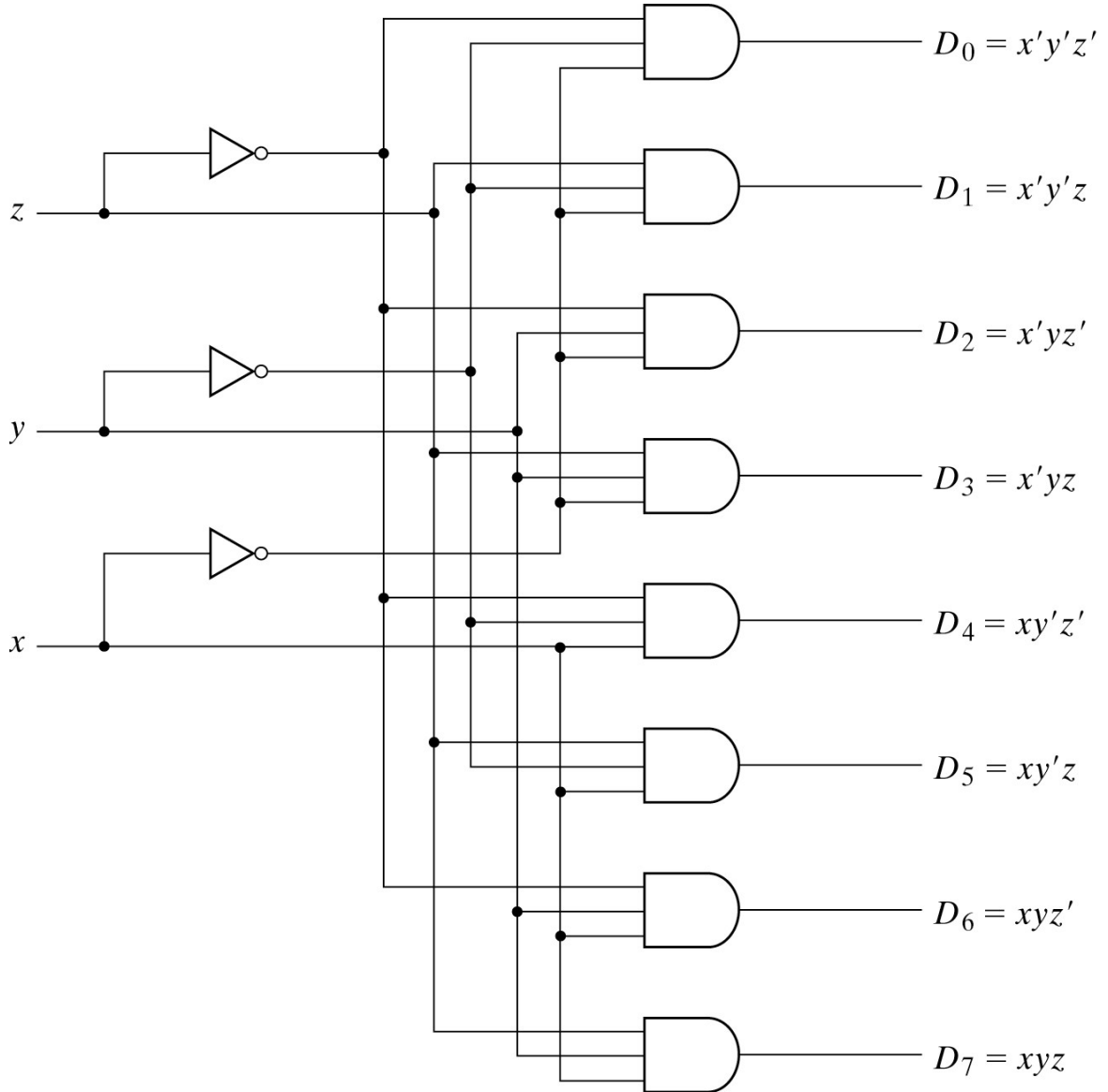


Kod Çözücüler ve Kodlayıcılar (Decoders, Encoders)

n bitlik bir ikili bilgi 2^n bağımsız kodlanmış veriyi temsil eder. 3 bitlik bir bilginin $2^3=8$ tane kombinasyonu vardır. Aşağıdaki devre 3'ten 8'e bir kod çözücüdür. Bu 8 tane kombinasyon minterimleri temsil eder. D ile belirtilen çıkışlar indeks numaralarına göre m ile belirtilen minterimlerdir.



3 girişli bir Boole fonksiyonunda olası minterimler $x'y'z'$, $x'y'z$, $x'yz'$, $x'yz$, $xy'z'$, $xy'z$, xyz' ve xyz 'dir. Bu şekilde girişten gelen kombinasyonun hangi minterimi temsil ettiği belirlenebilir.

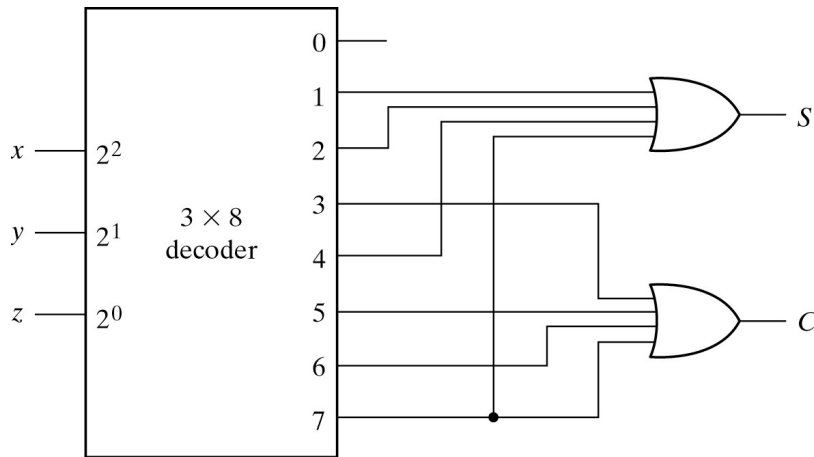
Girişler			Çıkışlar							
x	y	z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Kombinezonal Lojik Uygulaması

Bir kod çözücü girişine gelen değişkenlere göre minterimler oluşturur. O halde kombinezonal bir devre kod çözücüler ile dizayn edileceği zaman Boole fonksiyonu toplamaların çarpımı şeklinde ifade edilmeli ve uygun çıkışlar daha sonra bir VEYA kapısı ile çıkış fonksiyonu olarak kullanılmalıdır.

Örnek: Bir tam toplayıcıyı kod çözücüler kullanarak tasarlayınız.

Daha önce tam toplayıcı için toplam Boole fonksiyonu $S(x,y,z)=\Sigma(1,2,4,7)$ elde çıkış Boole fonksiyonu da $C(x,y,z)=\Sigma(3,5,6,7)$ olarak bulunmuştu. O halde kod çözücünün 1,2, 4 ve 7 nolu çıkışları bir VEYA kapısının girişi olacak şekilde toplam sonucunu oluşturur ve 3, 5, 6 ve 7 nolu çıkış uçları ise elde çıkışı üretecek şekilde bir VEYA kapsının girişine bağlanır. Bağlantılar aşağıdaki gibidir.

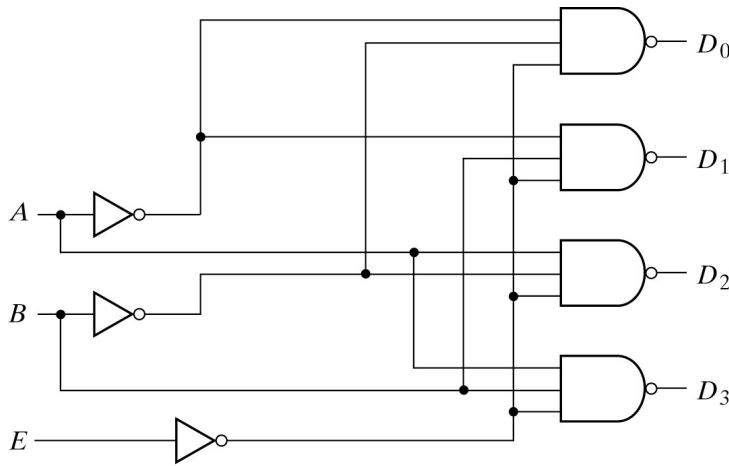


Çok sayıda değişkenden oluşan bir fonksiyonun minterimlerinin toplamını oluşturmak için çok girişli bir VEYA kapısı kullanmak gereklidir. k adet minterimli bir F fonksiyonu yerine 2^n-k minterimli F' tümlenmiş şekli de kullanılabilir. Sadece çıkışta istenilen fonksiyonu türetmek

için bir eviriciye gerek vardır. Bir fonksiyonun minterimleri sayısı $2^n/2$ 'den büyükse ($n=3$ ise 4'den büyük olmalı) bunun tümleyeni daha az minterim içereceğinden F' 'ne ait minterimlerin toplamı bulunur ve daha sonra bu toplamın tümleyeni alınarak F oluşturulur. Bütün bu işlemleri bir arada yürütmek için bir VEYADEĞİL kapısı kullanılır. Az sayıda minterimle ifade edilen uygulamalarda kod çözümler en iyi çözümü sağlar.

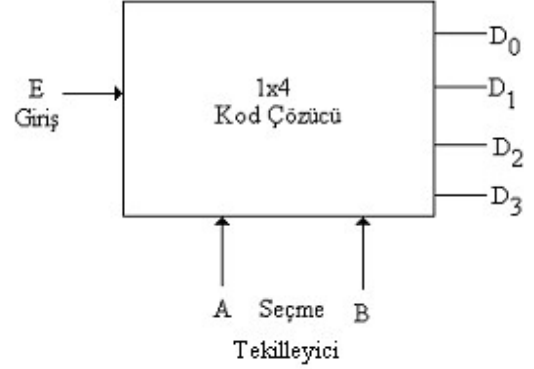
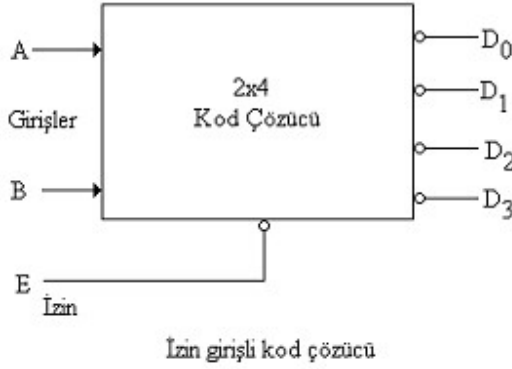
Tekilleyici (Demultiplexer)

Bazı entegre kod çözümler VEDEĞİL kapıları ile oluşturulurlar. Aşağıdaki devre izin girişi 2'den 4'e kod çözümler olarak bilinir. E girişi 1 olduğu sürece diğer girişlerin değeri ne olursa olsun çıkış sürekli 1 seviyesinde kalır. Eğer E girişi 0 ise kod çözümlerine izin verilmiş olur ve bu durumda girişteki kodlara göre tümlenmiş bir kod çözümler gibi çalışır. Kod çözme işlemi $E=0$ iken yapılır ve çıkışlar 0 durumdayken seçilir.

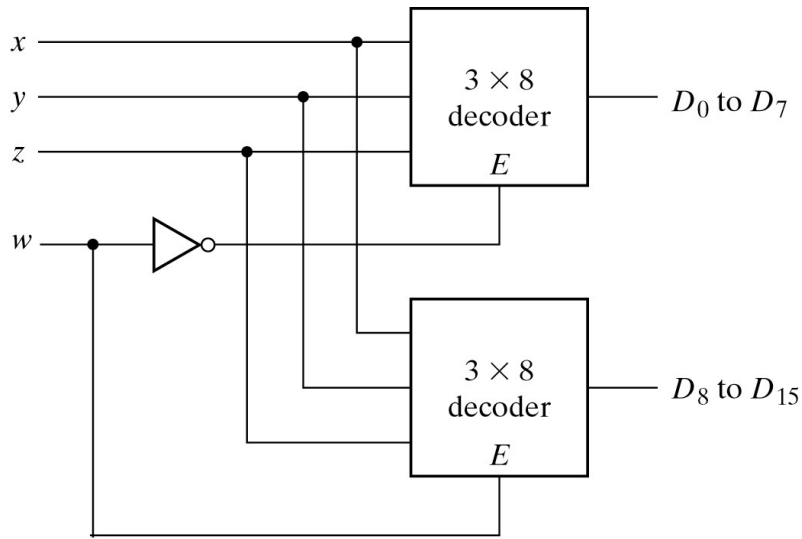


E	A	B	D ₀	D ₁	D ₂	D ₃
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Çıkıştaki küçük daireler tüm çıkışların tümleyeninin alındığını gösterir. İzin girişi bir kod çözümler bir veri tekilleyici olarak bilinir. Tekilleyici, girişteki veriyi alarak bunu 2^n hattan birine iletir. Belirli bir çıkış hattının seçilmesi n adet seçme hattıyla gösterilir. E giriş değişkenindeki bilgi iki seçme hattı A ve B'nin ikili değerlerine bağlı olarak çıkış hatlarından birine aktarılır. Örneğin, $AB=10$ için D_2 girişi E ile aynı değerde tutulurken diğer hatların tümü 1 değerinde tutulur.



Kod çözücü tekilleyici devreleri daha büyük bir kod çözücü tasarlamak üzere kullanılabilir. Örneğin, iki tane 3x8'lik kod çözücünden bir tane 4x16 kod çözücü elde edilebilir. 4x16 'lı bir kod çözücünün 4 tane girişi vardır. İlk 8 değer için en ağırlıklı bit (aşağıdaki şekilde w) sürekli 0 değerinde kalırken diğer 8 değer için sürekli 1 seviyesinde kalır. Bu bit, izin girişi olarak kullanılırsa aşağıdaki şekilde görüldüğü üzere iki tane 3x8'lik kod çözücü tekilleyiciden bir tane 4x16'lık kod çözücü elde edilebilir.



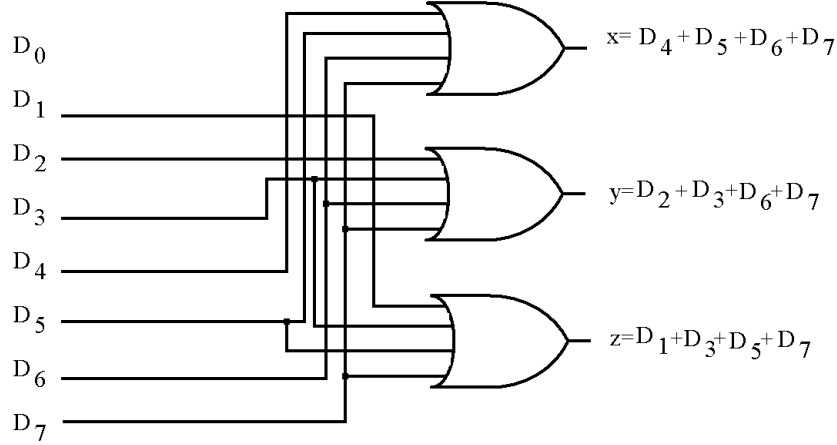
Kodlayıcılar(Encoders)

Bir kodlayıcı bir kod çözücünün tersi işlemi yapan sayısal devredir. 2^n girişten n adet çıkış üretir. Çıkış hatları, giriş değerine karşı düşen ikili kodu üretir. Sekizden ikiliye dönüşüm için doğruluk tablosu aşağıda verildiği gibidir. Görüldüğü üzere herhangi bir anda sadece bir giriş

1 değerine sahiptir. Birden fazla girişin 1 olması durumunda ise devrenin herhangi bir anlamlı çıkışa sahip olmadığı düşünülür.

Girişler								Çıkışlar		
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Bir kodlayıcı, girişleri doğruluk tablosundan elde edilen VEYA kapıları ile gerçekleştirilebilir.



Bu kodlayıcı da bazı belirsiz durumlar söz konusudur. Örneğin, her iki girişe aynı değer geldiği durumu göz önüne alalım. D₃ ve D₆ aynı anda 1'e eşit olursa, aynı anda tanımlı iki giriş olmadığı için kodlayıcı çıkışı 111 olup ne 3'ü ne de 6'yı verir. Bu karmaşayı çözmek için kodlayıcı devrelerde yalnız bir girişin kodlandığından emin olmak için öncelik oluşturulmalıdır. Büyük indeksli girişlerin daha öncelikli olduğu varsayılırsa aynı anda giriş olan D₃ ve D₆ için çıkış 110 olacaktır. Sekizliden ikiliye diğer bir belirsizlik de tüm girişlerin sıfır olduğu durumda tüm çıkışların sıfır olmasıdır. Burada problem D₀ 1 olduğu durumda tüm çıkışların sıfır olmasıdır. Bu belirsizlik, girişlerin hiçbirinin aktif olmadığını belirleyen ilave bir çıkış kullanılarak çözülebilir.

Öncelik Kodlayıcısı

Bir öncelik kodlayıcısı, öncelik fonksiyonunu da içeren bir kodlayıcıdır. Öncelik kodlayıcısının işlevi, iki veya daha fazla giriş aynı anda 1'e eşit olduğunda en yüksek öncelikli girişin göz önüne alınmasını sağlamaktadır. Dört girişli öncelikli bir kodlayıcının doğruluk tablosu aşağıda verilmiştir. X'ler dikkate alınmayan etkisiz koşullardır. D₃ girişi en öncelikli giriştir.

Girişler				Çıkışlar		
D0	D1	D2	D3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Eğer D₃ girişi 1 ise diğer girişlerin değerlerinin bir önemi yoktur. xy çıkışı 3 (11) değerini alır. D₃ girişi 0 ve D₂ girişi 1 ise diğer iki girişin bir önemi yoktur ve çıkış 10 (2) değerini alır. Burada geçerli çıkış göstergesi V ile gösterilmiş olup sadece bir veya daha fazla giriş 1 değerini aldığı anda değeri 1 olur. x ve y girişleri için doğruluk tablosu oluşturulursa;

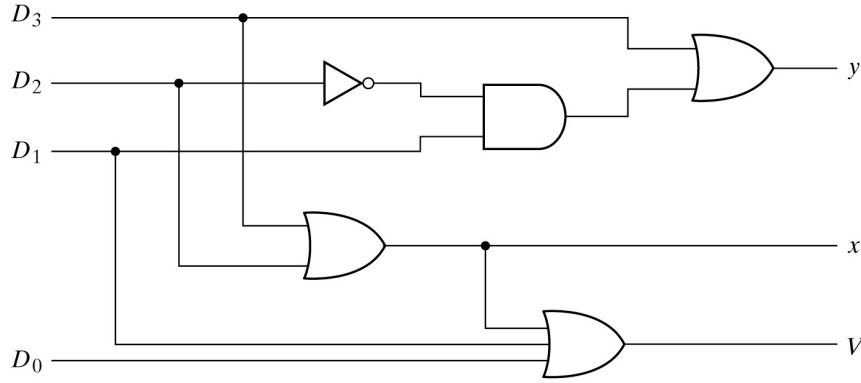
		x						y			
		D ₂ D ₃		D ₀ D ₁				D ₂ D ₃		D ₀ D ₁	
		00	01	11	10			00	01	11	10
00	00	X	1	1	1	00	00	X	1	1	
01	01		1	1	1	01	01	1	1	1	
11	11		1	1	1	11	11	1	1	1	
10	10		1	1	1	10	10		1	1	

$x = D_3 + D_2$ $y = D_3 + D'_2 D_1$

		D ₂ D ₃				V
D ₀ D ₁		00	01	11	10	
00			1	1	1	
01		1	1	1	1	
11		1	1	1	1	
10		1	1	1	1	

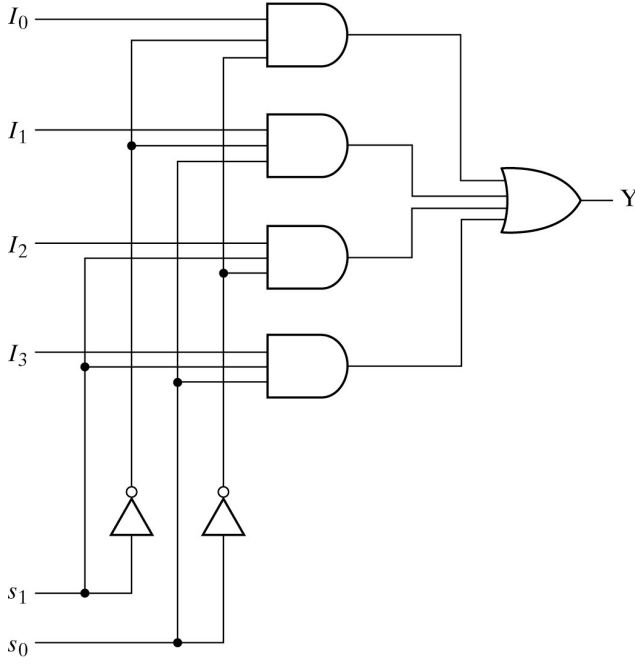
$$V = D_0 + D_1 + D_2 + D_3$$

Yukarıdaki doğruluk tablolarından elde edilen dört girişli öncelik kodlayıcısı aşağıda görülmektedir.



Veri Seçiciler (Multiplexer)

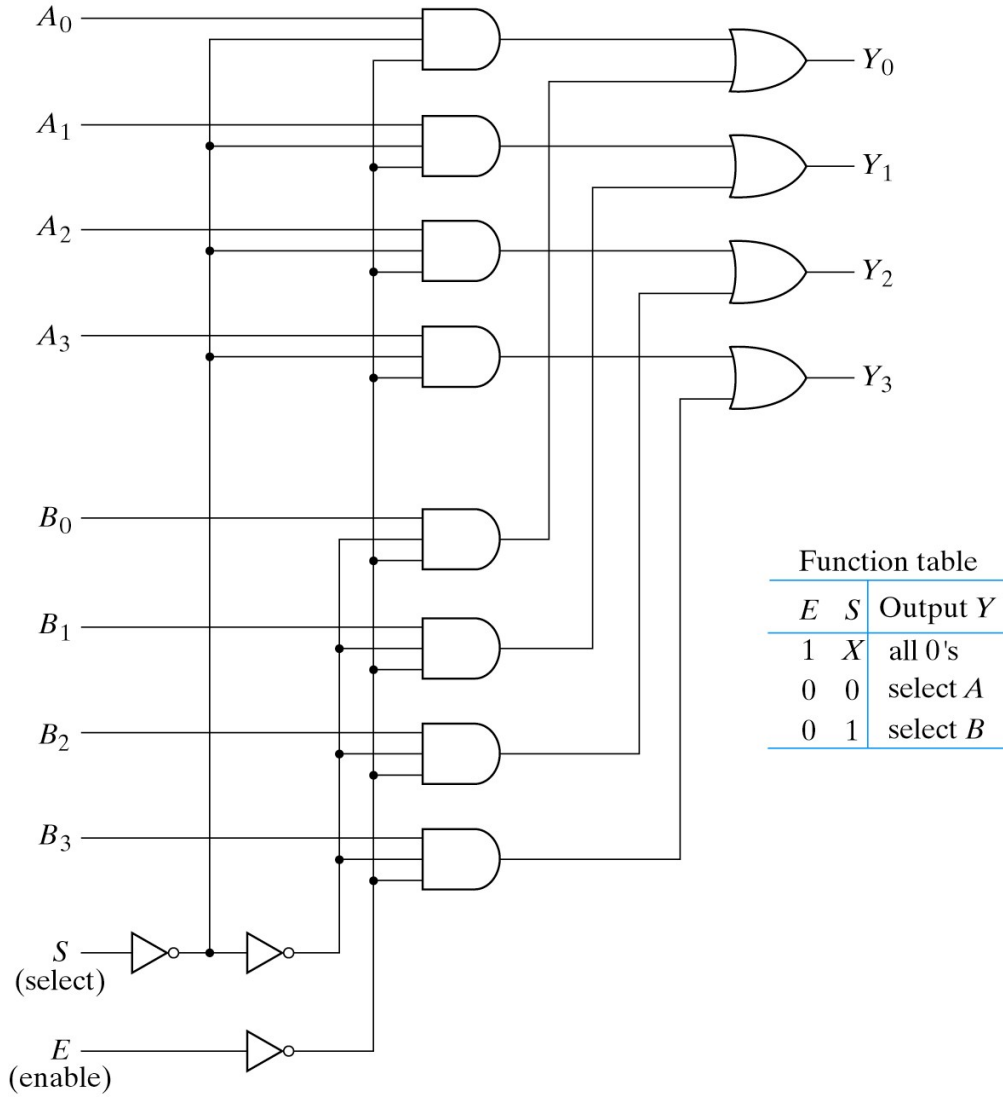
Veri seçmenin anlamı çok sayıdaki bilgi birimlerinin daha az sayıdaki kanallara veya yollara aktarılmasıdır. Sayısal bir veri seçici bir veya daha fazla giriş hattından ikili bilgiyi seçen ve bunu tek giriş hattına bağlayan kombinezonal devredir. Özel bir giriş hattının seçilmesi bir grup seçme hattı ile kontrol edilir. 2^n adet giriş hattı varsa n tane seçme ucu vardır. Aşağıda, 4'ten 1'e veri seçici lojik diyagramı, doğruluk tablosu ve blok diyagramı gösterimi görülmektedir. S_1 ve S_0 veri seçme uçlarına bağlı olarak girişlerden gelen I_0 , I_1 , I_2 ve I_3 verilerinden bir tanesi çıkışa aktarılır.



s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

Devrenin çalışmasını anlamak için $S_1S_0=10$ olduğu durumu düşünelim. VE kapılarının girişlerinin en az bir tanesi 0 olursa çıkış 0 olacağı için I_0 , I_1 ve I_3 girişlerinin uygulandıkları VE kapılarının bir veya iki girişi 0 olduğu için çıkışları da 0 olacaktır. O halde VEYA ile sonlandırılmış kombinezonal devre doğrudan I_2 girişinin değerini alacaktır. Böylece seçilen bir girişten çıkışa bir yol sağlanmış olacaktır.

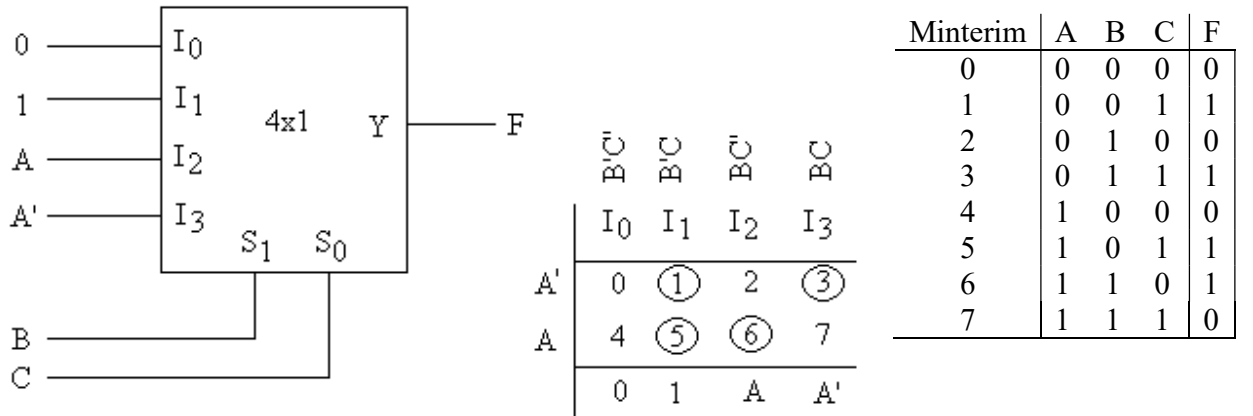


Veri seçicideki VE kapıları ve eviriciler kod çözücü bir devreye benzer ve giriş seçme hatlarının kodunu çözer. Veri seçiciler genellikle multiplexer kelimesinin kısaltması olan MUX ile adlandırılırlar. Tıpkı bir kod çözücüde olduğu gibi veri seçicide de bir izin girişi vardır. İzin girişinin belirli durumlarında tüm çıkışlar yasaklanır, diğer durumlarda ise devre veri seçme işlemini normal bir şekilde yürütür. İzin girişi daha büyük sayısal sistemlerde iki veya daha fazla veri seçme tüm devresinin bir birine bağlandığı durumlarda kullanılır. Aşağıdaki devrede dörtlü 2'den 1'e veri seçme tüm devresi görülmektedir. Eğer E , izin girişi 1 ise veri seçme işlemi yasaklanmıştır demektir. Bu durumda S girişinin aldığı değerlerin bir önemi yoktur. Devrede dört tane veri seçici vardır. Bu çıkışlar ya A ya da B değerlerinden bir tanesini alırlar. Seçme

işleminin yapılması için E girişi sürekli 0 seviyesinde tutulmalıdır. Eğer $S=0$ ise A girişleri, $S=1$ ise B girişleri seçilecektir demektir.

Boole Fonksiyonlarının Uygulanması

$n+1$ elemanlı bir Boole fonksiyonu varsa, bu değişkenlerden n tanesi seçme ucu olarak kullanılır, veri seçicinin diğer uçlarına ise kalan değişkenin değili, kendisi, 1 ve 0 uygulanır.



$F(A,B,C) = \Sigma(1,3,5,6)$ Boole fonksiyonunu göz önüne alalım.

3 değişkenli bir Boole fonksiyonu olduğuna göre B ve C değişkenleri seçme uçları olarak seçilir. A, A', 1 ve 0 girişleri ise veri seçicinin girişlerine uygulanır. Kullanılacak olan minterimler m_1 ($A'B'C$), m_3 ($A'BC$), m_5 ($AB'C$) ve m_6 (ABC')'dir. Veri seçicide seçme uçları $BC=00$ için I_0 girişi, $BC=01$ için I_1 , $BC=10$ için I_2 ve $BC=11$ için I_3 girişi seçilir. $BC=00$ için m_0 ve m_4 minterimleri gerçekleşecek ve her ikisi de 0 çıkışı ürettiği için I_0 0'a bağlanmalıdır. $BC=01$ için, m_1 ve m_5 minterimleri gerçekleşecek ve her ikisi de 1 çıkışı ürettiği için I_1 1'e bağlanmalıdır. $BC=10$ için, m_2 ve m_6 minterimleri gerçekleşecek ve m_2 için 0, m_6 için 1 değeri çıkış olduğu için A değeri I_2 girişine bağlanır. $BC=11$ için, m_3 ve m_7 minterimleri gerçekleşecek ve m_3 için 1, m_7 için 0 değeri çıkış olduğu için A' değeri I_3 girişine bağlanır.

Genel bir yöntem verilecek olursa;

Minterimler için seçilen değişken sıralaması, ABC... olsun. BCD.... Değişkenleri B'den başlanarak en yüksek ağırlıklı seçme ucundan S_0 'a doğru bağlanır. Aşağıdaki şekilde daire içine alınmış terimler kullanılan minterimlerdir. Eğer bir sütunda hiçbir terim daire içine alınmamışsa o giriş 0'a, tek bir terim daire içersine alınmış ise daire hangi satırda ise A ya da A' değişkenine, iki terim daire içersine alınmış ise 1'e bağlanmalıdır.

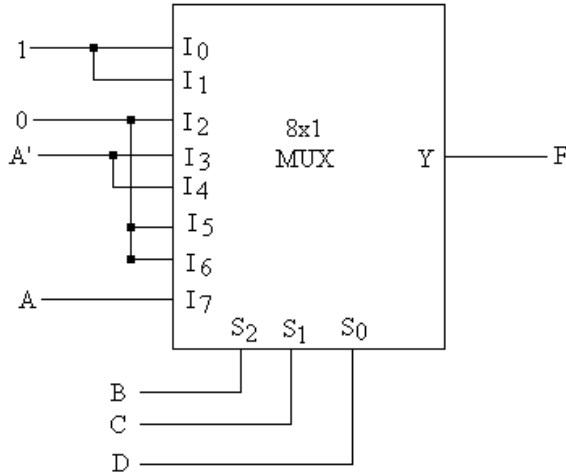
	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
	I_0	I_1	I_2	I_3
A'	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	A'

Örnek: $F(A,B,C,D)=\Sigma(0,1,3,4,8,9,15)$ minterimleri ile belirlenmiş Boole fonksiyonunu veri seçiciler kullanarak gerçekleştiriniz.

Seçme uçları $B(S_2)$, $C(S_1)$, $D(S_0)$ olarak belirlenir. n değişkenli olduğuna göre $2^{n-1} \times 1$ 'lık bir veri seçici gereklidir. Önce uygulama tablosu oluşturulur.

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	①	②	3	④	5	6	7	
A	⑧	⑨	10	11	12	13	14	⑮
	1	1	0	A'	A'	0	0	A

Bu uygulama tablosundan gerekli devre gerçekleştirilir.



Kombinezonal devrelerin uygulanması için kod çözücü ve veri seçici yönteminin karşılaştırılması:

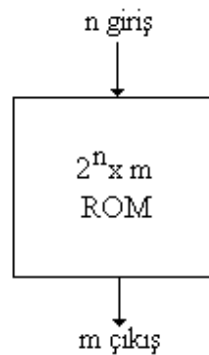
- ✓ Kod çözücü yöntemi her bir fonksiyon çıkışı için bir VEYA kapısı gerektirir. Fakat tüm minterimlerin oluşturulması için bir kod çözücü yeterlidir.

- ✓ Her bir fonksiyonun üretilmesi için ayrı bir veri seçici gereklidir ve çıkışlarının VEYA kapısı ile sonlandırılmasına gerek yoktur.
- ✓ Az çıkışlı bir devre veri seçici ile çok çıkışlı bir devre kod çözücü ile tasarlandığında daha az tüm devre elemanı kullanılır.
- ✓ Her iki eleman da kombinezonal devrelerin tasarımında kullanılabilmelerine rağmen kod çözücüler ikili bilgilerin çözülmesinde, veri seçiciler ise çok girişli ve tek çıkışlı sistemler arasındaki yolun seçilmesinde kullanılırlar.

Salt Okunur Bellek (Read-Only Memory:ROM)

Bir kombinezonal devrenin üretilmesi için minterimlerin toplamalarının üretilmesi yeterlidir. Bir kombinezonal devre, minterimler bir kod çözücü ile elde edilip toplamalar da bir VEYA kapısı kullanılarak gerçekleştirilebilir. Bir salt okunur bellek hem kod çözücünün hem de VEYA kapılarının içerisinde bulunduğu bir tümdevre kılıfıdır. ROM, karmaşık kombinezonal devrelerin tümdevre içinde gerçekleşmesini sağlayan veya ikili bilgiyi sürekli saklayan hafıza elemanı olarak kullanılır.

ROM esas olarak içinde ikili bilgileri sürekli olarak saklayan bir hafıza elemanıdır. Saklanacak ikili bilgi tasarımcı tarafından belirlendikten sonra bu ikili bilgiyi temsil eden desen eleman içerisine yerleştirilir. ROM'lar belirli bir yapı için programlanabilen iç elektronik sigortalar ile donatılmış olarak kullanıma hazırdırlar. Desen bir kere oluşturulduktan sonra enerjisi kesilse dahi desen aynı şekilde aygıtın içerisinde kalır. Bir ROM'a ilişkin blok diyagramı aşağıda gösterilmiştir.

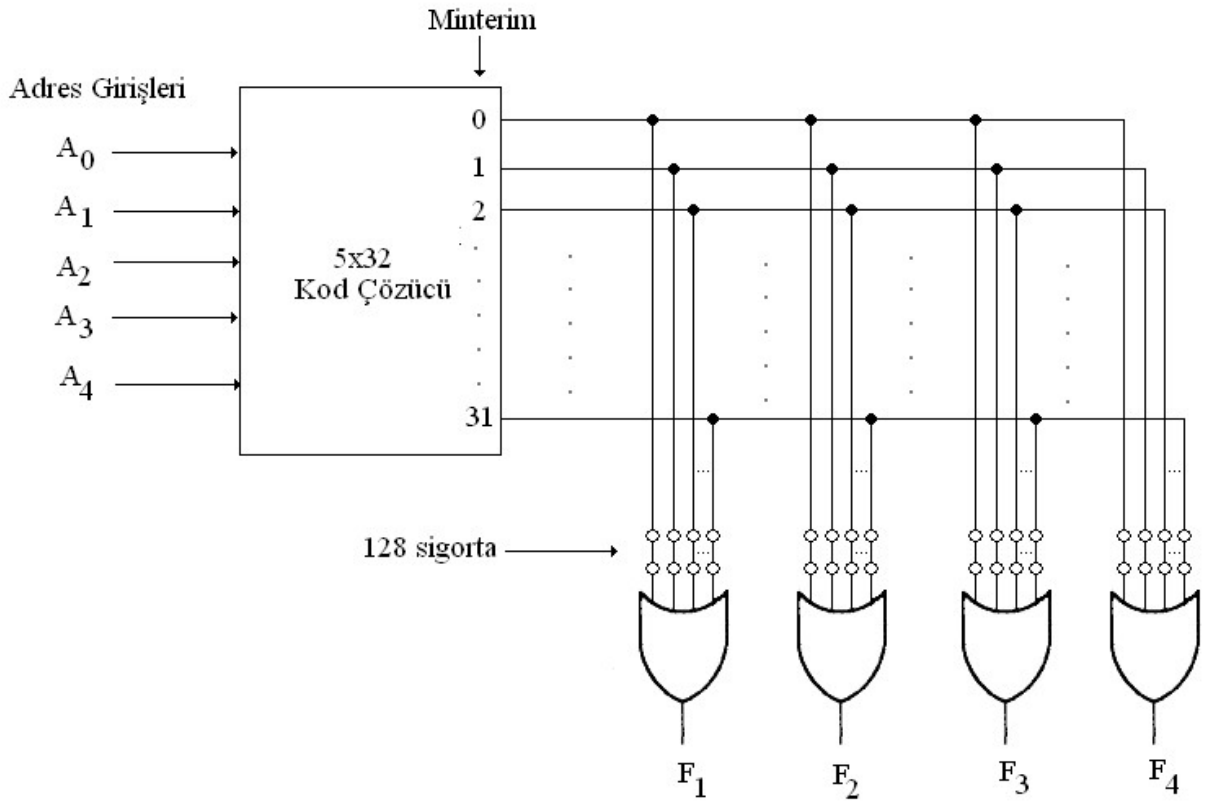


Bu ROM n adet giriş, m adet çıkış hattından oluşmuştur. Giriş değişkenlerinin oluşturduğu bit kombinasyonları *adres*, çıkış değişkenlerinin oluşturduğu bit kombinasyonları ise *sözcük* (*kelime*) olarak adlandırılır. Sözcük başına düşen bit sayısı çıkış hattı sayısı olan m 'e

eşittir. Bir adres aslında n değişkenli minterimlerden birini gösteren bir ikili sayıdır. Dolayısıyla, n giriş değişkenine bağlı 2^n tane adres vardır. Bir çıkış sadece bir adres tarafından seçilebildiği için ROM'da 2^n tane de farklı sözcük vardır. Bu sözcükler birimde saklanır ve herhangi bir anda girişteki adres değerlerine bağlı olarak çıkış hattından elde edilir. Bir ROM'da sözcük sayısı 2^n ile sözcük başına düşen bit sayısı da m ile karakterize edilir.

32×8 'li bir ROM'u göz önüne alalım. Birim, her biri 8 bit olan 32 farklı sözcüğe sahiptir. Bunun anlamı, 8 tane çıkış hattı olduğu ve her birimde 32 farklı sözcük saklandığıdır. Bu sözcüklerden herhangi biri çıkış hattına uygulanabilir. 32×8 'lik bir ROM'da $2^5=32$ olmak üzere 5 giriş vardır. Bu girişler adresleri belirlerler. 00000 girişi için 0 adresi, 11111 girişi için ise 31 adresi seçilip çıkış hattına bu adresteki sözcük iletilir.

ROM bir kod çözücü ve sigortalı VEYA kapıları içerir. Çıkıştaki sigortalar isteğe bağlı olarak atılarak minterimlerin toplamı şeklinde ifade elde edilebilir. Aşağıdaki şekilde 32×4 'lük bir ROM görülmektedir. 5 giriş hattı kod çözücü ile adresleri belirlendikten sonra çıkış hattı üzerindeki sigortalı VEYA kapılarında bulunan toplam 128 sigortadan istenilenler atırılıp desen oluşturulur.



Kombinezonal Lojik Uygulamaları

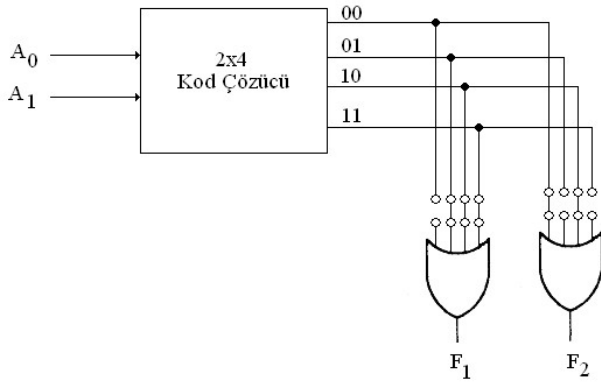
ROM'un lojik diyagramına bakılacak olursa, çıkışların her birinde n adet giriş değişkenine bağlı olarak tüm minterimlerin toplamı yer almaktadır. Daha önceki konulardan hatırlanacağı gibi her bir fonksiyon minterimlerin toplamı şeklinde ifade edilebilir. Elde edilmek istenen fonksiyonda bulunmayan minterimler çıkış hattı üzerindeki sigortalar atırılarak toplamdan çıkartılabilir. Bu şekilde ROM'un çıkışlarında istenen çıkışı temsil eden bir Boole fonksiyonu ifadesi elde edilebilir. n girişli m çıkışlı bir kombinezonal devre için $2^n \times m$ boyutunda bir ROM'a ihtiyaç vardır. Sigortaların atırılması ROM'un programlanması anlamına gelir.

Örnek olarak aşağıdaki devreyi inceleyelim. İki girişli, iki çıkışlı bir kombinezonal devre tasarlanmak istenmektedir. Boole fonksiyonunun minterimleri aşağıdaki gibi verilmiştir.

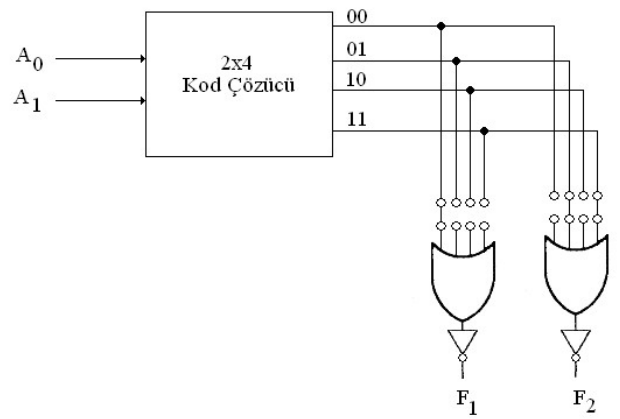
$$F_1(A_1, A_0) = \Sigma(1, 2, 3)$$

$$F_2(A_1, A_0) = \Sigma(0, 2)$$

A_1	A_0	F_1	F_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	0



VE-VEYA kapıları ile ROM



VE-VEYA-EVİRME kapıları ile ROM

Pratikte böyle bir uygulama için kesinlikle ROM kullanılmaz. Çünkü sistemin yapısı oldukça basittir. ROM'un kullanılmasının avantajı daha karmaşık uygulamalarda ortaya çıkmaktadır. Boole fonksiyonu ile belirlenmiş ifadede iki giriş (n) (A_1, A_0) ve iki çıkış (m) (F_1, F_2) olduğuna göre $2^n \times m$, yani 4×2 'lik bir ROM'a ihtiyaç vardır. Aynı zamanda girişte 2×4 'lük bir

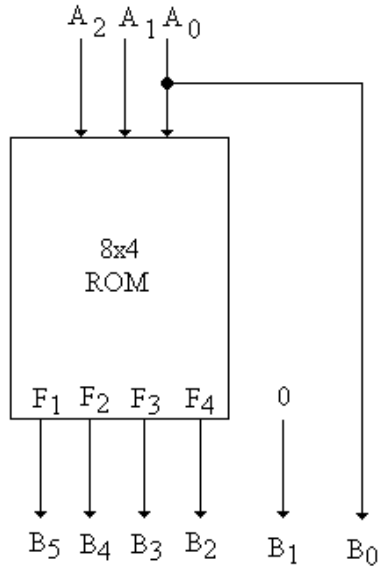
kod çözücüsü olmalıdır. Doğruluk tablosu oluşturmak hangi minterimlerin toplama katılacağını görmek açısından kolaylık sağlar. Şekilde görüldüğü üzere F_1 ve F_2 için minterimlerden faydalanılarak doğruluk tabloları oluşturulmuştur. F_1 için toplamaların çarpımı ifadesinde kullanılan minterimler 1, 2 ve 3 olduğuna göre kod çözücünün 01, 10 ve 11 kodlu çıkışları ROM'un çıkışındaki sigortalı VEYA kapısına giriş olarak bağlanmalıdır. Kullanılmayan minterim ise sigortalı VEYA kapısının girişindeki sigorta attırılarak toplama katılması engellenir. F_2 için toplamaların çarpımı ifadesinde kullanılan minterimler 0 ve 2 olduğuna göre kod çözücünün 00 ve 10 kodlu çıkışları ROM'un çıkışındaki sigortalı VEYA kapısına giriş olarak bağlanmalıdır. Kullanılmayan minterimler, 01 ve 11 ise sigortalı VEYA kapısının girişindeki sigortalar attırılarak toplama katılması engellenir.

Örnek: Girişine uygulanan 3 bitlik bir sayının karesinin ikili karşılığını çıkışta verebilecek bir kombinezonal devreyi ROM kullanarak tasarlayınız.

İlk olarak kombinezonal devre için doğruluk tablosu oluşturulup kullanılacak olan minterimler bulunmalıdır.

Girişler			Çıkışlar						Onlu
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
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

Doğruluk tablosu oluşturmak ROM ile yapılacak uygulamalarda yeterlidir. Bazı durumlarda doğruluk tablosundaki değerlere bağlı olarak ROM'un doğruluk tablosu daha az elemanlı bir tablo olabilir. 3 giriş ve 6 çıkış için istenilen özellikleri sağlayan ROM, 3 girişli olmalı ve 6 çıkışlı olmalıdır. Fakat doğruluk tablosuna bakıldığında çıkışlardan B_0 , A_0 giriş değişkeni ile aynı değerleri almaktadır. Dolayısıyla çıkışa A_0 giriş değişkeni verilebilir. Aynı şekilde B_1 çıkışı ise sürekli 0 olduğu için bu çıkışta sürekli lojik 0 seviyesinde tutulması yeterlidir. O halde 3 girişli 4 çıkışlı 8×4 'lük bir ROM yeterli olacaktır.



Blok diyagram

A ₂	A ₁	A ₀	F ₁	F ₂	F ₃	F ₄
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

Doğruluk Tablosu

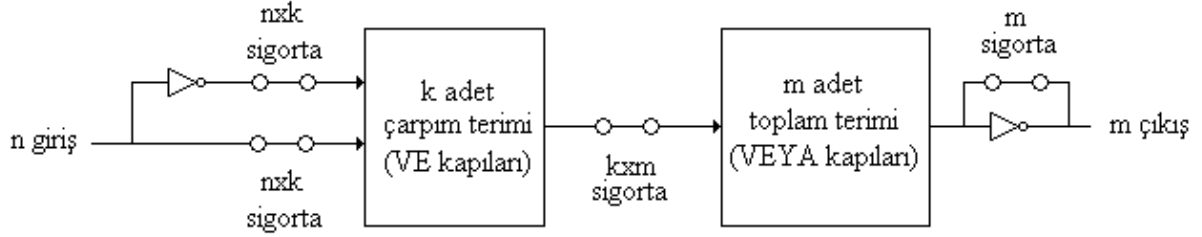
ROM Tipleri

ROM'lar iki şekilde programlanabilir. Birisi maske programlama olarak bilinen ve üretim esnasında üreticiye verilen doğruluk tablosu yardımı ile hazır olarak istenilen fonksiyonu gerçekleştiren bir maske üretilerek ROM bu şekilde programlanır. Bu durumda özel bir üretim olduğu için fiyat da ona göre yüksek olacaktır. Bu durum ancak oldukça fazla üretim gerçekleştiğinde faydalı bir yöntem olacaktır. Daha küçük miktarlar için PROM (Programlanabilir ROM) kullanılır. Attırılmak istenen sigortanın çıkışından bir miktar akım uygulanarak sigorta attırılır. Bu tip elemanlar ROM ve PROM bir kere programlandıktan sonra bir daha kesinlikle desen değişmez ve programlanamaz. Küçük bir hatada artık ROM veya PROM kullanılamaz hale gelir. Bu durumda üçüncü bir eleman olan EPROM (silinebilir-erasable PROM) olarak adlandırılan bir eleman kullanılır. Mor ötesi ışık altına tutulduklarında içerisine yüklenmiş olan desen silinebilir. Bu silme işlemi oldukça uzun süren ve silme işlemini garanti etmeyen bir yoldur. Bu nedenle bu eleman yerine EEPROM olarak adlandırılan ve elektriksel olarak silinebilir PROM'lar geliştirilmiştir.

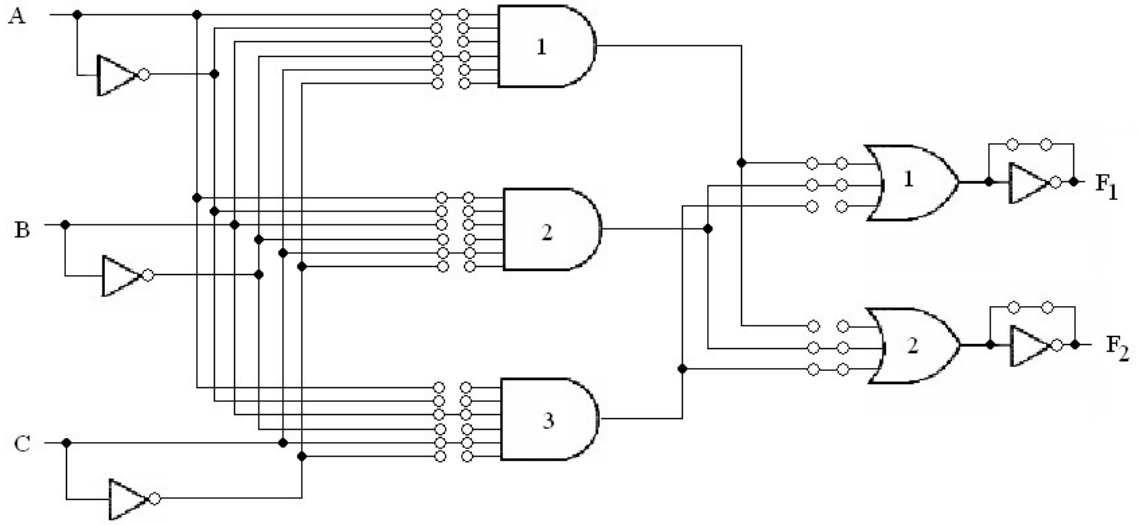
ROM'lar iki amaçla kullanılırlar. Herhangi bir kombinezonal devrenin tasarımında kullanılabildikleri gibi bilgi saklama elemanı olarak da kullanılırlar. Girişteki değişkenler adres değerini belirler ve o adresteki bilgi çıkışa aktarılır.

Programlanabilir Lojik Dizi (PLA)

Etkisiz koşulların olduğu bir kombinezonal devre ROM kullanılarak gerçekleştirilirse bu etkisiz koşullar hiçbir zaman gerçekleşmeyecek bir adres girişine karşılık düşer. Bu adreslerde yer alan sözcüklerin programlanmasına gerek yoktur ve orijinal değerlerinde bırakılabilirler. Etkisiz koşulların fazla olduğu kombinezonal devreleri gerçekleştirmek için ROM yerine programlanabilir lojik dizi veya PLA adı verilen ikinci tip LSI elemanları kullanmak daha ekonomik olacaktır. Bir PLA, ROM'a benzer fakat PLA'da değişkenlerin tümünün kodlanması sağlanmaz ve tüm minterimler üretilmez. Girişinde ROM'da olduğu gibi bir kod çözücü içermez, bunun yerine bir VE kapı dizisi içerir. Bu kapılardan her biri, giriş değişkenlerinden bir çarpım terimi üretilmesi sağlar. PLA içerisinde VE ve VEYA kapıları arasında bir sigorta bulunmaktadır. Çarpımların toplamı şeklindeki ifadelerin şeklindeki özel Boole fonksiyonları uygulandığında istenilen çıkışı sağlayan sigortalar olduğu gibi bırakılırken diğerleri atılır.



Yukarıdaki şekilde bir PLA'ya ait blok diyagramı görülmektedir. n adet giriş için, k adet sigorta değişkenlerin kendisi, k adet sigorta değişkenin tümleyeni ve k adet VE giriş kapısı, m adet VEYA kapısı girişi için sigorta ve m adet de çıkış hattı ve çıkışın tümleyeni için sigorta bulunur. Aşağıdaki şekilde özel bir PLA yapısı görülmektedir.



Burada kullanılan giriş değişkenleri A, B ve C dir. Girişteki VE kapıları ile AB' , AC ve BC çarpım terimleri oluşturulup $F_1=AB'+AC$, ve $F_2=AC+BC$ çıkış fonksiyonları tanımlanmıştır. Kullanılmayan terimlere ait tüm sigortalar atırılmıştır.

PLA çok girişli ve çok çıkışlı kombinezonal devrelerin yerine düşünülmelidir. Basit bir örnek üzerinde PLA'nın programlanmasını düşünelim. Burada seçilen örnek PLA kullanımına uygun olmayıp sadece çalışmasını ve programlanmasını anlamak üzere seçilmiştir. ROM'da minterimlerin toplamı şeklinde bir yapı göz önüne alınırken PLA'da çarpımların toplamı şeklinde bir ifade göz önüne alınmaktadır. O halde 3 girişli herhangi bir kombinezonal devreye ait doğruluk tablosunun aşağıdaki gibi verildiğini varsayalım.

A	B	C	F ₁	F ₂
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

PLA program tablosu

A	BC			
	00	01	11	10
0				
1	1	1	1	

$$F_1 = AB' + AC$$

A	BC			
	00	01	11	10
0			1	
1		1	1	

$$F_2 = AC + BC$$

	Çarpım Terimi	Girişler			Çıkışlar	
		A	B	C	F1	F2
AB'	1	1	0	-	1	-
AC	2	1	-	1	1	1
BC	3	-	1	1	-	1
		T	T	T/C		

Üç değişkenli bir kombinezonal devrenin PLA ile gerçekleştirilmesi için PLA program tablosunun oluşturulması gerekmektedir. Yukarıdaki program tablosundan üç terimin çarpılacağı anlaşılmaktadır. Dolayısıyla girişte 3 tane VE kapısına ihtiyaç vardır. Girişler sütununda ABC değişkenlerinin altında eğer 1 yazıyorsa değişkenin kendisi, 0 yazıyorsa değişkenin tümleyeni, - yazıyor ise değişkenin çarpım terimi içinde yer almayacağı anlaşılr. Bu çarpım terimlerinin hangisinin ya da hangilerinin çıkışa verileceği ise çıkışlar sütunu altında yer alan çıkış göstergeleri ile belirlenir. Eğer 1 yazıyorsa o terim çıkıştaki VEYA kapısının girişine verilecek, - yazıyor ise o çıkışa verilmeyecektir. O sütunun en altında yer alan T ve C değerleri ise şu anlama gelir:Eğer çıkışta fonksiyonun kendisi üretilecek ise T, tümleyeni üretilecekse C ile gösterilir. T, True anlamında C ise Complement anlamında kullanılmaktadır. Bu da çıkışta bulunan DEĞİL kapılarına paralel bağlı sigortaların atırılıp atırılmayacağını belirtmek için kullanılır. Eğer tasarımda çok sayıda çarpım terimi oluşuyor ise bu durumda değişkenin kendisi değil de tümleyeninin bulunması yoluna gidilir. Çünkü, PLA sınırlı sayıda çarpım terimi üzerinde işlem görür.

Örnek: Bir kombinezonal devre aşağıdaki gibi tanımlanmıştır. Devreyi PLA kullanarak gerçekleştiriniz.

$$F_1(A,B,C)=\Sigma(3,5,6,7)$$

$$F_2(A,B,C)=\Sigma(0,2,4,7)$$

Minimum değişkenli fonksiyonlar Karnaugh diyagramı yardımıyla bulunacak olursa;

$$F_1=AC+AB+BC$$

$$F'_1=B'C'+A'C'+A'B'$$

$$F_2= B'C'+A'C'+ABC$$

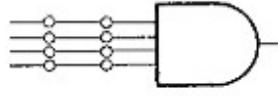
$$F'_2=B'C+A'C+ABC'$$

Eğer F_1 ve F_2 PLA ile gerçekleştirilmek istenirse, toplam 6 tane ve birbirinden farklı terim içerecektir. Bunun yerine F'_1 ile F_2 gerçekleştirilip çıkışta F'_1 'nın tümleyeni alınırsa yine aynı işlem yerine getirilmiş olur. Fakat bu durumda gerçekleştirilecek çarpım işlemi ortak terimleri fazla olduğu için 4 olacaktır.

	Çarpım Terimi	Girişler			Çıkışlar	
		A	B	C	F ₁	F ₂
B'C'	1	-	0	0	1	1
A'C'	2	0	-	0	1	1
A'B'	3	0	0	-	1	-
ABC	4	1	1	1	-	1
					C	T
					T/C	

Programlanabilir Dizi Lojiği (PAL)

Programlanabilir lojik devreler içerisinde sigortalar ile birbirine bağlı yüzlerce kapı devresi içerirler. Böyle devrelerin iç lojik çizimi için özlü bir gösterim olarak dizi lojiği'ni referans almak uygun olacaktır. Alışlagelmiş gösterimde bir kapıya gelen girişler sigortaları ayrı ayrı gösterilerek tüm girişler gösterilir. Fakat dizi lojiği gösteriminde yatay olarak gelen tek bir hat ve girişler ise dikey hatlar ile gösterilir. Yatay hat ile dikey hattın her kesiştiği nokta bir sigortayı tanımlar.

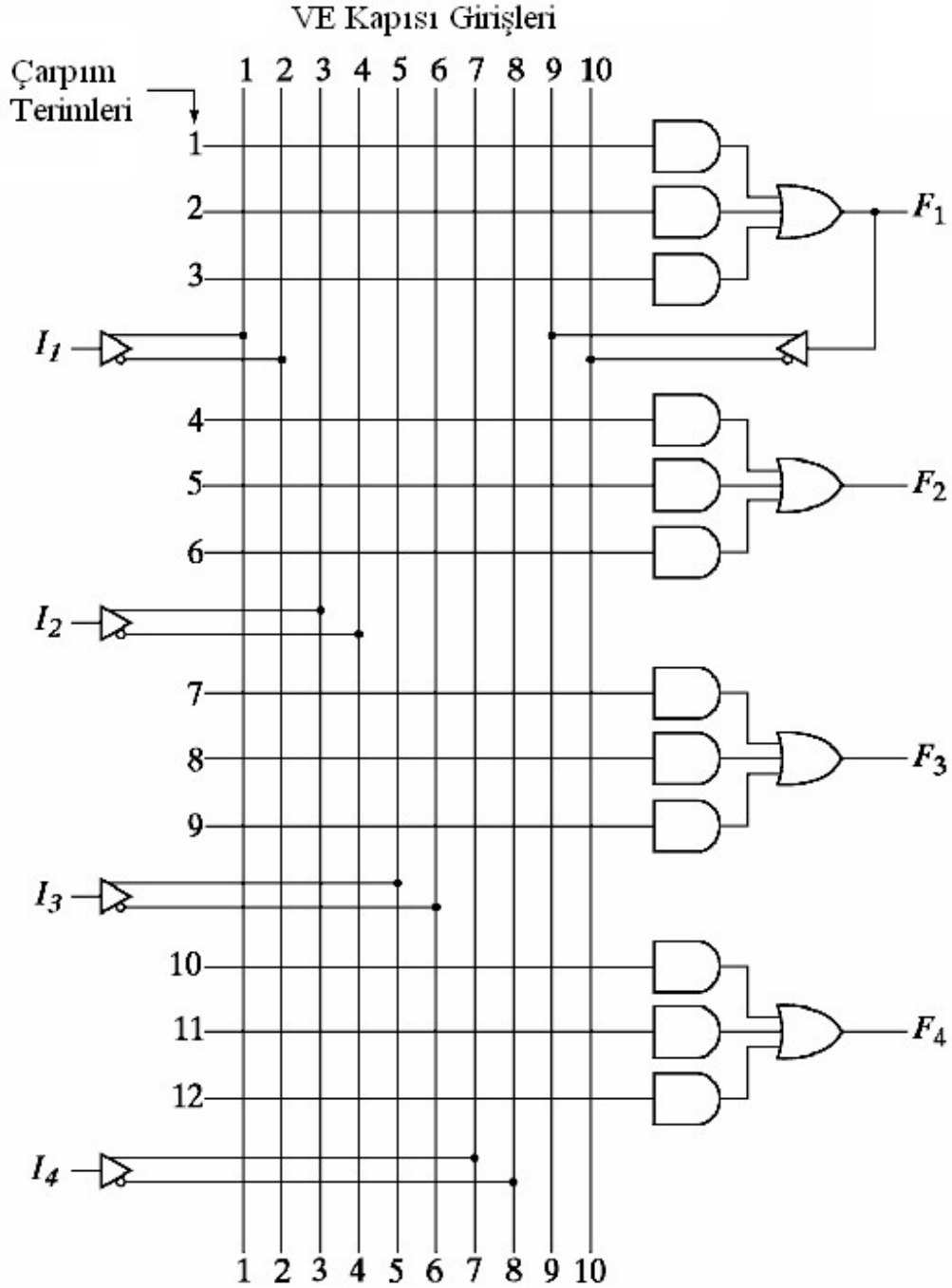


Alışlagelmiş gösterim



Dizi lojiği gösterimi

PAL'lar sabit VEYA kapıları ve programlanabilir VE dizileri ile programlanabilir bir cihazdır. Sadece VE kapıları programlanabilir olduklarından dolayı PAL'ın programlanması daha kolaydır fakat PLA'lar kadar esnek değildir. Aşağıda 4 girişli 4 çıkışlı tipik bir PAL dizi lojiği devresi görülmektedir. Her giriş bir tampon ve evirici devresine sahiptir. Bu iki kapı normal ve tümleyen çıkışları birleştirilmiş bir gösterim ile çizilmiştir. Devre toplam dört kattan oluşmaktadır. Dikey çizgiler VE kapılarının girişlerini bir başka deyişle tüm girişleri ve DEĞİL'lerini temsil etmektedir. Her bir VE-VEYA katı üç genişliktir. Her bir dikey ve yatay çizginin kesiştiği noktada bir sigorta var olduğu kabul edilir eğer sigorta atılmış ise + ile atılmamış ise kesişim noktasına X işareti konur. Durumu korunan tüm sigortalar X ile gösterilir. Yatay çizgiler ise VE kapılarının çok girişli olduğu anlamına gelir. Çıkışlarının birisi Tampon-Evirici devresi ve sigortalar ile VE kapılarının girişlerine bağlanmıştır. Normalde PAL devreleri şekilde gösterilenden daha fazla giriş ve çıkışa sahiptirler. Tipik bir PAL devresi her biri 8 girişe ve 8 çıkışa sahip 8 genişlikli VE-VEYA bölgesinden oluşurlar.



PAL kullanılarak yapılan tasarımlarda Boole fonksiyonları öncelikle sadeleştirilmelidir. PLA'daki gibi bir çarpım terimi iki ya da daha fazla VEYA kapısı tarafından paylaşılamaz. Bu nedenle her bir terim ortak terimler göz önüne alınmadan kendi başlarına sadeleştirilmelidir. Yapıdaki VE kapılarının sayısı sabittir eğer çok sayıda terim içeriyorsa çıkışlar ayrı ayrı katlarda hesaplanarak toplanmalıdır.

Bir PAL uygulaması olarak aşağıdaki Boole fonksiyonunu göz önüne alalım.

$$w(A, B, C, D) = \sum(2, 12, 13)$$

$$x(A, B, C, D) = \sum(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$y(A, B, C, D) = \sum(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$z(A, B, C, D) = \sum(1, 2, 8, 12, 13)$$

Bu dört fonksiyonun basitleştirilmesi sonucu aşağıdaki ifadeler elde edilir.

$$w = ABC' + A'B'CD'$$

$$x = A + BCD$$

$$y = A'B + CD + B'D'$$

$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$

$$= w + AC'D' + A'B'C'D$$

z terimine dikkat edilecek olursa, w terimini de içerdiği görülmektedir. Bu nedenle bu terimleri tekrar bir çarpım kapısı girişine vermektense w çıkış fonksiyonunu PAL'ın çıkışlarını tümleyen ve normal çıkış veren buffer ucu kullanılırsa tekrar hesaplamaya gerek kalmaz. Aşağıda PAL program tablosu görülmektedir. Eğer bir VE kapsının girişine gelen tüm sigortalar korunur, atırlmazsa değişkenin kendisi ve tümleyeni çarpılacağı için AA' her zaman 0 olacağından kapı çıkışı hep 0 da kalır ve fonksiyonun değeri değişmez.

Çarpım Terimleri	VE girişleri					Çıkışlar
	A	B	C	D	W	
1	1	1	0	-	-	$w = ABC' + A'B'CD'$
2	0	0	1	0	-	
3	-	-	-	-	-	
4	1	-	-	-	-	$x = A + BCD$
5	-	1	1	1	-	
6	-	-	-	-	-	
7	0	1	-	-	-	$y = A'B + CD + B'D'$
8	-	-	1	1	-	
9	-	0	-	0	-	
10	-	-	-	-	1	$z = w + AC'D' + A'B'C'D$
11	1	-	0	0	-	
12	0	0	0	1	-	

