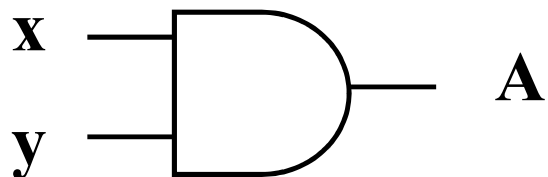




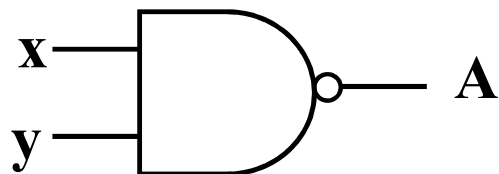
گیت ها (دریچه ها)

AND



x	y	$A = x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

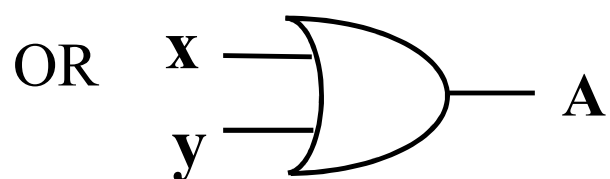
Nand



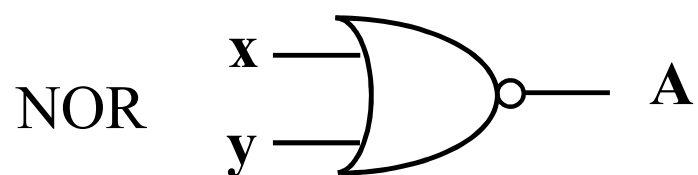
x	y	$A = (x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0



گیت ها (دریچه ها)



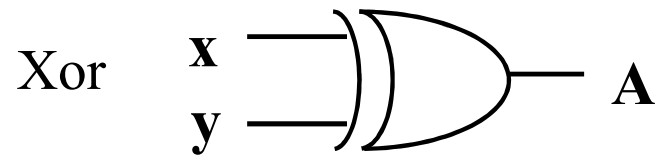
x	y	$A = x + y$
0	0	0
0	1	1
1	0	1
1	1	1



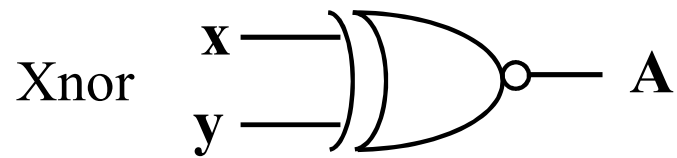
x	y	$A = (x + y)'$
0	0	1
0	1	0
1	0	0
1	1	0



گیت ها



x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

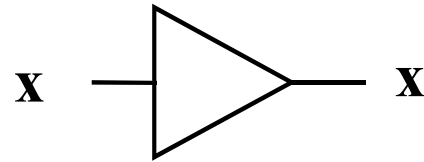


x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

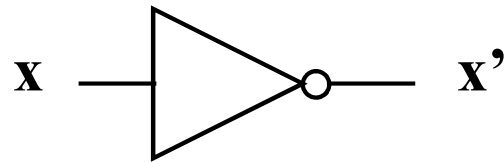


گیت ها

تقویت کننده
Buffer



متعم NOT



x	x'
0	1
1	0

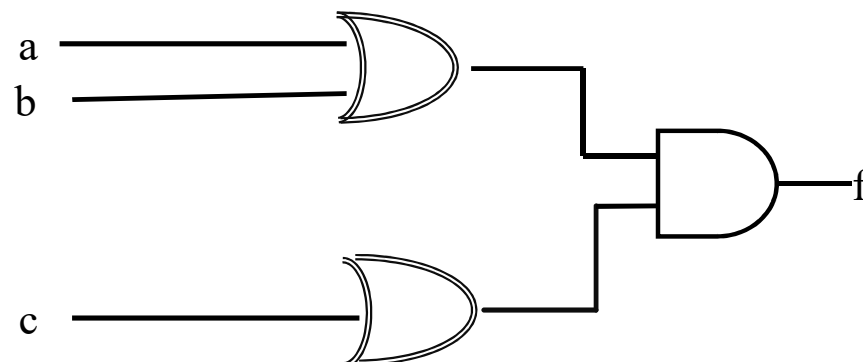
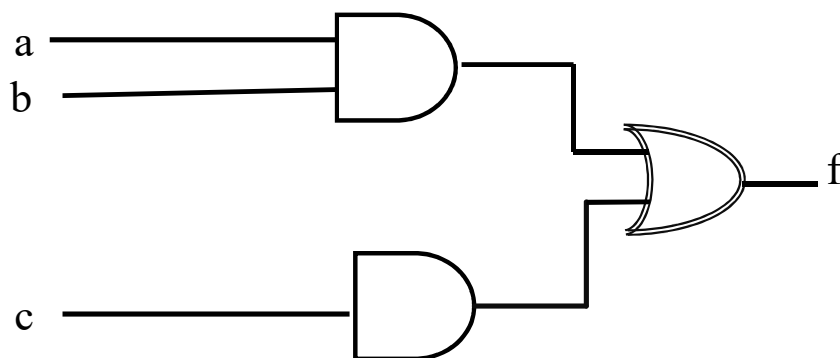


روش های پیاده سازی توابع منطقی در سطح گیت (implementaion)

نمایش های “AND – OR” و “OR-AND”

نمایش به صورت AND – OR ← کافی است تابع را به صورت SOP نمایش دهیم. در این نمایش سطح اول را گیت های AND و سطح دوم را گیت های OR تشکیل می دهند.

نمایش به صورت OR – AND ← کافی است تابع را به صورت POS نمایش دهیم. در این نمایش سطح اول را گیت های OR و سطح دوم را گیت های AND تشکیل می دهند.



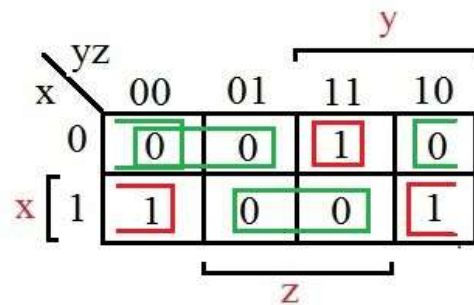


نمایش های AND – OR و OR – AND

مثال: یک تابع منطقی با سه ورودی و دو خروجی در نظر بگیرید. جدول درستی آن در زیر آمده است.

این مدار را به صورت “AND – OR” و “OR – AND” پیاده سازی کنید.

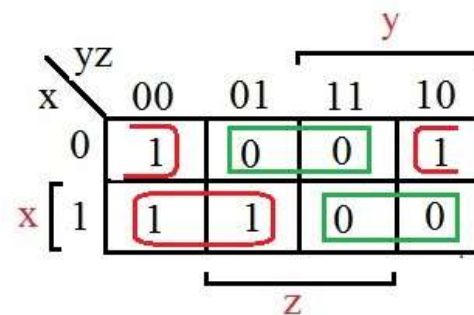
x	y	z	F1	F2
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0



$$F1 = x\bar{z} + \bar{x}.yz \text{ (SOP)}$$

$$\overline{F1} = xz + \bar{x}.\bar{y} + \bar{x}.\bar{z}$$

$$F1 = (\bar{x} + \bar{z}).(x + y).(x + z) \text{ (POS)}$$



$$F2 = x\bar{y} + \bar{x}.\bar{z} \text{ (SOP)}$$

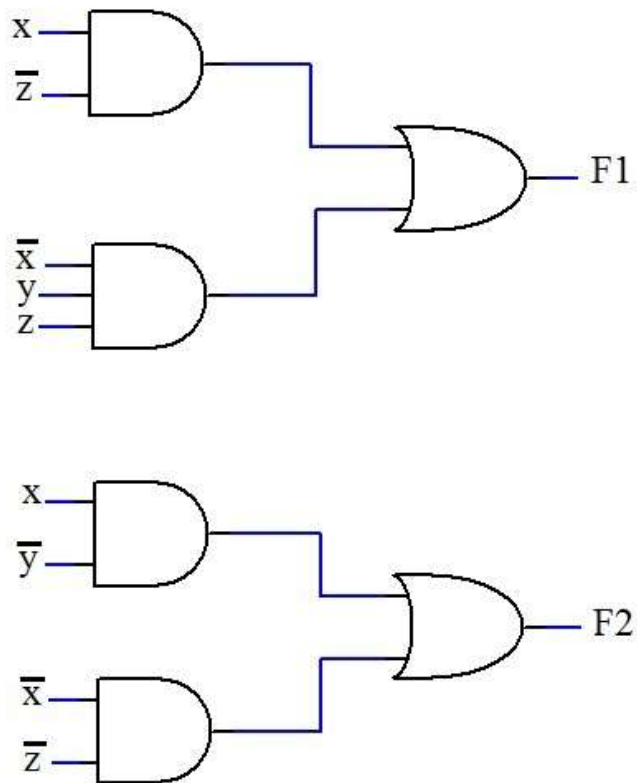
$$\overline{F2} = \bar{x}.z + xy$$

$$F2 = (x + \bar{z}).(\bar{x} + \bar{y}) \text{ (POS)}$$

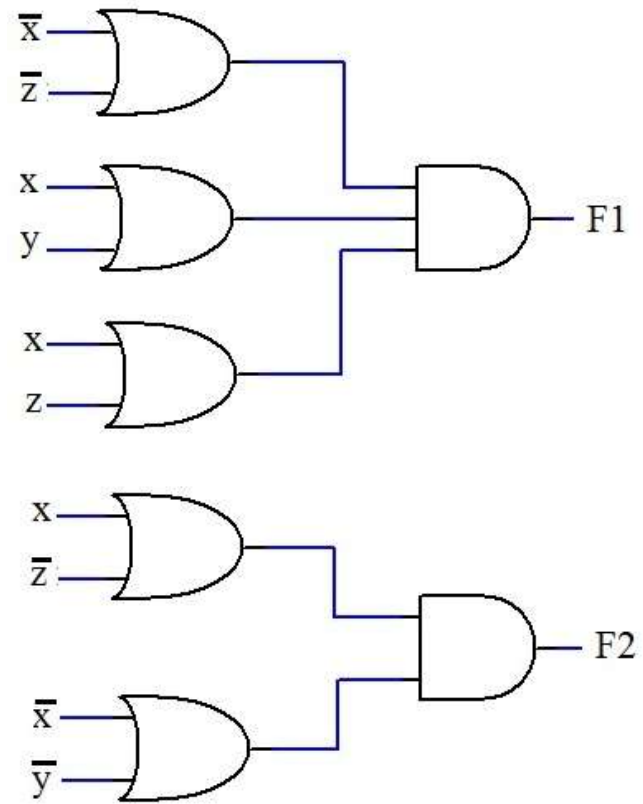


نمایش های AND - OR و OR - AND

نمایش AND - OR



نمایش OR-AND





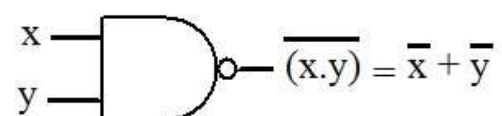
- همون گیت‌های پایه ای مثل **AND** و **OR** و **NOT** اگر بخواهیم در ۳ طبقه پیاده سازی کنیم
 - فقط با گیت‌های **NAND**
 - فقط با گیت‌های **NOR**
 - با استفاده از گیت‌های **open collector**
- این نوع پیاده سازی را **wired -And** هم می‌گوییم چون خروجی ۲ تا گیت را به هم می‌بندیم
- در نتیجه ← گیت **And** را داریم

روش‌های پیاده سازی :

- با استفاده از گیت‌های **NOR** از نوع **Emitter coupled logic**
- این نوع پیاده سازی را **wired -Or** هم می‌گوییم چون خروجی ۲ تا گیت را به هم می‌بندیم
- در نتیجه ← گیت **Or** را داریم
- روش **And-Or- invert(AOI)**
 - روش **Or-And- invert(OAI)**



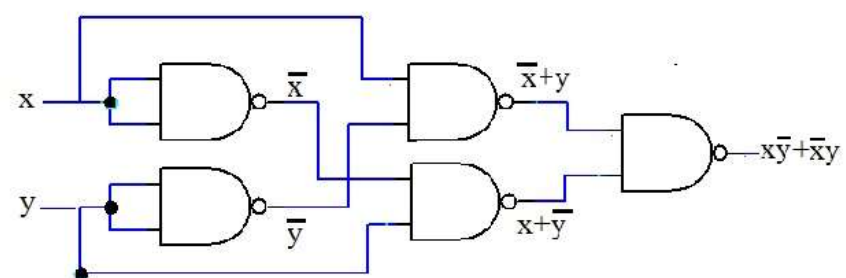
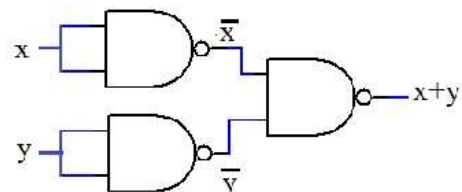
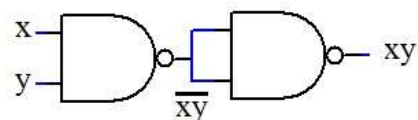
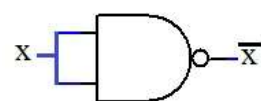
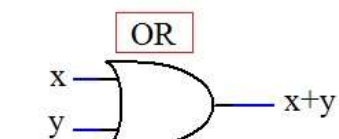
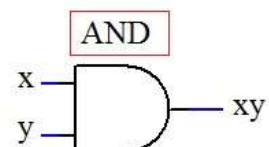
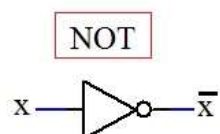
گیت NAND



AND - Invert



Invert - OR





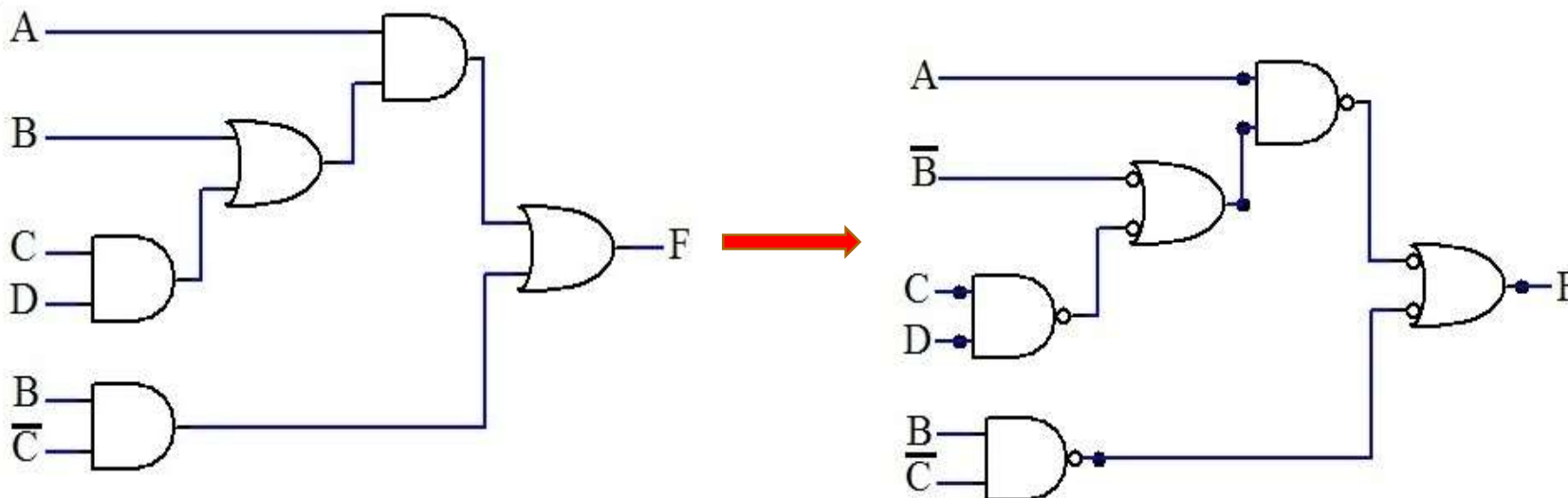
پیاده سازی مدارات منطقی با گیت NAND

ابتدا تابع را با استفاده از گیت های AND و OR پیاده سازی می کنیم.

همه ی گیت های AND (با سمبل AND – Invert) و OR (با سمبل OR – Invert) به گیت NAND تبدیل می شوند.

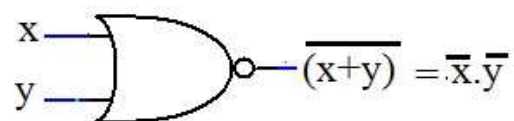
همه ی حباب های مدار چک می شوند. هرجایی که در مسیر جبران نشده اند، یک وارونگر اضافه می شود یا لیترال ورودی متمم می گردد.

مثال: تابع $F = A.(B+CD) + B\bar{C}$ را با گیت های NAND پیاده سازی کنید.

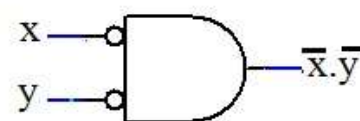




گیت NOR

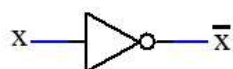


OR - Invert

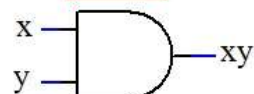


Invert - AND

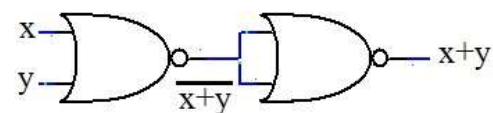
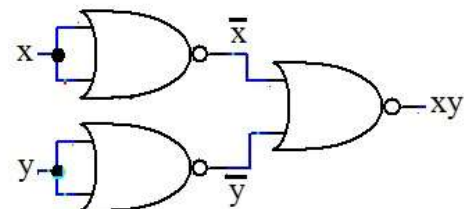
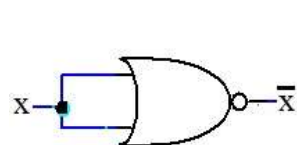
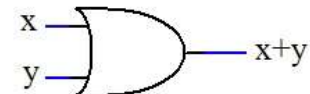
NOT



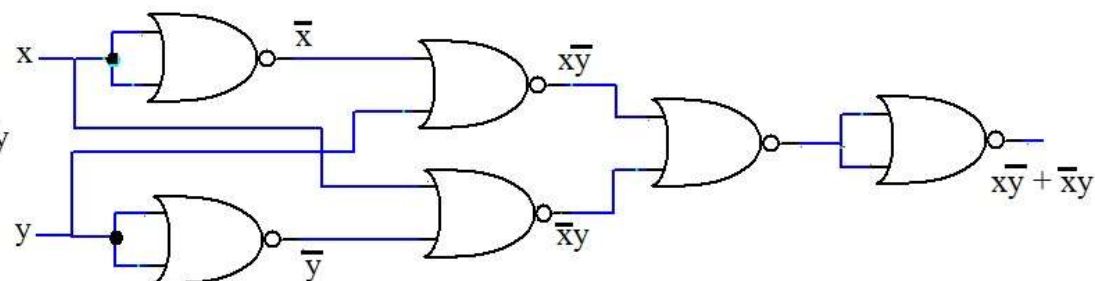
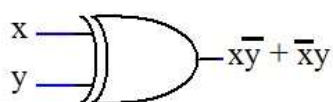
AND



OR



XOR



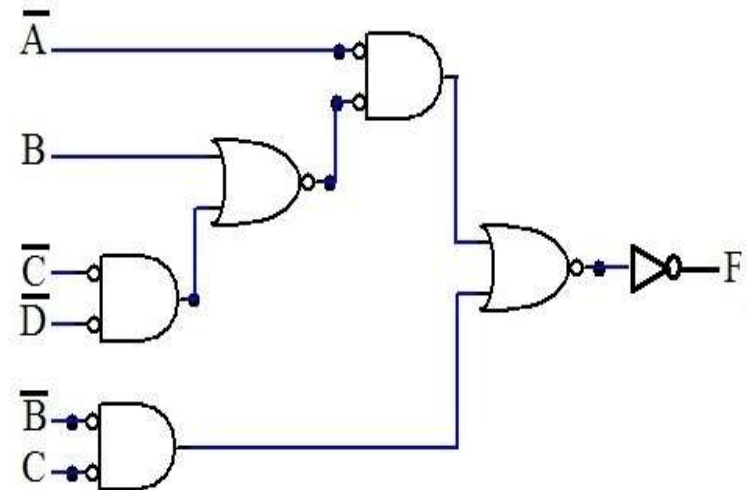
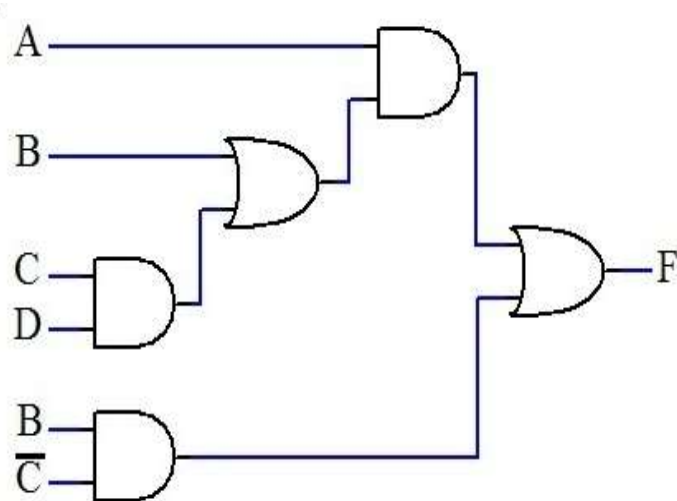


پیاده سازی مدارات منطقی با گیت NOR

ابتدا تابع را با استفاده از گیت های AND و OR پیاده سازی می کنیم.
همه ی گیت های AND (با سمبل Invert - AND) و OR (با سمبل OR - Invert) به گیت NOR تبدیل می شوند.

همه ی حباب های مدار چک می شوند. هرجایی که در مسیر جبران نشده اند، یک وارونگر اضافه می شود یا لیترال ورودی متمم می گردد.

مثال: تابع $F = A.(B+CD) + B\bar{C}$ را با گیت های NOR پیاده سازی کنید.





منطق Wired (Wired-Logic)

در برخی گیت های NAND و NOR (نه همگی آنها) می توان خروجی دو گیت را بهم متصل کرد تا تابع خاصی را پیاده سازی کرد. این نوع اتصال را Wired گویند.

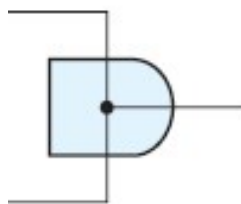
۲ نوع گیت Wired وجود دارد:

Wired – AND

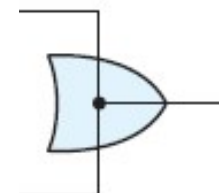
Wired – OR

گیت های Wired-AND و Wired-OR گیت فیزیکی نیستند، بلکه تنها یک سمبل هستند که معرف نحوه ی عملکرد اتصال Wired مربوطه می باشند.

Wired – AND



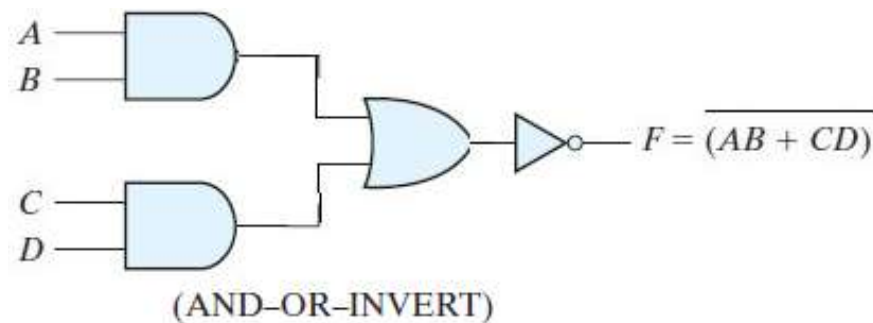
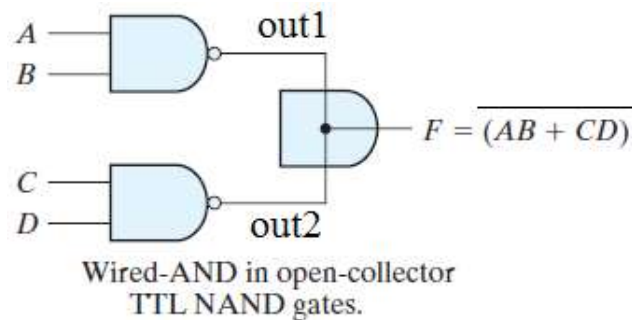
Wired – OR





منطق Wired (Wired-Logic)

اتصال خروجی دو گیت NAND کلکتور باز ساخته شده با تکنولوژی TTL، به صورت یک گیت AND عمل می کند. (یعنی در شکل زیر اگر $out1 = out2 = 1$ باشد، $F = 1$ و در غیر این صورت $F = 0$ می باشد. بنابراین این اتصال را با سمبل زیر که معرف گیت Wired-AND می باشد، نشان می دهند.



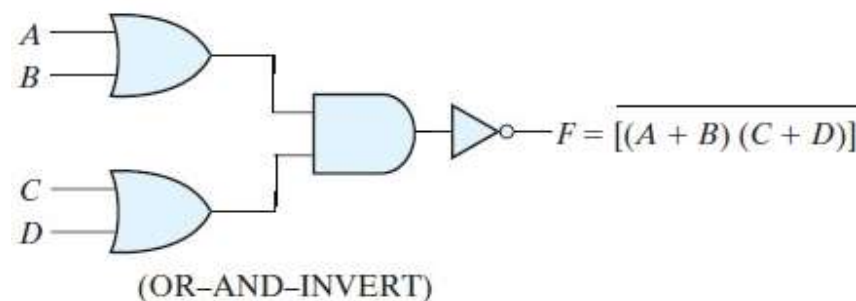
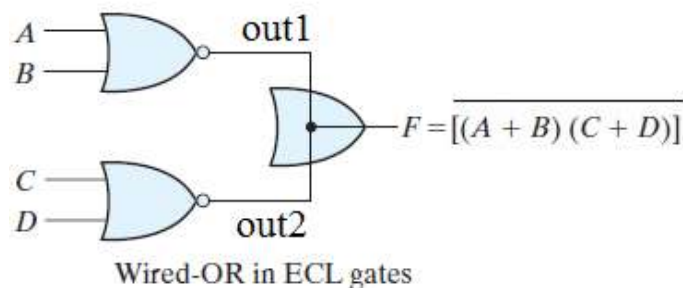
$$out1 = \overline{AB} \quad out2 = \overline{CD}$$

$$F = out1.out2 = \overline{AB}.\overline{CD} = \overline{AB + CD}$$



منطق Wired (Wired-Logic)

اتصال خروجی دو گیت NOR ساخته شده با تکنولوژی ECL، به صورت یک گیت OR عمل می کند. (یعنی در شکل زیر اگر $out1 = out2 = 0$ باشد، $F = 0$ و در غیر این صورت $F = 1$ می باشد. بنابراین این اتصال را با سمبل زیر که معرف گیت Wired-OR می باشد، نشان می دهند.

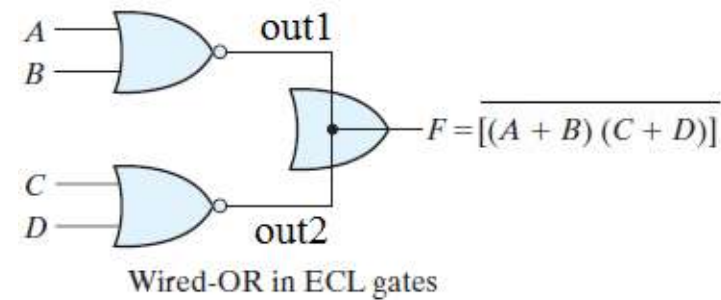
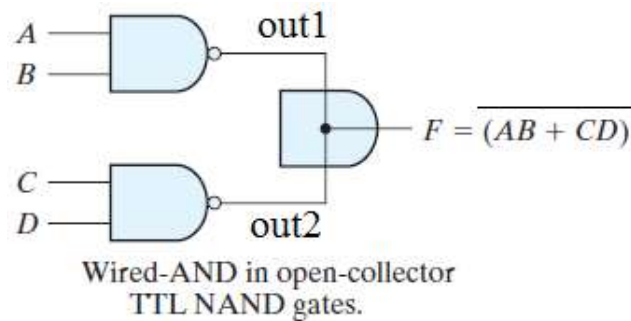


$$\begin{aligned} out1 &= \overline{(A + B)} \quad out2 = \overline{(C + D)} \\ F = out1 + out2 &= \overline{(A + B)} + \overline{(C + D)} = \overline{[(A + B).(C + D)]} \end{aligned}$$



منطق Wired (Wired-Logic)

اگرچه گیت های Wired – AND و Wired – OR های فیزیکی نیستند ولی دو پیاده سازی زیر پیاده سازی های دو سطحی در نظر گرفته می شوند.





مثال : تابع زیر را در هشت فرم مختلف در سطح گیت نمایش دهید.

$$F(w,x,y,z) = \sum(0,2,4,5,7,13,15)$$

		y			
		00	01	11	10
w	00	1	0	0	1
	01	1	1	1	0
	11	0	1	1	0
	10	0	0	0	0

yz

wx

$\bar{w}\bar{x}\bar{z}$

$\bar{w}y\bar{z}$

x

z

xz

$$F = \bar{w}\bar{x}\bar{z} + \bar{w}y\bar{z} + xz$$

(SOP)

		y			
		00	01	11	10
w	00	1	0	0	1
	01	1	1	1	0
	11	0	1	1	0
	10	0	0	0	0

yz

wx

$\bar{x}z$

$xy\bar{z}$

x

z

$w\bar{x}$

$w\bar{y}\bar{z}$

$$\bar{F} = \bar{x}z + w\bar{x} + xy\bar{z} + w\bar{y}\bar{z}$$

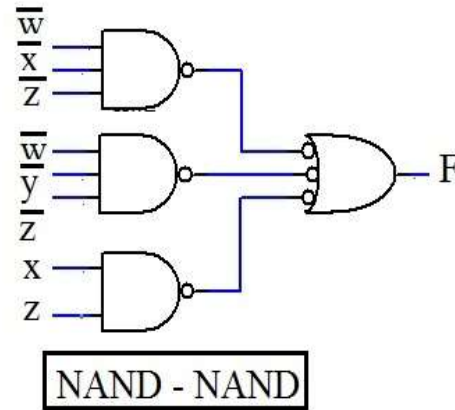
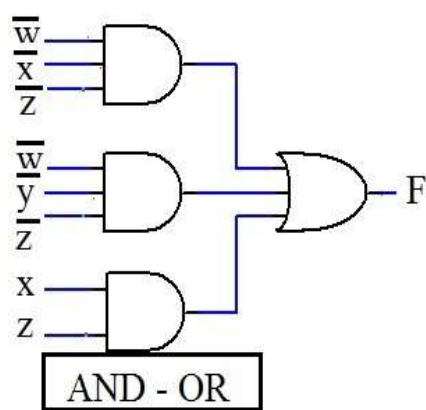
$$F = (x+\bar{z}).(\bar{w}+x).(\bar{x} + \bar{y}+z).(\bar{w}+y+z)$$

(POS)

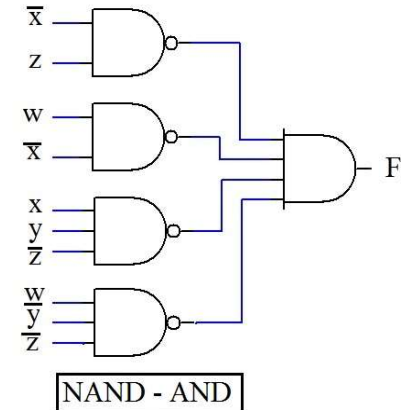
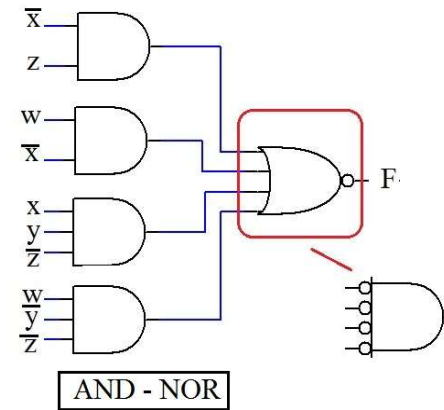


دانش تهران

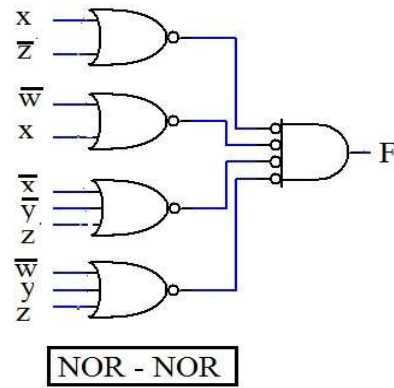
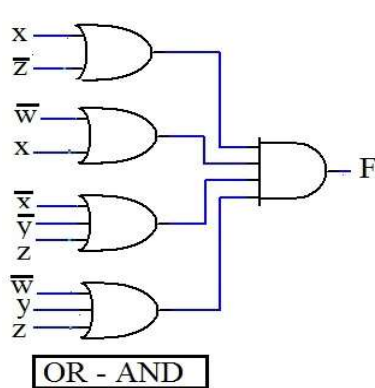
$$F = \bar{w}\bar{x}\bar{z} + \bar{w}\bar{y}\bar{z} + xz$$



$$F = \bar{\bar{F}} = \overline{(\bar{x}z + w\bar{x} + xy\bar{z} + w\bar{y}\bar{z})}$$



$$F = (x + \bar{z}).(\bar{w} + x).(\bar{x} + \bar{y} + z).(\bar{w} + y + z)$$



$$F = \overline{(w + x + z)} + \overline{(w + y + z)} + \overline{(\bar{x} + \bar{z})}$$

