

Mantık Devreleri
5. Bölüm

Tümleşik Bileşenlerle Tasarım

Feza Kerestecioğlu

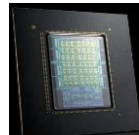
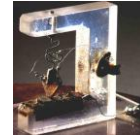
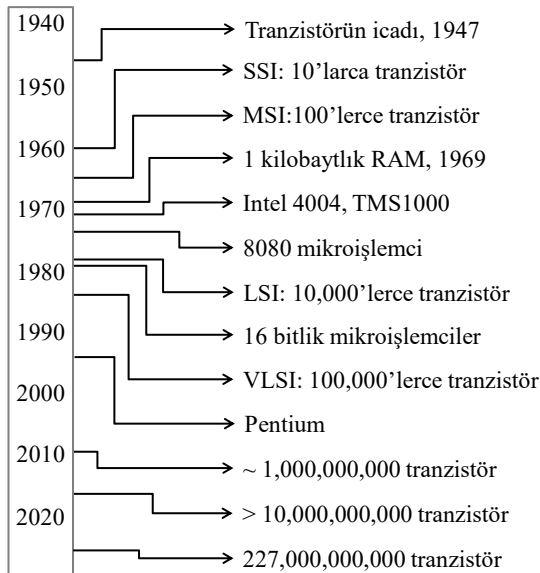


Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

1

Tümleşik Bileşenlerin Kısa Bir Tarihi



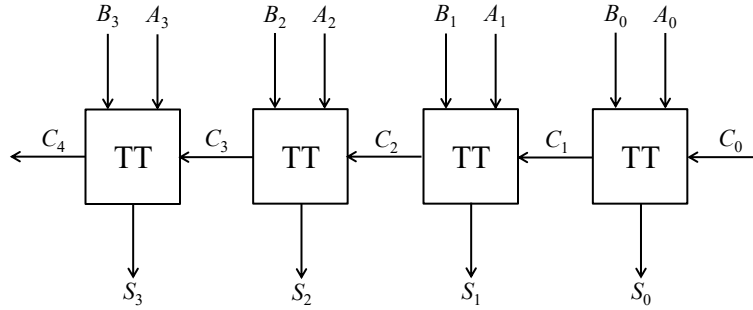
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

2

Paralel Toplayıcılar

İki tane 4 bitlik sayıyı ($A_3 A_2 A_1 A_0$ ve $B_3 B_2 B_1 B_0$), toplamak için, dört tane 1-bitlik tam toplayıcıyı paralel olarak bağlayabiliriz.



4 bitlik, 2 bitlik ve 1 bitlik tam toplayıcıları birleştirerek, istenen basamak sayısını içeren tam toplayıcılar oluşturabiliriz.



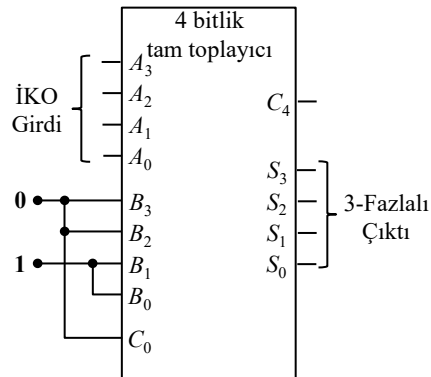
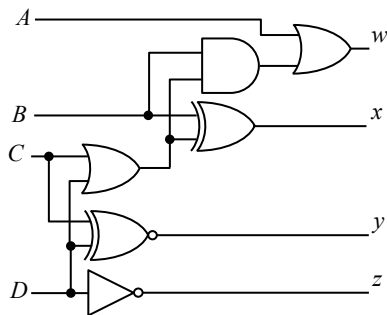
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

3

Paralel Toplayıcılar

Örnek: İKO'dan 3-Fazlı'ya dönüştürme



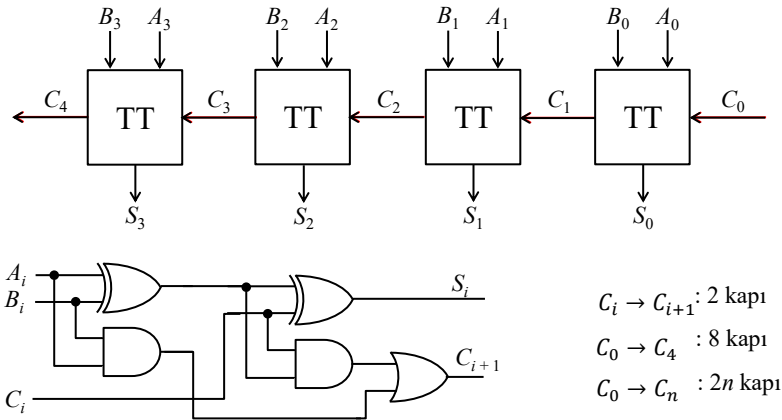
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

4

Öngörülü Elde Üreteci

Sorun: 4 bitlik tam toplayıcının yayılma gecikmesi, eldenin C_0 'dan C_4 'e olan yayılmasıyla belirlenir. Bu gecikme, toplayıcının basamak sayısıyla orantılıdır.



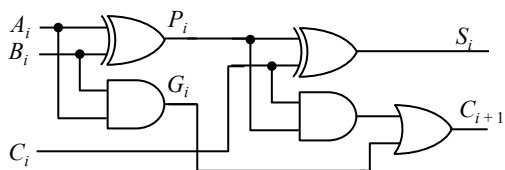
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

5

Öngörülü Elde Üreteci

Çözüm: Öngörülü Elde



$$C_1 = G_0 + P_0 C_0$$

$$\begin{aligned} C_2 &= G_1 + P_1 C_1 \\ &= G_1 + P_1 (G_0 + P_0 C_0) \\ &= G_1 + P_1 G_0 + P_1 P_0 C_0 \end{aligned}$$

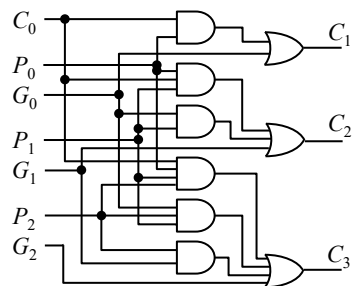
$$\begin{aligned} C_3 &= G_2 + P_2 C_2 \\ &= G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0) \\ &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \end{aligned}$$

$$P_i = A_i \oplus B_i \quad (\text{elde yayılımı})$$

$$G_i = A_i B_i \quad (\text{elde üretimi})$$

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$



74182: 4 bitlik öngörülü elde üretici

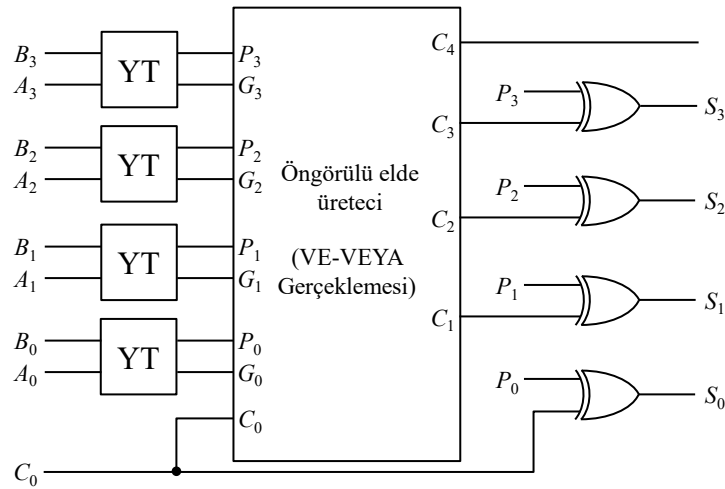


Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

6

Öngörülü Elde Üreteci



Dört kapılık bir yayılma gecikmesi (2 Özel-VEYA, 1 VE, 1 VEYA)

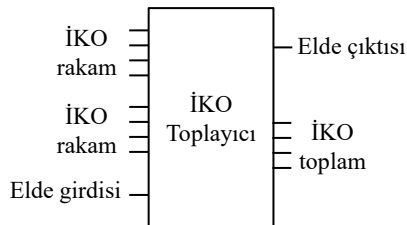


Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

7

İKO Toplayıcı



$$Z \leq 9 \text{ ve } K = 0 \Rightarrow \begin{cases} S = Z \\ C = K \end{cases}$$

$$Z > 9 \text{ veya } K = 1 \Rightarrow \begin{cases} S = Z + 6 \\ C = 1 \end{cases}$$

$$Z = \boxed{Z_3} \boxed{Z_2} \boxed{Z_1} \boxed{Z_0}$$

$$Z > 9 \Leftrightarrow Z_3 \cdot (Z_2 + Z_1) = 1$$

$$K + Z_3 \cdot (Z_2 + Z_1) = 1 \Rightarrow \begin{cases} S = Z + 6 \\ C = 1 \end{cases}$$

Ondalık Sonuç	Tam Toplayıcı Çıktısı		İKO Çıktı	
	K (Elde)	Z (Toplam)	C (Elde)	S (Toplam)
0	0	0000	0	0000
1	0	0001	0	0001
⋮	⋮	⋮	⋮	⋮
9	0	1001	0	1001
10	0	1010	1	0000
11	0	1011	1	0001
⋮	⋮	⋮	⋮	⋮
15	0	1111	1	0101
16	1	0000	1	0110
17	1	0001	1	0111
18	1	0010	1	1000
19	1	0011	1	1001

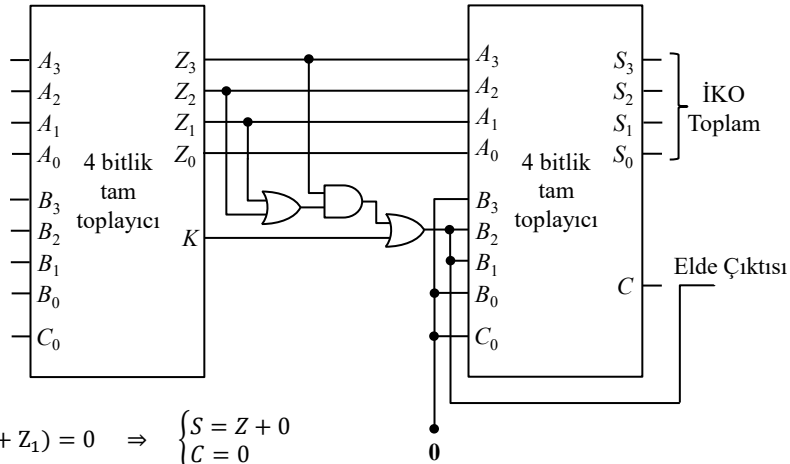


Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

8

İKO Toplayıcı



$$K + Z_3 \cdot (Z_2 + Z_1) = 0 \Rightarrow \begin{cases} S = Z + 0 \\ C = 0 \end{cases}$$

$$K + Z_3 \cdot (Z_2 + Z_1) = 1 \Rightarrow \begin{cases} S = Z + 6 \\ C = 1 \end{cases}$$



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

9

Büyüklik Karşılaştırıcısı

Girdiler: $A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0$ Çıktılar: E, L_{AB}, S_{AB}

$$A = B \Leftrightarrow E = 1$$

$$A > B \Leftrightarrow L_{AB} = 1$$

$$A < B \Leftrightarrow S_{AB} = 1$$

$$E = 1 \Leftrightarrow A_i = B_i \quad i = 0, \dots, 3$$

$$E = x_3 \cdot x_2 \cdot x_1 \cdot x_0 \quad x_i = A_i B_i + A'_i B'_i = A_i \odot B_i$$

$$L_{AB} = A_3 B'_3 + x_3 A_2 B'_2 + x_3 x_2 A_1 B'_1 + x_3 x_2 x_1 A_0 B'_0$$

$$S_{AB} = A'_3 B_3 + x_3 A'_2 B_2 + x_3 x_2 A'_1 B_1 + x_3 x_2 x_1 A'_0 B_0$$

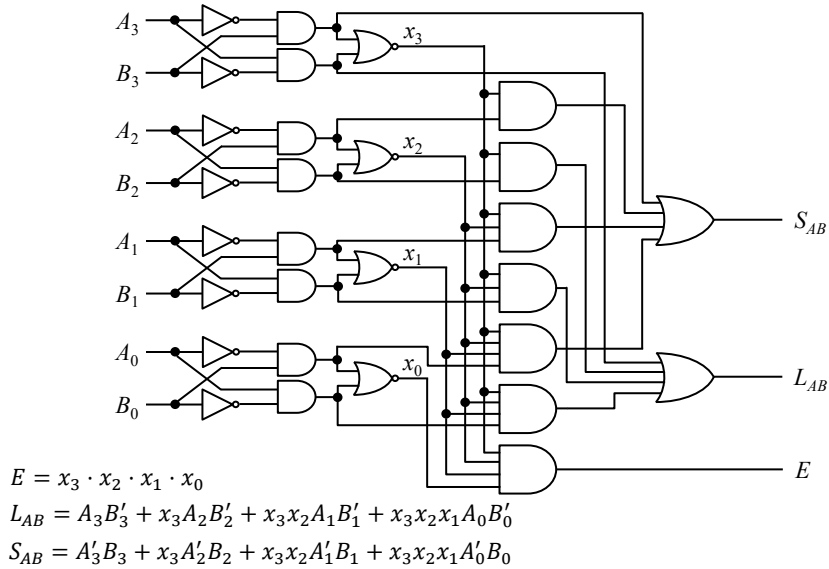


Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

10

Büyüklik Karşılaştırıcısı



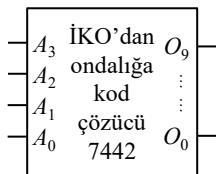
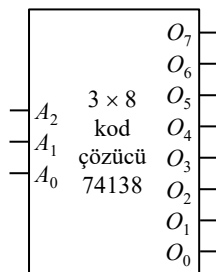
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

11

Kod Çözücüler

Kod çözücü (asal terim üretici), n tane girdi hattından gelen ikili bilgiyi en fazla 2^n sayıda olabilen çıktı hatlarına dönüştüren bir mantık devresidir.



$$\left. \begin{array}{l} O_i = 1 \\ O_j = 0 \quad j \neq i \end{array} \right\} A_2A_1A_0 \text{ ikili } i \text{ sayısını temsil ediyorsa}$$

Girdiler			Çıktılar							
A_2	A_1	A_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



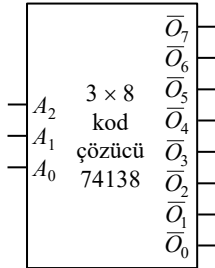
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

12

Kod Çözücüler

Düşük aktif kod çözücüler:



$$\left. \begin{array}{l} \bar{O}_i = 0 \\ \bar{O}_j = 1 \quad j \neq i \end{array} \right\} A_2 A_1 A_0 \text{ ikili } i \text{ sayısını temsil ediyorsa}$$

Girdiler			Çıktılar							
A_2	A_1	A_0	$\bar{O}_0$	$\bar{O}_1$	$\bar{O}_2$	$\bar{O}_3$	$\bar{O}_4$	$\bar{O}_5$	$\bar{O}_6$	$\bar{O}_7$
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

13

Kod Çözücüler

Asal terimlerin toplamı şeklindeki Boole fonksiyonları kod çözücüler ve VEYA kapıları ile gerçekleştirilebilir.

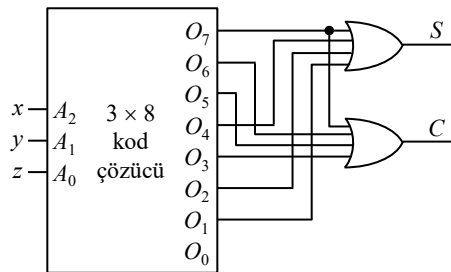
Örnek: Bir basamaklık tam toplayıcının kod çözücü ve VEYA kapıları ile gerçekleştirilmesi

	yz	00	01	11	10
x	0		1		1
	1	1		1	

$$S(x, y, z) = x \oplus y \oplus z = \sum(1, 2, 4, 7)$$

	yz	00	01	11	10
x	0			1	
	1		1	1	1

$$C = xy + xz + yz = \sum(3, 5, 6, 7)$$



Fonksiyondaki asal terim sayısı $2^n/2$ 'den fazla ise, F' asal terimlerini birleştiren bir VEYA-DEĞİL kapısı kullanılabilir.



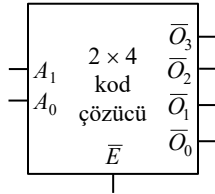
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

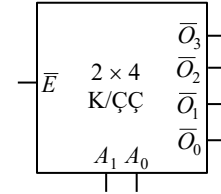
14

Çoğullama Çözücü Olarak Kod Çözümler

Etkinleştirici girdiler:

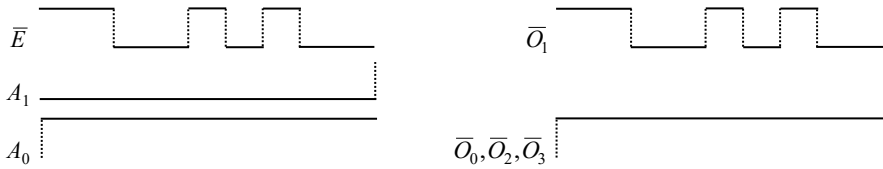


Girdiler			Çıktılar			
$\bar{E}$	A_1	A_0	$\bar{O}_0$	$\bar{O}_1$	$\bar{O}_2$	$\bar{O}_3$
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



Etkinleştirici girdileri olan kod çözucüler **çoğullama çözücü** olarak kullanılabilirler.

Bir çoğullama çözücü, tek girdi hattından gelen bilgiyi, seçme girdilerindeki verilere göre 2^n çıktıdan birine iletir.



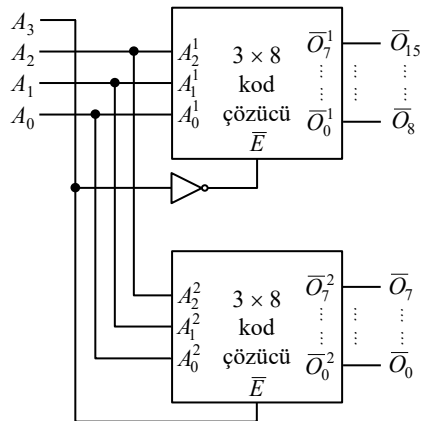
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

15

Kod Çözümler

Kod çözucüler daha büyük kod çözucüler elde etmek için birleştirilebilirler.



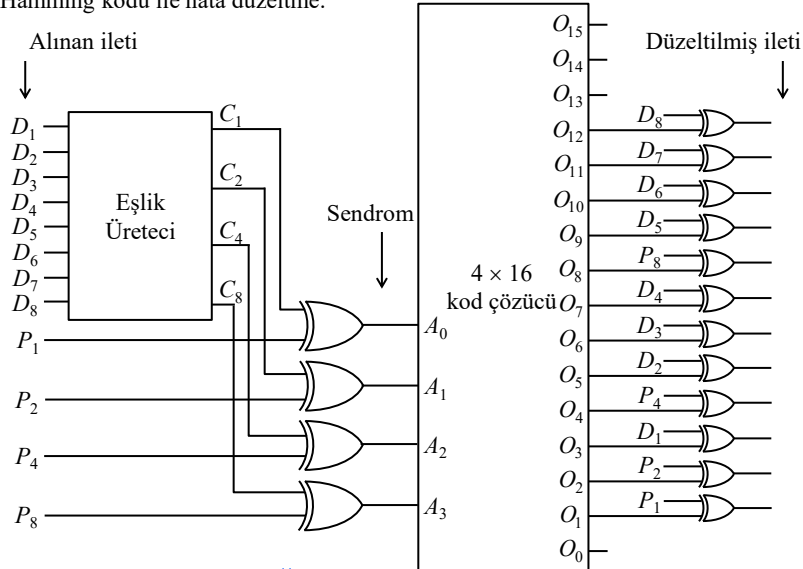
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

16

Kod Çözücüler

8 bitlik veri için Hamming kodu ile hata düzeltme:



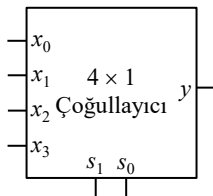
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

17

Çoğullayıcılar

Dijital **çoğullayıcı**, girdi hatlarının birinden gelen bilgiyi seçerek tek çıktı hattına yönlendiren bir mantık devresidir.



s_1	s_0	y
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

$$y = x_3 s_1 s_0 + x_2 s_1 s'_0 + x_1 s'_1 s_0 + x_0 s'_1 s'_0$$



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

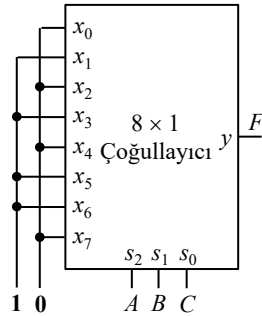
18

Çoğullayıcılar

Boole fonksiyonlarını gerçekleştirme:

Örnek:

A	B	C	$F(A,B,C)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

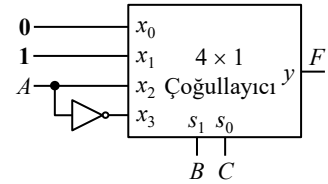


$$BC = 00 \Rightarrow F = 0$$

$$BC = 01 \Rightarrow F = 1$$

$$BC = 10 \Rightarrow F = A$$

$$BC = 11 \Rightarrow F = A'$$



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

19

Çoğullayıcılar

Boole fonksiyonlarını gerçekleştirme:

Örnek:

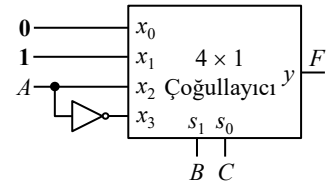
Fonksiyonun asal terimlerini bu tür bir tabloda gösterebiliriz:

A	B	C	$F(A,B,C)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

	I_0	I_1	I_2	I_3
A'	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	A'

I_i 'ler A dışındaki değişkenlerin literallerini içeren terimler ($I_0 = B'C'$, ..., $I_3 = BC$) olacak şekilde F 'nin asal terimleri $A'I_i$ veya AI_i olarak yazılabilir.

$$x_i = \begin{cases} 0 & I_i \text{ sütununda asal terim yoksa} \\ 1 & I_i \text{ sütunundaki iki asal terim de varsa} \\ A' & I_i \text{ sütununda sadece } A' \text{ satırındaki asal terim varsa} \\ A & I_i \text{ sütununda sadece } A \text{ satırındaki asal terim varsa} \end{cases}$$



Nöroteknolojik Çözümler Platformu

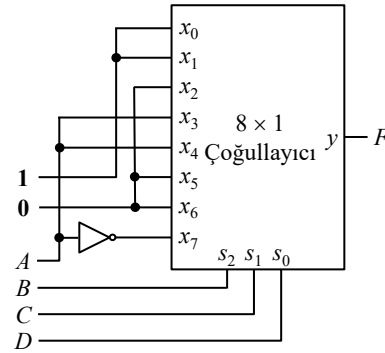
Boğaziçi Üniversitesi Ar-Ge Okulu

20

Çoğullayıcılar

Örnek: $F(A,B,C,D) = \sum(0,1,7,8,9,11,12)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	A	A	0	0	A'



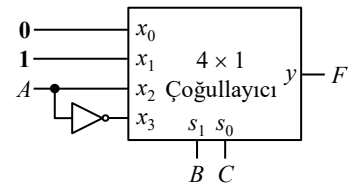
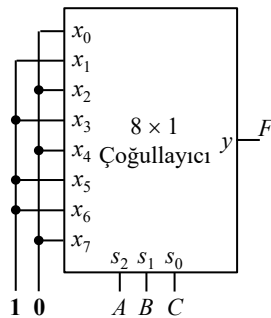
Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

21

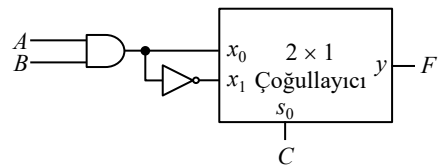
Çoğullayıcılar

A	B	C	$F(A,B,C)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



$$C = 0 \Rightarrow F = A \cdot B$$

$$C = 1 \Rightarrow F = (A \cdot B)'$$



Nöroteknolojik Çözümler Platformu

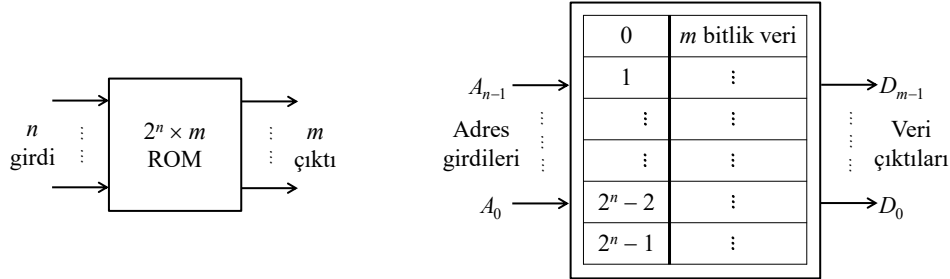
Boğaziçi Üniversitesi Ar-Ge Okulu

22

Salt Okunur Bellek

Salt okunur bellek (ROM), sabit bir ikili bilgi kümesinin depolandığı bir bellek aygıtıdır.

İşlevsel blok diyagramı:



m bitlik çıktı, adres girdisiyle belirtilen konumda depolanan veridir.



Nöroteknolojik Çözümler Platformu

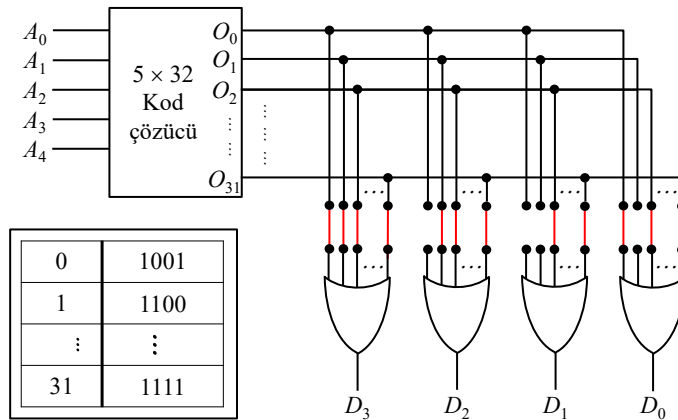
Boğaziçi Üniversitesi Ar-Ge Okulu

23

Salt Okunur Bellek

Mantık yapısı:

Örnek: 32×4 ROM



Bağlantılar, ROM'da saklanması gereken verilere göre koparılır. Belirli bir adres kombinasyonu için $D_i = 0$ ise, kod çözücünün ilgili asal terim çıkışından D_i 'yi üreten VEYA kapısına giden bağlantı koparılır.



Nöroteknolojik Çözümler Platformu

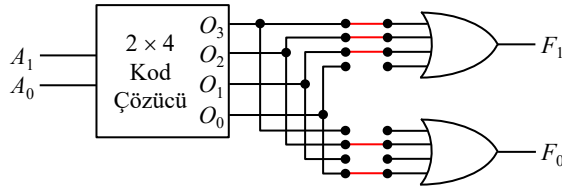
Boğaziçi Üniversitesi Ar-Ge Okulu

24

Salt Okunur Bellek

Mantık fonksiyonlarının gerçekleştirilmesi:

Örnek: $F_0(A_0, A_1) = \Sigma(0,2)$ $F_1(A_0, A_1) = \Sigma(1,2,3)$



ROM Tablosu

Adres			Veri	
	A_1	A_0	F_1	F_0
0	0	0	0	1
1	0	1	1	0
2	1	0	1	1
3	1	1	1	0



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

25

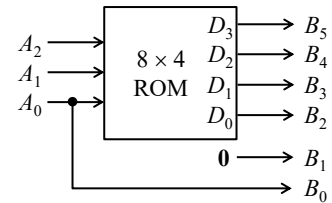
Salt Okunur Bellek

Örnek: Kare üretici

Girdiler				Çıktılar					
A_2	A_1	A_0		B_5	B_4	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	1	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	1	0	0
7	1	1	1	1	1	0	0	0	1

ROM Tablosu

Adres	Veri
0	0
1	0
2	1
3	2
4	4
5	6
6	9
7	C



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

26

Öğrenme Çıktıları

Neler öğrendik:

- Bir Boole fonksiyonunun (ya da aynı değişkenlerin birden fazla fonksiyonunun)
 - mantık kapıları
 - kod çözücüler
 - çoğullayıcılar
 - salt okunur bellek
 - duruma has yaklaşımlar
 kullanılarak gerçekleştirilmesi
- Çok seviyeli kombinezonsal devrelerdeki yayılım gecikmesini hesaplama
- Daha büyüklerini elde etmek için kod çözücülerini birleştirme



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

27

Devam edecek...

kerestec@khas.edu.tr



Nöroteknolojik Çözümler Platformu

Boğaziçi Üniversitesi Ar-Ge Okulu

28