

Chapter 1

Digital Systems and Binary numbers

Analog Sinyal: Sürekli değişim gösteren sinyaldir.

Digital Sinyal: Belirli aralıklarda değişim gösterir.

→ Digitallesmenin en önemli amacları analog sinyali özetleyerek gürültiden arıtmaya ve daha iyi işlenmesini sağlamaktır.

→ örnekleme yaparken ne kadar fazla frekans yani kuantum detaylı takasak o kadar iyi (doğru) sonuc alırsın.

→ Kuantalama veya örnekleme hattası çözünürlük ortasına uzatır.

Örnek

Binary Numbers: Ord 1

① each digital represents a power of 2, starting w 2^0

Hexadecimal: base 16

① digits (0-9) and A-F (10-15)

② each digit represent a power of 16, starting w 16^0

③ to convert to/from binary: convert each hex digit to 4 binary digit

④ when convert from binary/hex/octal \rightarrow multiply

Octal : base 8

- Digits : 0-7
 - each digit represent a Power of 8
 - to convert to/from binary : convert each octal to 3 binary

Binary addition method:

- Align numbers vertically and add corresponding bits, from right
 - Carry over to the next bit if the sum is greater than or equal to 2

Binary subtraction methods:

→ One's complement:

- negative numbers r represented by (hafha bir kalmasi) ^{cide}

$$\begin{array}{r}
 26 \\
 -11 \\
 \hline
 15
 \end{array}
 \quad
 \begin{array}{r}
 11010 \text{ original} \\
 1011 \quad +1 \\
 \hline
 0100
 \end{array}
 \quad
 \begin{array}{r}
 11010 \\
 +1 \\
 \hline
 1110
 \end{array}
 \quad
 \left. \begin{array}{r}
 1111 \\
 =15
 \end{array} \right\}$$

$11110 \rightarrow$ odd carry

- if we have no carry we take the one comp to the result
such negatif

→ Two's complement:

$$\begin{array}{r} 1010 \\ - 0111 \\ \hline \end{array} \quad \begin{array}{r} 10 \\ - 7 \\ \hline \end{array} \quad \left. \begin{array}{r} 1 \\ + 1000 \\ \hline 1001 \end{array} \right\} \text{2's complement} \quad \rightarrow \quad \begin{array}{r} 1010 \\ + 1001 \\ \hline \end{array} \quad \boxed{\text{result}} \\ \text{Ignore } \boxed{1} \rightarrow 0011 \rightarrow 0011 \rightarrow \boxed{3}$$

- if we have no carry we take the 2's comp. to the final result
sova negative \uparrow

Chapter - 2 -

Logik Devre Temelleri:

Sayısal Kodlama:

- ① BCD - binary Coded Decimal: ondalık sayının her hanesinin ikili olarak kodlanmasıdır.

$$(25)_{10} \rightarrow \underbrace{0010}_2 + \underbrace{0101}_5 = \boxed{0010\ 0101}$$

- ② Üç Fazlılık Kodu: ikili sistemin üç fazası alınarak oluşturulmuş. Simetrikdir.

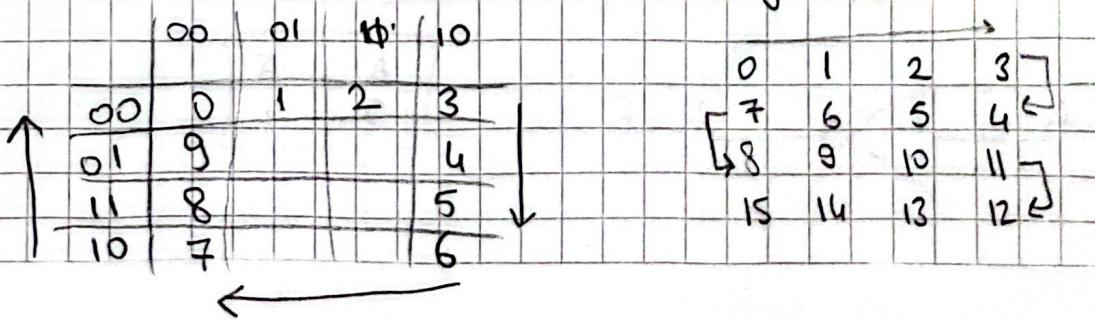
0	0011	
g	1100	simetrik
1	0100	
8	1011	

- ③ Aiken Kodu: (0-4) arasındaki ilk beş sayının ikili kodlamaya eşit, (5-9) arasındaki beş sayının ise ilk beş sayının 1'e tamleyenin olduğu söylenebilir.

$$\begin{aligned} 1 &\rightarrow 0001 \\ 8 &\rightarrow 1110 \end{aligned}$$

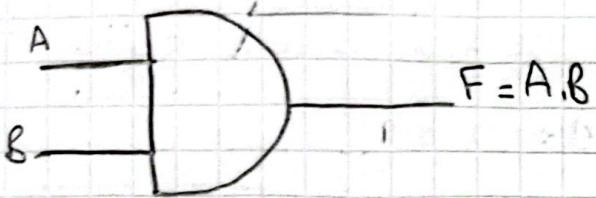
- ④ Bitişik Kodlar (Gray): birbirini izleyen sayıları kapsıltı alırak ikili kod sözcükleri arasındaki uzaklığı (hamming) 1 ise bitişik kod denir.

Gray Fork:

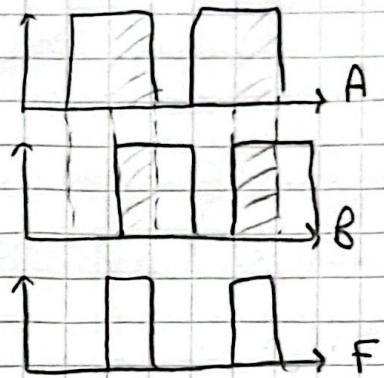
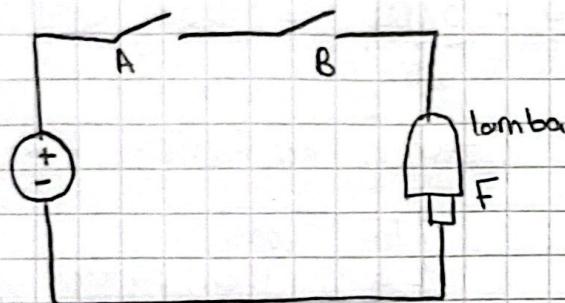


Boole Cebri:

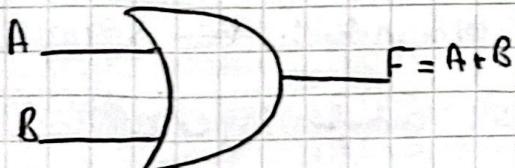
① Ve islemi AND:



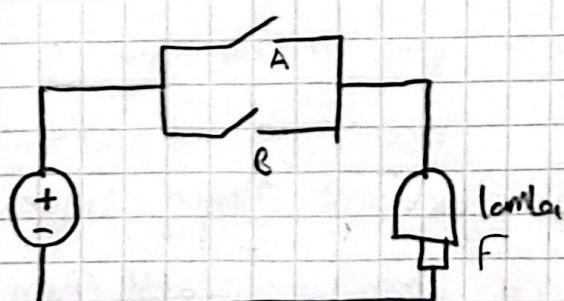
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



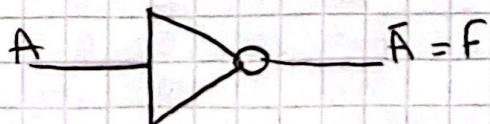
② Veya islemi OR:



A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



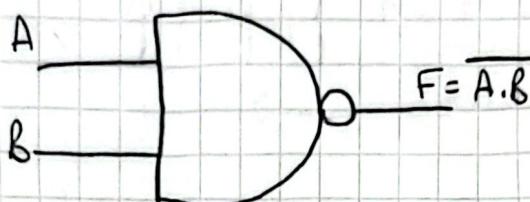
③ Tumleme islemi (NOT):



A	F
0	1
1	0

④ Tümleyen Ve işlemi (NAND) :

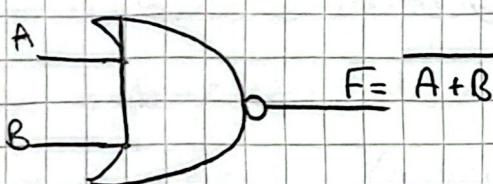
Ve' nin değili:



A	B	$F = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

⑤ Tümleyen Veya işlemi (NOR) :

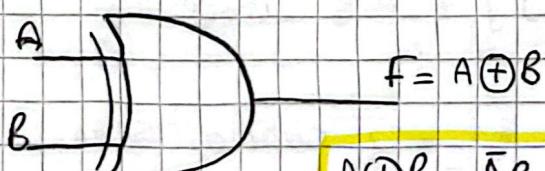
Veya'nın değili:



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

⑥ Ya Da - özel Veya - (XOR) :

farklı olduklarında 1 aynı olduklarında 0

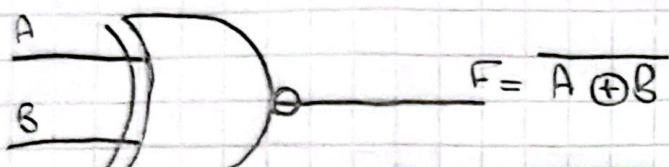


$$A \oplus B = \overline{AB} + A\overline{B}$$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

⑦ T4a Da (anak ve anak) islemi: (XNOR):

oynilarsa 1 forklilarsa 0



A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$$A \odot B = AB + \bar{A}\bar{B}$$

- Soru -

A	1	1	0	1	0	1
B	1	1	0	0	0	1
F	1	1	1	0	1	1

XNOR

oyni $\rightarrow 1$ forklı $\rightarrow 0$

→ Basit Kavramlar

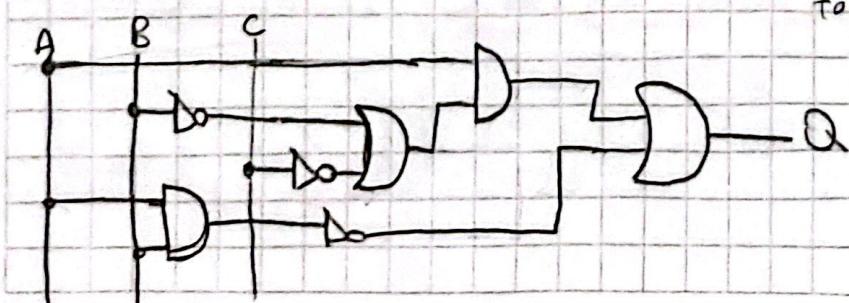
$$A + \bar{A} = 1 \quad 1 + A = 1 \quad 0 + A = A$$

$$A \cdot \bar{A} = 0 \quad 1 \cdot A = A \quad 0 \cdot A = 0$$

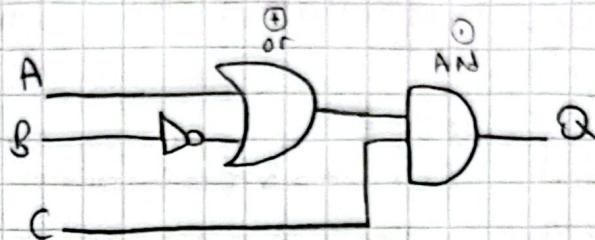
Soru:

$Q = [A \cdot (\bar{B} + \bar{C})] + \bar{A} \cdot \bar{B}$ ifadesinin
lojik diyagramini çiziniz

Carpma \rightarrow ve
toplana \rightarrow veya



Soru: Aşağıdaki lojik devresinin ifadesini yazınız.



$$[(A + \bar{B}) \cdot C] = Q$$

Teoremler:

$$\rightarrow a + b + c = (a + b) + c$$

$$\rightarrow a \cdot b \cdot c = (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

$$\rightarrow a + (b \cdot c) = (a + b) \cdot (a + c)$$

$$\rightarrow a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$\rightarrow a + \overline{a} \cdot b = a$$

$$\rightarrow a \cdot (a + b) = a$$

birlesme

Dağılma

yutma

DeMorgan Kuralı

$$\overline{(a + b + c + \dots)} = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \dots$$

$$\overline{(a \cdot b \cdot c \cdot \dots)} = \bar{a} + \bar{b} + \bar{c} + \dots$$

$$\text{Örnek: } Q = B \cdot C + B \cdot (C + A) + C \cdot (B + A)$$

ifadesini boolean teoremleri ile sadeleştir.

$$(B \cdot C) + (B \cdot C) + (B \cdot A) + (C \cdot B) + (C \cdot A)$$

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

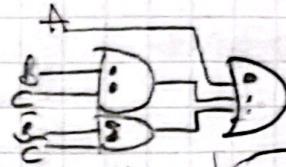
$$\overline{A \cdot B \cdot (A + C)}$$

Circuit Optimization:

→ Literal cost (L)

→ Gate input cost (G)

→ Gate input cost with Not's (GN)



Literal Cost : (L) → Sadece bilinmeyenler \rightarrow Sadece bilinmeyenler
the number of gates \rightarrow inversion + the normal one

$$\text{Gate input cost (G)} : \rightarrow 5 + 2 = 7$$

$\overbrace{\quad\quad\quad}^{5}$ $\underbrace{\quad\quad\quad}_{2}$

$\overbrace{B \cdot \bar{C} + \bar{B} \cdot C}^{7+2=9}$

Gate input cost with Not's (GN) : Not koflansın
deşay

$$\underline{7+2=9}$$

Minterms :

Fonksiyonun doğru olduğu durumları ifade eder. (1)

$$F(A, B, C) = \sum(0, 2, 5, 7) \rightarrow \text{çikısları 1 olan durumlar}$$

\rightarrow çarpımların toplamı olarak hesaplanır (SOP)

Maxterms :

Fonksiyonun yanlış olduğu durumları ifade eder (0)

$$F(A, B, C) = \sum(1, 3, 4, 6) \rightarrow \text{çikısları '0' olan durumlar.}$$

\rightarrow toplamların çarpımı olarak hesaplanır (POS) $\rightarrow 0$

chapter - 3 -

Karnaugh Tablosu :

$\rightarrow 2^2 = 4$

iki girīeli

	B	
A'	$A'B'$	$A'B$
A	AB'	AB

$2^3 = 8$

3 girīeli

	BC			
A'	$B'C'$	$B'C$	BC	BC'
A	$A'B'C'$	$A'B'C$	$A'BC$	$A'BC'$

$2^4 = 16$

	CD				
AB	$C'D'$	$C'D$	CD'	CD	
$A'B'$	$A'B'C'D'$	$A'B'C'D$	$A'B'CD'$	$A'B'CD$	
$A'B$	$A'B'C'D'$	$A'B'C'D$	$A'BCD'$	$A'BCD$	
$A'B'$	$ABC'D'$	$ABC'D$	$ABC'D'$	$ABC'D$	
AB	$ABC'D'$	$ABC'D$	$ABC'D'$	$ABC'D$	

! Sadelestirme nasıl yapılır ?

	B'	B
A'	0	1
A	1	1

$f_2 = B$

$f_1 = A$

$$Q = \bar{A}B + AB + A\bar{B} \rightarrow Q = f_1 + f_2 = A + B$$

$A + B$

	$B'C'$	$B'C$	BC'	BC
A'	0	1	1	0
A	1	0	1	1

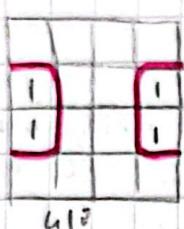
$$Q = \bar{A}BC + \bar{A}\bar{B}C + AB\bar{C} + ABC$$

$$Q = \bar{A}C + AB$$

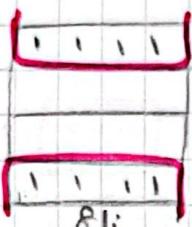
④ kirmizi olan gerekstir grup

$$A'C + AB$$

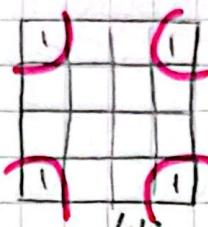
Karnaugh gruplandırımları:



61:



81:



61:

→ Simetrik bir şekilde gruptanılabılır

Quine-McCluskey yöntemi: 6 ve daha yüksek girisler için

- $m(0, 1, 2, 5, 7)$

- $m_0 = 000$
 $[m_1 = 001]$ 1 bit farklı

$m_2 = 010$

$\begin{matrix} 1 \text{ bit} \\ \text{farklı} \end{matrix} \quad [m_5 = 101]$

$m_7 = 111$

- $(m_0, m_1, m_2)_{\text{grup } 1} \rightarrow (m_1, m_5, m_7)_{\text{grup } 2}$
 (sadeleştir 11 bit)

- $(m_0, m_2), (m_1, m_5, m_7)$

- $Q = m_0m_2 + m_1m_5m_7$

Chapter - 4-

Combinational Circuit

Bu devrelerde akışlar yalnızca onluğunu girişlere bağlıdır ve devrenin geçmiş durumu ile ilgili değildir.

Bellek durumu tozmetler, örneğin; AND, OR, NOT gibi basit kapların birleşmesiyle oluşan devredere örnektir.

① Basit Kapları:

- AND, OR, NOT

- AND tüm girişler yüksek (1) çıkışta yüksek, OR'da herhangi bir giriş yüksek dursa çıkış yüksek olur.

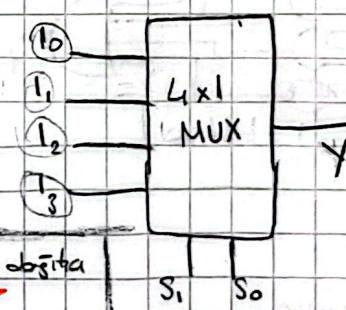
- İki girişli AND \rightarrow çıkış = A ve B

② MUX (multiplexer): Seçici

- birden fazla giriş arasında seçim yapmak için kullanılır. Kontrol sinyali hangi girişin çıkışa ileteceğini söyler

- İki girişli bir MUX:

$$\text{Çıkan} = \begin{cases} \text{Kontrol} = 0 \rightarrow \text{Giriş} = 1 \\ \text{Kontrol} = 1 \rightarrow \text{Giriş} = 2 \end{cases}$$



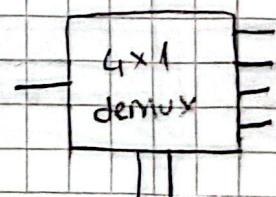
S ₀	S ₁	Y
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃

③ DEMUX (demultiplexer):分布式

- Demux tek giriş, farklı çıkış. Kontrol sinyali hangi çıkışın girişe bağlanacağını belirler.

- tek girişli iki farklı çıkış gösteren demux:

$$\text{Giriş} = \begin{cases} \text{Kontrol} = 0 \rightarrow \text{Çıkan} = 1 \\ \text{Kontrol} = 1 \rightarrow \text{Çıkan} = 2 \end{cases}$$



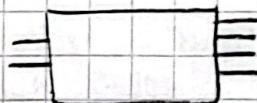
4 Encoder ve Decoder (Kodlayıcılar ve Gözüçüler),

Encoder: birden fazla girişi daha az girişe kodlar

Decoder: kodlanmış bilgiyi orijinal gireklere dönüştür. low active \rightarrow inverter high active \rightarrow normal dm

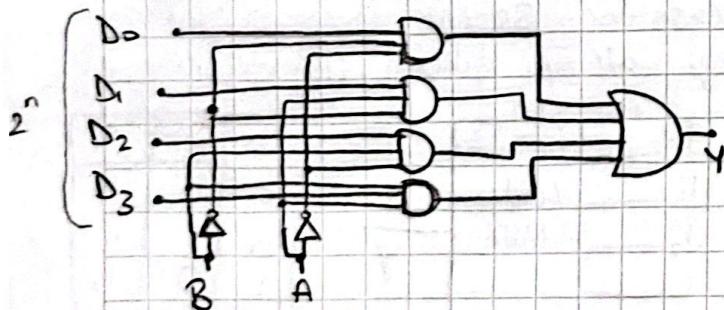
Örneğin: 4-2 Kodlayıcı \rightarrow 4 girişli bir veri setini 2 girişli bir veri setine kodlar.

2-4 Gözüçü \rightarrow 2 girişli bir veri setini 4 farklı (2 giriş - 4 çıkış) çıkışa dönüştüren bir devredir.



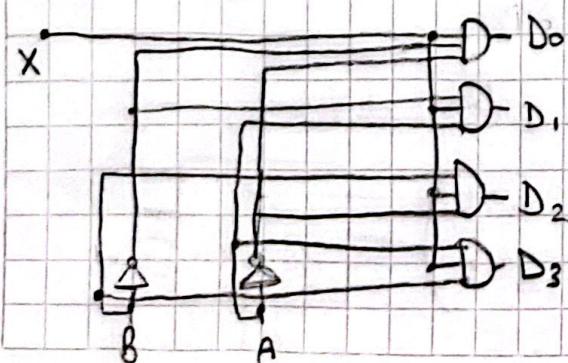
ÖRNEKLER:

① 4 to 1 MUX



$D_0 = \bar{A} \cdot \bar{B}$	D_0	Y	B	A
$D_1 = \bar{B} \cdot A$	D_1		0 0	D_0
$D_2 = B \cdot \bar{A}$	D_2		0 1	D_1
$D_3 = A \cdot B$	D_3		1 0	D_2
			1 1	D_3

② 1 to 4 DEMUX



B	A	D ₀	D ₁	D ₂	D ₃
0	0	X	0	0	0
0	1	0	X	0	0
1	0	0	0	X	0
1	1	0	0	0	X



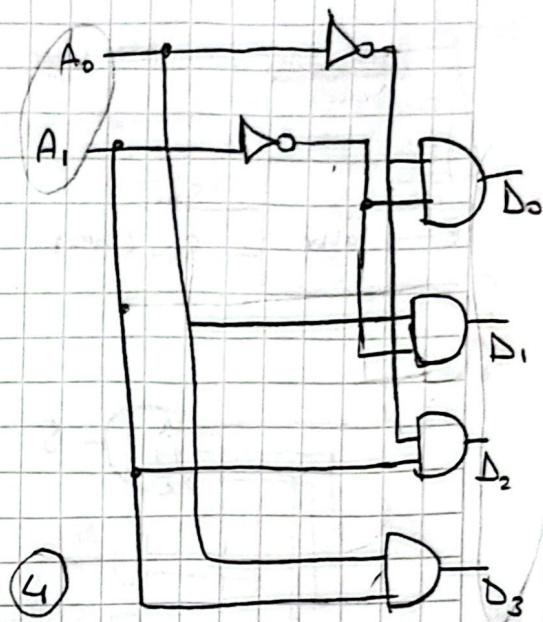
binary counter & binary decoder

Decoder Examples:

2 to 4

A ₁	A ₀	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$n \rightarrow m, m = 2^n \rightarrow 2^2 = 4$$

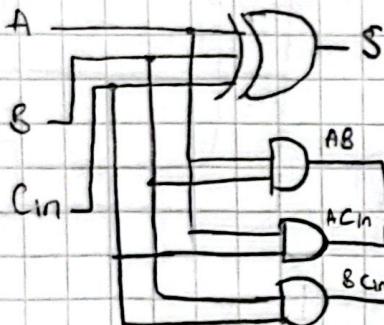
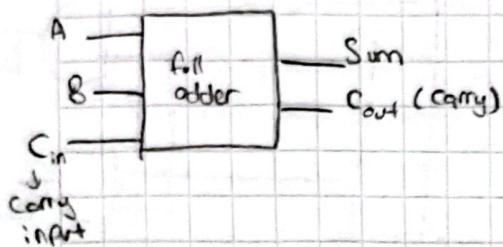


Encoding Example

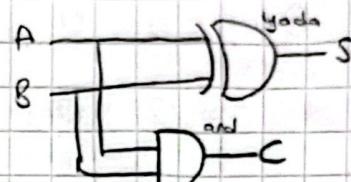
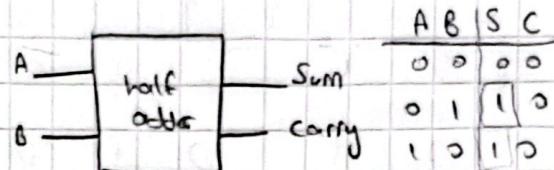
Arithmetic addition - Subtraction circuits:

① Arithmetic addition circuit:

full-adder: compute larger numbers



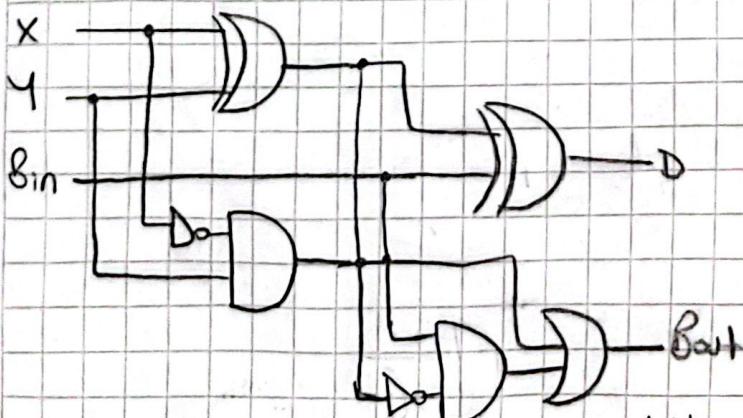
half-adder: compute single digit numbers



A	B	Cin	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

② Arithmetic Subtraction Circuit:

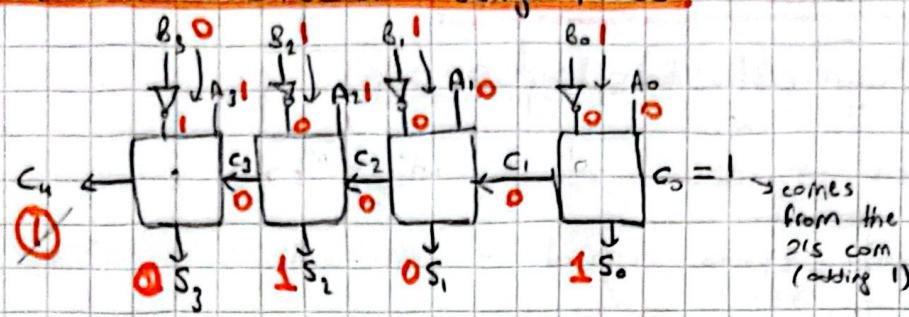
full-subtractor:



X	Y	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\begin{array}{r}
 14 \\
 - 5 \\
 \hline
 9
 \end{array}
 \quad
 \begin{array}{r}
 1110 \\
 - 0101 \\
 \hline
 1001
 \end{array}
 \rightarrow \text{two's comp. } (1011) \rightarrow
 \begin{array}{r}
 1110 \\
 - 1011 \\
 \hline
 01001 \rightarrow 9
 \end{array}$$

4 bit Subtractor using Adder :



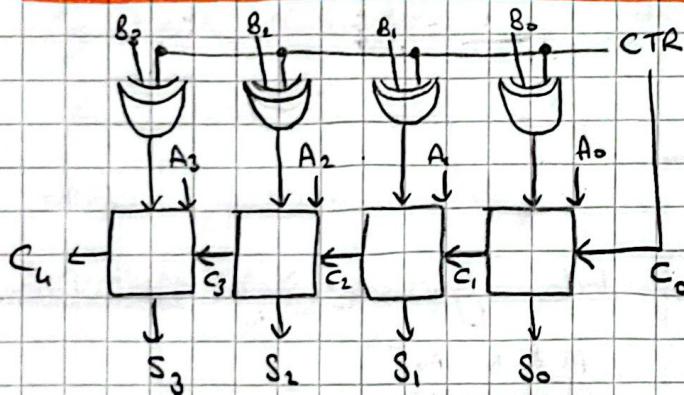
$$\begin{array}{r}
 12 \\
 - 7 \\
 \hline
 5
 \end{array}
 \quad
 \begin{array}{r}
 1100 \\
 - 0111 \\
 \hline
 1010
 \end{array}$$

↓ 2's comp.

$$\begin{array}{r}
 1100 \\
 + 1001 \\
 \hline
 1010
 \end{array}$$

$$\text{Solution} = 0101 = 5$$

4 bit Adder / Subtractor :



$$\begin{array}{r}
 0101 = A \\
 0001 = B \\
 \hline
 \begin{array}{r}
 111 \\
 0101 \\
 + 1111 \\
 \hline
 1010 = 4
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 12 \\
 + 7 \\
 \hline
 \begin{array}{r}
 100 \\
 1100 \\
 + 0111 \\
 \hline
 1001
 \end{array}
 \end{array}$$

Sum

$$\begin{array}{r}
 \text{Adder} \\
 \begin{array}{r}
 111 \\
 + 0011 \\
 \hline
 1010
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 12 \\
 - 7 \\
 \hline
 \begin{array}{r}
 1100 \\
 - 0111 \\
 \hline
 1010
 \end{array}
 \end{array}$$

Subtract

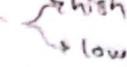
Subtractor

$$\begin{array}{r}
 10 \\
 - 1100 \\
 \hline
 \begin{array}{r}
 1010 \\
 + 0100 \\
 \hline
 1110 = ???
 \end{array}
 \end{array}$$

no carry

complicated
Case

Chapter - 5-

3-state buffer 

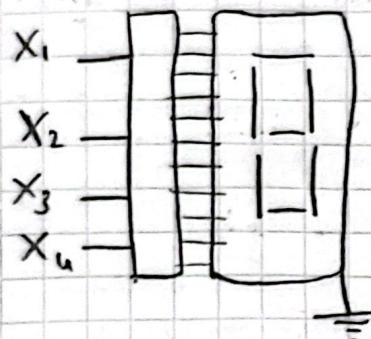
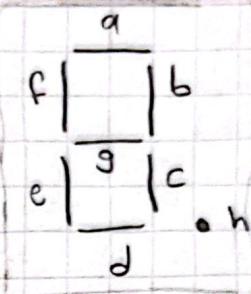
ALU (arithmetic logic unit)

Seven segment Display

1 sayısi \rightarrow bc |

2 Σ sayısi (2) \rightarrow abged

gibi

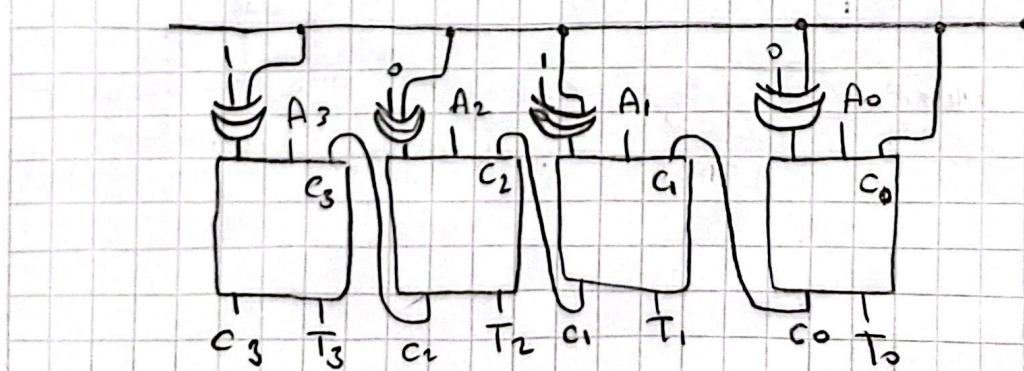
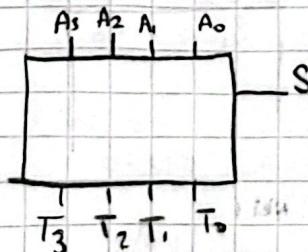


örnek:

bü devresi: tam toplayıcıları yapınız, XOR kullanılsın

$$S_0 \rightarrow T = A + 2$$

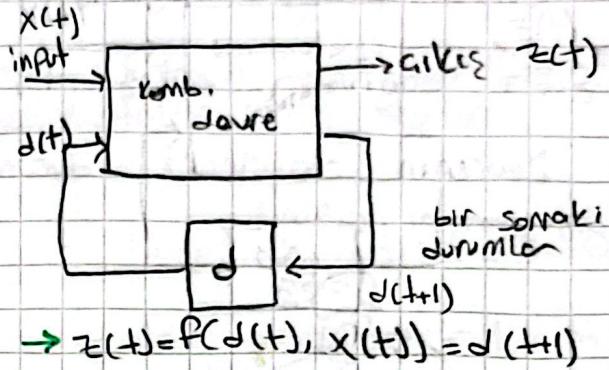
$$S_1 \rightarrow T = A - 2$$



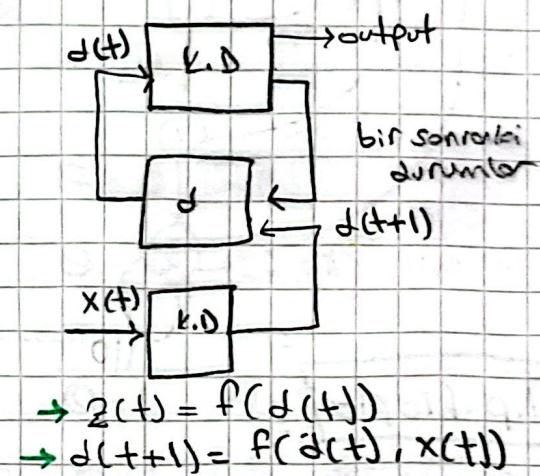
Chapter - 6-

Sequential Circuits

- Mealy makinalarında
t anındaki çıkış değeri
oonda durumlara ve
girişlere bağlıdır.

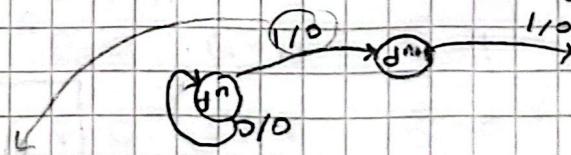


- Moore Makinalarında
ise t anındaki çıkış
değeri yalnızca oonda
durumlara bağlıdır.



Durum diyagramı:

Ardısal devrenin durumlarını ve durumlar arası geçişini gösteren graftır



geçiş koşulu / çıkış değeri

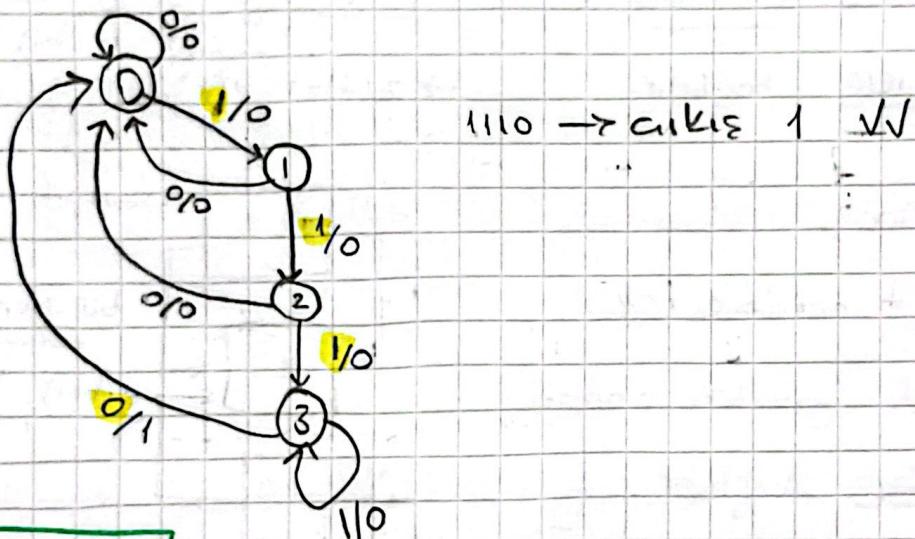
Örnek:

Sifre devresi tasarmı

Sifrenin lojik değerde 1110 olduğu varsayılmaktadır.

Sifre doğrusa çıkış 1, değilse 0 olmalı

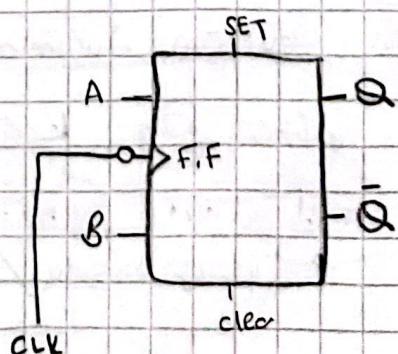
Devrenin obbilecek durumlarını belirleyen ve durum diyagramını çiziniz.



Flip-floplor:

Flip floplor 1 bittlik saklama birimleridir.

Saklayıcı, Sayıcı, bellek vb. birimler oluşturabilir

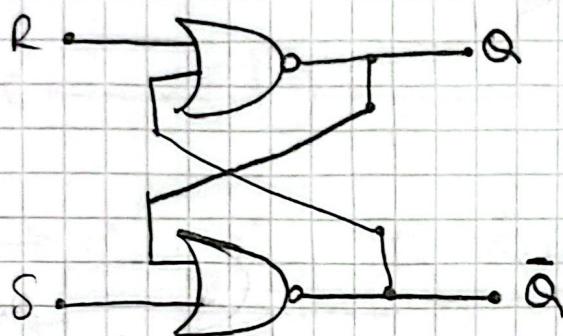


(Set - Reset Latch):

- NOR
- NAND

• basic digital circuit element that can store one bit. 1 or 0.

• also known as a flip-flop or bistable multivibrator



Reset \rightarrow output = 0 = Q

Set \rightarrow output = 1 = Q

Case 1:

$$S=0 \quad R=1 \quad Q=0 \quad \bar{Q}=1$$

$$S=0 \quad R=0 \quad Q=0 \quad \bar{Q}=1 \quad \text{memory}$$

Case 2:

$$S=1 \quad R=0 \quad Q=1 \quad \bar{Q}=0$$

$$S=0 \quad R=0 \quad Q=1 \quad \bar{Q}=0 \quad \text{memory}$$

Case 3 NOT USED !

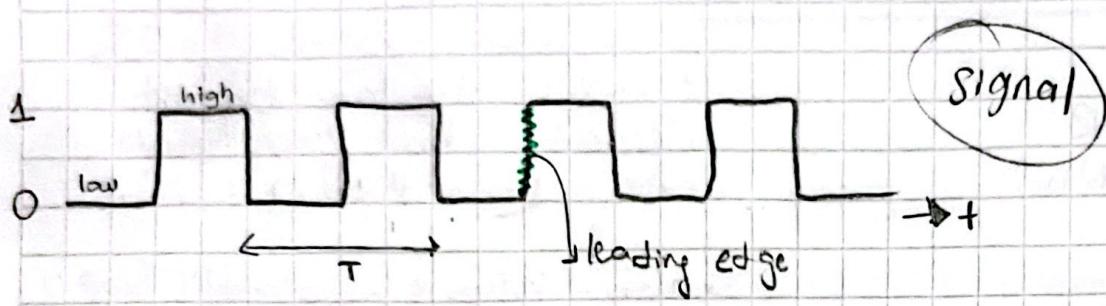
$(1,1) \rightarrow$ for Nor
 $(0,0) \rightarrow$ for Nand

$$S=1 \quad R=1 \quad Q=0 \quad \bar{Q}=0 \quad (X)$$

$$S=0 \quad R=0 \quad Q=0 \quad \bar{Q}=1 \quad] ???$$

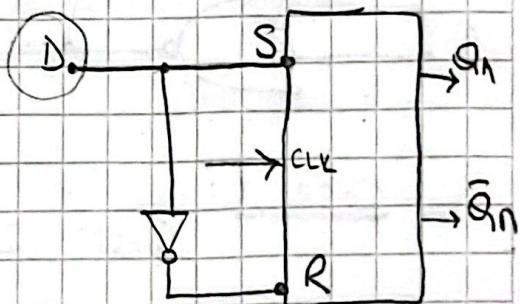
$$S=0 \quad R=0 \quad Q=1 \quad \bar{Q}=0$$

what is clock?



D flip flop (data) ^{or delay}

CLK	S	R	Q_{n+1}
0	x	x	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	invalid



truth table

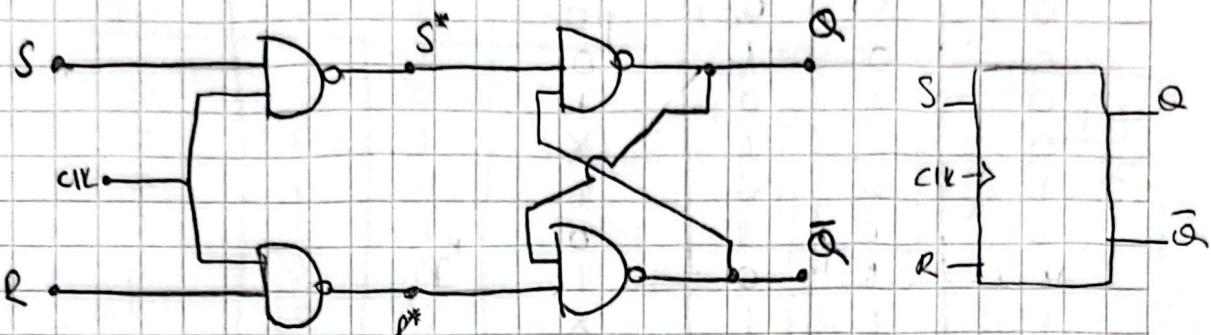
clk	D	Q_{n+1}
0	x	Q_n
1	0	0
1	1	1

characteristic table

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

$$Q_{n+1} = D = S$$

SR flip-flop :



S^*	R^*	Q	\bar{Q}
0	0	not used (Nand) (in Nor (1,1))	
0	1	1	0
1	0	0	1
1	1	memory	

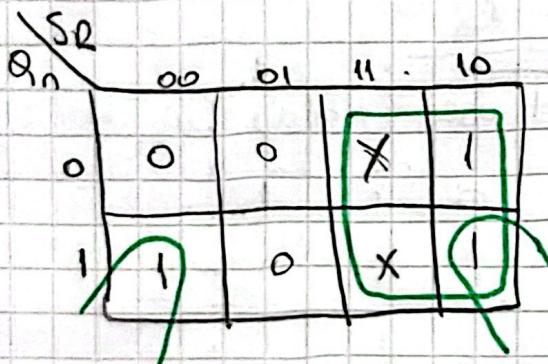
* $S^* = \overline{(S \cdot CLK)} = \overline{S} + \overline{CLK}$ $R^* = \overline{(R \cdot CLK)} = \overline{R} + \overline{CLK}$

CLK	S	R	Q	\bar{Q}
0	x	x	memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	not used	
1	0	0	memory	

Characteristic table:

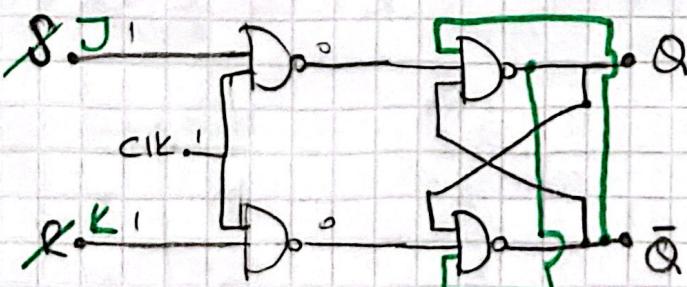
SR flip-flop

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X



$$Q_{n+1} = S + Q_n \bar{R}$$

JK flip flop



Clk	J	K	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	\bar{Q}_n (toggle)

characteristic table (JK):

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	$0' = 1$
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	$1' = 0$

Q_n	00	01	11	10
J_K	0	0	1	1
0	0	0	1	1
1	1	0	0	1
1	0	0	0	1

$$Q_{n+1} = \overline{Q_n J} + Q_n K$$

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

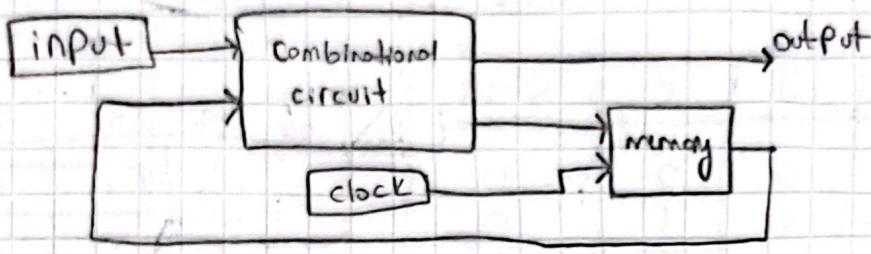
Q_n	0	1
Q_{n+1}	0	1
0	0	x
1	x	x

$$J = Q_{n+1}$$

Q_n	0	1
Q_{n+1}	0	1
0	x	x
1	1	0

$$K = \overline{Q_{n+1}}$$

→ Triggering methods in flip-flops



→ Difference between Latch and flip-flop

→ there is enable in latch



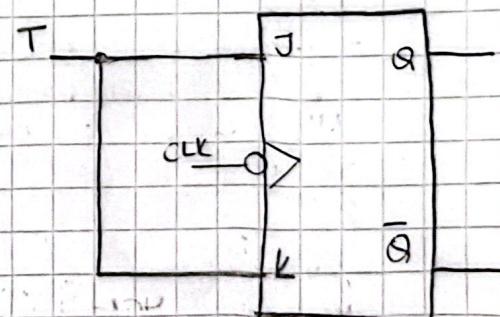
→ there is clock in flip flop



T flip-flop (toggle)

CLK	T	Q _{n+1}
0	X	Q _n
1	0	Q _n
1	1	$\overline{Q_n}$ (toggle)

memory



* ch. table

Q _n	T	Q _{n+1}
0	0	0
0	1	1 → 0'
1	0	1 → 1'
1	1	0 → 1'

(XOR)

* Excitation table

Q _n	Q _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

T=0 → memory

T=1 → $\overline{Q_n}$ → toggle state

$$Q_{n+1} = Q_n \oplus T$$

Full-adder

C	a	b	S	C'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

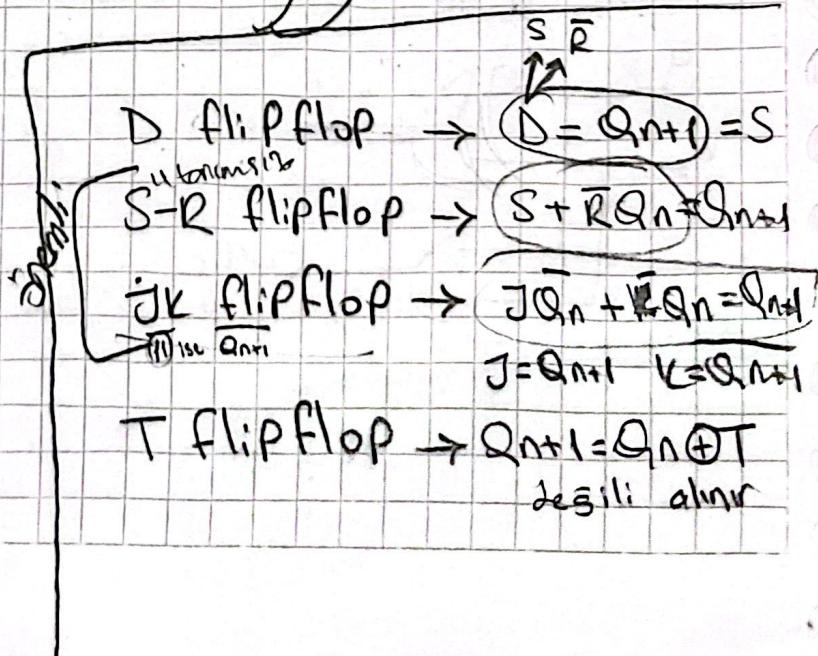
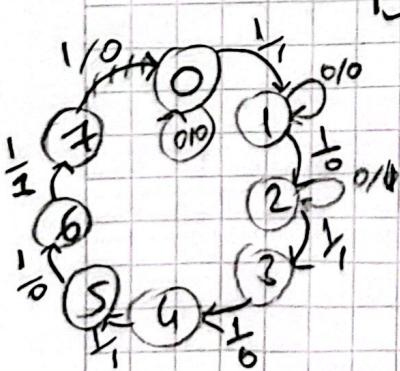
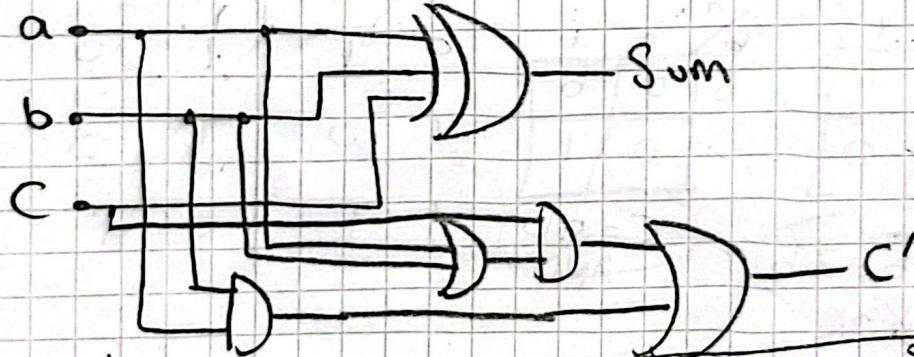
$C/a\ b$

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

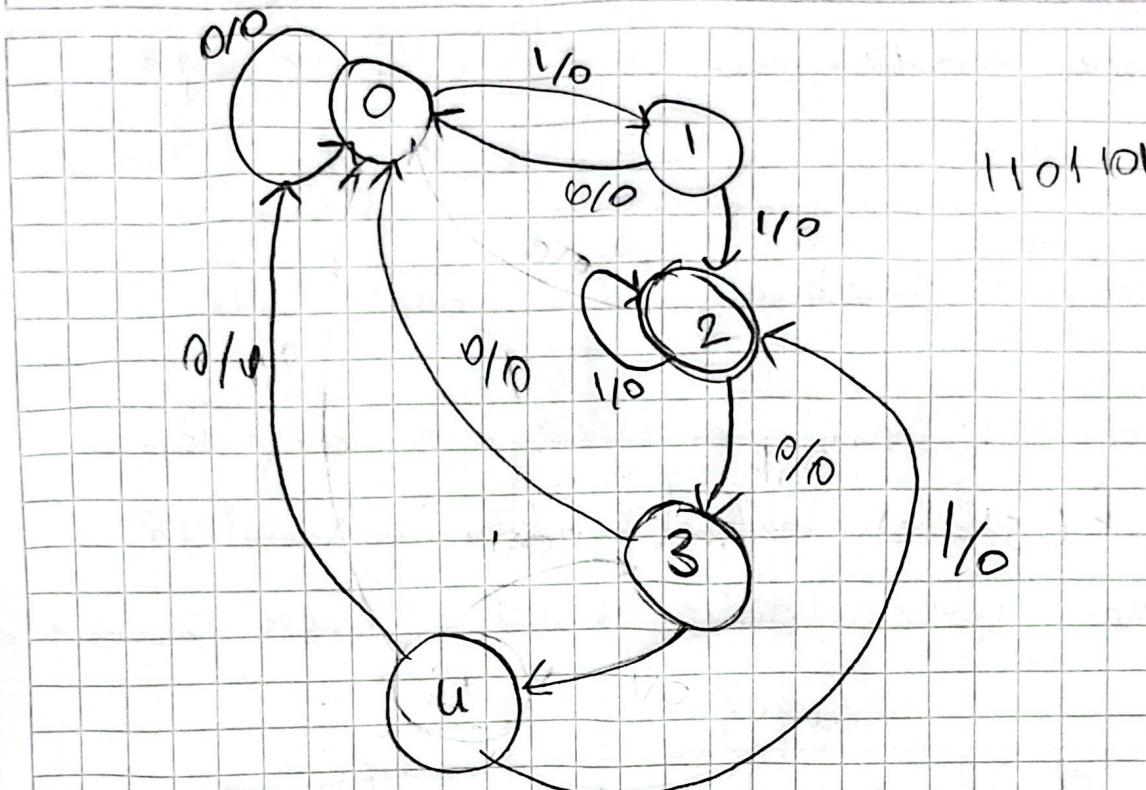
$$S = C + a + b \quad \text{XOR}$$

$$C' = ab + cb + ac$$

$$C' = ab + c(ab)$$

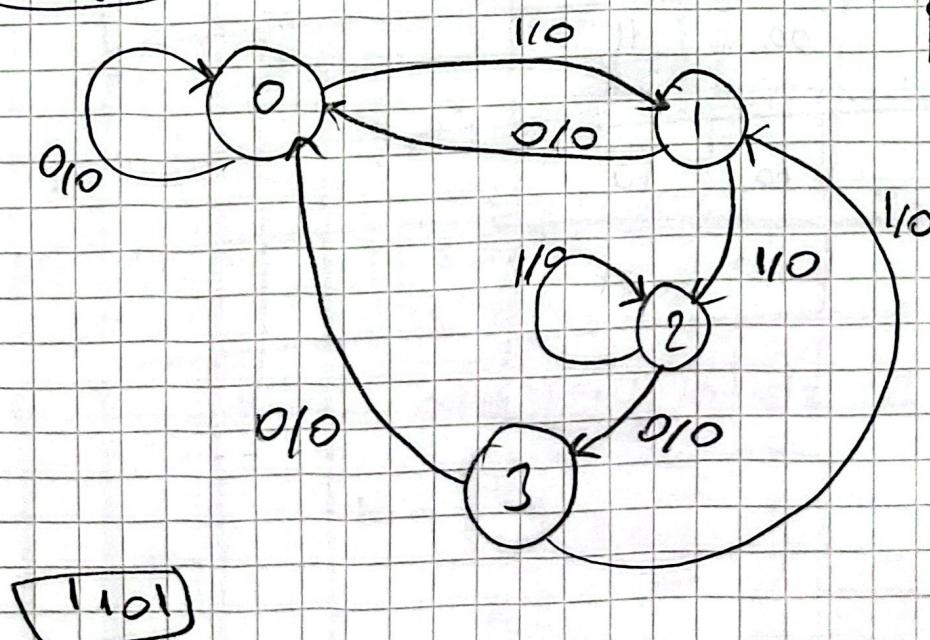


1101101



hyol

2 bit

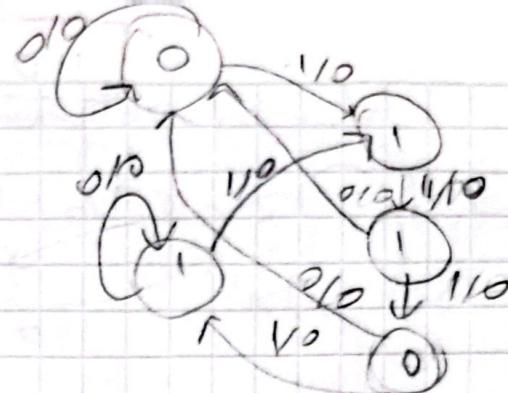


hafta 12 #

meal

moore → output step

clock



moore gösteriminde çıkış değişebilir.

Moore : sonuc yani çıkış aynı olduğu için

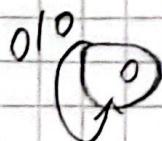
$f(state) = output$ olmamak yerine yazabiliyor

ama baba çıkış farklı olursa 2. durum

durus.

Durum 0	Durum 1	Cıktı
11	X=0 00	X=1 10
01	00	11
10	00	10
00	00	01

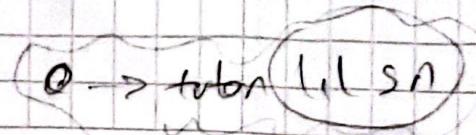
Example 3: state diagram



S0
B1

01 → döngüsünü taban devre

S1
B1



S01 ① + S0
S0f

stroke

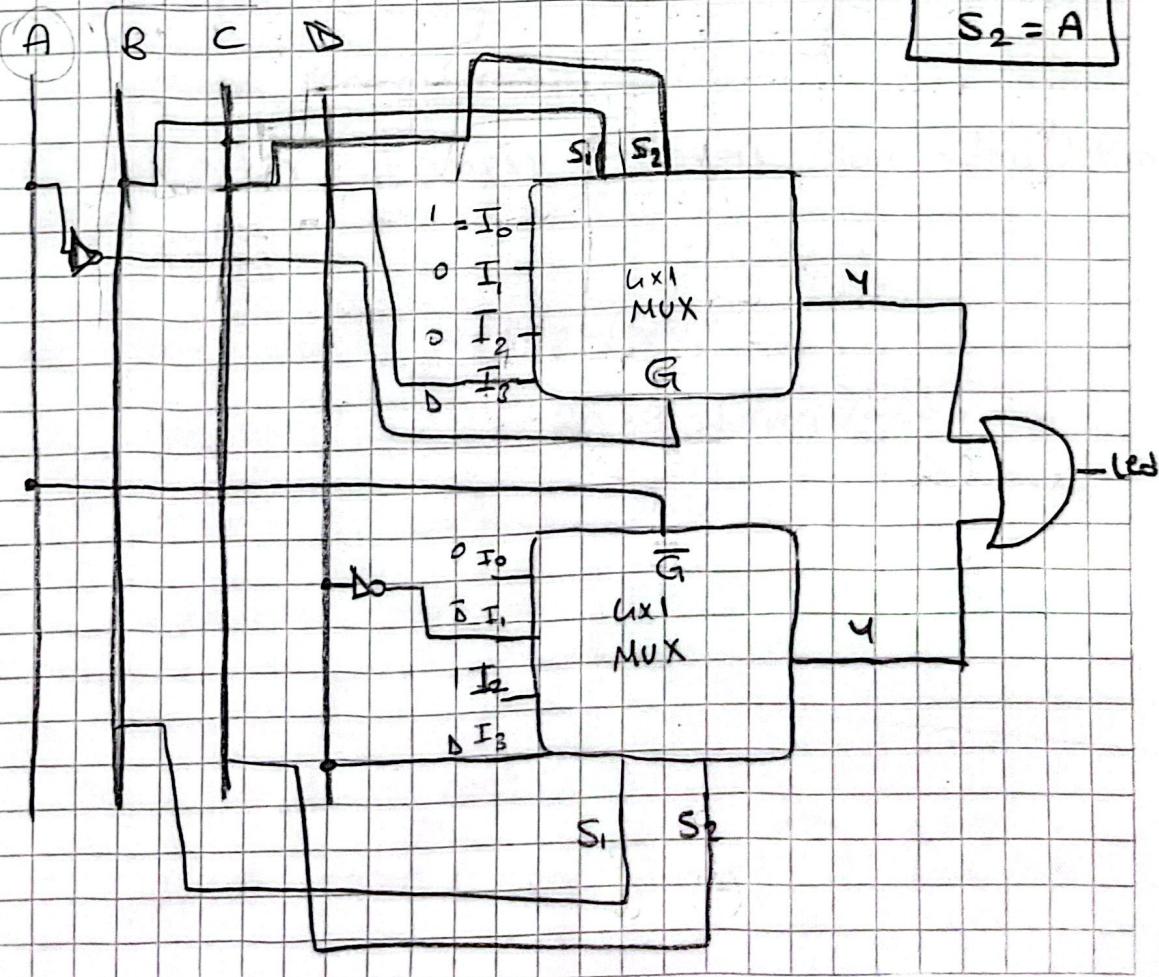
$$\begin{array}{l} I = SV \\ O = GND \end{array}$$

G (B) (A)

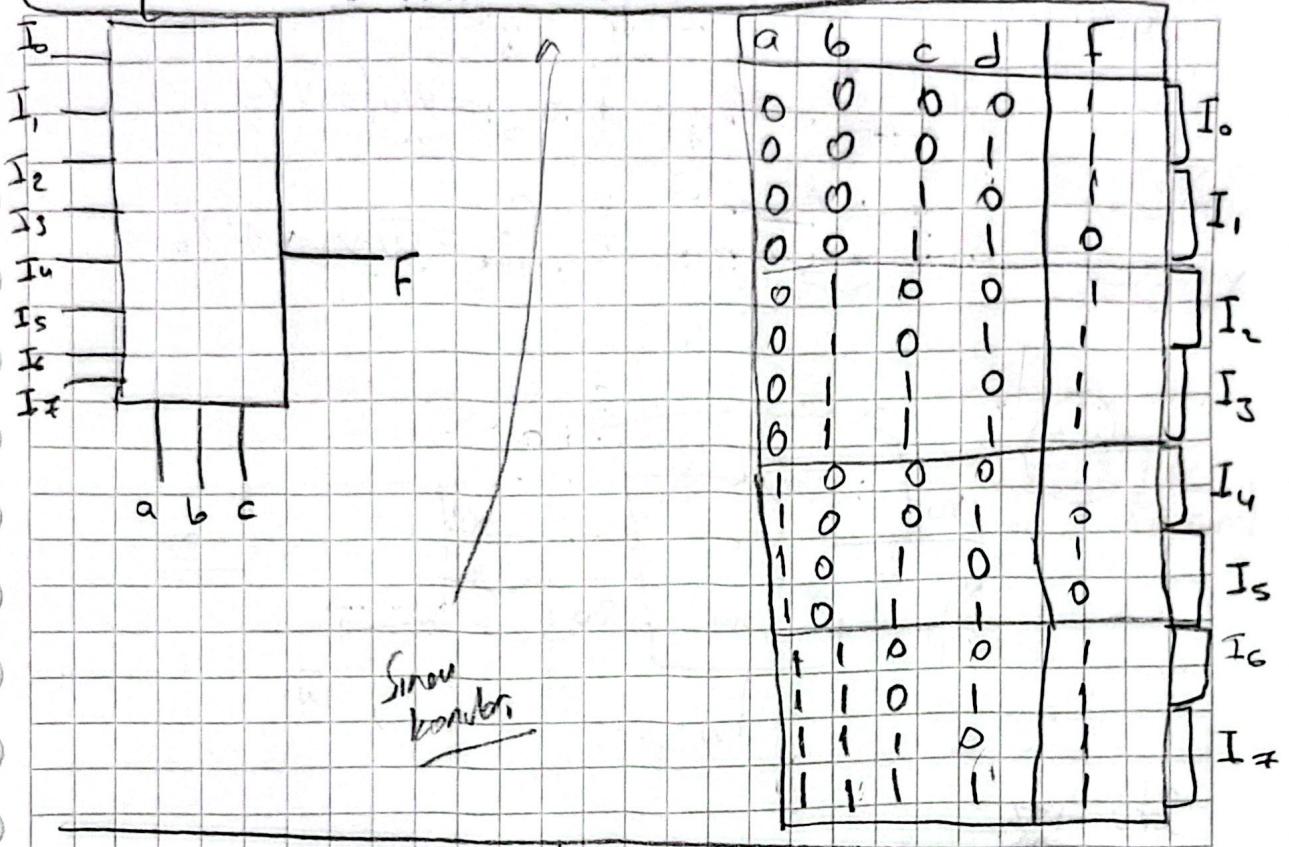
	A	B	C	D	F	
$I = I_0$	0	0	0	0	1	1
$O = I_1$	0	0	1	0	0	0
$O = I_2$	0	1	0	0	0	0
$D = I_3$	0	1	0	1	0	0
$O = I_0$	1	0	0	0	0	0
$I = I_1$	1	0	0	1	0	0
$I = I_2$	1	1	0	0	1	1
$D = I_3$	1	1	1	0	0	0

devret

$$\begin{array}{l} S_1 = B \\ S_2 = A \end{array}$$



1- Combin
 derle
 8-cast
 1 minterm nooterm
 2 sayısal devamlar
 6 mux decoder
 3 konus
 4 logic senler
 5 " analiz"



Sequential Circuits

→ depolaran elementi (muntigil)

+ outputun çıkışmasını sağlayan son adım (istem)
sakları.

[Adım]

→ Storage element → Flip-flop

→ mealy \leftarrow \rightarrow önceki durum / geçmi's

→ moore \rightarrow sadece sıradı

→ geçikme zamanını kullanıp eski
değerle eşitmek atıyoğlanır.

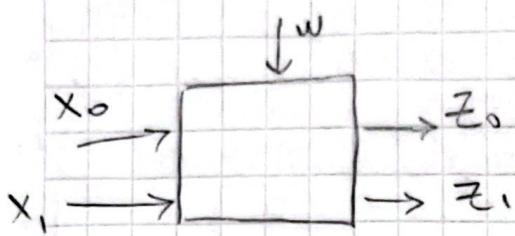
Starting state (kontinut)

→ önemli element (clock) \rightarrow S

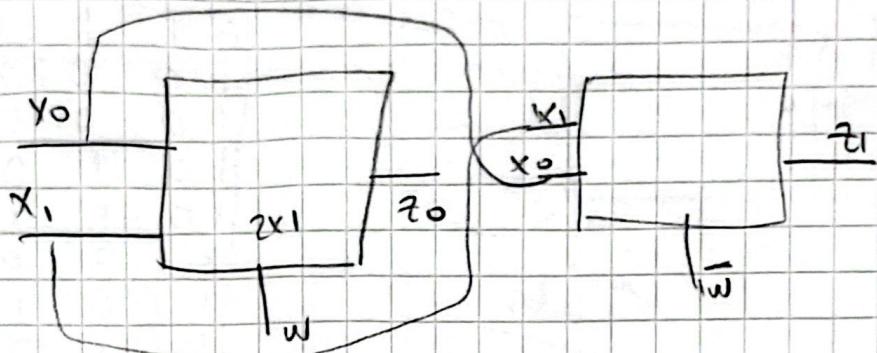
sonra
başka
değişik.

hafta 10

Soru 6)

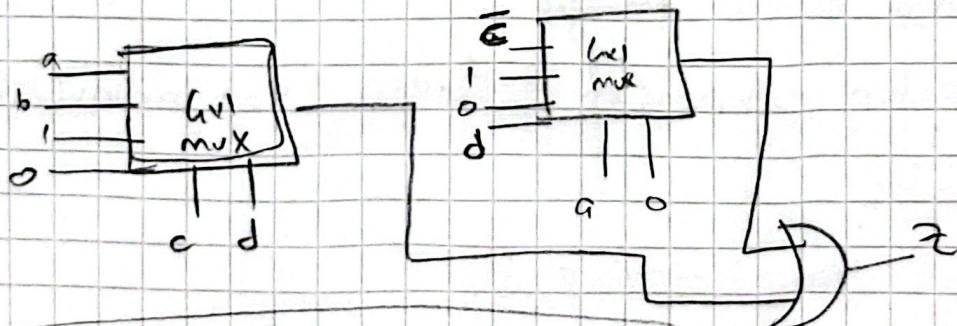


mux



Soru -17

2 toone mux ve bir veyan . kapisı



$$f_{\text{mux}_1} = \overline{c} \overline{d} a + \overline{c} d b + c \overline{d} b + c d a$$

$$+ c d o$$

$$f_{\text{mux}_2} = \overline{a} \overline{b} \overline{c} + \overline{a} \overline{b} l + a \overline{b} o + a b d$$

$\rightarrow a b = y_i$ Seçme girişleri doğrudur 8x1 lik

MUX yapısı

$$2^3 = 8$$

$$3 = \text{segment}$$