

مدار منطقی

تبدیل بین مبناهای 2، 16 و 8

تبدیل بین مبناهای 2، 8 و 16 به سادگی دسته بندی رقم ها و جایگزینی آنها با رقمهای متناظر در سیستم دیگر می باشد.

1	2	7	5	4	3	Octal
1	0	1	0	1	1	Binary
A	F	6	3			Hexa

تبدیل اعداد در مبنای 10 به مبنای دیگر

• تبدیل از مبنای R به مبنای 10

$$A = a_{n-1} a_{n-2} a_{n-3} \dots a_0 . a_{-1} \dots a_{-m}$$

$$V(A) = \sum a_k R^k$$

$$(736.4)_8 = 7 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1}$$

$$= 7 \times 64 + 3 \times 8 + 6 \times 1 + 4/8 = (478.5)_{10}$$

$$(110110)_2 = \dots = (54)_{10}$$

$$(110.111)_2 = \dots = (6.785)_{10}$$

$$(F3)_{16} = \dots = (243)_{10}$$

$$(0.325)_6 = \dots = (0.578703703 \dots)_{10}$$

تبدیل دسیمال به مبنای R

• اعداد را به دو قسمت صحیح و کسری تقسیم می کنیم و هر قسمت را جداگانه تبدیل می کنیم

• تبدیل قسمت صحیح به مبنای R.

– با تقسیمات متوالی بر R و گرد آوری باقیمانده ها به عنوان رقم های مبنای R.

• تبدیل قسمت کسری به مبنای R.

– با ضرب متوالی در R و گرد آوری اعداد صحیح تولید شده به عنوان رقم های مبنای جدید R

تبدیل دسیمال به مبنای R

مثال: عدد 41.6875 در مبنای 10 می باشد. آنرا به مبنای 2 تبدیل کنید.

حل: ابتدا قسمت اعشار را از صحیح جدا کرده عمل تبدیل را برای هر یک جداگانه انجام می دهیم.

اعشار 0.6875 =

$$\begin{array}{r} 0.6875 \\ \times 2 \\ \hline 1.3750 \\ \times 2 \\ \hline 0.7500 \\ \times 2 \\ \hline 1.5000 \\ \times 2 \\ \hline 1.0000 \end{array}$$

صحیح 41 =

$$\begin{array}{r|l} 41 & \\ 20 & 1 \\ 10 & 0 \\ 5 & 0 \\ 2 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}$$

$$(41)_{10} = (101001)_2$$

$$(0.6875)_{10} = (0.1011)_2$$

$$(41.6875)_{10} = (101001.1011)_2$$

اصول جبر بول

- تعریف جبر بول: یک ساختار جبری تعریف شده روی اعضای یک مجموعه بسته B همراه با دو عملگر + و • که به ترتیب جمع و ضرب نامیده می شوند.
- اصل 1 (P1): عضو خنثی

$$a + 0 = a \quad (+) \quad a \bullet 1 = a \quad (\bullet)$$
 عضو همانی برای (b) عضو همانی برای (a)
- اصل 2 (P2): جابجایی

$$a + b = b + a \quad , \quad a \bullet b = b \bullet a$$
- اصل 3 (P3): شرکت پذیری

$$a \bullet (b \bullet c) = (a \bullet b) \bullet c \quad (a + (b + c) = (a + b) + c)$$
- اصل 4 (P4): توزیع پذیری

$$a \bullet (b + c) = a \bullet b + a \bullet c \quad (a + (b \bullet c) = (a + b) \bullet (a + c))$$
- اصل 5 (P5): مکمل

$$a + a' = 1 \quad (b) \quad a \bullet a' = 0$$

قضایای جبر بول

• قضیه 1 (T1):

$$(b) aa = a$$

$$(a) a + a = a$$

• قضیه 2 (T2):

$$(b) a0 = 0$$

$$(a) a + 1 = 1$$

• قضیه 3 (T3):

$$\overline{\overline{a}} = a$$

• قضیه 4 (T4):

$$(b) (ab)' = a' + b'$$

$$(a) (a+b)' = a' b'$$

• قضیه 5 (T5):

$$(b) a(a+b)' = a$$

$$(a) a + ab = a$$

بعضی از خواص صفرو یک در جبر بول:

Complement (مکمل)	AND (ضرب)	OR (جمع)
$0' = 1$	$a0 = 0$	$a + 0 = a$
$1' = 0$	$a1 = a$	$a + 1 = 1$

• مثال: با استفاده از قضایای و اصول جبر بول تساوی زیر را ثابت کنید:

$$B + AB' C' D = B + AC' D$$

حل

$$\begin{aligned}
 B + AB'C'D &= (B + ABC'D) + AB'C'D \quad \bullet \\
 &= B + (ABC'D + AB'C'D) \\
 &= B + AC'D (B+B') \\
 &= B + AC'D
 \end{aligned}$$

• **مثال:** رابطه زیر را ثابت کنید:

$$(X + Y)((X + Y)' + Z) = (X + Y)Z$$

حل:

$$\begin{aligned}
 (X + Y)((X + Y)' + Z) &= \\
 &= (X + Y)(X+Y)' + (X+Y)Z \\
 &= 0 + (X+Y)Z \\
 &= (X+Y)Z
 \end{aligned}$$

• **قضیه 6 (T6):**

$$(b) (a + b)(a + b') = a$$

$$(a) ab + ab' = a$$

مثال

ثابت کنید

$$(W' + X' + Y' + Z')(W' + X' + Y' + Z)(W' + X' + Y + Z')(W' + X' + Y + Z) \\ = (W' + X')$$

حل

$$(W' + X' + Y' + Z')(W' + X' + Y' + Z)(W' + X' + Y + Z')(W' + X' + Y + Z) \\ = (W' + X' + Y')(W' + X' + Y + Z')(W' + X' + Y + Z) \\ = (W' + X' + Y')(W' + X' + Y) \\ = (W' + X')$$

قضیه 7 (T7):

$$(b) (a + b)(a + b' + c) = (a + b)(a + c)$$

$$ab + ab'c = ab + ac \quad (a)$$

(b) مثال: عبارت بولی زیر را ساده کنید

$$wy' + wx'y + wxyz + wxz'$$

حل

$$\begin{aligned} [T7(a)] &= wy' + wx'y + wxy + wxz' & wy' + wx'y + wxyz + wxz' &- \\ [T7(a)] &= wy' + wy + wxz' \\ [T7(a)] &= w + wxz' \\ [T7(a)] &= w \end{aligned}$$

جدول درستی توابع بولي

- براي نشان دادن درستی قضایا و یا روابط بولي مي توان از جدول درستی استفاده کرد.
- جدول درستی براي دو عملگر AND و OR آورده شده است:

ab	$f(a,b)=a+b$	ab	$f(a,b)=ab$	a	$f(a)=a'$
00	0	00	0	0	1
01	1	01	0	1	0
10	1	10	0		
11	1	11	1		

مثال : جدول درستی را براي تابع $f(A,B,C) = AB + A'C + AC'$ بدست آورید.

ABC	$f(A,B,C)$	ABC	$f(A,B,C)$
000	0	FFF	F
001	1	FFT	T
010	0	FTF	F
011	1	FTT	T
100	1	TFF	T
101	0	TFT	F
110	1	TTF	T
111	1	TTT	T

نحوه نمایش توابع بولي

- توابع بولي را مي توان به چند طريق نمايش داد:
- بصورت ترکيبي نامرتب از عبارات و متغيرهاي بولي.
- بصورت مجموع جملات مي نيمم
- بصورت حاصلضرب جملات ماکزيمم
- *چند جمله اي مي نيمم:* يك عبارت حاصلضرب است که همه متغيرها يا مکمل آنها حتماً وجود دارد.

- براي حالي که سه متغير داشته باشيم، جدول جملات مي نيمم در زیر آورده شده است:

Minterm	Minterm Code	Minterm Number
$A'B'C'$	000	m_0
$A'B'C$	001	m_1
$A'BC'$	010	m_2
$A'BC$	011	m_3
$AB'C'$	100	m_4
$AB'C$	101	m_5
ABC'	110	m_6
ABC	111	m_7

• مثال:

جدول درستی تابع $f_1(A,B,C) = m_2 + m_3 + m_6 + m_7$ که بصورت
 $f_1(A,B,C) = \Sigma m(2,3,6,7)$ هم نشان داده می شود بصورت زیر است:

Row No. (i)	Inputs ABC	Outputs $f_1(A,B,C) = \Sigma m(2,3,6,7)$	Complement $f_1'(A,B,C) = \Sigma m(0,1,4,5)$
0	000	0	1 $\leftarrow m_0$
1	001	0	1 $\leftarrow m_1$
2	010	1 $\leftarrow m_2$	0
3	011	1 $\leftarrow m_3$	0
4	100	0	1 $\leftarrow m_4$
5	101	0	1 $\leftarrow m_5$
6	110	1 $\leftarrow m_6$	0
7	111	1 $\leftarrow m_7$	0

نکته: برای یافتن مکمل يك تابع که با جملات می نیمم نشان داده شده، کافی است آن دسته از جملات می نیمم را که در تابع بکار نرفته اند، استفاده کرد.

مثال: مکمل تابع زیر را بدست آورید:

$$f(A,B,Q,Z) = \Sigma m(0, 1, 6, 7)$$

$$= m_0 + m_1 + m_6 + m_7$$

$$= A'B'Q'Z' + A'B'Q'Z + A'BQZ' + A'BQZ$$

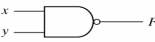
برای بدست آوردن مکمل تابع به اینصورت عمل می کنیم:

$$= m_2 + m_3 + m_4 + m_5 + m_8 + m_9 + m_{10} + m_{11} + m_{12} \quad f'(A,B,Q,Z)$$

$$+ m_{13} + m_{14} + m_{15}$$

$$= \Sigma m(2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15)$$

گیت های منطقی

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

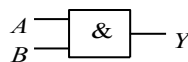
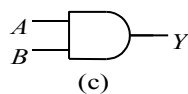
گیت AND

a	b	$f_{AND}(a, b) = ab$
0	0	0
0	1	0
1	0	0
1	1	1

(a)

A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

(b)



(a) جدول درستی يك AND

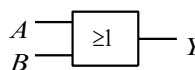
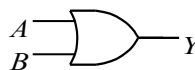
(a) استفاده از سطح ولتاژهای H و L برای نشان دادن صفرو يك منطقی

(b) سمبل استاندارد گیت

(d) سمبل گیت در استاندارد IEEE

گیت OR

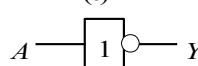
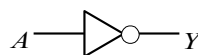
a	b	$f_{OR}(a, b) = a + b$	A	B	Y	
0	0	0	L	L	L	(c)
0	1	1	L	H	H	
1	0	1	H	L	H	
1	1	1	H	H	H	(d)



- (a) جدول درستی يك OR
- (b) استفاده از سطح ولتاژهاي H و L براي نشان دادن صفرو يك منطقي
- (c) سمبل استاندارد گیت
- (d) سمبل گیت در استاندارد IEEE

گیت NOT

a	$f_{NOT}(a) = \bar{a}$	A	Y	
0	1	L	H	(c)
1	0	H	L	(d)



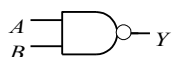
- (a) جدول درستی يك NOT
- (b) استفاده از سطح ولتاژهاي H و L براي نشان دادن صفرو يك منطقي
- (c) سمبل استاندارد گیت
- (d) سمبل گیت در استاندارد IEEE

گیت NAND

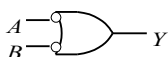
a	b	$f_{NAND}(a, b) = \overline{ab}$	A	B	Y
0	0	1	L	L	H
0	1	1	L	H	H
1	0	1	H	L	H
1	1	0	H	H	L

(a)

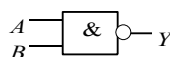
(b)



(c)



(d)



(e)

(a) جدول درستی يك NAND

(b) استفاده از سطح ولتاژهاي H و L براي نشان دادن صفر و يك منطقي

(c) سمبل استاندارد گیت

(d) گیت OR معادل با ورودیهاي مناسب

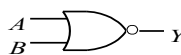
(e) سمبل گیت در استاندارد IEEE

گیت NOR

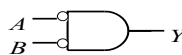
a	b	$f_{NOR}(a, b) = \overline{a + b}$	A	B	Y
0	0	1	L	L	H
0	1	0	L	H	L
1	0	0	H	L	L
1	1	0	H	H	L

(a)

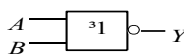
(b)



(c)



(d)



(e)

(a) جدول درستی يك NOR

(b) استفاده از سطح ولتاژهاي H و L براي نشان دادن صفر و يك منطقي

(c) سمبل استاندارد گیت

(d) گیت AND معادل با ورودیهاي مناسب

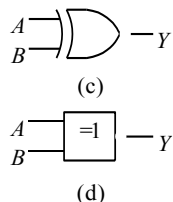
(e) سمبل گیت در استاندارد IEEE

گیت XOR

a	b	$f_{XNOR}(a, b) = a \odot b$	A	B	Y
0	0	1	L	L	H
0	1	0	L	H	L
1	0	0	H	L	L
1	1	1	H	H	H

(a)

(b)



(a) جدول درستی يك XOR

(b) استفاده از سطح ولتاژهاي H و L براي نشان دادن صفر و يك منطقي

$$a \oplus b = \bar{a}b + a\bar{b}$$

$$= \bar{a}a + \bar{a}b + a\bar{b} + b\bar{b}$$

$$= \bar{a}(a+b) + \bar{b}(a+b)$$

$$= (\bar{a} + \bar{b})(a+b)$$

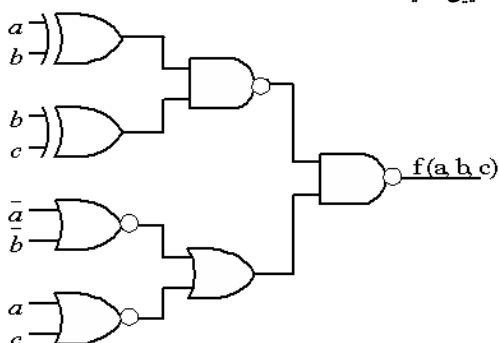
(c) سمبل استاندارد گیت

(d) سمبل گیت در استاندارد IEEE

طراحی و تحلیل مدارهاي ترکیبي

- تحلیل مدار: با استفاده از روابط جبر بول و شکل مدار و روابط گیت‌ها، رابطه‌اي بین ورودی‌ها و خروجی مدار بدست می‌آید.

- مثال: تابع بولي مدار زیر را تعیین کنید:



حل

- با استفاده از قوانین جبر بول و روابط گیتها داریم:

$$\begin{aligned}
 f(a,b,c) &= \overline{(a \oplus b)(b \oplus c) \cdot (\bar{a} + \bar{b} + a + c)} \\
 &= \overline{(a \oplus b)(b \oplus c) + \bar{a} + \bar{b} + a + c} \\
 &= (a \oplus b)(b \oplus c) + (\bar{a} + \bar{b})(a + c) \\
 &= (a\bar{b} + \bar{a}b)(b\bar{c} + \bar{b}c) + (\bar{a} + \bar{b})(a + c) \\
 &= a\bar{b}b\bar{c} + a\bar{b}\bar{b}c + \bar{a}bb\bar{c} + \bar{a}\bar{b}\bar{b}c + \bar{a}a + \bar{a}c + a\bar{b} + \bar{b}c \\
 &= a\bar{b}c + \bar{a}b\bar{c} + \bar{a}c + a\bar{b} + \bar{b}c \\
 &= \bar{a}b\bar{c} + \bar{a}c + a\bar{b} + \bar{b}c \\
 &= \bar{a}b\bar{c} + \bar{a}c + a\bar{b} \\
 &= \bar{a}b + \bar{a}c + a\bar{b} \\
 &= \bar{a}c + a \oplus b
 \end{aligned}$$

تحليل مدار با استفاده از جدول درستي

- میتوان با مقارنه‌ی ورودی‌های مدار خروجی را بدست آورد و براساس مقادیر خروجی، تابع بولی مدار را بدست آورد.

$$f(a,b,c) = \bar{a}c + a \oplus b$$

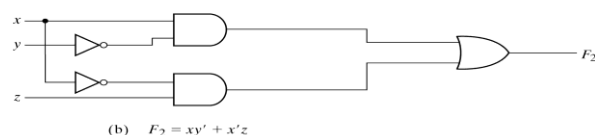
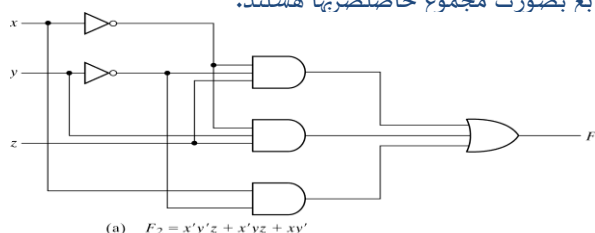
- برای مثال قبلی:

abc	$\bar{a}c$	$a \oplus b$	$f(a,b,c)$
000	0	0	0
001	1	0	1
010	0	1	1
011	1	1	1
100	0	1	1
101	0	1	1
110	0	0	0
111	0	0	0

ساخت مدارهاي ترکیبي

يکي از ساده‌ترین روشها براي طراحی و ساخت مدار ترکیبي، تحقق AND-OR-NOT می‌باشد. در این روش ابتدا تابع را بصورت مجموع حاصلضربها نوشته و سپس با استفاده از سه نوع گیت AND و OR و NOT آنرا پیاده‌سازی می‌کنیم. گاهی اوقات نیز تابع F' را با استفاده از AND و OR پیاده کرده و سپس با استفاده از گیت NOT خروجی را معکوس کرده و تابع F را می‌سازیم. معمولاً ابتدا از گیتهاي AND و سپس از OR استفاده می‌شود.

- مثال: دو تابع بولي متفاوت توسط گیتهاي AND و OR و NOT ساخته شده‌اند. توجه کنید که توابع بصورت مجموع حاصلضربها هستند.



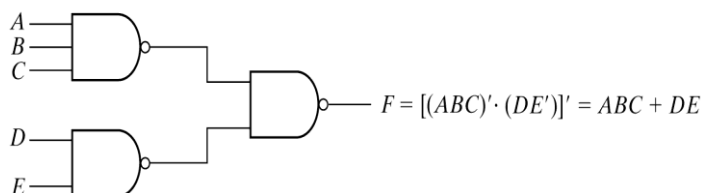
مثال

- تابع زیر را توسط گیتهای NAND پیاده سازید:

$$f(A,B,C,D,E)=ABC+DE$$

حل:

- ابتدا دو گیت NAND در تراز اول قرار می‌دهیم. برای هر جمله یک گیت انتخاب می‌شود.



مثال

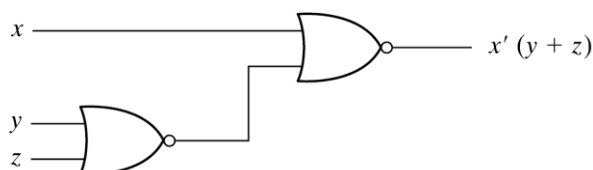
- تابع زیر را توسط گیتهای NOR پیاده سازید:

$$f(x,y,z)=x'y+x'z$$

حل:

$$f(x,y,z)=x'y+x'z=x'(y+z)$$

- می‌توان دو گیت NOR در تراز اول قرار داد ولی از آنجا که یکی از جملات تنها یک حرف است، تنها یک گیت در تراز اول بکار برده شد.



روشهای ساده سازی توابع

- استفاده از قضایا و اصول جبر بول
- استفاده از جدول کارنو
- هر تابع بولی را می توان بصورت مجموعی از جملات می نیمم نشان داد.
- جدول کارنو از مربعیایی تشکیل شده است که هریک نشان دهنده یکی از جملات می نیمم است.
- برای توابع دو متغیری، جدول کارنو دارای چهار خانه است.
- بطور کلی برای توابع با n متغیر، جدول کارنو دارای 2^n خانه است.

جدول کارنو برای دو متغیر

- نحوه قرار گرفتن جملات می نیمم در جدول کارنو
- هر خانه جدول با خانه مجاور خود تنها در یک رقم تفاوت دارد.

m_0	m_1
m_2	m_3

(a)

		y	
		0	1
x	0	$x'y'$	$x'y$
	1	xy'	xy

(b)

مثال

- توابع زیر را توسط جدول کارنو نمایش دهید.

$$f_1(x, y) = xy' + xy + x'y$$

$$f_2(x, y) = x + y$$

حل: همانگونه که دیده می‌شود، هر دو تابع به یک صورت در جدول ظاهر شده‌اند. همچنین با استفاده از قوانین جبر بول می‌توان f_1 را ساده کرد و به f_2 رسید.

		y	
		0	1
x	0		1
	1	1	1
		$x + y$	

جدول کارنو برای سه متغیر

- برای سه متغیر، هشت عدد جمله می‌نیمم وجود دارد و بنابراین جدول کارنو باید هشت خانه داشته باشد.

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

		y			
		xz	00	01	11
x	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	$xy'z$	xyz	xyz'
		z			

(b)

(a)

(b)

کاربرد جدول کارنو در ساده سازی توابع

- یکی از بهترین ابزارها برای ساده سازی توابع، جدول کارنو است.
- خانه هایی از جدول که مقدراتابع در آنها برابر با یک می باشد را مشخص می کنیم.
- خانه هایی که دارای یک هستند و مجاور یکدیگر هم هستند بصورت دودو انتخاب کرده و ساده می کنیم.
- خانه های لبه بالا و پایین هم مجاور یکدیگر هستند، اگرچه در کنار یکدیگر قرار ندارند.

مثال

- تابع زیر را ساده کنید:

$$f(x, y, z) = \sum(2, 3, 4, 5)$$

حل: با استفاده از جملات می نیمم، جدول کارنو را پر می کنیم:

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

- حال خانه‌های مجاور را همانگونه که در شکل نشان داده شده است انتخاب کرده و ساده می‌کنیم:

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

$$F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$$

مثال

- تابع زیر را ساده کنید:

$$F(x, y, z) = \Sigma(3, 4, 6, 7)$$

حل: با استفاده از جملات می‌نیم، جدول کارنو را پر می‌کنیم:

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

$$F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$$

مثال

- تابع زیر را ساده کنید:

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$$

حل: با استفاده از جملات می‌نیمم، جدول کارنورا پر می‌کنیم. در این مثال ستونهای کناری نیز مجاور یکدیگر هستند:

		y			
		yz		11	10
x	0	1			1
	1	1	1		1

z

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$$

مثال

- تابع زیر را ساده کنید:

$$F(x, y, z) = A'C + A'B + AB'C + BC$$

حل: با استفاده از جملات می‌نیمم، جدول کارنورا پر می‌کنیم. در این مثال چهار خانه مجاور انتخاب و ساده شده‌اند. هر چه تعداد بیشتری خانه انتخاب شود بهتر است.

		B			
		BC		11	10
A	0		1	1	1
	1		1	1	

C

$$F(x, y, z) = A'C + A'B + AB'C + BC = C + A'B$$

جدول کارنو برای چهار متغیر

- برای چهار متغیر، شانزده عدد جمله می‌نیمم وجود دارد و بنابراین جدول کارنو باید شانزده خانه داشته باشد.

				yz		y			
				0 0		0 1	11 10		
				wx	00	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'
					01	w'xy'z'	w'xy'z	w'xyz	w'xyz'
					11	wxy'z'	wxy'z	wxyz	wxyz'
					10	wx'y'z'	wx'y'z	wx'yz	wx'yz'
						z			
(a)				(b)					

مثال

- تابع زیر را ساده کنید:

$$F(w, x, y, z) = \sum (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

حل: با استفاده از جملات می‌نیمم، جدول کارنو را پر می‌کنیم. در این مثال هشت خانه مجاور انتخاب و ساده شده‌اند. هر چه تعداد بیشتری خانه انتخاب شود تابع بیشتر ساده می‌شود.

		yz		y	
		0 0	0 1	1 1	1 0
w	x				
0 0		1	1		1
0 1		1	1		1
1 1		1	1		
1 0		1	1		

$$\begin{aligned}
 F(w, x, y, z) &= \\
 &= \Sigma (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14) \\
 &= y' + w'z' + xz'
 \end{aligned}$$

مثال طراحی نیم جمع کننده

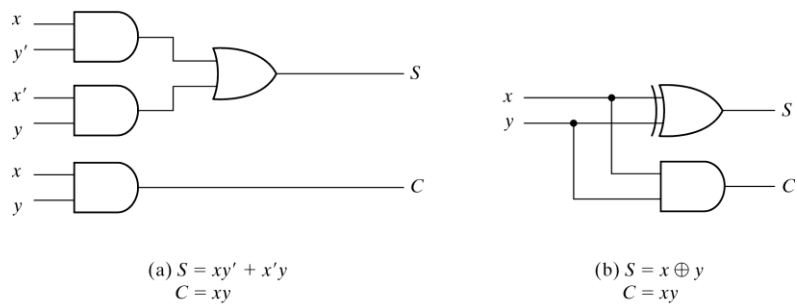
- نیم جمع کننده مداری است که دو بیت ورودی و دو خروجی دارد.
- جدول درستی این مدار بصورت زیر است.
- توابع بولی نیز با استفاده از جدول درستی بدست می آیند.

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = x'y + xy'$$

$$C = xy$$

دو صورت متفاوت پیاده‌سازی مدار نیم جمع کننده



طراحی تمام جمع کننده

- تمام جمع کننده مداری است که عمل جمع را بین دو بیت انجام می‌دهد با این احتمال که ممکن است یک بیت نقلی از مرحله قبل باشد. بنابراین مدار سه ورودی و دو خروجی دارد.
- جدول درستی و توابع بولی آن بصورت زیر است:

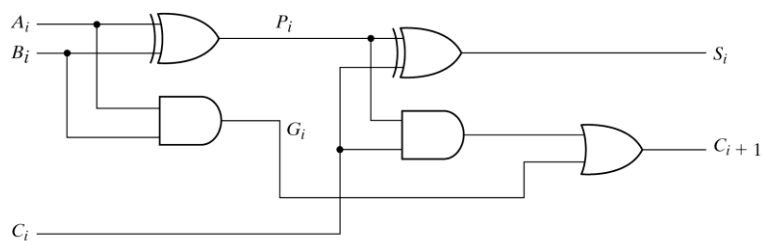
طراحی تمام جمع کننده

x	y	z	C	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

$$C = x'yz + x'yz' + xy'z' + xyz$$

$$S = xy + xz + yz$$

نمودار منطقی تمام جمع کننده



طراحی تمام تفریق کننده

- تمام تفریق کننده مداری است که عمل تفریق را بین دو بیت انجام می دهد با این احتمال که ممکن است یک 1 از مرحله پایین تر قرض گرفته شده باشد. بنابراین مدار سه ورودی و دو خروجی دارد.
- جدول درستی و توابع بولی آن بصورت زیر است:

طراحی نیم تفریق کننده

- مدار دارای دو ورودی و دو خروجی است.
- جدول درستی آن بصورت زیر است.
- توابع بولی هر یک از خروجی ها نیز از روی جدول بدست می آیند.
- نکته:** توابع بدست آمده مشابه حالت نیم جمع کننده است با این تفاوت که در تابع B متغیر x' به کار رفته است ولی در تابع C متغیر X استفاده می شود.

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$D = x'y + xy'$$

$$B = x'y$$

طراحی تمام تفریق کننده

- تمام تفریق کننده مداری است که عمل تفریق را بین دو بیت انجام می دهد با این احتمال که ممکن است یک 1 از مرحله پایین تر قرض گرفته شده باشد. بنابراین مدار سه ورودی و دو خروجی دارد.
- جدول درستی و توابع بولی آن بصورت زیر است:

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

روابط بدست آمده برای D و B مشابه همان روابط S و C هستند که برای تمام جمع کننده هم بدست آمده است. تفاوت این روابط تنها بین C و B است. در یکی x و در دیگری x' بکار می رود.

$$D = x'yz + x'yz' + xy'z' + xyz$$

$$B = x'y + x'z + yz$$

طراحی مدار مقایسه‌کننده

- این مدار دو عدد n بیتی را بعنوان ورودی پذیرفته و آنها را با یکدیگر مقایسه کرده نتیجه را از طریق سه بیت اعلام می‌کند.
- هر کدام از خروجی‌ها به ترتیب برای نشان دادن بزرگی، تساوی و کوچکی عدد اول نسبت به دوم است.
-

55

نحوه مقایسه دو عدد

- فرض کنید دو عدد چهار بیت و بصورت زیر باشند:

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$
- برای مقایسه آنها از رقم با ارزش‌تر شروع می‌کنیم. تابع B'_i تابع $x_i = A_iB_i + A'_i$ را تابع تساوی می‌نامیم. هرگاه x_i برابر با 1 شود یعنی دو رقم با هم مساوی هستند. اگر $x_i = 0$ به این معنی است که دو رقم مساوی نیستند.
- شرط تساوی دو عدد چهار بیت A و B اینست که:

$$(A=B) = x_3x_2x_1x_0$$

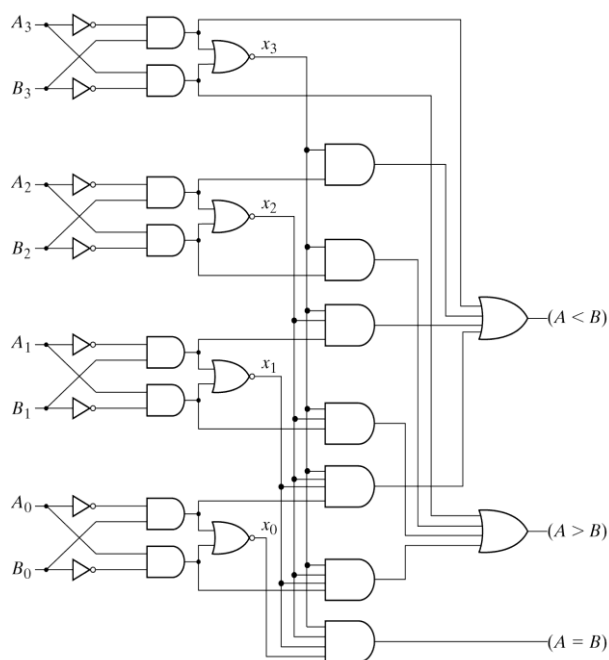
طراحی مدار مقایسه کننده

- برای حالات بزرگتر و کوچکتر توابع زیر را تعریف می کنیم:

$$(A > B) = A_3 B'_3 + x_3 A_2 B'_2 + x_3 x_2 A_1 B'_1 + x_3 x_2 x_1 A_0 B'_0$$

$$(A < B) = A'_3 B_3 + x_3 A'_2 B_2 + x_3 x_2 A'_1 B_1 + x_3 x_2 x_1 A'_0 B_0$$

حال با توجه به این تعاریف، مداری طراحی می شود که عمل مقایسه دو عدد چهار بیتی را انجام دهد.



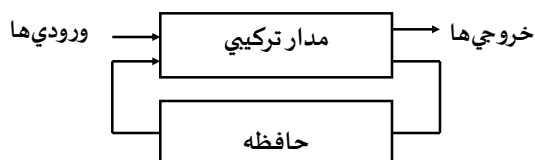
مدار مقایسه کننده

مدارهای ترتیبی



- تا کنون تمام مداراتی که دیدیم مدارات منطقی ترکیبی بودند که هیچ عنصر حافظه‌ای در آنها وجود نداشت.
- در مدارات منطقی ترکیبی یک ورودی یکسان منجر به یک خروجی می‌شود.
- این وضعیت مشابه عملکرد توابع می‌باشد. که در آن هر ورودی تنها یک خروجی تولید می‌کند.

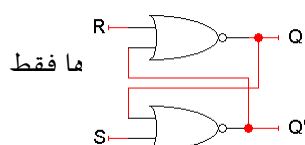
مدارهای منطقی ترتیبی



- در مقابل مدارهای منطقی ترکیبی که خروجی تنها به ورودی بستگی داشت در مدارهای منطقی ترتیبی خروجی نه تنها به ورودی بلکه به حالت فعلی مدار و یا مقدار فعلی حافظه نیز بستگی دارد.
- این موضوع درک عملکرد مدار را دشوار می‌کند چرا که اعمال یک خروجی می‌تواند (بسته به مقدار ذخیره شده در حافظه) منجر به تولید خروجی‌های متفاوت گردد.
- محتویات حافظه نیز می‌تواند بسته به مقدار ورودی تغییر کند.
- برای تجزیه و تحلیل و طراحی این مدارات به روش‌های دیگری علاوه بر روش‌های مورد استفاده در مدارات ترکیبی نیازمندیم.

لچ RS

- در مدار زیر به جاي استفاده از گیت هاي NOT از گیت هاي NOR استفاده مي كنيم. به مدار زیر يك لچ RS مي گويند .
- با استفاده از ورودي هاي R و S مي توان خروجي هاي Q و Q' را كنترل كرد.

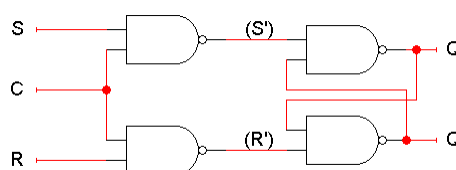


- در اين مدار خروجي هاي Q و Q' به ورودي هاي مدا خروجي نيستند بلكه ورودي نيز هستند.

يك RS ا صلاح شده

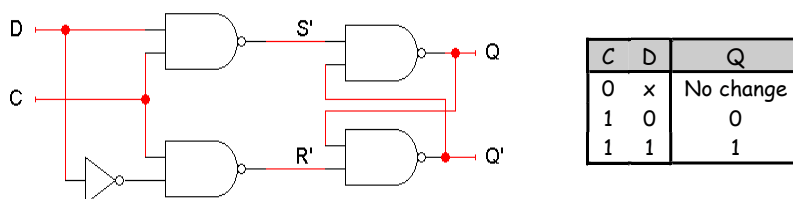
- در اينجا يك لچ با يك ورودي كنترلي C آورده شده است.
- در صورتي كه مقدار ورودي كنترلي 1 باشد ورودي هاي RS مي توانند همانند گذشته عمل كنند و حالت مدار را با توجه به مقادير خود تغيير دهند. در صورتي كه مقدار $C=0$ باشد ورودي ها نهي توانند حالت مدار را تغيير دهند.

C	S	R	S'	R'	Q
0	x	x	1	1	No change
1	0	0	1	1	No change
1	0	1	1	0	0 (reset)
1	1	0	0	1	1 (set)
1	1	1	0	0	Avoid!



لچ D

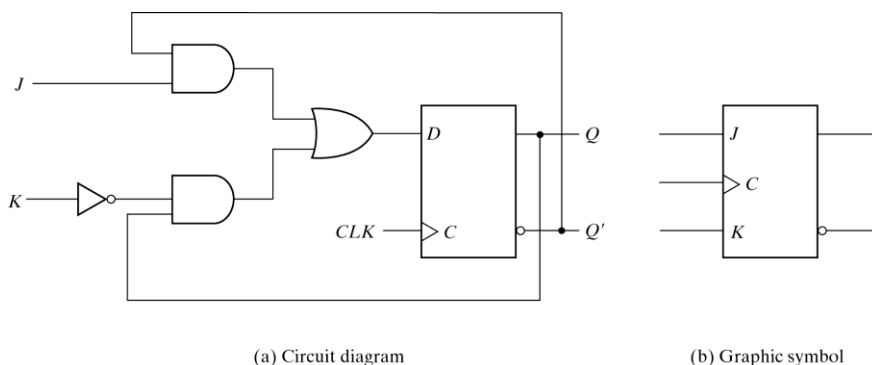
- لچ D با استفاده از يك لچ RS ساخته مي شود. وظیفه گیتهاي اضافي توليد سيگنالهاي مناسب براي ورودي هاي S و R بر اساس ورودي هاي D (data) و C (control) مي باشد.
- هنگامی که $C=0$ باشد S و R هر دو يك مي شوند و بنابر اين مقدار Q تغيير نمي کند



فلیپ فلاپ JK

- بسیار شبیه به فلیپ فلاپ RS می باشد.
- حالت نامعین فلیپ فلاپ RS رفع اشکال شده است.
- ورودی J بري ست کردن و ورودی K براي پاک کردن فلیپ فلاپ بکار می رود.
- اگر هر دو ورودی J و K با هم يك شوند، خروجي به حالت مکمل خود می رود.
- اگر هر دو ورودی J و K برابر با صفر باشند، خروجي بدون تغییر می ماند.

فلیپ فلاپ نوع JK



فلیپ فلاپ T

- يك زیرگروه تك ورودی از فلیپ فلاپ JK است.
- اگر هر دو ورودی J و K فلیپ فلاپ JK بهم وصل شوند، فلیپ فلاپ T ساخته می‌شود.
- اگر ورودی T برابر با يك باشد، در هر پالس ساعت خروجی یکبار مکمل می‌شود.
- اگر ورودی T فلیپ فلاپ برابر با صفر باشد، خروجی فلیپ فلاپ ثابت و بدون تغییر می‌ماند.

ساختار فلیپ فلاپ T

- دو صورت ساخت فلیپ فلاپ T در زیر آورده شده است:

