

Tümleştirilmiş Kombinezonsal Devre Elemanları

Sayısal sistemlerin gerçekleştirilmesinde çokça kullanılan lojik devreler, lojik bağlaçların bir araya getirilmesiyle tümleştirilmiş devre (entegre devre, tümdevre) (*integrated circuit -IC*) olarak üretilirler ve satılırlar.

Bağlaçlar yerine bu hazır devrelerin kullanılması tasarımları kolaylaştırır.

Tümdevreler içerdikleri kapı sayısına göre çeşitli gruplara ayrılırlar.

Tümleştirme düzeylerine göre gruplama:

- Küçük Ölçekli Tümleştirme (*Small-Scale Integration SSI*):

Bu gruptaki tümdevreler 10 taneden az lojik kapı içerirler. Örneğin 7400 4 adet TVE kapısı içerir.

- Orta Ölçekli Tümleştirme (*Medium-Scale Integration MSI*):

Bu gruptaki tümdevreler 10 ile 1000 tane arasında lojik kapı içerirler. Toplayıcı, veri seçici, kod çözücü elemanlar bu gruba girer.

- Büyük Ölçekli Tümleştirme (*Large-Scale Integration LSI*):

Bu gruptaki tümdevreler binler mertebesinde lojik kapı içerirler. Mikroişlemciler, bellekler bu grupta yer alırlar.

- Çok Büyük Ölçekli Tümleştirme (*Veri Large-Scale Integration VLSI*):

Bu gruptaki tümdevreler yüzbinlerce ve daha fazla sayıda lojik kapı içerirler.

Örnek: Gelişmiş mikroişlemciler ve büyük bellek tümdevreleri.

Yarım Toplayıcı (Half Adder):

İki adet birer bitlik sayıyı toplayan bir devredir.



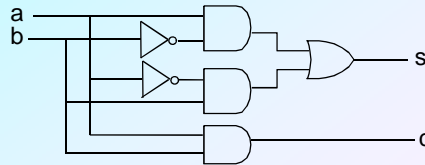
a: Birinci Sayı
b: İkinci Sayı
s: Sonuç
c: Elde Çıkışı

a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Doğruluk tablosundan devrenin ifadesi elde edilir.

$$s = ab' + a'b$$

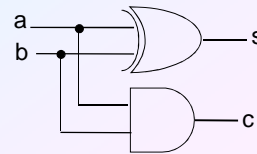
$$c = ab$$



Bu devre yanda gösterildiği gibi YA DA (DARVEYA) (EXOR) bağlacı kullanılarak da gerçekleştirilebilir.

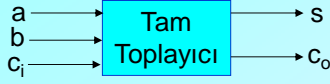
$$s = a \oplus b$$

$$c = ab$$



Tam Toplayıcı (Full Adder):

İki adet birer bitlik sayıyı eldeli olarak toplayan devredir.



a: Birinci Sayı
 b: İkinci Sayı
 c_i: Elde Girişi (Carry in)
 s: Sonuç
 c_o: Elde Çıkışı (Carry out)

a	b	c _i	c _o	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	1	0
1	1	1	1	1

s	bc _i	00	01	11	10
a	0	0	1	0	1
a	1	1	0	1	0

$$s = a'b'c_i + a'bc_i + ab'c_i + abc_i$$

$$s = a \oplus (b \oplus c_i)$$

$$s = a \oplus b \oplus c_i$$

c _o	bc _i	00	01	11	10
a	0	0	0	1	0
a	1	0	1	1	1

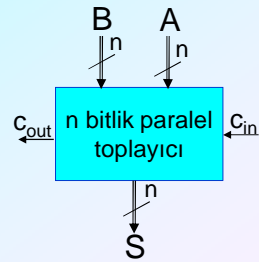
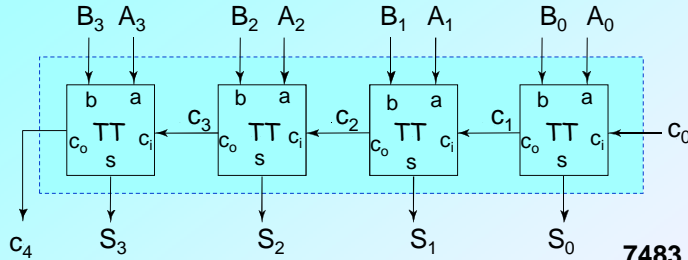
$$c_o = ac_i + bc_i + ab$$

n-Bitlik İkili Paralel Toplayıcı:

İki adet n bitlik 2'li sayıyı toplayan devredir.

Toplanmak istenen sayıların basamak sayısına bağlı olarak bir bitlik tam toplayıcılar peş peşe bağlanarak ikili paralel toplayıcılar gerçekleştirilebilir.

Aşağıda 4 bitlik bir ikili toplayıcının iç yapısı gösterilmiştir.

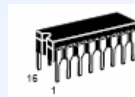
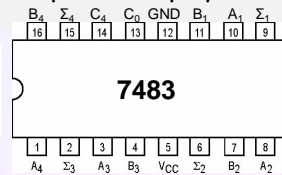
1. Sayı: A₃A₂A₁A₀2. Sayı: B₃B₂B₁B₀Sonuç: S₃S₂S₁S₀Elde Girişi: c₀Elde Çıkışı: c₄**Örnek:**

1. Sayı: 0110

2. Sayı: 1100

Sonuç: 0010

Elde : 1

**7483** tümdevresi 4 bitlik bir ikili paralel toplayıcıdır.

Çıkarma Devresi

Çıkarma işlemi "2'ye tümleyeni ile toplama" şeklinde gerçekleştirilir.

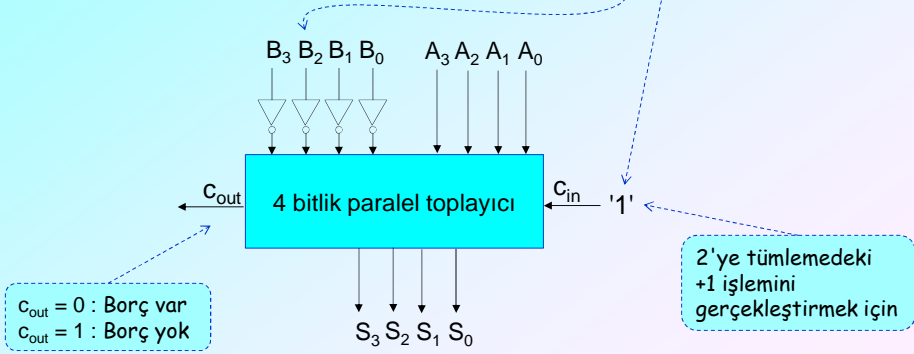
Çıkarma devresi n bitlik bir tam toplayıcı ve tümleme kapıları ile gerçekleştirilebilir.

Örnek: 4 bitlik çıkarma devresi

$$S = A - B$$

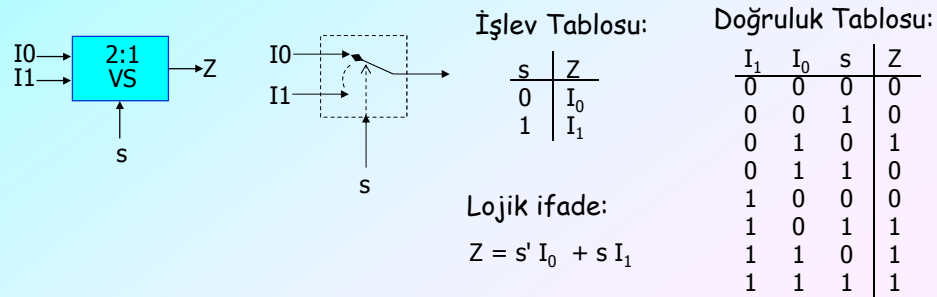
B sayısının 2'ye tümleyeni A ile toplanır.

$$S = A - B = A + 2\text{'ye tümleme}(B) = A + (B' + 1)$$

**Veri Seçiciler (Multiplexer):**

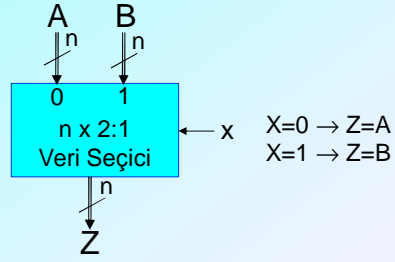
- 2^n adet veri girişi (I), n adet seçme (denetim) girişi (S), 1 adet çıkışı (Z) vardır.
- Seçme girişlerine gelen değere göre, veri girişlerinden birindeki değer çıkışa aktarılır.
- Seçme girişlerindeki n bitlik ikili sayı hangi veri girişinin seçileceğini belirler.
- Veri seçiciler giriş sayılarına göre m:1 olarak adlandırılır. Burada m veri girişlerinin sayısını gösterir.

Örnek: 2:1 Veri seçici ("İkiye bir veri seçici" olarak okunur.)

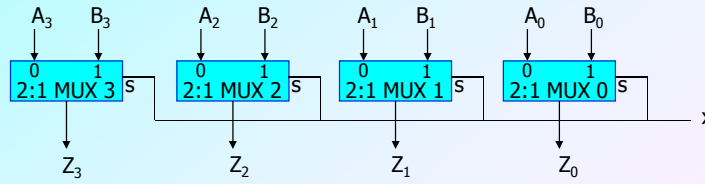


Veri seçicilerin paralel bağlanması:

İki adet n bitlik veri arasında seçme yapmak için n adet 2:1 veri seçici paralel olarak bağlanır.
Yanda blok diyagramı verilen devre, x seçme girişinin değerine bağlı olarak n bitlik A ya da B sayılarından birini Z çıkışına aktarmaktadır.



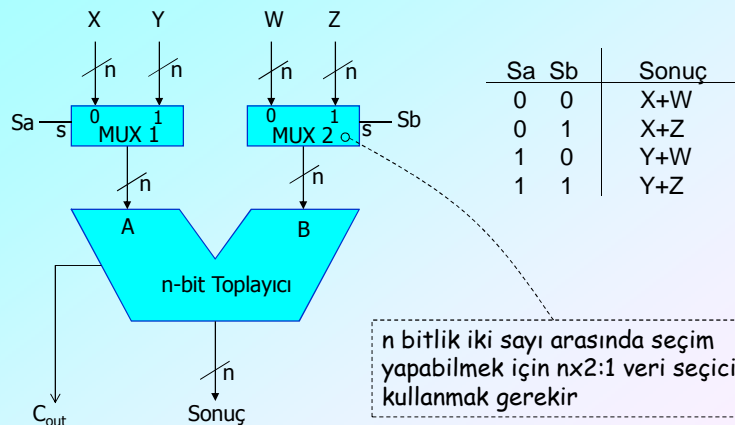
Örnek: 4 bitlik A ya da B sayılarından birini Z çıkışına aktaran devre



Bu devrede tüm veri seçicilerin seçme uçları ortaktır (kısır devre).

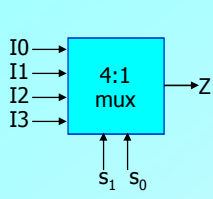
Veri seçicilerin kullanımına bir örnek:

Bir toplayıcının girişine isteğe bağlı olarak farklı sayılar uygulanabilir.

**Örnek Tümdevre:**

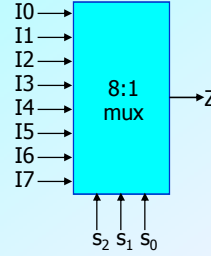
74151 içinde bir adet 8:1 veri seçici bulunduran bir tümdevredir.

Diğer Veri Seçici (MUX) Örnekleri:



İşlev Tablosu:

$s_1 s_0$	Z
0 0	I_0
0 1	I_1
1 0	I_2
1 1	I_3



İşlev Tablosu:

$s_2 s_1 s_0$	Z
0 0 0	I_0
0 0 1	I_1
0 1 0	I_2
0 1 1	I_3
1 0 0	I_4
1 0 1	I_5
1 1 0	I_6
1 1 1	I_7

Lojik İfadeler:

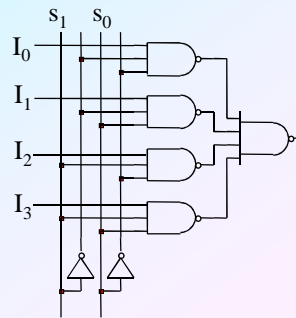
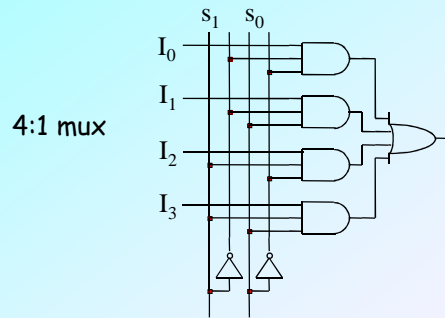
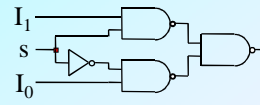
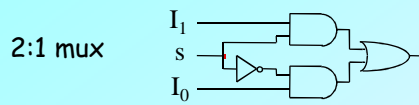
2:1 mux: $Z = s' I_0 + s I_1$

4:1 mux: $Z = s_1' s_0' I_0 + s_1' s_0 I_1 + s_1 s_0' I_2 + s_1 s_0 I_3$

8:1 mux: $Z = s_2' s_1' s_0' I_0 + s_2' s_1' s_0 I_1 + s_2' s_1 s_0' I_2 + s_2' s_1 s_0 I_3 + s_2 s_1' s_0' I_4 + s_2 s_1' s_0 I_5 + s_2 s_1 s_0' I_6 + s_2 s_1 s_0 I_7$

Genel İfade (k:1 Mux): $Z = \sum_{j=0}^{k-1} (m_j I_j) \quad k=2^n, m_j = j. \text{ minterim}$

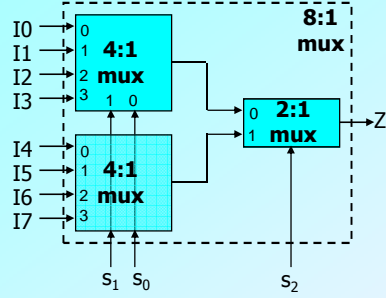
Veri Seçiciler lojik bağlaçlar kullanılarak aşağıdaki gibi gerçekleştirilebilirler.



Büyük boyutlardaki veri seçiciler, daha küçüklerin uygun şekilde bağlanmasıyla gerçekleştirilebilir.

Aşağıda 8:1 veri seçicinin 2 farklı şekilde gerçekleştirilmesi gösterilmiştir.

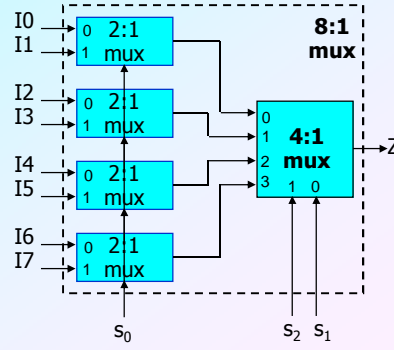
1. Yöntem:



Burada s_0 ve s_1 seçme girişleri 4:1 veri seçicileri için ortaktır. İki veri seçicinin de aynı girişi seçilir.

Hangi veri seçicinin çıkışının seçileceğini ise s_2 belirler.

2. Yöntem:



Veri Seçiciler ile Genel Amaçlı Lojik Devre Tasarımı 1:

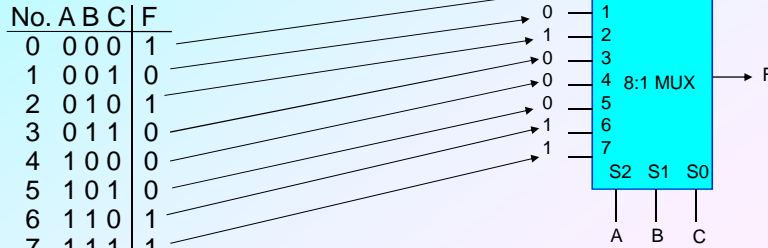
$2^n:1$ boyutlu bir adet veri seçici kullanılarak n girişli, bir çıkışlı herhangi bir lojik devre başka bir bağlaç kullanmadan gerçekleştirilebilir.

Yöntem:

- Tasarlanacak olan fonksiyonun değişkenleri (devrenin girişleri) veri seçicinin seçme uçlarına bağlanır.
- Her seçme değeri bir giriş kombinasyonuna karşı düştüğüne göre, tasarlanmak istenen fonksiyonun doğruluk tablosuna göre veri seçicinin veri girişlerine lojik "0" veya "1" sabitleri bağlanır.

Örnek:

$$F(A,B,C) = m_0 + m_2 + m_6 + m_7 = \cup_1(0,2,6,7)$$



Veri Seçiciler ile Genel Amaçlı Lojik Devre Tasarımı 2:

$2^{n-1}:1$ boyutlu bir adet veri seçici kullanılarak n girişli, bir çıkışlı herhangi bir lojik devre ek olarak **sadece bir adet tümlleme bağlacı kullanılarak** gerçekleştirilebilir.

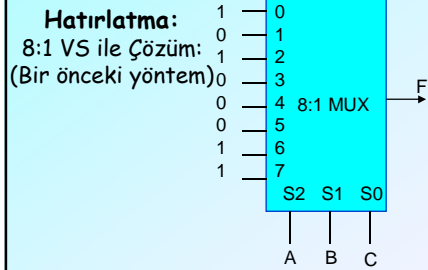
Yöntem:

- Tasarlanacak olan fonksiyonun değişkenlerinden $n-1$ tanesi veri seçicinin seçme uçlarına bağlanır.
- Arta kalan değişkenin kendisi ya da tümlenyeni, doğruluk tablosuna göre veri seçicinin veri girişlerine bağlanır.

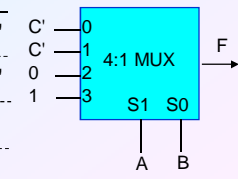
Örnek:

$$F(A,B,C) = m_0 + m_2 + m_6 + m_7 = \cup_1(0,2,6,7)$$

4:1 VS ile Çözüm:



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

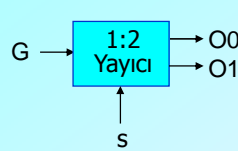


Burada her iki c' değeri de aynı tümlleme kapısından elde edilebilir.

Yayıcı Makas (Demultiplexer):

- 1 adet veri girişi, n adet seçme (denetim) girişi, 2^n adet çıkışı vardır.
- Seçme girişlerine gelen değere göre, veri girişindeki değer çıkışlardan birine aktarılır. Diğer çıkışlar "0" değerini alır. Seçme girişlerindeki n bitlik ikili sayı girişteki değerin hangi çıkışa aktarılacağını belirler.
- Yayıcılar çıkış sayılarına göre $1:m$ olarak adlandırılır. Burada m çıkış sayısını gösterir.

Örnek: 1:2 Yayıcı Makas ("Bire iki yayıcı" olarak okunur)

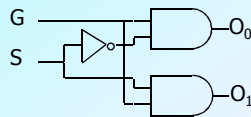


İşlev Tablosu:

s	O ₁	O ₀
0	0	G
1	G	0

Doğruluk Tablosu:

s	G	O ₁	O ₀
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

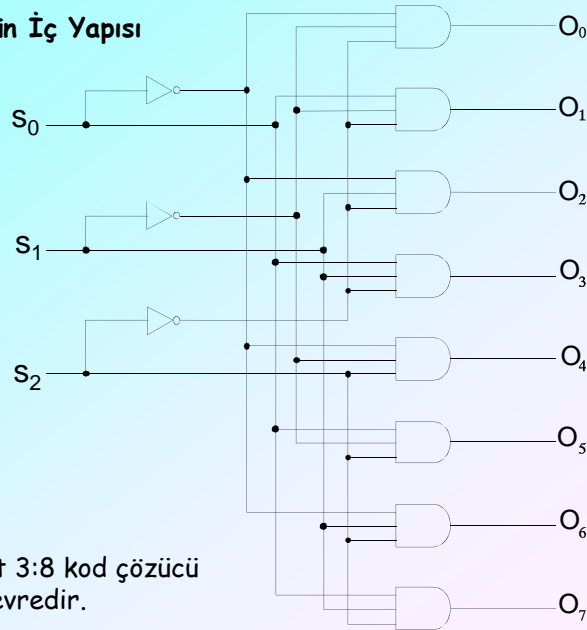


Kod Çözücüler (Decoder):

- n adet seçme (denetim) girişi, 2^n adet çıkışı vardır.
- Seçme girişlerine gelen değere göre, çıkışlardan bir tanesi "1" değerini, diğerleri "0" değerini alır. Seçme girişlerindeki n bitlik ikili sayı hangi çıkışın "1" değerini alacağını belirler.
- Kod çözücü, girişine sabit "1" değeri verilmiş bir yayıcı makas gibi düşünülebilir.
- Kod çözücüler seçme girişi ve çıkış sayılarına göre $n:2^n$ olarak adlandırılır. Burada n seçme girişi sayısı, 2^n çıkış sayısıdır.

Örnek: 3:8 Kod Çözücü

	S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	1	0	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

3:8 Kod Çözücünün İç Yapısı**Örnek Tümdevre:**

74138 içinde bir adet 3:8 kod çözücü bulunduran bir tümdevredir.

Kod Çözücüler ile Genel Amaçlı Lojik Devre Tasarımı:

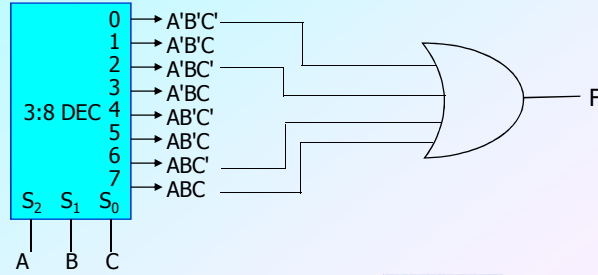
$n:2^n$ boyutlu bir kod çözücü kullanılarak n girişli m çıkışlı herhangi bir genel fonksiyon ek olarak **VEYA bağlaçları** kullanılarak gerçekleştirilebilir.

Yöntem:

- Tasarlanacak olan fonksiyonun değişkenleri (devrenin girişleri) kod çözücünün seçme uçlarına bağlanır.
- Kod çözücünün her çıkışı bir minterime karşı düşer. Gerçeklenecek olan fonksiyonu oluşturan minterimlere ilişkin çıkışlar **VEYA** kapıları ile toplanır.

Örnek:

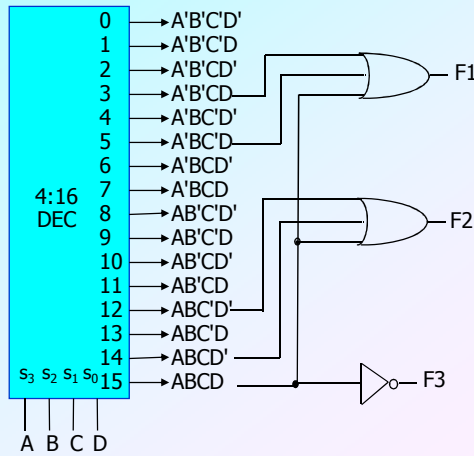
$$F(A,B,C) = m_0 + m_2 + m_6 + m_7 = \cup_1(0,2,6,7)$$

**Örnek: 4 girişli 3 çıkışlı genel fonksiyon tasarımı**

$$F1(A,B,C,D) = A' B C' D + A' B' C D + A B C D$$

$$F2(A,B,C,D) = A B C' D' + A B C$$

$$F3(A,B,C,D) = (A' + B' + C' + D')$$



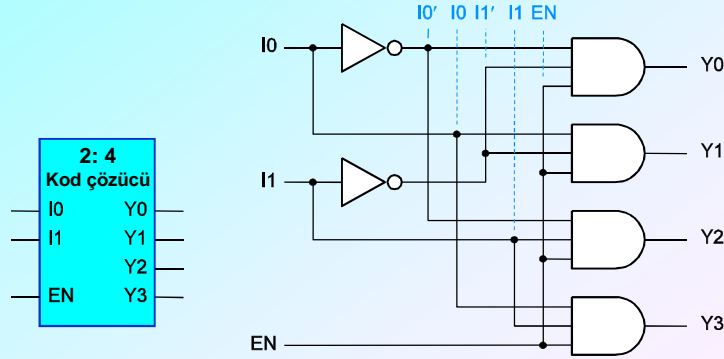
İzin Girişli (Enable -EN) Kod Çözücü:

Kod çözücülerde seçme girişlerine ek olarak izin girişi de (Enable -EN) olabilir.

EN girişi lojik "1" olduğunda kod çözücü normal işlevini görür.

EN girişi lojik "0" olduğunda kod çözücünün tüm çıkışları "0" olur.

Aşağıda izin girişli bir 2:4 kod çözücü gösterilmiştir:

**Kod çözücülerin kullanımına ilişkin bir örnek**

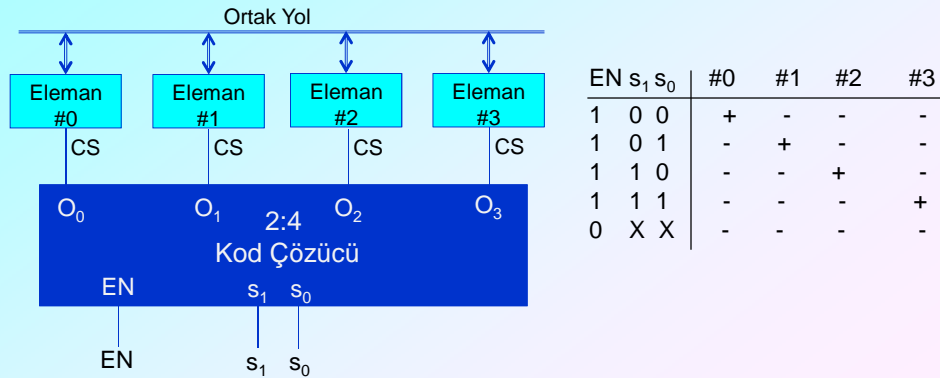
Bazı sistemlerde bir grup elemandan sadece bir tanesinin belli bir anda etki olması istenir.

Diğer bir ifadeyle aynı anda iki eleman etkin (aktif) olamaz.

Bu tür etkin/etkisiz yapılabilen elemanların seçme (chip select -CS) girişleri bulunur.

Kod çözücüler istenen elemanı seçerek etkin yapmak için kullanılırlar.

Örnek: 4 adet elemanı (birimi) kontrol eden kod çözücü



Programlanabilir Lojik Elemanlar (*Programmable Logic Device- PLD*)

Günümüzde karmaşık sayısal devreler programlanabilir lojik elemanlar kullanılarak gerçekleştirirler.

Bu elemanlar, içinde çok sayıda lojik bağlaç bulunduran tümdevrelerdir (birkaç yüz - birkaç milyon). Bazılarının içinde bellek elemanları da (flip-flop) bulunur.

Tasarımcı bir "programlama" dili ve cihazı kullanarak bu bağlaçların arasında belli sınırlar içinde istediği bağlantıları gerçekleştirebilir.

Böylece sadece tek bir tümdevre kullanılarak karmaşık lojik devreler gerçekleştirilebilir.

Programlanabilir lojik elemanların çeşitli türleri vardır:

- *Programmable Logic Array - PLA*
- *Programmable Array Logic - PAL*
- *Generic Array Logic - GAL*
- *Complex PLD - CPLD*
- *Field-Programmable Gate Array - FPGA*

Programlama:

Bu elemanların iç yapılarının düzenlenebilmesi (programlama) için lojik kapılar arasındaki bağlantı noktalarında "sigortalar" (*fuse*) bulunur.

İlk PLD türlerinde (PLA, PAL) bipolar transistörler (Bkz. Bölüm 9) kullanılmıştır. Bu elemanlarda sigortalar sadece bir defa kopartılarak programlama yapılabilir.

Günümüz elemanlarında (GAL, CPLD, FPGA) CMOS transistörler ve programlama için bellek elemanları kullanılmaktadır. Bu elemanlar defalarca silinip programlanabilirler.

Bu elemanları programlamak için çeşitli donanım betimleme dilleri (*Hardware Description Language - HDL*) ve cihazlar kullanılır.

HDL örnekleri:

PALASM

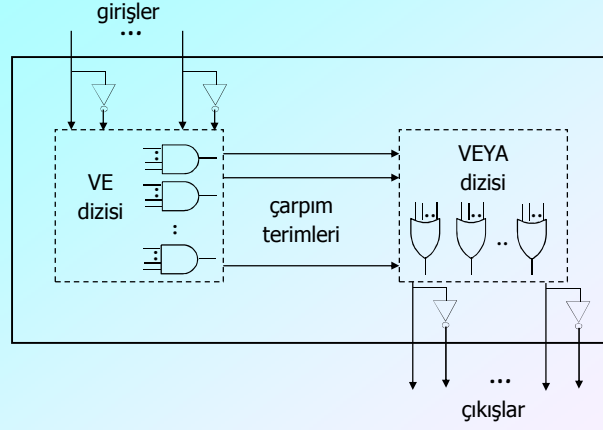
ABEL

Verilog

VHDL (*Veri high speed integrated circuits* HDL)

Programlanabilir Lojik Dizi (Programmable Logic Array - PLA)

Girişlerinde VE (çarpım) çıkışlarında ise VEYA (toplama) elemanları bulunur.



PLA'lar VE, VEYA gruplarının esnek olarak programlanabildiği elemanlardır.

PLA'ların sınırlarını belirleyen parametreleri şunlardır:

Giriş sayısı: n

Çıkış sayısı: m

VE kapısı sayısı: p

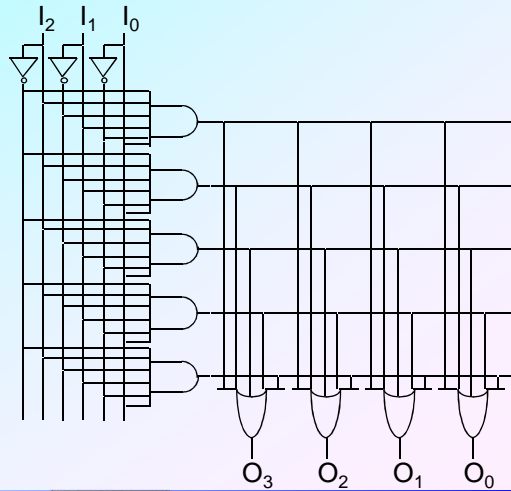
Bu tür bir eleman, " p çarpımlı $n \times m$ PLA" olarak adlandırılır.

Yandaki şekilde örnek olarak 5 çarpımlı 3×4 bir PLA gösterilmiştir.

PLA'larda yüz civarında bağlaç bulunur.

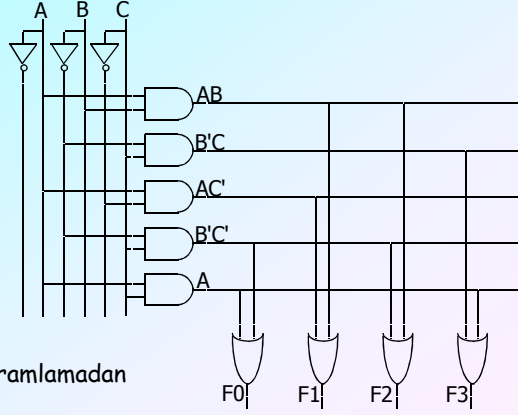
Örnek: 82S100

16 giriş, 8 çıkış, 48 çarpım (VE)



Örnek:

$$\begin{aligned}
 F0 &= A + B' C' \\
 F1 &= A C' + A B \\
 F2 &= B' C' + A B \\
 F3 &= B' C + A
 \end{aligned}$$



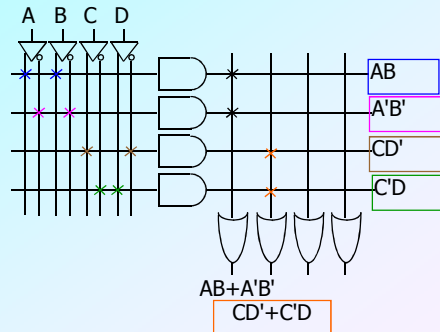
3x4PLA'nın iç bağlantıları, programlamadan sonra bu şekilde oluşur.

Basit Gösterilim: Çizimleri karmaşık hale getirmemek için PLA çizimlerinde tüm hatlar gösterilmez.

Onun yerine ilgili kapının girişine hangi hatlar bağlanacaksa o hattın üstüne X konur.

Örnek:

$$\begin{aligned}
 F0 &= A B + A' B' \\
 F1 &= C D' + C' D
 \end{aligned}$$

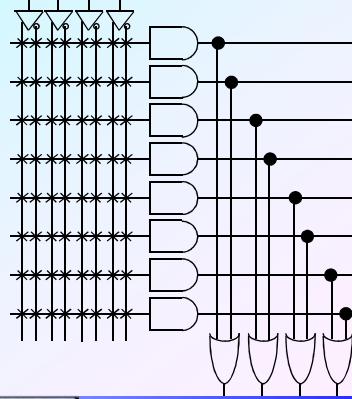


Programlanabilir Dizi Lojiği (Programmable Array Logic - PAL)

VE Bağlaçlarının girişleri PLA'larda olduğu gibi esnek bir biçimde programlanabilir. Ancak VEYA bağlaçlarının girişleri esnek değildir. Her VEYA bağlacının girişine sadece belli VE bağlaçlarının çıkışları bağlıdır. Örneğin ilk VEYA bağlacının girişine sadece ilk iki VE bağlacının çıkışları gelebilir. PAL'ler daha kolay programlanabilirler, daha ucuzdurlar, daha çok eleman içerebilirler.

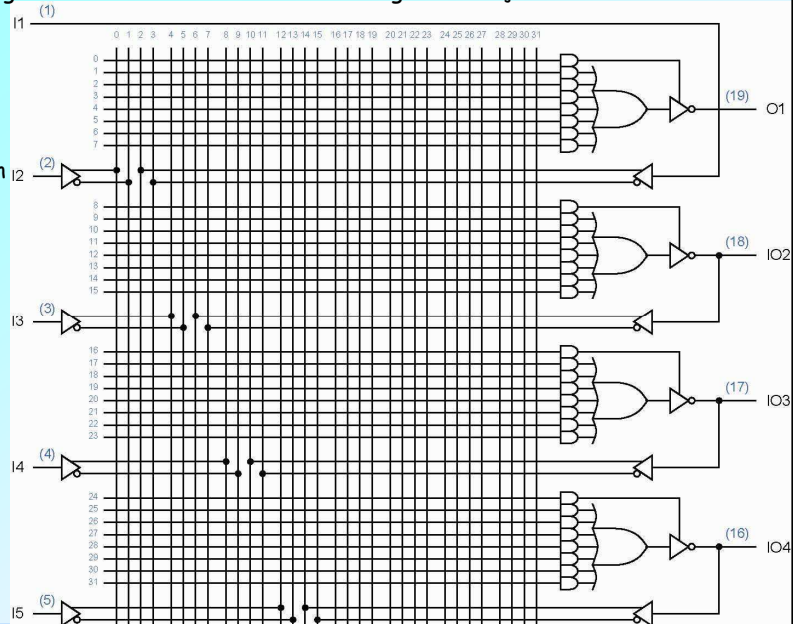
İlk olarak Monolithic Memories, Inc (MMI) firması tarafından üretilmiştir.

MMI, daha sonra Advanced Micro Devices (AMD) tarafından satın alınmıştır.

**Örnek: Aşağıda PAL 16L8 elemanın bir kısmı gösterilmiştir:**

16 giriş,
8 çıkış,
64 çarpım (VE)
Her VE kapısının
2x16 girişi
(kendisi ve
tümleyeni)
vardır.

PAL16L8		
1	I1	O1
2	I2	O2
3	I3	O3
4	I4	O4
5	I5	O5
6	I6	O6
7	I7	O7
8	I8	O8
9	I9	O9
10	I10	O10



Genel Dizi Lojiği (*Generic Array Logic - GAL*)

PAL ile benzer özellikler taşır.

İç yapısı CMOS transistörlerden oluşmaktadır. Defalarca silinerek tekrar programlanabilir.

İlk olarak Lattice Semiconductor firması tarafından oluşturulmuştur.

Örnek: GAL16V8

Karmaşık PLD (*Complex PLD - CPLD*)

Aynı tümleşik devrenin içinde birden fazla PLD (*macro cell*) bulunur.

Her bir PLD, GAL özelliklerine sahiptir.

Toplam kapı sayısı birkaç bin ile birkaç yüz bin arasındadır.

Hem PLD'lerin iç yapıları hem de aralarındaki bağlantılar programlanabilir.

Örnek: Atmel ATF1500

32 giriş/çıkış + 4 giriş

32 adet PLD (*macro cell*) içerir.

Sahada Programlanabilir Kapı Dizisi (*Field-Programmable Gate Array - FPGA*)

Çok sayıda işlevsel blok ve bloklar arasındaki bağlantılardan oluşurlar.

Defalarca silinerek tekrar programlanabilir.

Toplam kapı sayısı birkaç bin ile birkaç milyon arasındadır.

Karmaşık sayısal devrelerin (örneğin özel amaçlı mikroişlemciler) gerçekleştirilmesinde kullanılırlar.

CPLD'lere göre daha esnek ve daha yeteneklidirler ancak gecikmeleri ve maliyetleri daha yüksektir.

Örnek: Atmel AT6010

204 giriş/çıkış

30000 bağlaç