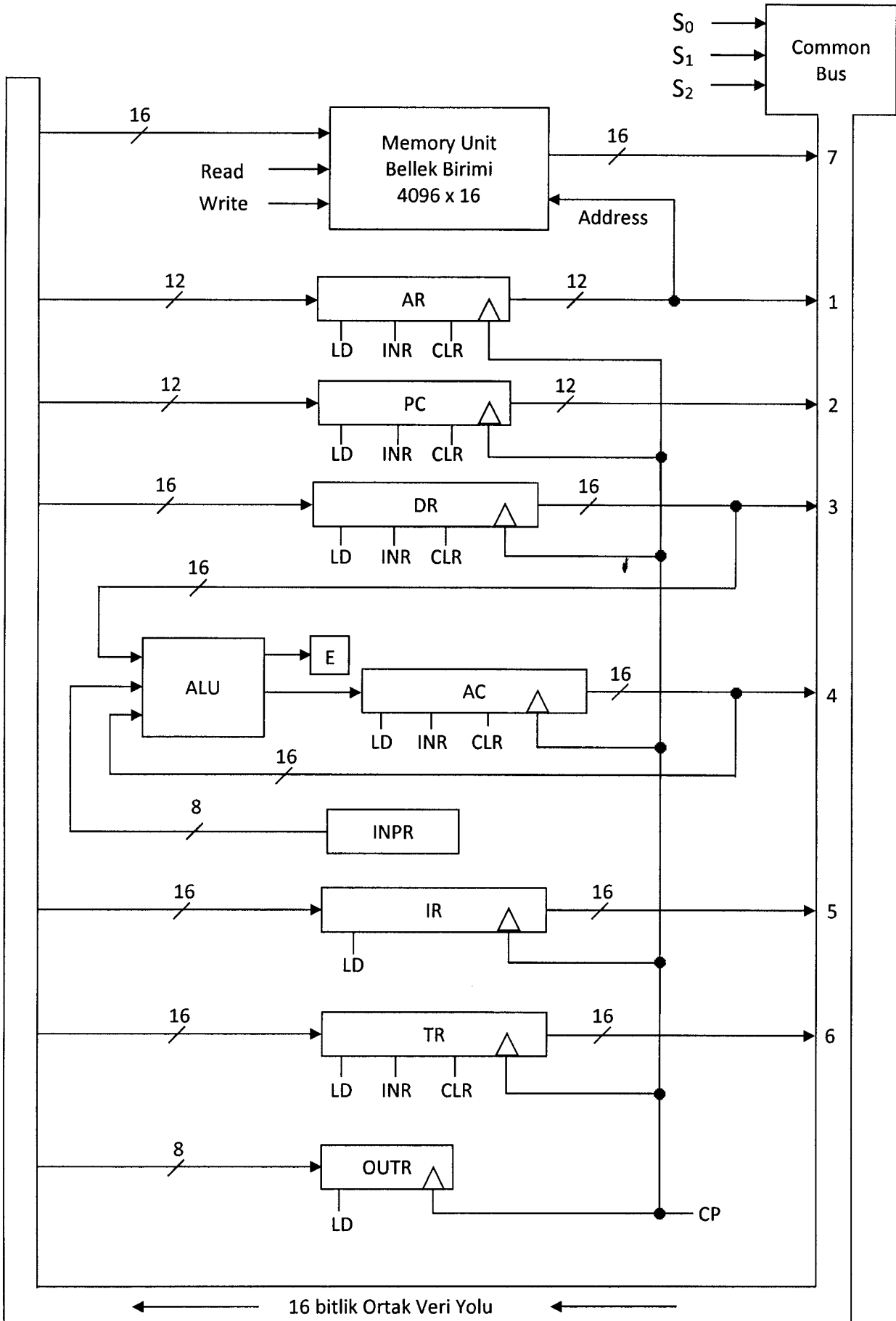
- Sıfır adres buyruklu komut kümesi olan bir dil ile yazınız.
- Bir adres buyruklu komut kümesi olan bir dil ile yazınız.
- İki adres buyruklu komut kümesi olan CISC mimari kullanan bir dil ile yazınız.
- İki adres buyruklu komut kümesi olan RISC mimari kullanan bir dil ile yazınız.

Soru 4

S_3	S_2	S_1	S_0	C_{gtr}	İşlem
0	0	0	0	0	$F = A + B$
0	0	0	0	1	$F = A + B + 1$
0	0	0	1	0	$F = A + \bar{B}$
0	0	0	1	1	$F = A + \bar{B} + 1$
0	0	1	0	0	$F = A$
0	0	1	0	1	$F = A + 1$
0	0	1	1	0	$F = A - 1$
0	0	1	1	1	$F = A$
0	1	0	0	X	$F = A \wedge B$
0	1	0	1	X	$F = A \vee B$
0	1	1	0	X	$F = A \oplus B$
0	1	1	1	X	$F = \bar{A}$
1	0	X	X	X	$F = shr A$
1	1	X	X	X	$F = shl A$

Fonksiyon tablosu yanda verilen Aritmetik Mantık Birimi (ALU) devresinin bir bitlik kısmını aşağıdaki aşamaları dikkate alarak tasarlayınız.
Bir bitlik Aritmetik Devre Aşaması
Bir bitlik Mantık Devresi Aşaması
Bir bitlik ALU Aşaması

Temel Bilgisayar Yazacılarının bir Ortak Veri Yoluna Bağlanması
Basic computer registers connected to a common bus



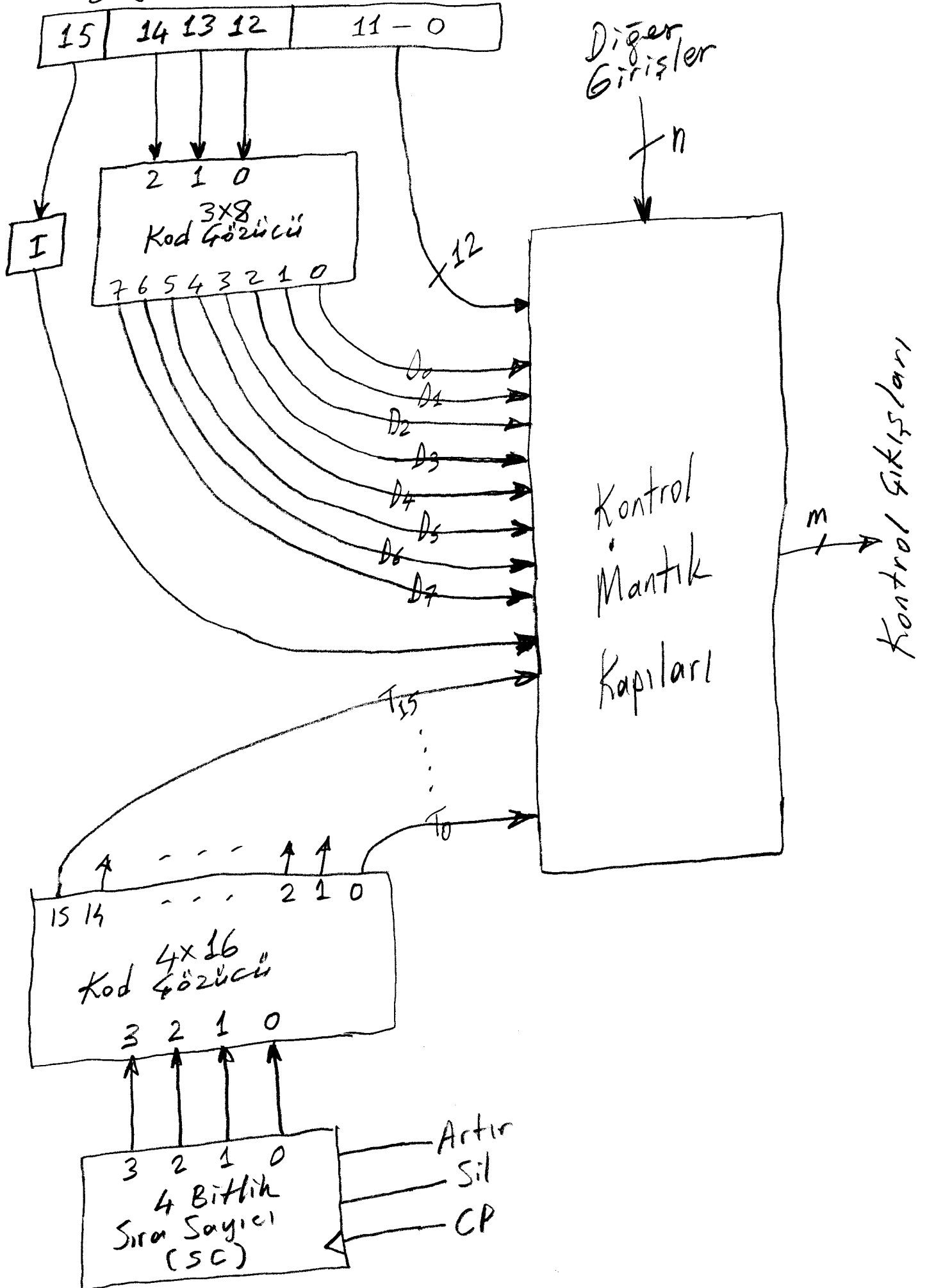
Temel Bilgisayar için Denetim Fonksiyonları ve Mikroişlemler
Control functions and microoperations for the basic computer

Fetch – Al Getir	$R' T_0 : AR \leftarrow PC$ $R' T_1 : IR \leftarrow M[AR], PC \leftarrow PC + 1$
Decode – Kod Çöz	$R' T_2 : I \leftarrow IR(15), D_7 \dots D_0 \leftarrow \text{Decode } IR(14-12), AR \leftarrow IR(11-0)$
Indirect – Dolaylı	$D_7' I T_3 : AR \leftarrow M[AR]$
Interrupt – Kesme	$(T_0 + T_1 + T_2)' \cdot IEN \cdot (FGI + FGO) : R \rightarrow 1$ $RT_0 : AR \leftarrow 0, TR \leftarrow PC$ $RT_1 : M[AR] \leftarrow TR, PC \leftarrow 0$ $RT_2 : PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$
Memory Reference Instructions – Bellek Adreslemeli Buyruklar	
AND	$D_0 T_4 : DR \leftarrow M[AR]$ $D_0 T_5 : AC \leftarrow AC \wedge DR, SC \leftarrow 0$
ADD	$D_1 T_4 : DR \leftarrow M[AR]$ $D_1 T_5 : AC \leftarrow AC + DR, E \leftarrow C_{out}, SC \leftarrow 0$
LDA	$D_2 T_4 : DR \leftarrow M[AR]$ $D_2 T_5 : AC \leftarrow DR, SC \leftarrow 0$
STA	$D_3 T_4 : M[AR] \leftarrow AC, SC \leftarrow 0$
BUN	$D_4 T_4 : PC \leftarrow AR, SC \leftarrow 0$
BSA	$D_5 T_4 : M[AR] \leftarrow PC, AR \leftarrow AR + 1$ $D_5 T_5 : PC \leftarrow AR, SC \leftarrow 0$
ISZ	$D_6 T_4 : DR \leftarrow M[AR]$ $D_6 T_5 : DR \leftarrow DR + 1$ $D_6 T_6 : M[AR] \leftarrow DR, \text{If } DR = 0 \text{ then } PC \leftarrow PC + 1, SC \leftarrow 0$
Register Reference Instructions – Yazaç Adreslemeli Buyruklar	
$D_7' I T_3 = r, \quad IR(i) = B_i (i = 0, 1, 2, \dots, 11)$ $r : SC \leftarrow 0$	
CLA	$rB_{11} : AC \leftarrow 0$
CLE	$rB_{10} : E \leftarrow 0$
CMA	$rB_9 : AC \leftarrow AC'$
CME	$rB_8 : E \leftarrow E'$
CIR	$rB_7 : AC \leftarrow shr AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$rB_6 : AC \leftarrow shl AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	$rB_5 : AC \leftarrow AC + 1$
SPA	$rB_4 : \text{If } AC(15) = 0 \text{ then } PC \leftarrow PC + 1$
SNA	$rB_3 : \text{If } AC(15) = 1 \text{ then } PC \leftarrow PC + 1$
SZA	$rB_2 : \text{If } AC = 0 \text{ then } PC \leftarrow PC + 1$
SZE	$rB_1 : \text{If } E = 0 \text{ then } PC \leftarrow PC + 1$
HLT	$rB_0 : S \leftarrow 0$
Input Output Instructions – Giriş Çıkış Buyrukları	
$D_7' I T_3 = p, \quad IR(i) = B_i (i = 6, 7, 8, 9, 10, 11)$ $p : SC \leftarrow 0$	
INP	$pB_{11} : AC(7-0) \leftarrow INPR, FGI \leftarrow 0$
OUT	$pB_{10} : OUTR \leftarrow AC(7-0), FGO \leftarrow 0$
SKI	$pB_9 : \text{If } FGI = 1 \text{ then } PC \leftarrow PC + 1$
SKO	$pB_8 : \text{If } FGO = 1 \text{ then } PC \leftarrow PC + 1$
ION	$pB_7 : IEN \leftarrow 1$
IOF	$pB_6 : IEN \leftarrow 0$

Temel Bilgisayarın Denetim Birimi

Bayrak Yazıcı (IR)

(3)

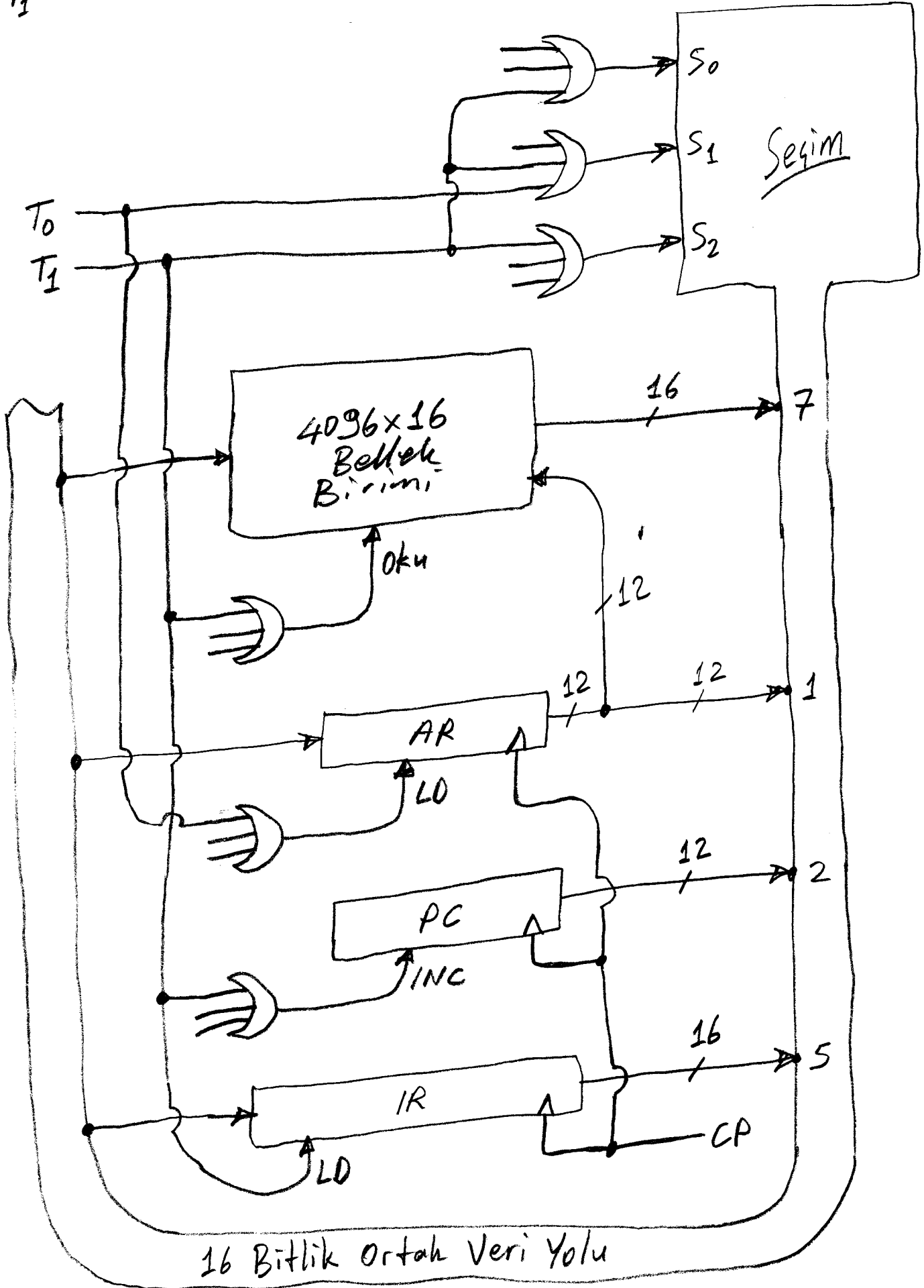


Al-Getir Evresi için Yazag Aktarımları

(4)

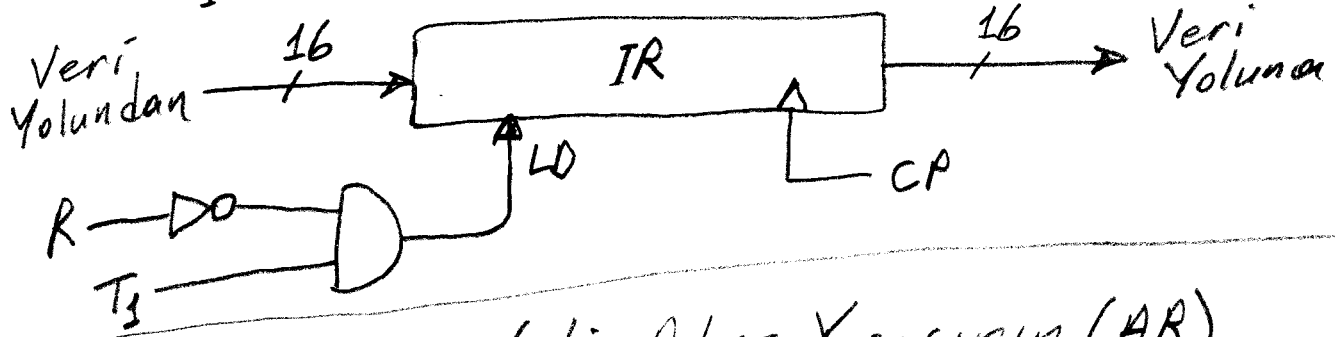
$T_0: AR \leftarrow PC$

$T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$



Temel Bilgisayardaki Buyruk Yazacının (IR) LD denetimi için kapı tasarımını çiziniz. (5)

$$R'T_1 : IR \leftarrow M[AR]$$



Temel Bilgisayardaki Adres Yazacının (AR) LD, INC, CLR denetimi için kapı tasarımını çiziniz.

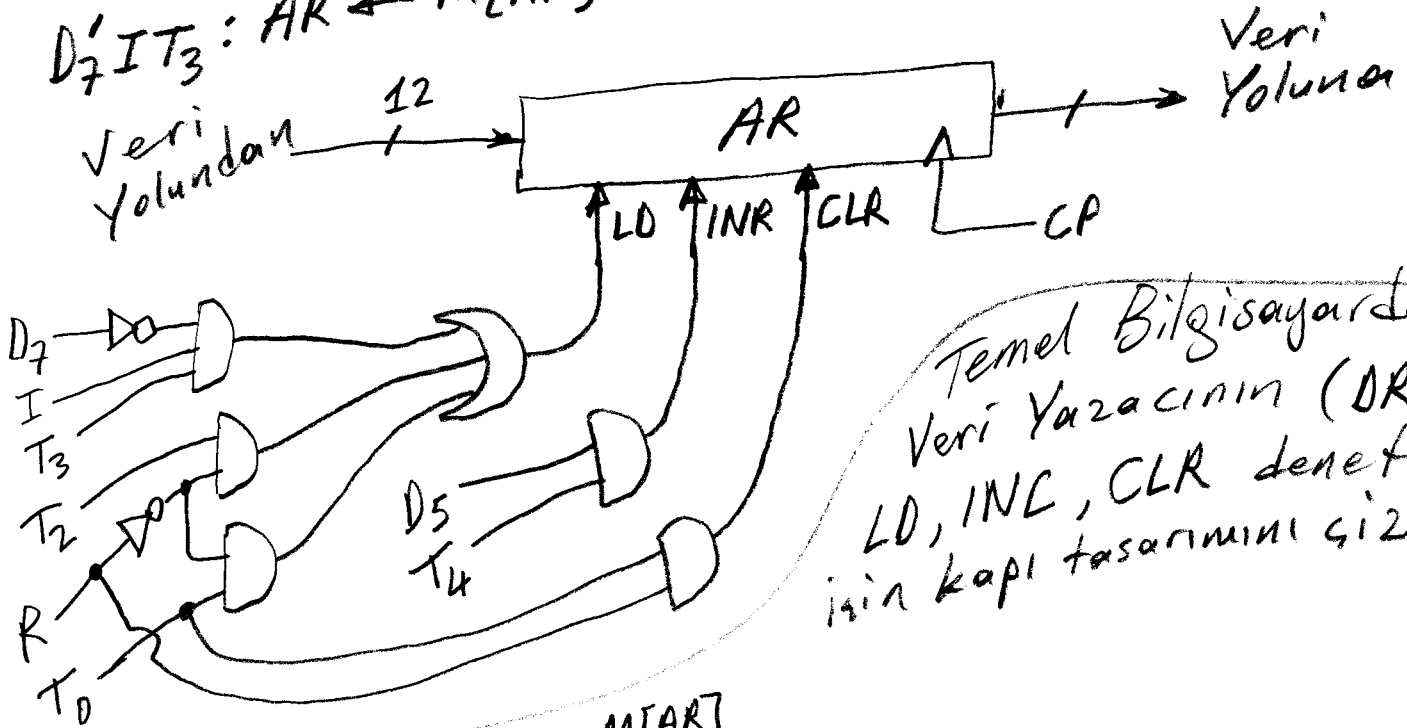
$$R'T_0 : AR \leftarrow PC$$

$$R'T_2 : AR \leftarrow IR(11-0)$$

$$D_7'IT_3 : AR \leftarrow M[AR]$$

$$D_5T_4 : AR \leftarrow AR + 1$$

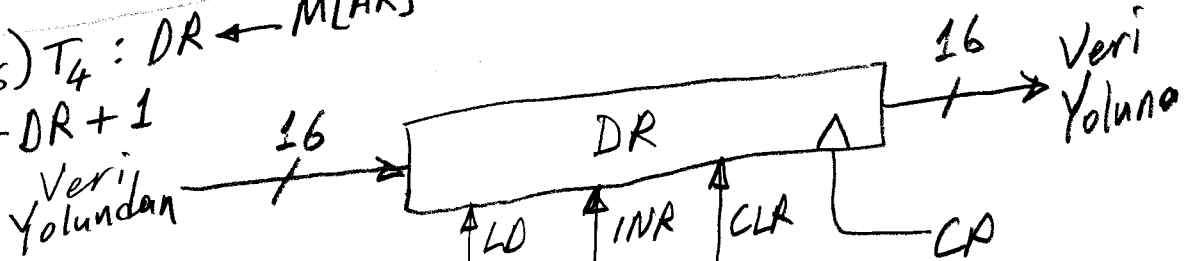
$$RT_0 : AR \leftarrow 0$$



Temel Bilgisayardaki Veri Yazacının (DR) LD, INC, CLR denetimi için kapı tasarımını çiziniz.

$$(D_0 + D_1 + D_2 + D_6)T_4 : DR \leftarrow M[AR]$$

$$D_6T_5 : DR \leftarrow DR + 1$$



Temel Bilgisayardaki belleği ve yazaçları ortak veri yolunda bağlayan kodlayıcının girişlerini: ⑥
 $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ ve çıkışlarını (S_0, S_1, S_2) bulunuz.

$$\left. \begin{array}{l} D_4 T_4 : PC \leftarrow AR \\ D_5 T_5 : PC \leftarrow AR \end{array} \right\} x_1 = D_4 T_4 + D_5 T_5$$

$$\left. \begin{array}{l} R' T_0 : AC \leftarrow PC \\ R T_0 : TR \leftarrow PC \\ D_5 T_4 : M[AR] \leftarrow PC \end{array} \right\} x_2 = T_0 + D_5 T_4$$

$$\left. \begin{array}{l} D_2 T_5 : AC \leftarrow DR \\ D_6 T_6 : M[AR] \leftarrow DR \end{array} \right\} x_3 = D_2 T_5 + D_6 T_6$$

$$\left. \begin{array}{l} D_3 T_4 : M[AR] \leftarrow AC \\ PB_{10} : OUTR \leftarrow AC(7-0) \end{array} \right\} x_4 = D_3 T_4 + PB_{10}$$

$P = D_7 I T_3, B_{10} = IR(10)$

$$R' T_2 : I \leftarrow IR(15), AR \leftarrow IR(11-0) \left\} x_5 = R' T_2$$

$$R T_1 : M[AR] \leftarrow TR \left\} x_6 = R T_1$$

$$R' T_1 : IR \leftarrow M[AR]$$

$$D_7 I T_3 : AR \leftarrow M[AR]$$

$$(D_0 + D_1 + D_2 + D_6) T_4 : DR \leftarrow M[AR]$$

$$\left. \begin{array}{l} x_7 = R' T_1 + D_7 I T_3 \\ + (D_0 + D_1 + D_2 + D_6) T_4 \end{array} \right\}$$

X	S ₂	S ₁	S ₀
x ₁	0	0	1
x ₂	0	1	0
x ₃	0	1	1
x ₄	1	0	0
x ₅	1	0	1
x ₆	1	1	0
x ₇	1	1	1

$$S_0 = x_1 + x_3 + x_5 + x_7$$

$$S_1 = x_2 + x_3 + x_6 + x_7$$

$$S_2 = x_4 + x_5 + x_6 + x_7$$

Sembol	İşlem Kodu	Sembolik Gösterim
OR	000	$AC \leftarrow AC \vee M[AR]$
SUB	001	$AC \leftarrow AC - M[AR]$
XCH	010	$AC \leftarrow M[AR], M[AR] \leftarrow AC$
ADM	011	$M[AR] \leftarrow M[AR] + AC$

Temel Bilgisayardaki bazı buyrukları yukarıdaki buyruklar ile değiştirelim. ALU devresinde herhangi bir depoziklik yapılmıyor. Her bir komut için T_4 anından başlayarak gerekli mikro işlemleri sırasıyla yazınız.

OR $D_0T_4: DR \leftarrow M[AR], AC \leftarrow \overline{AC}$
 $D_0T_5: DR \leftarrow AC, AC \leftarrow DR$
 $D_0T_6: AC \leftarrow \overline{AC}$
 $D_0T_7: AC \leftarrow AC \wedge DR$
 $D_0T_8: AC \leftarrow \overline{AC}, SC \leftarrow 0$

$$A \vee B = (A' \wedge B')'$$

$$A + B = A + \overline{B} + 1$$

SUB $D_1T_4: DR \leftarrow M[AR]$
 $D_1T_5: DR \leftarrow AC, AC \leftarrow DR$
 $D_1T_6: AC \leftarrow \overline{AC}$
 $D_1T_7: AC \leftarrow AC + 1$
 $D_1T_8: AC \leftarrow AC + DR, E \leftarrow C_{out}, SC \leftarrow 0$

XCH $D_2T_4: DR \leftarrow M[AR]$
 $D_2T_5: M[AR] \leftarrow AC, AC \leftarrow DR, SC \leftarrow 0$

ADM $D_3T_4: DR \leftarrow M[AR]$
 $D_3T_5: DR \leftarrow AC, AC \leftarrow AC + DR, E \leftarrow C_{out}$
 $D_3T_6: M[AR] \leftarrow AC, AC \leftarrow DR, SC \leftarrow 0$

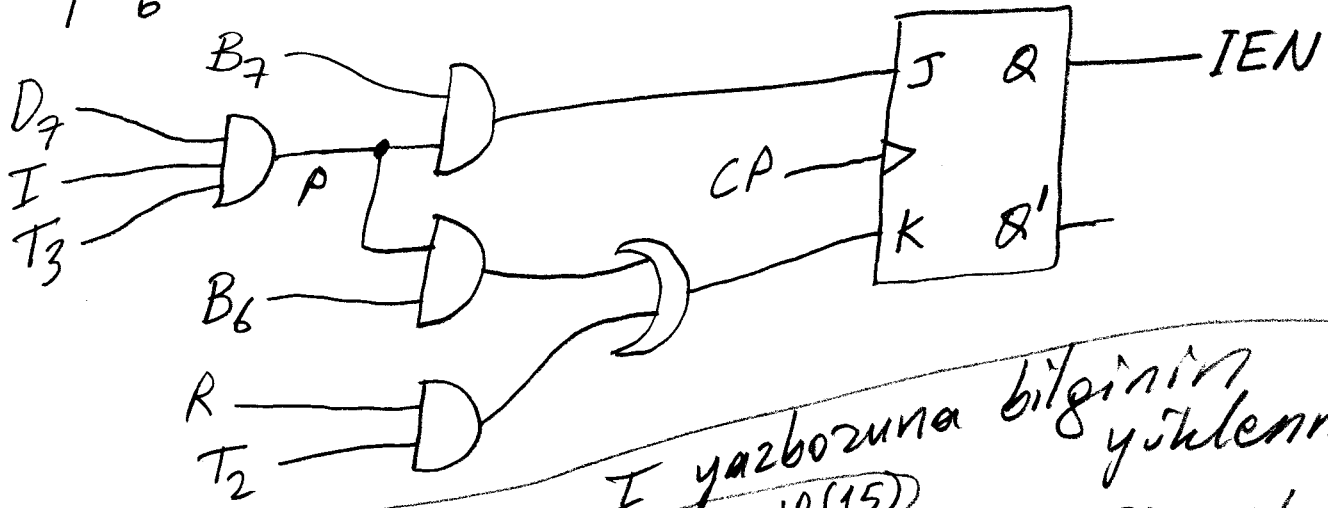
IEN için Denetim Girişleri

(8)

$$PB_7 : IEN \leftarrow 1$$

$$PB_6 : IEN \leftarrow 0$$

$$RT_2 : IEN \leftarrow 0$$

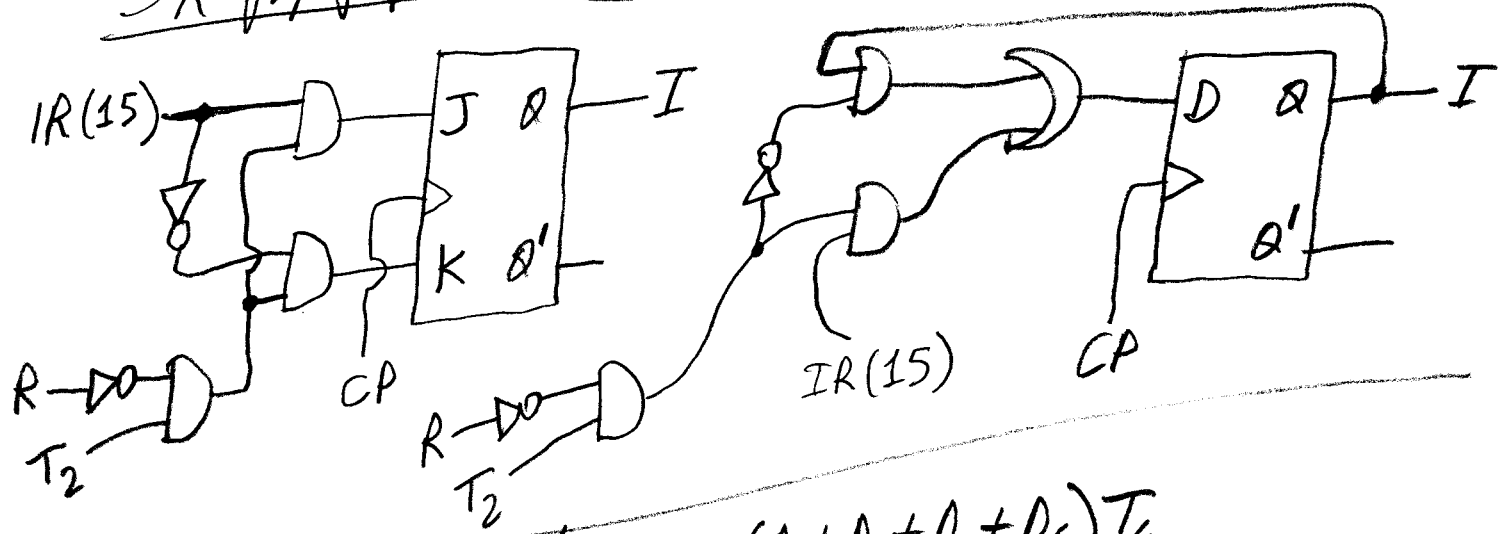


I yazbozuna bilginin yüklenmesi

JK flip-flop ile

$$R'T_2 : I \leftarrow IR(15)$$

D flip-flop ile



$$\text{Bellek (Oku)} = R'T_1 + D_7'I T_3 + (D_0 + D_1 + D_2 + D_6)T_4$$

$$\text{Bellek (Yaz)} = RT_1 + (D_3 + D_5)T_4 + D_6T_6$$

$$SC (CLR) = RT_2 + (D_0 + D_1 + D_2 + D_5)T_5 + (D_3 + D_4)T_4 + D_6T_6 + p + r$$

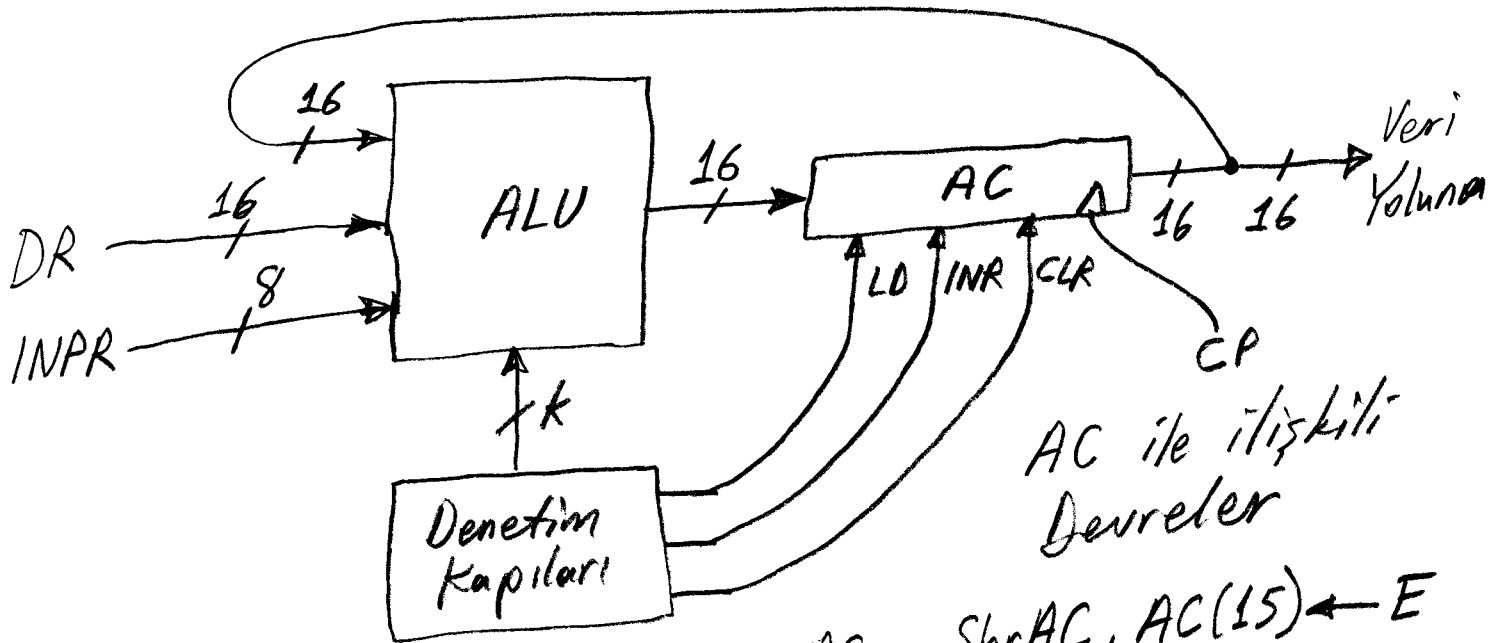
$$p + r = D_7IT_3 + D_7I'T_3 = D_7T_3(I + I') = D_7T_3$$

$$PC(INR) = R'T_1 + RT_2 + D_6T_6Z_{DR} + rB_4AC(15)' + rB_3AC(15) + rB_2Z_{AC} + rB_1E' + pB_9FGI + pB_8F60$$

$$r = D_7I'T_3, \quad p = D_7IT_3, \quad B_i = IR(i)$$

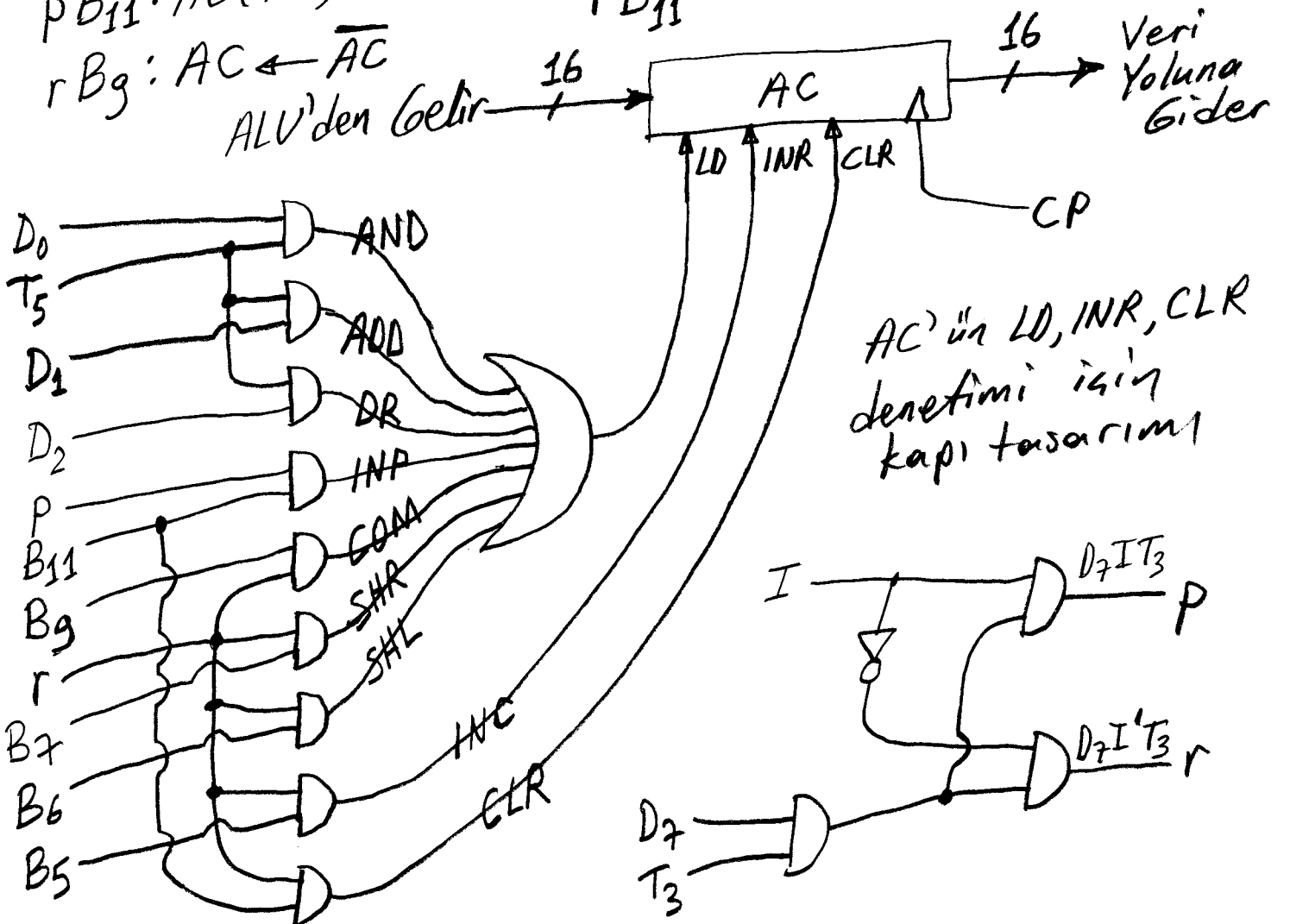
İşlemci Yazıcı Mantık Tasarımı

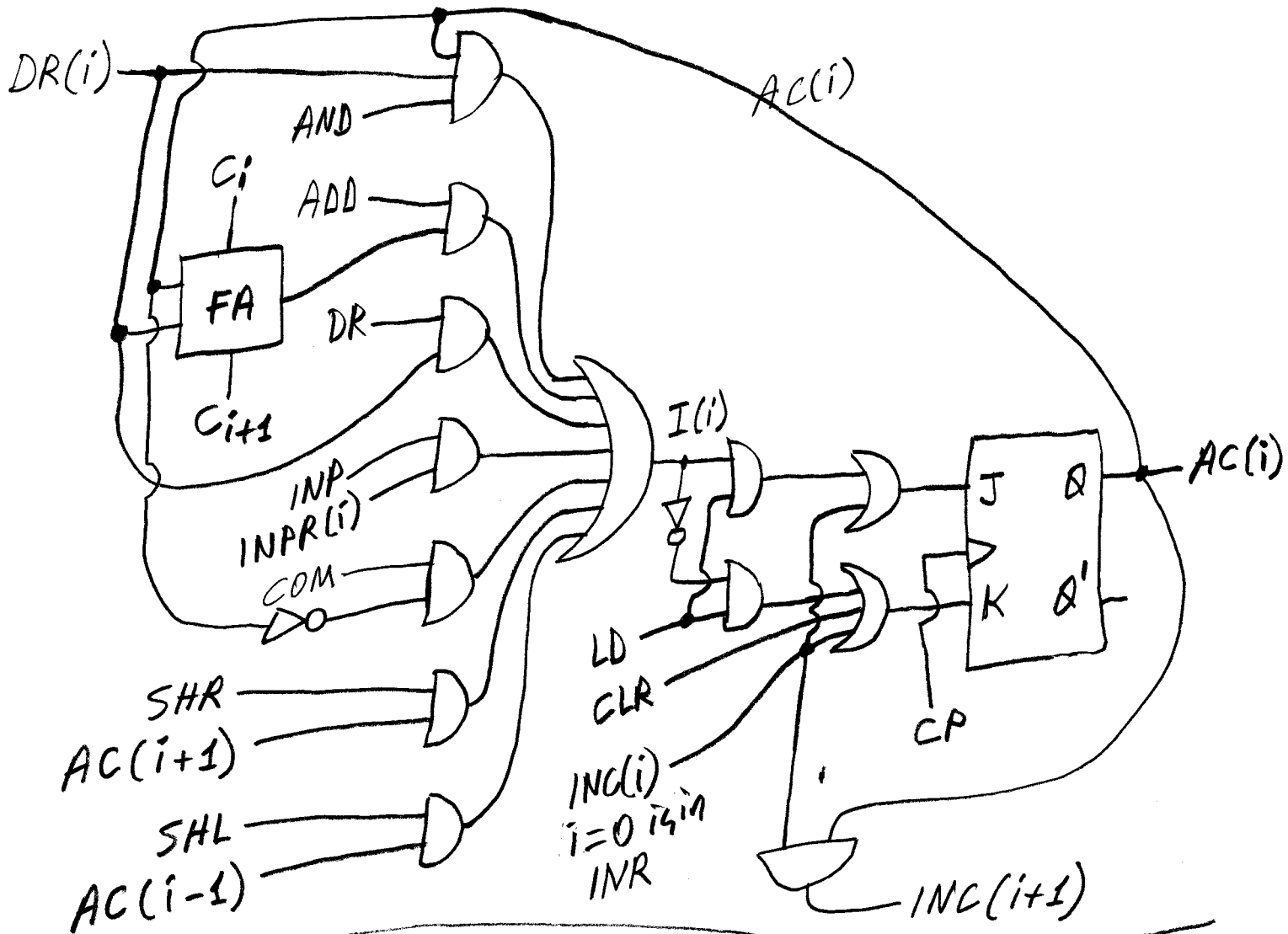
9



$D_0T_5: AC \leftarrow AC \wedge DR$
 $D_1T_5: AC \leftarrow AC + DR$
 $D_2T_5: AC \leftarrow DR$
 $PB_{11}: AC(7-0) \leftarrow INPR$
 $rB_9: AC \leftarrow \overline{AC}$
 ALU'den Gelir

$rB_7: AC \leftarrow ShrAC, AC(15) \leftarrow E$
 $rB_6: AC \leftarrow ShlAC, AC(0) \leftarrow E$
 $rB_5: AC \leftarrow AC + 1$
 $rB_{11}: AC \leftarrow '0$





$$r = D_7 I' T_3$$

$$rB_6 : E \leftarrow AC(15)$$

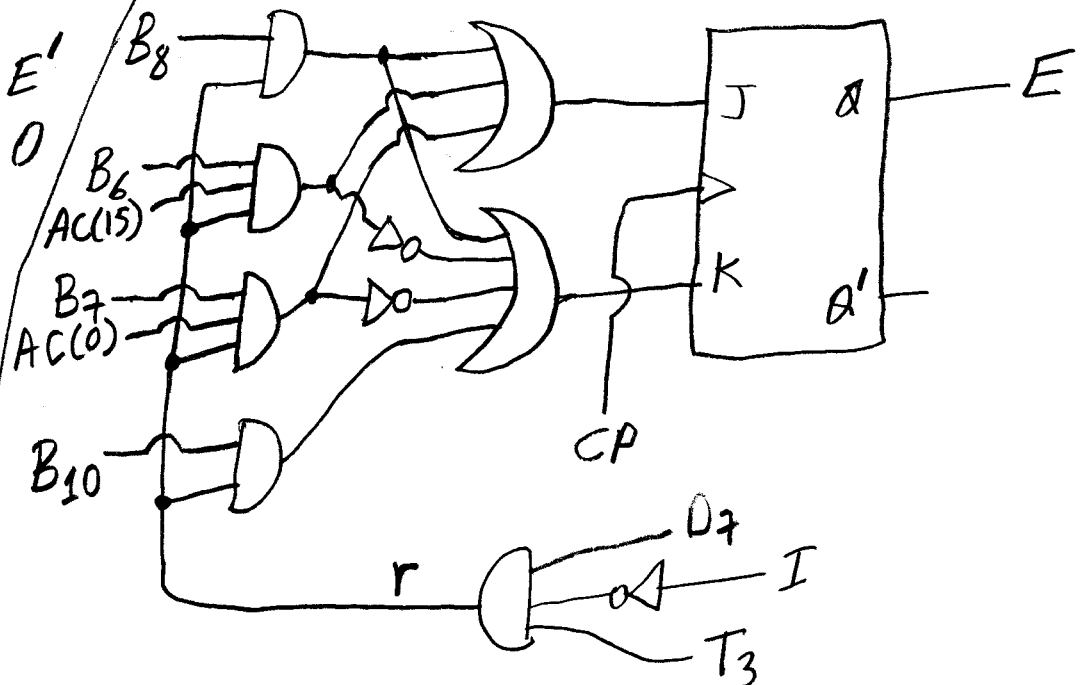
$$rB_7 : E \leftarrow AC(0)$$

$$rB_8 : E \leftarrow E'$$

$$rB_{10} : E \leftarrow 0$$

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

Temel Bilgisayardaki
E yazbozunu tasarla



Temel Bilgisayarın Buyrukları

11

a) Bellek Adreslemeli Buyruklar

Sembol	Onaltılı Kod	Açıklama
	I=0 I=1	
AND	0XXX 8XXX	$AC \leftarrow AC \wedge M[AR]$
ADD	1XXX 9XXX	$AC \leftarrow AC + M[AR], E \leftarrow \text{Cont}$
LOA	2XXX AXXX	$AC \leftarrow M[AR]$
STA	3XXX BXXX	$M[AR] \leftarrow AC$
BUN	4XXX CXXX	$PC \leftarrow AR$ Şartsız Dallar
BSA	5XXX DXXX	$M[AR] \leftarrow PC, PC \leftarrow AR+1$ Dallar ve geri dönüş adresini sakla
ISZ	6XXX EXXX	$M[AR] \leftarrow M[AR] + 1, M[AR] + 1 = 0$ ise $PC \leftarrow PC + 1$ Artır ve eğer sıfır ise atla

b) Yazma Adreslemeli Buyruklar

Sembol	Onaltılı Kod	Açıklama
CLA	7800	$AC \leftarrow 0$ Akümülatörü Sil
CLE	7400	$E \leftarrow 0$ Elderi Sil
CMA	7200	$AC \leftarrow \overline{AC}$ Akümülatörün Tümleyenini Al
CME	7100	$E \leftarrow \overline{E}$ Eldenin Tümleyenini Al
CIR	7080	$AC \leftarrow \text{Shr } AC, AC(15) \leftarrow E, E \leftarrow AC(0)$ Dairesel Sağa Kaydır
CIL	7040	$AC \leftarrow \text{Shl } AC, AC(0) \leftarrow E, E \leftarrow AC(15)$ Dairesel Sola Kaydır
INC	7020	$AC \leftarrow AC + 1$
SPA	7010	$AC(15) = 0$ ise $PC \leftarrow PC + 1$ AC pozitif ise sonraki buyruğu atla
SNA	7008	$AC(15) = 1$ ise $PC \leftarrow PC + 1$ AC negatif ise sonraki buyruğu atla
SZA	7004	$AC = 0$ ise $PC \leftarrow PC + 1$
SZE	7002	$E = 0$ ise $PC \leftarrow PC + 1$
HLT	7001	$S \leftarrow 0$ Programı Durdur

c) Giriş - Çıkış Adreslemeli Buyruklar

Sembol	Onaltılı Kod	Açıklama
INP	F800	$AC(7-0) \leftarrow INPR, FGI \leftarrow 0$
OUT	F400	$OUTR \leftarrow AC(7-0), FGO \leftarrow 0$
SKI	F200	$FGI = 1$ ise $PC \leftarrow PC + 1$
SKO	F100	$FGO = 1$ ise $PC \leftarrow PC + 1$
ION	F080	$IEN \leftarrow 1$ Kesmeyi Aktif Yap
IOF	F040	$IEN \leftarrow 0$ Kesmeyi Pasif Yap

Temel Bilgisayarın Programlanması

(12)

Temel Bilgisayarın komut kümesini kullanarak aşağıdaki işleri yapan altprogramları gerçekleştir

$$C \leftarrow A + B$$

TPL,
LDA A
ADD B
STA C
BUN TPL, I

$$C \leftarrow A \wedge B$$

VE,
LDA A
AND B
STA C
BUN VE, I

$$C \leftarrow A \oplus B$$

$$\leftarrow A' \wedge B \vee A \wedge B'$$
$$\leftarrow ((A' \wedge B)' \wedge (A \wedge B')')$$

EXR,

LDA A
CMA
AND B
CMA
STA C
LDA B
CMA
AND A
CMA
AND C
CMA
STA C
BUN EXR, I

$$C \leftarrow A - B$$
$$\leftarrow A + \bar{B} + 1$$

CKR,
LDA B
CMA
INC
ADD A
STA C
BUN CKR, I

$$C \leftarrow A \vee B$$
$$\leftarrow (A' \wedge B')'$$

VYA,
LDA A
CMA
STA C
LDA B
CMA
AND C
CMA
STA C
BUN VYA, I

$$F \leftarrow A \vee B \wedge C$$
$$\leftarrow (A' \wedge (B \wedge C'))'$$

LOJ,
LDA A
CMA
STA F
LDA B
AND C
CMA
AND F
CMP
STA F
BUN LOJ, I

$$F \leftarrow A - B - C$$
$$\leftarrow A - (B + C)$$

ISL,
LDA B
ADD C
CMA
INC
ADD A
STA F
BUN ISL, I

$A > B$ ise YR1'e git
 $A = B$ ise YR2'e git
 $A < B$ ise YR3'e git

LDA B
CMA
INC
ADD A
SPA
BUN YR3
SZA
BUN YR1
BUN YR2

B ← A-1 işlemini yapan DEC isimli altprogramı 45 farklı yollarla yazınız. (13)

DEC, _____
LDA A
CMA
INC
CMA
STA B
BUN DEC, I

DEC, _____
CLA
CMA
ADD A
STA B
BUN DEC, I

DEC, _____
CLA
INC
CMA
INC
ADD A
STA B
BUN DEC, I

BSA MOV
HEX 100
HEX 200
DEC -16
HLT

PT1, _____
PT2, _____
CTR, _____

MOV, HEX 0
LDA MOV, I
STA PT1
ISZ MOV
LDA MOV, I
STA PT2
ISZ MOV
LDA MOV, I
STA CTR
ISZ MOV
LOP, LDA PT1, I
STA PT2, I
ISZ PT1
ISZ PT2
ISZ CTR
BUN LOP
BUN MOV, I

Herhangi bir dil ile yazılmış kod parçaları

$F = 0$ → CLA
STA F

$F = F + A + B$

→ LDA F
ADD A
ADD B
STA F

$A = A - B$

→ LDA B
CMA
INC
ADD A
STA A

$B = A + B$

→ LDA B
ADD A
STA B

Yanda verilen program temel bilgisayarın bellek biriminde saklıdır.

Her buyruk çalıştıktan sonra onaltılık olarak AC, PC, IR yazarsanızın içeriklerini yazınız.

Adres	Komut
100	CLA
101	ADD 106
102	BUN 104
103	HLT
104	AND 107
105	BUN 103
106	A2DC
107	7EA9

Adres	Komut	AC	PC	IR
100	CLA	0000	101	7800
101	ADD 106	A3E5	102	1106
102	BUN 104	A3E5	104	4104
103	HLT	8041	104	7001
104	AND 107	8041	105	0107
105	BUN 103	8041	103	4103
106	A3E5			
107	9C4B			

A3E5 → 1010 0011 1110 0101
9C4B → 1001 1100 0100 1011

1000 0000 0100 0001
8 0 4 1

Altprogram kullanan bir program

(15)

Adres	Program
	ORG 100
100	LDA X
101	BSA SH4
102	STA X
103	LDA Y
104	BSA SH4
105	STA Y
106	HLT
107	X, HEX 1234
108	Y, HEX 4321

Adres	Program
109	SH4, HEX 0
10A	CIL
10B	CIL
10C	CIL
10D	CIL
10E	AND MSK
10F	BUN SH4
110	MSK, HEX FFF0
	END

Temel bilgisayarın komut kümesini kullanarak

a) Bir karakter alma işlemini yapan altprogram

b) Bir karakter basma işlemini yapan altprogram

CIF, _____
 LOP, SKI
 BUN LOP
 INP
 OUT
 STA CHR
 BUN CIF, I

CHR, _____
 COF, _____
 LDA CHR
 LOP, SKO
 BUN LOP
 OUT
 BUN COF, I

İki karakter alma işlemi yapan bir altprogram

IN2, _____
 FST, SKI
 BUN FST
 INP
 OUT
 BSA SH4
 BSA SH4
 SCD, SKI
 BUN SCD

INP
 OUT
 BUN IN2, I
 MSK, HEX FFF0
 SH4, _____
 CIL
 CIL
 CIL
 CIL
 AND MSK
 BUN SH4, I

Bir Kesme Servis Programı

(16)

Adres	Program
0	ZRO, —
1	BUN SRV
...	...
100	CLA
101	ION
102	LDA X
103	ADD Y
104	STA Z
...	...

→ Dönüş adresi burada saklanır
→ Servis yordamına şartsiz daller.

→ Kesme burada oluşur.
→ Kesmeden sonra program buraya döner.

200	SRV, STA SAC
	CIR
	STA SE
	SKI
	BUN NXT
	INP
	OUT
	STA PT1, I
	ISZ PT1
NXT, SKO	
BUN	EXT
LDA	PT2, I
OUT	
ISZ	PT2
EXT, LDA	SE
CIL	
LDA	SAC
ION	
BUN	ZRO, I

SAC, —
SE, —
PT1, —
PT2, —

Kesme Servis Yordamı

SAC Akümülatörün saklandığı adres

SE Elden saklandığı adres

PT1 Giriş Buffer Göstergesi

PT2 Çıkış Buffer Göstergesi

→ Daha önce çalıştırılan programa geri dsn.

iki pozitif sayının çarpımını
ile ilgili program

(17)

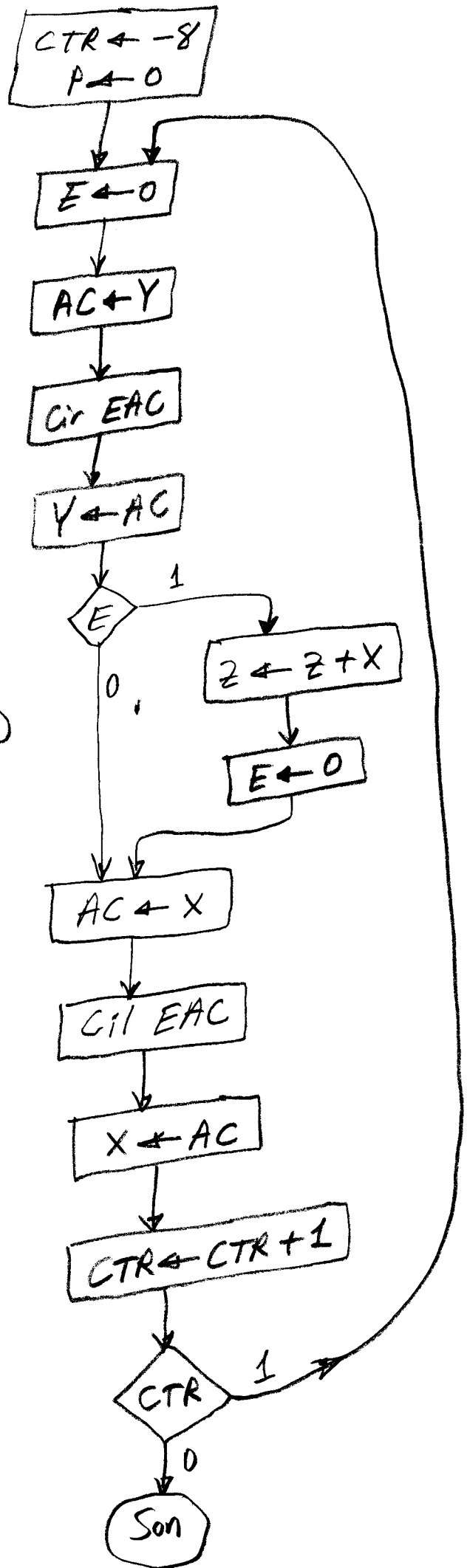
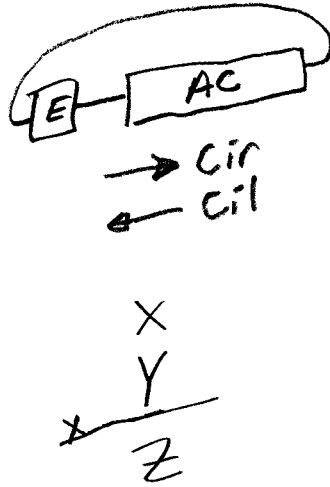
```

ORG 100
LOP, CLE
LDA Y
CIR
STA Y
SZE
BUN ONE
BUN ZRO
ONE, LDA X
ADD Z
STA Z
CLE
ZRO, LDA X
CIL
STA X
ISZ CTR
BUN LOP
HLT

```

CTR, DEX -8
X, HEX C7
Y, HEX 5A
Z, HEX 0

$Z = X * Y$
 ↓ ↓
 Sonuç Çarpılan
 Çarpılan



LOA AL AL, —
 ADD BL AH, —
 STA CL BL, —
 CLA BH, —
 CIL CL, —
 ADD AH CH, —
 ADD BH
 STA CH
 HLT

AH	AL
----	----

BH	BL
----	----

+

CH	CL
----	----

 $F \leftarrow A * B$

Tekrarlı toplama
 mantığına göre
 altprogramı yaz
 $27 * 13$ için
 13 kez 27'nin toplamı

CRP, —
 LOA A
 SZA
 BUN NZR
 BUN ZRO
 NZR, CMA
 INC

STA C
 CLA
 LOP, ADD B
 ISZ C

BUN LOP
 ZRO, STA F
 BUN CRP, I

CRP, —
 CLE
 CLA
 INC
 CIL
 CIL
 CIL
 CIL
 CMA
 INC
 STA C
 LOA A
 STA D
 LOA B
 STA E
 CLA
 STA F

LOP, CLE
 LOA E
 CIR
 STA E
 SZE
 BUN ONE
 BUN ZRO

ONE, LOA D
 ADD F
 STA F
 CLE
 ZRO, LOA D
 CIL
 STA D
 ISZ C
 BUN LOP
 BUN CRP, I

 $F \leftarrow A * B$

A, —
 B, —
 C, —
 D, —
 E, —
 F, —



Veri bloğunun taşınması ile ilgili program

```
BSA MVE
HEX 100
HEX 200
DEC -16
HLT

PT1, _____
PT2, _____
CTR, _____
MVE, _____

LDA MVE, I
STA PT1
ISZ MVE
LDA MVE, I
STA PT2
ISZ MVE
LDA MVE, I
STA CTR
ISZ MVE
LOP, LDA PT1, I
STA PT2, I
ISZ PT1
ISZ PT2
ISZ CTR
BUN LOP
BUN MVE, I
```

Veriler, veri sayısı ve adresler korunuyor.

100 sayının toplanmasını yapan sembolik program (19)

```
ORG 100
100, LDA ADS
STA PTR
LDA NBR
STA CTR
CLA
LOP, ADD PTR, I
ISZ PTR
ISZ CTR
BUN LOP
STA SUM
HLT
ADS, HEX 150
PTR, HEX 0
NBR, DEC -100
CTR, HEX 0
SUM, HEX 0
```

ORG 150
150, DEC 75 ilk veri

DEC 23 Son veri
END → Sembolik programın sonu

A adresindeki
veri bozulmuyor.
B ← içindeki bit sayısı (A)

SAY, _____
CLE
CLA
STA B
LDA A
SZA
BUN G01
BUN G02

G01, CIL
SZE
BUN G03
BUN G01

G03, CLE
ISZ B
SZA
BUN G01

G02, BUN SAY, I

B ← Bitset Ters Çevir (A) (20)

C Sayı 9

G Geçici bellek adresi

A adresindeki veri
bozulmuyor.

TRS, _____

LDA A

STA G

CLE

CLA

INC

CIL

CIL

CIL

CIL

CMA

INC

STA C

LOP, LDA G

CIL

STA G

LDA B

CIR

STA B

ISZ C

BUN LOP

BUN TRS, I

T.C.
DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MİMARİSİ DERSİ FİNAL SINAVI

18.05.2018, Cuma
Saat 15³⁰

Not: Toplam Süre 45 dakikadır. Her bir soru 35 puandır.
Sadece 3 soru çözülecektir. Sorular öğrencide kalabilir.

Soru 1: Temel bilgisayarın belleğini ve yazaçlarını ortak veri yoluna bağlayan kodlayıcı için $x_1, x_2, x_3, x_4, x_5, x_6, x_7, S_0, S_1, S_2$ giriş değerlerini bulunuz.

Soru 2: Aşağıda verilen bazı devre elemanlarının bazı girişleri için karşılık gelen eşitlikleri bulunuz. Derste anlatılan temel bilgisayar mantığına göre çözünüz ve tasarım yapmayınız.

- a-) Bellek (Oku) = ? b-) Bellek (Yaz) = ?
c-) PC (Artır) = ? d-) SC (Sil) = ?
e-) AR (Yükle) = ? f-) IR (Yükle) = ?

Soru 3: Aşağıdaki işlemi yapan program parçasını temel bilgisayarın komut kümesini kullanarak yazınız. *If A > B then goto YR1, elseif A = B then goto YR2, else goto YR3*

Soru 4: Derste anlatılan temel bilgisayardaki komut kümesini kullanarak

- a) Toplama işlemini yapan altprogramı yazınız ($C \leftarrow A + B$)
b) Çıkarma işlemini yapan altprogramı yazınız ($C \leftarrow A - B$)
c) Ve işlemini yapan altprogramı yazınız ($C \leftarrow A \wedge B$)
d) Veya işlemini yapan altprogramı yazınız ($C \leftarrow A \vee B$)
e) Exor işlemini yapan altprogramı yazınız ($C \leftarrow A \oplus B$)

Soru 5: Derste anlatılan temel bilgisayardaki bazı buyrukları aşağıdaki buyruklar ile değiştirelim. ALU devresinde herhangi bir değişiklik yapılmıyor. Her bir komut için T_4 anından başlayarak gerekli mikro işlemleri sırasıyla yazınız.

Sembol	İşlem Kodu	Sembolik Gösterim
OR	000	$AC \leftarrow AC \vee M[AR]$
SUB	001	$AC \leftarrow AC - M[AR]$
XCH	010	$AC \leftrightarrow M[AR]$
ADM	011	$M[AR] \leftarrow M[AR] + AC$

T.C.
DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİLGİSAYAR MİMARİSİ DERSİ FİNAL BÜTÜNLEME SINAVI

06.06.2018, Çarşamba
Saat 17⁰⁰

Not: Toplam Süre 45 dakikadır. Her bir soru 35 puandır.
Sadece 3 soru çözülecektir. Sorular öğrencide kalabilir.

Soru 1: Temel bilgisayardaki *ALU* (Aritmetik Mantık Birimi) devresinin bir bitlik kısmını tasarlayınız.

Soru 2: Temel bilgisayardaki İşlemci Yazacının (*AC*) *LD* , *INC* , *CLR* denetimi için kapı tasarımını gerçekleyiniz.

Soru 3: Temel bilgisayarın komut kümesini kullanarak A adresindeki veriyi bozmadan içindeki birlerin sayısı bulup sonucu B adresine atayan altprogramı yazınız.

Soru 4: Temel bilgisayarın komut kümesini kullanarak A adresindeki veriyi bozmadan içindeki verinin bitlerini ters çevirip sonucu B adresine atayan altprogramı yazınız. Geçici bellek adresi olarak G değişkenini kullanınız.

Soru 5: Aşağıda verilen program temel bilgisayarın bellek biriminde saklıdır. Her buyruk çalıştıktan sonra onaltılık olarak AC, PC ve IR kaydedicilerinin içeriklerini bu alanda gösteriniz.

Adres	Komut	AC	PC	IR
100	CLA	?	?	?
101	ADD 106	?	?	?
102	BUN 104	?	?	?
103	HLT	?	?	?
104	AND 107	?	?	?
105	BUN 103	?	?	?
106	A3DC	?	?	?
107	7EA9	?	?	?