

مبحث سوم

حداقل سازی سطح گیت

فهرست مطالب

- ← مقدمه‌ای بر حداقل‌سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog

مقدمه

یادآوری بازنمایی‌های مختلف دیجیتال (ویژگی‌ها و تبدیل):

بین فرم‌های مختلف عبارت منطقی تابع، کدام ارجح است؟



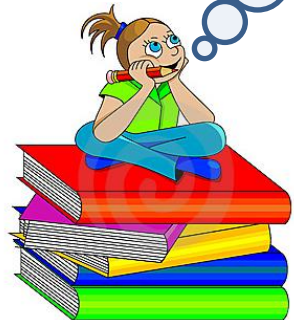
مفهوم و معیارهای حداقل‌سازی؟

اهمیت روش نقشه برای حداقل‌سازی؟

• حداقل جملات
• کمترین لیترال
• کمترین سطوح

روش نقشه کمک می‌کند
تا ساده‌سازی را **قاعدهمند**
و **ساده‌تر** انجام دهیم.

ساده‌ترین عبارت
منحصر به فرد نیست.



فهرست مطالب

مقدمه‌ای بر حداقل سازی

روش نقشه 

سیستماتیک کردن انتخاب ترکیب‌ها در نقشه

ساده‌سازی به فرم ضرب حاصل جمع‌ها

حالت‌های بی‌اهمیت

پیاده‌سازی با NAND و NOR

دیگر پیاده‌سازی‌های دو سطحی

تابع OR انحصاری

زبان توصیف سخت‌افزار Verilog

عناوین این بخش:

➤ نقشه‌ی دو متغیره

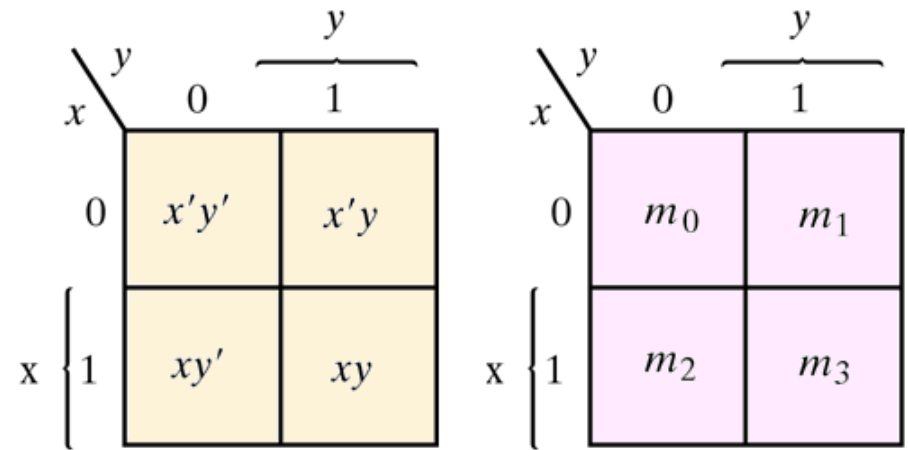
➤ نقشه‌ی سه متغیره

➤ نقشه‌ی چهار متغیره

➤ نقشه‌ی پنج متغیره

نقشه‌ی دو متغیره

x	y	minterm	Designation
0	0	$x'y'$	m_0
0	1	$x'y$	m_1
1	0	xy'	m_2
1	1	xy	m_3



در سطرها و ستون‌های با
برچسب صفر، متغیر به
صورت مکمل ظاهر می
گردد و ...



مثال: ساده‌سازی با نقشه دو متغیره

تابع های منطقی زیر را با استفاده از نقشه ساده کنید.

$$F_1(x,y) = \sum(1,2,3)$$

$$F_2(x,y) = \sum(3)$$

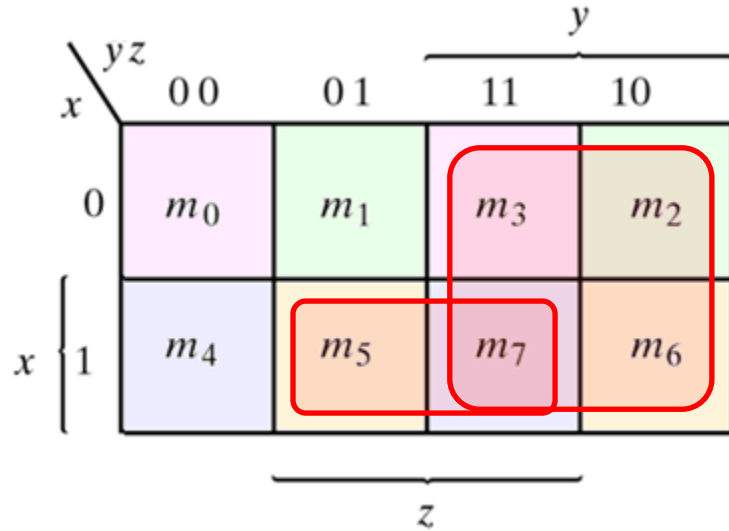
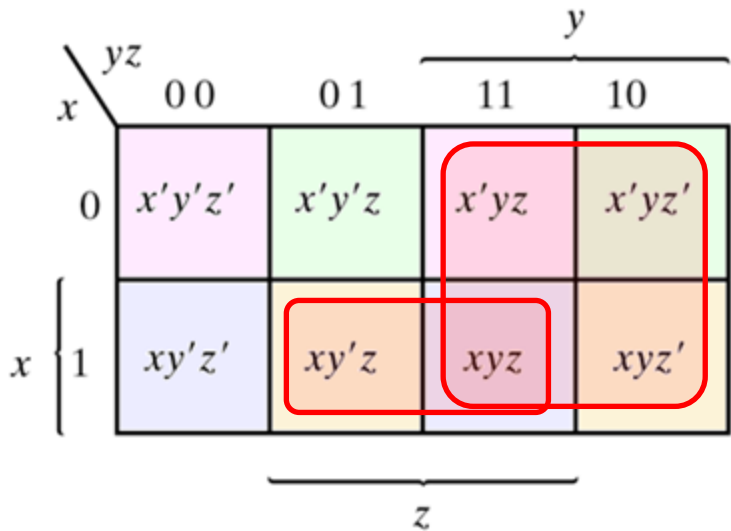
		y	
		0	1
x	0	0	1
	1	1	1

$$F_1(x,y) = y + x$$

		y	
		0	1
x	0	0	0
	1	0	1

$$F_2(x,y) = xy$$

نقشه‌ی سه متغیره



Gray Code
00
01
11
10

چرا چیدمان خانه‌ها اینچنین انتخاب شده است؟

در دو خانه‌ی مجاور هم، تمام لیترال‌ها، جز یک لیترال، یکسان هستند. در نتیجه با فاکتورگیری می‌توان آن لیترال را ساده نمود.

$$m_5 + m_7 = xy'z + xyz = xz(y' + y) = xz$$

$$m_3 + m_2 + m_7 + m_6 = x'yz + x'yz' + xyz + xyz' = x'y + xy = y$$

مثال:



تعداد خانه‌های مجاور که می‌توان آنها را ترکیب نمود همواره توانی از ۲ است:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

⋮

برای یک تابع سه متغیره:

یک خانه: جمله‌ای با سه لیترال (یک minterm)

دو خانه‌ی مجاور: جمله‌ای با دو لیترال

چهار خانه‌ی مجاور: جمله‌ای با یک لیترال

هشت خانه‌ی مجاور: تابع برابر یک است.

مثال: ساده‌سازی با نقشه سه‌متغیره

تابع‌های زیر را ساده کنید.



اطلاعات را وارد نقشه نمایید.

$$F_1(x,y,z) = \sum (2,3,4,5)$$

$$F_2(x,y,z) = \sum (3,4,6,7)$$

$$F_3(x,y,z) = \sum (0,2,4,5,6)$$

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

Groupings: $x'y$ (red box), xy' (green box)

$$F_1(x,y,z) = x'y + xy'$$

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

Groupings: yz (red box), xz' (green box)

$$F_2(x,y,z) = yz + xz'$$

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

Groupings: z' (red box), xy' (green box)

$$F_3(x,y,z) = z' + xy'$$

درستی ساده‌سازی خانه‌های سبز را بررسی کنید.



درستی ساده‌سازی خانه‌های قرمز را بررسی کنید.

مثال: نقشه‌ی سه‌متغیره

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

تابع روبرو را در نظر بگیرید.

الف: با استفاده از نقشه‌ی کارنو، تابع را به صورت جمع minterm ها بازنمایی کنید.

ب: تابع را ساده نمایید.

$$F(A,B,C) = \sum (1,2,3,5,7)$$

		B			
		00	01	11	10
A	0	0	1	1	1
	1	0	1	1	0

Diagram illustrating the Karnaugh map for the function $F(A,B,C)$. The map is a 2x4 grid with rows labeled A (0, 1) and columns labeled BC (00, 01, 11, 10). The values in the cells are: (0,00)=0, (0,01)=1, (0,11)=1, (0,10)=1, (1,00)=0, (1,01)=1, (1,11)=1, (1,10)=0. The cells containing 1s are grouped into three regions: a red box covering (0,01), (0,11), (1,01), and (1,11); a green box covering (0,11) and (0,10); and a blue box covering (0,01) and (1,01). Brackets labeled B, C, and A indicate the groupings.

$$F_1(A,B,C) = C + A'B$$

نقشه‌ی چهار متغیره

		y			
		yz	01	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

		y			
		yz	01	11	10
wx	00	m_0	m_1	m_3	m_2
	01	m_4	m_5	m_7	m_6
	11	m_{12}	m_{13}	m_{15}	m_{14}
	10	m_8	m_9	m_{11}	m_{10}

z

توجه کنید که خانه های m_0 ،
 m_2 ، m_8 و m_{10} مجاور هم
 هستند.



مثال: ساده‌سازی با نقشه‌ی چهار متغیره

تابع زیر را ساده کنید.

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

wx \ yz		yz		y	
		00	01	11	10
w	00	1	1	0	1
	01	1	1	0	1
	11	1	1	0	1
	10	1	1	0	0

Diagram illustrating the simplification of the function $F(w,x,y,z)$ using a 4-variable Karnaugh map. The map is a 4x4 grid with variables w and x on the vertical axis, and y and z on the horizontal axis. The cells containing 1s are grouped into three prime implicants: a red group covering $y=01$ and $y=11$ (simplifying to y'), a blue group covering $w=00$ and $w=01$ (simplifying to $w'z'$), and a green group covering $x=01$ and $x=11$ (simplifying to xz').

$$F(w,x,y,z) = y' + w'z' + xz'$$

مثال: ساده‌سازی با نقشه‌ی چهار متغیره

تابع زیر را ساده کنید.



$$F(A,B,C,D) = A'B'C' + B'CD' + A'BCD' + AB'C'$$

اطلاعات را وارد نقشه نمایید.

		C			
		00	01	11	10
A	00	1	1	0	1
	01	0	0	0	1
	11	0	0	0	0
	10	1	1	0	1

The Karnaugh map shows four groups of 1s: a red group for $B'C'$ (top-left 2x2), a blue group for $B'D'$ (right column), a green group for $A'CD'$ (top-left 1s), and another green group for $A'CD'$ (bottom-right 1s). Brackets labeled A, B, C, and D indicate the variable ranges for each group.

$$F(A,B,C,D) = B'C' + B'D' + A'CD'$$

نقشه‌ی پنج متغیره

		$A = 0$			
		DE		D	
BC		00	01	11	10
B	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10
		E			

		$A = 1$			
		DE		D	
BC		00	01	11	10
B	00	16	17	19	18
	01	20	21	23	22
	11	28	29	31	30
	10	24	25	27	26
		E			

توجه کنید که خانه های m_0 ،
 m_{16} و ... مجاور هم هستند.



مثال: ساده‌سازی با نقشه‌ی پنج متغیره

$$F(A,B,C,D,E) = \sum(0,2,4,6,9,13,21,23,25,29,31)$$

تابع روبرو را ساده کنید.



اطلاعات را وارد نقشه نمایید.

A=0

BC \ DE		D			
		00	01	11	10
00	00	1	0	0	1
	01	1	0	0	1
11	11	0	1	0	0
	10	0	1	0	0

Groupings: A red box highlights the 1s in the first two rows (C=0). A green box highlights the 1s in the third and fourth rows (C=1). Brackets indicate groupings for variables C, D, and E.

A=1

BC \ DE		D			
		00	01	11	10
00	00	0	0	0	0
	01	0	1	1	0
11	11	0	1	1	0
	10	0	1	0	0

Groupings: A blue box highlights the 1s in the second and third rows (C=1). A green box highlights the 1s in the second and fourth rows (C=0). Brackets indicate groupings for variables C, D, and E.

$$F(A,B,C,D,E) = A'B'E' + BD'E + ACE$$

مثال: ترکیب توابع در نقشه

$$F(w,x,y,z) = w'xy' + w'yz + xz + wxy + wyz + wx'y'z'$$

$$G(w,x,y,z) = xz + yz + w'xy + wxy' + wx'y + wx'y'z'$$

تابع های F و G داده شده است.

با استفاده از روش نقشه، تابع های

زیر را ساده نمایید.

الف: $H(w,x,y,z) = F \cdot G$

ب: $K(w,x,y,z) = F + G$



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

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

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

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

$$H(w,x,y,z) = yz + xz$$

$$K(w,x,y,z) = yz + x + w$$

اطلاعات تابع های F و G را وارد

نقشه نمایید.



نقشه ی تابع های H و K چه ارتباطی

با نقشه ی تابع های F و G دارد؟

فهرست مطالب

مقدمه‌ای بر حداقل‌سازی

روش نقشه

سیستماتیک کردن انتخاب ترکیب‌ها در نقشه 

ساده‌سازی به فرم ضرب حاصل جمع‌ها

حالت‌های بی‌اهمیت

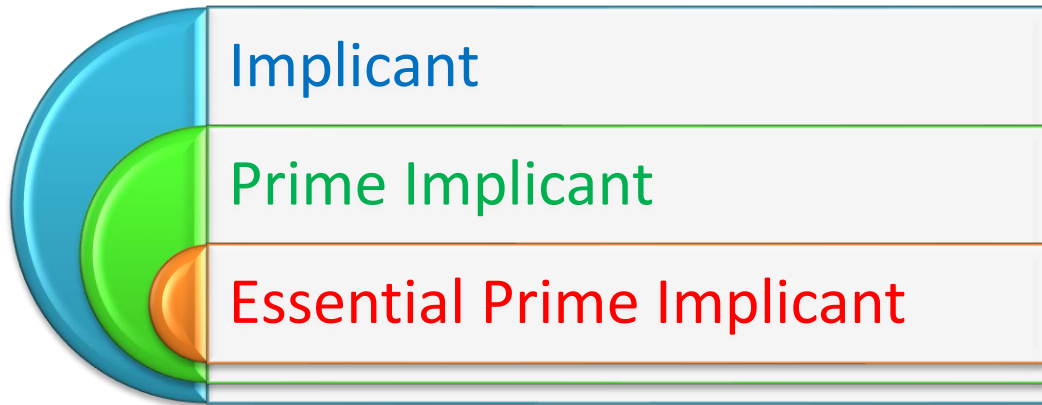
پیاده‌سازی با NAND و NOR

دیگر پیاده‌سازی‌های دو سطحی

تابع OR انحصاری

زبان توصیف سخت‌افزار Verilog

سیستماتیک کردن انتخاب ترکیب‌ها در نقشه



جمله‌ی حاصلضربی که از ترکیب بیشترین خانه‌های مجاور هم در نقشه حاصل می‌گردد.

اگر یک **minterm** تنها با یک موجب اصلی پوشش داده شود به آن موجب اصلی، موجب اصلی ضروری گویند.

مثال: مفهوم موجب اصلی ضروری

$$F(A,B,C,D) = \sum(0,2,3,5,7,8,9,10,11,13,15)$$

تابع روبرو را در نظر بگیرید.

		C			
		00	01	11	10
A	yz	00	01	11	10
	AB	00	01	11	10
	00	1	0	1	1
	01	0	1	1	0
11	0	1	1	0	
10	1	1	1	1	
		D			

الف: موجب های اصلی ضروری تابع را مشخص نمایید.

ب: گزینه های مختلف موجب های اصلی برای پوشش

دادن دیگر minterm ها را ذکر کنید.

ج: گزینه های مختلف ساده سازی را لیست نمایید.

Essential Prime Implicant: $B'D'$, BD

Prime Implicant: $m3 : CD$, $B'C$

$m9 : AD$, AB'

$m11 : CD$, $B'C$, AB' , AD

$$F = B'D' + BD + CD + AD$$

$$= B'D' + BD + CD + AB'$$

$$= B'D' + BD + B'C + AD$$

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

چرا این دو ترکیب، ترکیب اصلی ضروری است؟



منظورم این بود که گفتم:

«ساده ترین عبارت منحصر

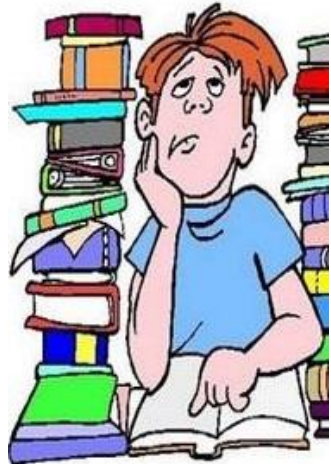
به فرد نیست.»



فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها 
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog

ساده‌سازی به فرم ضرب حاصل جمع‌ها



برای ساده‌سازی به فرم ضرب حاصل جمع‌ها با استفاده از نقشه چه پیشنهادی دارید؟



برای این منظور:

- I. با در نظر گرفتن صفرها در نقشه‌ی کارنو، مکمل تابع را به صورت جمع حاصل ضرب‌ها به دست می‌آوریم.
- II. با مکمل گرفتن از این عبارت، خود تابع را به فرم ضرب حاصل جمع‌ها بازنویسی می‌کنیم.

مثال: ساده‌سازی به فرم ضرب حاصل جمع‌ها

$$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$$

تابع روبرو را به صورت‌های خواسته شده ساده کنید.

شماتیک مداری ساده شده را نیز برای هر حالت ترسیم نمایید.

AB \ CD		C			
		00	01	11	10
A	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1

Groupings in the Karnaugh map:
 - A red box covers cells (00,00), (01,00), (10,00), (10,10) representing $B'C'$.
 - A blue box covers cells (01,00), (01,01), (10,00), (10,10) representing $B'D'$.
 - Green circles highlight cells (00,00), (10,00), (10,10), and (10,10) representing $A'C'D$.

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

AB \ CD		C			
		00	01	11	10
A	00	1	1	0	1
	01	0	1	0	0
	11	0	0	0	0
	10	1	1	0	1

Groupings in the Karnaugh map:
 - A red box covers cells (11,00), (11,01), (11,10), (11,11) representing CD .
 - A green box covers cells (00,00), (00,01), (10,00), (10,01) representing BD' .
 - A blue box covers cells (11,00), (11,01), (10,00), (10,01) representing AB .

الف: به صورت جمع حاصل ضرب‌ها

ب: به صورت ضرب حاصل جمع‌ها

$$F'(A,B,C,D) = CD + BD' + AB$$

$$\Rightarrow F(A,B,C,D) = (C'+D')(B'+D)(A'+B')$$

مثال: ساده‌سازی به فرم ضرب حاصل جمع‌ها

$$F(x,y,z) = \prod(0,2,5,7)$$

تابع روبرو را به صورت‌های خواسته شده ساده کنید.

شماتیک مداری ساده شده را نیز برای هر حالت ترسیم نمایید.

اطلاعات را وارد نقشه نمایید.



الف: به صورت جمع حاصل ضرب‌ها

ب: به صورت ضرب حاصل جمع‌ها

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

$$F(x,y,z) = x'z + xz'$$

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

$$F'(x,y,z) = x'z' + xz$$

$$\Rightarrow F(x,y,z) = (x+z)(x'+z')$$

مثال: وارد کردن تابعی به فرم ضرب حاصل جمع‌ها در نقشه

تابع $F(w,x,y,z)$ را با استفاده از روش نقشه ساده نمایید.

$$F(w,x,y,z) = (w+x+y)(w+x+z)(x+y+z)(w+x'+y+z)(w'+x'+y'+z)$$

اطلاعات تابع را وارد نقشه نمایید.



$$F'(w,x,y,z) = w'x'y' + w'x'z' + x'y'z' + w'xy'z' + wxyz'$$

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

Brackets in the original image indicate groupings: a vertical bracket on the right for 'x' (rows 01, 11, 10), a horizontal bracket at the bottom for 'z' (columns 01, 11, 10), and a horizontal bracket at the top for 'y' (columns 11, 10).

$$F(w,x,y,z) = yz + xz + w'xy + wxy' + wz + wx'y$$

فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت 
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog

حالت‌های بی‌اهمیت

تابع غیر کامل (incompletely specified function)

تابعی که خروجی آن به ازای برخی ترکیب‌های ورودی نامشخص باشد.

حالت‌های بی‌اهمیت

mintermهای مشخص نشده در تابع غیر کامل

مثال: کاربرد حالت‌های بی‌اهمیت در ساده‌سازی

تابع $F(w,x,y,z)$ با حالت‌های بی‌اهمیت d را به فرم‌های خواسته شده ساده کنید.

$$F(w,x,y,z) = \sum(1,3,7,11,15)$$

$$d(w,x,y,z) = \sum(0,2,5)$$

اطلاعات را وارد نقشه نمایید.



الف: به صورت جمع حاصل ضرب‌ها

ب: به صورت ضرب حاصل جمع‌ها

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

$$F(w,x,y,z) = w'z + yz$$

Alternative: $F(w,x,y,z) = w'x' + yz$

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

$$F'(w,x,y,z) = z' + wy'$$

$$\Rightarrow F(w,x,y,z) = z(w'+y)$$

استفاده از حالت‌های بی‌اهمیت چه فایده‌ای دارد؟



فهرست مطالب

مقدمه‌ای بر حداقل سازی

روش نقشه

سیستماتیک کردن انتخاب ترکیب‌ها در نقشه

ساده‌سازی به فرم ضرب حاصل جمع‌ها

حالت‌های بی‌اهمیت

پیاده‌سازی با NAND و NOR 

دیگر پیاده‌سازی‌های دو سطحی

تابع OR انحصاری

زبان توصیف سخت‌افزار Verilog

عناوین این بخش:

➤ معرفی گیت یونیورسال

➤ تعبیری جدید از قضیه‌ی دمورگان

➤ پیاده‌سازی با گیت‌های NAND

➤ پیاده‌سازی با گیت‌های NOR

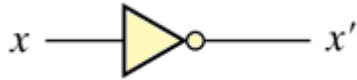
➤ مدارهای NAND و NOR چندسطحی

معرفی گیت یونیورسال

تعریف: گیت یونیورسال (universal gate) به گیتی اطلاق می‌گردد که بتوان گیت‌های AND ، OR ، و NOT را تنها با استفاده از آن نوع گیت پیاده‌سازی نمود.

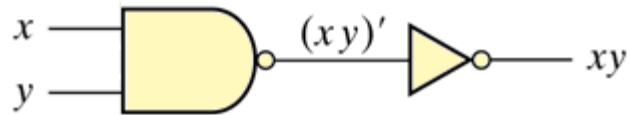
مثال: نشان دهید گیت NAND یک گیت یونیورسال است.

NOT:



گیت NAND یک ورودی مانند NOT عمل می‌کند.

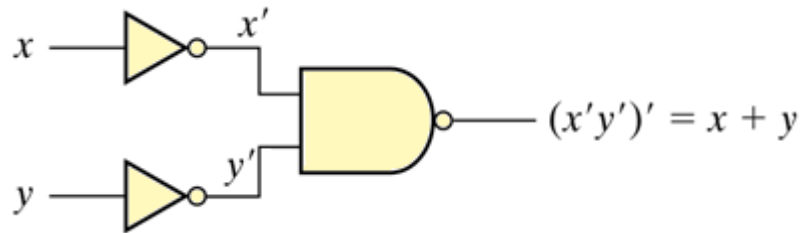
AND:



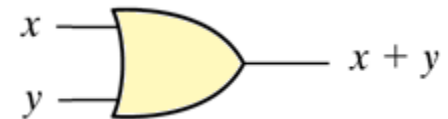
≡



OR:



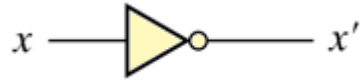
≡





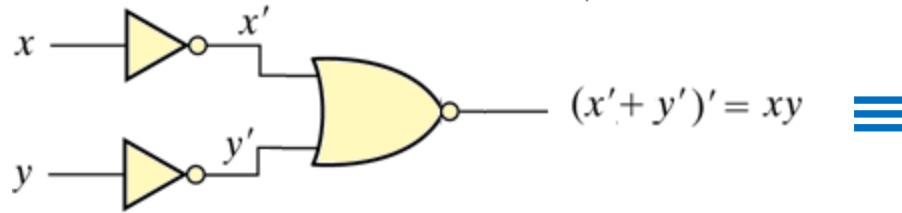
نشان دهید گیت NOR یک گیت یونیورسال است.

NOT:

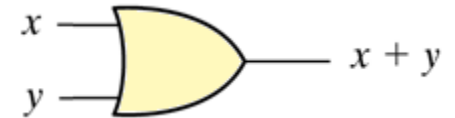
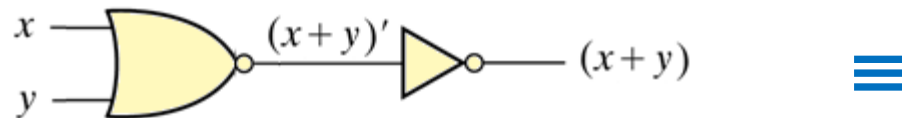


گیت NOR یک ورودی مانند NOT عمل می کند.

AND:

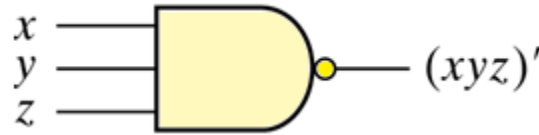


OR:

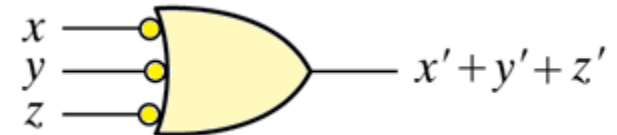


تعبیری دیگر از قضیه‌ی DeMorgan

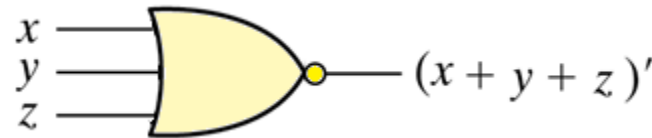
$$(xyz)' = x' + y' + z'$$



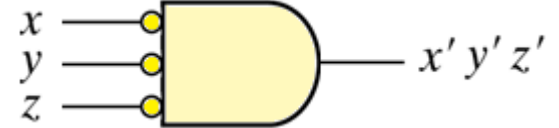
≡



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



≡

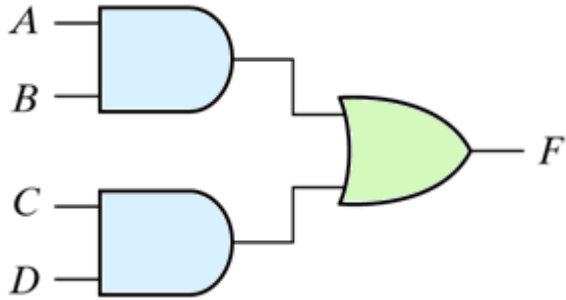


حباب NOT را می‌توان با تغییر
عمل گیت، از گیت عبور داد!

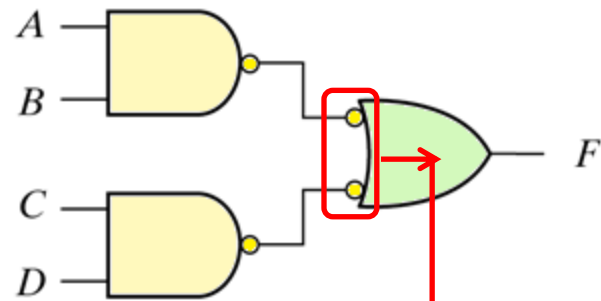


پیاده‌سازی تابع با NAND

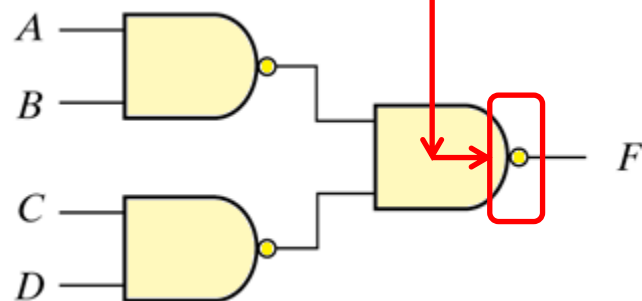
برای پیاده‌سازی تابع با گیت‌های NAND (دستیابی به ساختار NAND-NAND):



با استفاده از روش نقشه، تابع را به فرم جمع حاصل ضرب‌ها (آرایه‌ی AND-OR) پیاده‌سازی می‌کنیم.



در خروجی گیت‌های AND و ورودی‌های گیت‌های OR حساب اضافه می‌کنیم.



حساب ورودی گیت‌های OR را با تغییر عمل این گیت‌ها به AND به خروجی آن منتقل می‌کنیم.



اکنون پیاده‌سازی به فرم NAND-NAND است.



مثال: تابع $F(x,y,z)$ را با گیت‌های NAND پیاده‌سازی نمایید.

$$F(x,y,z) = \sum(1,2,3,4,5,7)$$

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

yz

z

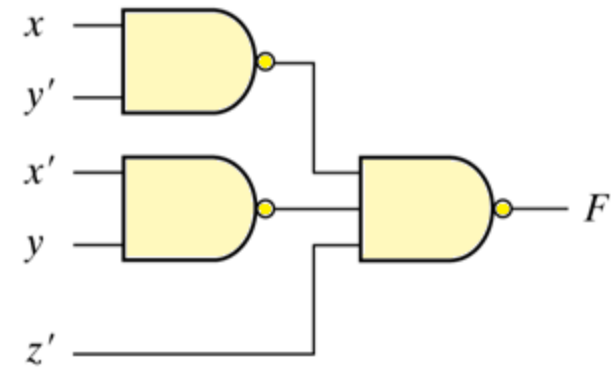
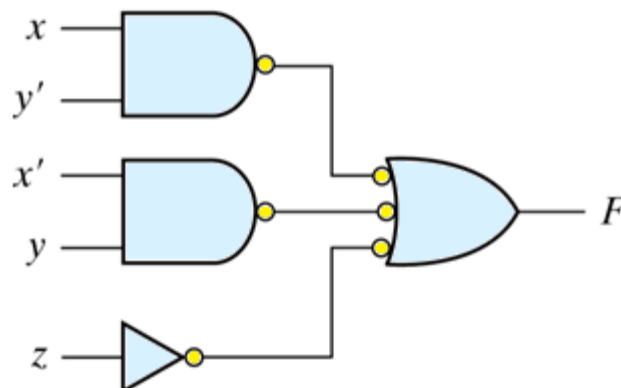
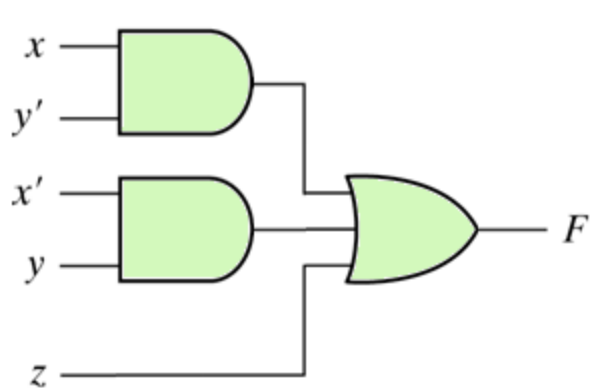
x

$$F(x,y,z) = z + x'y + xy'$$

$$= ((z + x'y + xy'))'$$

$$= (z' \cdot (x'y)' \cdot (xy'))'$$

$$F(x,y,z) = z + x'y + xy'$$



پیاده‌سازی تابع با NOR

برای پیاده‌سازی تابع با گیت‌های NOR (دستیابی به ساختار NOR-NOR):

- (1) با استفاده از روش نقشه، تابع را به فرم ضرب حاصل جمع‌ها (آرایه‌ی OR-AND) پیاده‌سازی می‌کنیم.
- (2) در خروجی گیت‌های OR و ورودی‌های گیت‌های AND حباب اضافه می‌کنیم.
- (3) حباب ورودی گیت‌های AND را با تغییر عمل این گیت‌ها به OR به خروجی آن منتقل می‌کنیم. اکنون پیاده‌سازی به فرم NOR-NOR است.

تابع $F(x,y,z)$ را با گیت‌های NOR پیاده‌سازی نمایید.

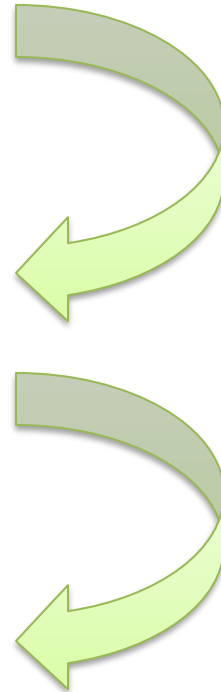
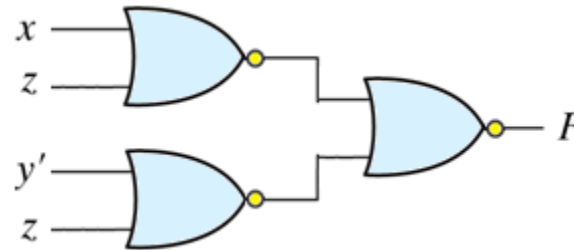
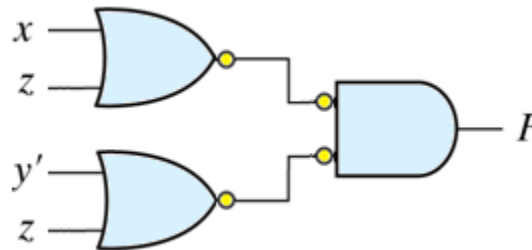
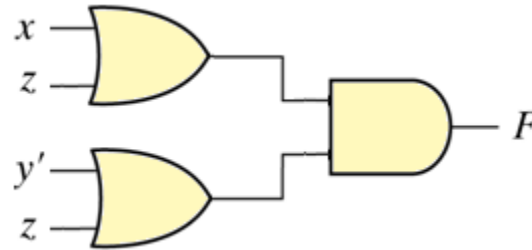
$$F(x,y,z) = \sum(1,3,4,5,7)$$

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

$$F'(x,y,z) = x'z' + yz'$$

$$F(x,y,z) = (x'z' + yz')'$$

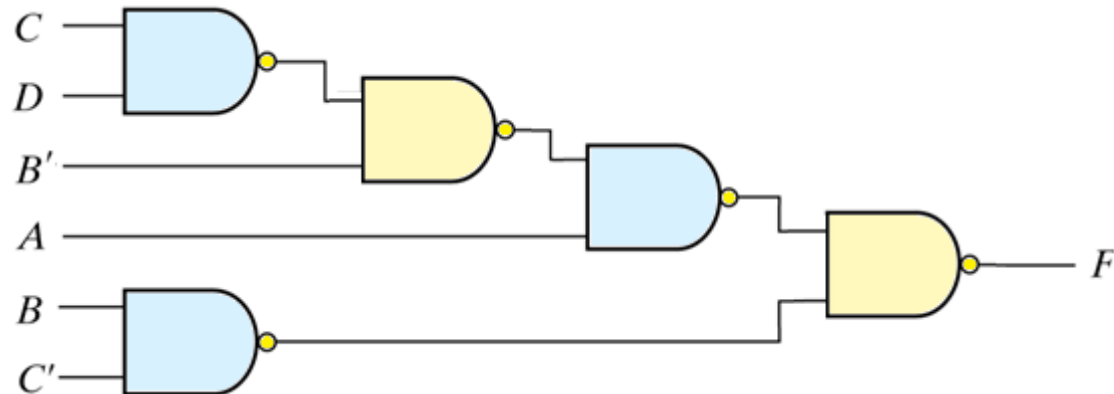
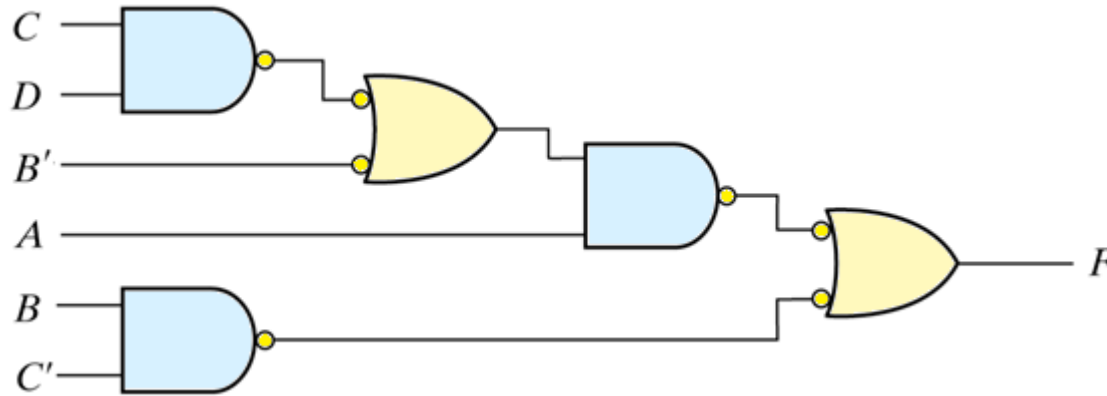
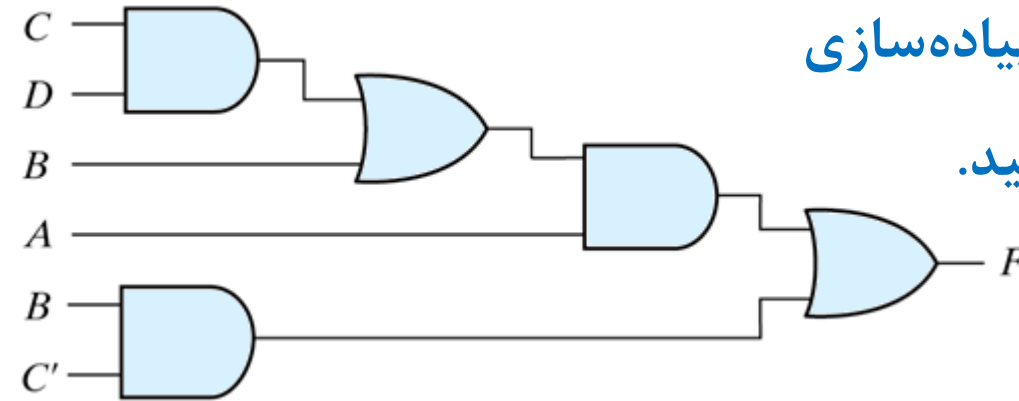
$$= (x+z)(y'+z)$$



مدارهای NAND چندسطحی

مثال: شماتیک مدار شکل روبرو را برای پیاده‌سازی

فقط با گیت‌های NAND بازنمایی کنید.



مدارهای NAND و NOR چندسطحی

مثال: شماتیک مدار شکل روبرو را به پیاده‌سازی‌های

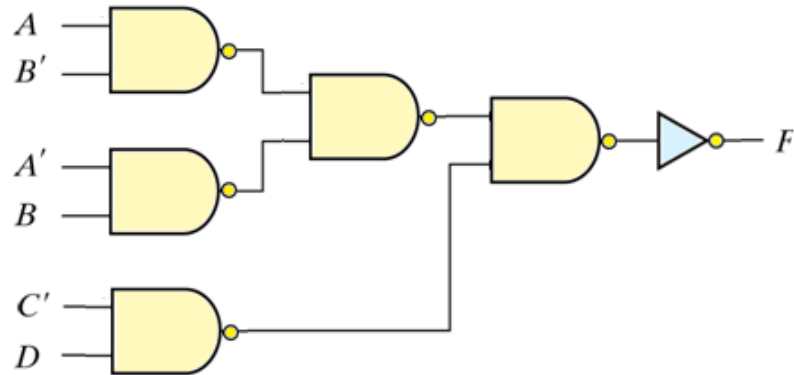
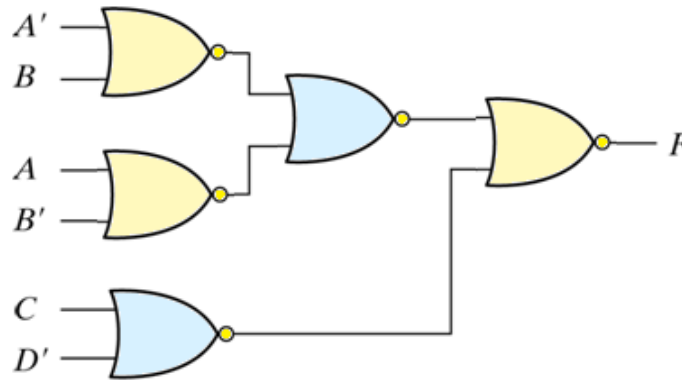
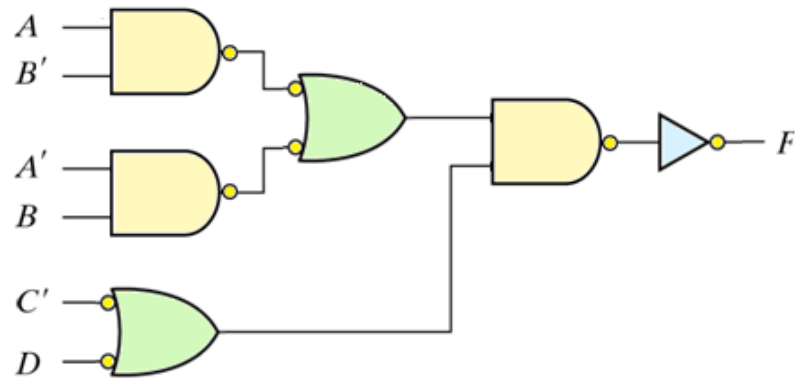
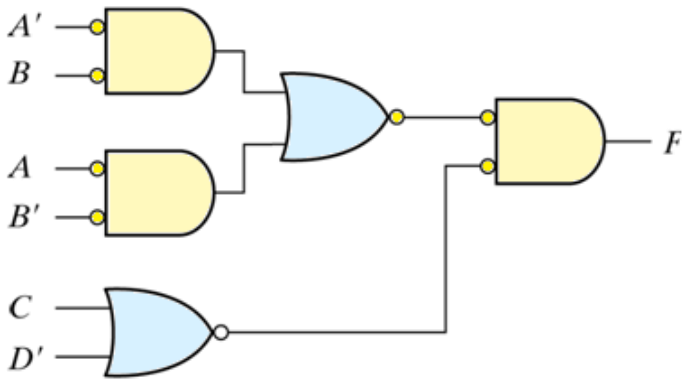
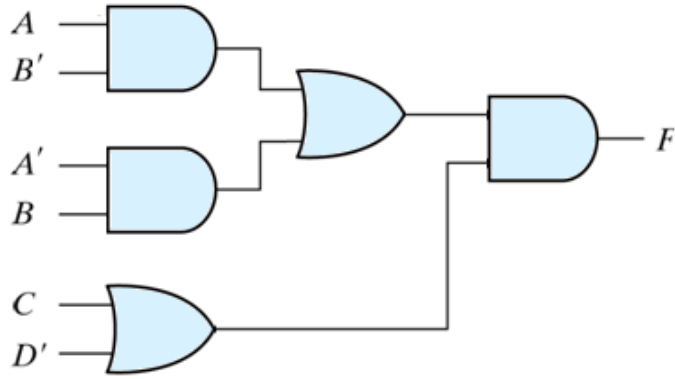
خواسته شده تبدیل نمایید.

الف: پیاده‌سازی با NOR

ب: پیاده‌سازی با NAND

الف:

ب:



فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی 
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog

دیگر پیاده‌سازی‌های دوسطحی

ترکیب‌های ممکن برای چهار گیت AND ، OR ، NAND و NOR :

~~AND - AND~~ بحث شد.

~~AND - NAND~~

~~AND - NOR~~

~~OR - AND~~ بحث شد.

~~OR - OR~~

~~OR - NAND~~

~~OR - NOR~~

~~NAND - AND~~

~~NAND - OR~~

~~NAND - NAND~~ بحث شد.

~~NAND - NOR~~

~~NOR - AND~~

~~NOR - OR~~

~~NOR - NAND~~

~~NOR - NOR~~ بحث شد.

\equiv AND - OR - INVERT

\equiv OR - AND - INVERT

مثال: تابع $F(x,y,z)$ را به فرم‌های خواسته شده پیاده‌سازی نمایید.

$$F(x,y,z) = \sum(0,6)$$

الف: AND – NOR

ب: NAND – AND

ج: OR – NAND

د: NOR – OR

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

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

$$F'(x,y,z) = z + x'y + xy'$$

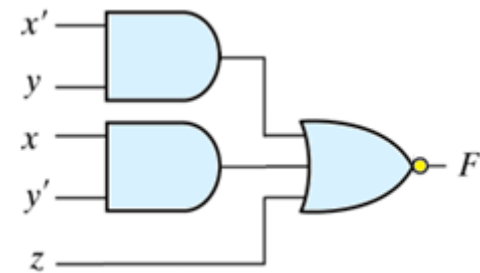
$$F(x,y,z) = (z + x'y + xy')'$$

$$F(x,y,z) = (z)'(x'y)'(xy)'$$

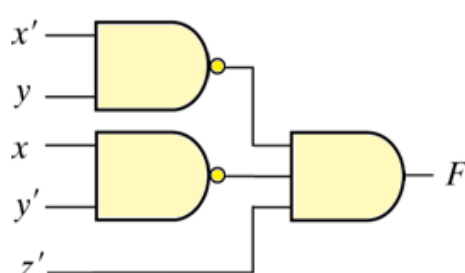
$$F(x,y,z) = x'y'z' + xyz'$$

$$F(x,y,z) = ((x'y'z' + xyz')')' = ((x+y+z)(x'+y'+z))'$$

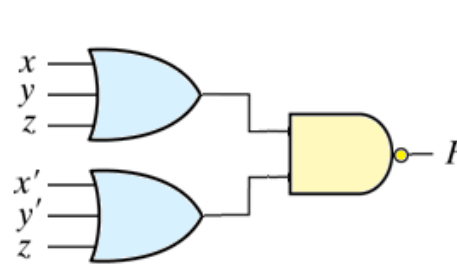
$$F(x,y,z) = (x+y+z)' + (x'+y'+z)'$$



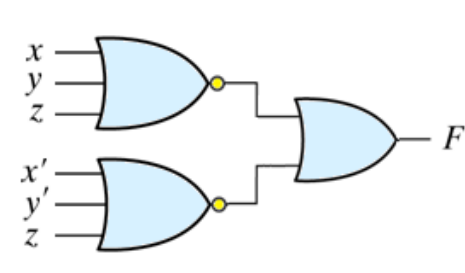
الف:



ب:



ج:



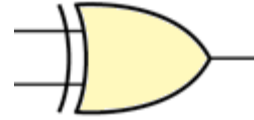
د:

فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری 
- زبان توصیف سخت‌افزار Verilog

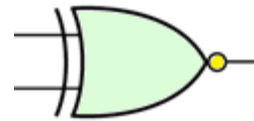
OR انحصاری (Exclusive OR - XOR)

XOR: $x \oplus y = x'y + xy'$



		y	
		0	1
x	0	0	1
	1	1	0

XNOR: $(x \oplus y)' = (x'y + xy)'$
 $= (x+y')(x'+y)$
 $= xx' + xy + x'y' + yy'$
 $= xy + x'y'$



		y	
		0	1
x	0	1	0
	1	0	1

OR انحصاری (Exclusive OR - XOR)

ویژگی‌ها:

$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

$$(x \oplus y)' = x \oplus y' = x' \oplus y$$

Commutative: $x \oplus y = y \oplus x$

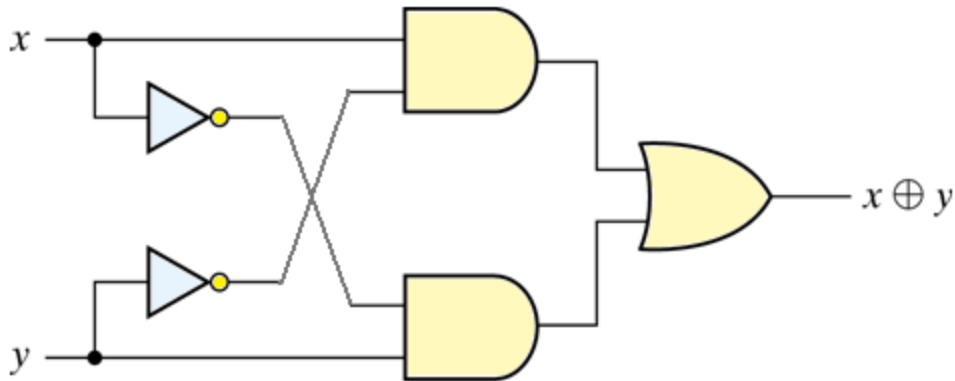
Associative: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

از XOR می‌توان به عنوان NOT کنترل شده استفاده نمود.

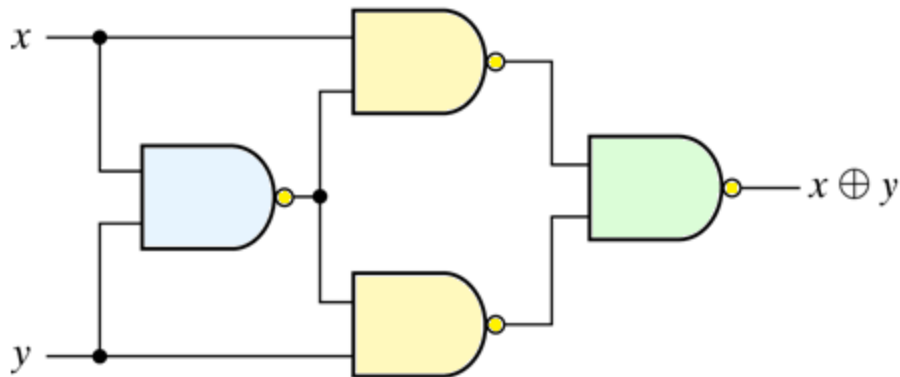
بنابراین شرط‌های گسترش ورودی‌های گیت برای XOR برقرار است.

پیاده‌سازی XOR

پیاده‌سازی با گیت‌های AND ، OR ، NOT :



پیاده‌سازی با گیت‌های NAND :



$$\begin{aligned} F &= \{ [x (xy)']' [y (xy)']' \}' \\ &= [x (xy)'] + [y (xy)'] \\ &= [x (x' + y')] + [y (x' + y')] \\ &= xx' + xy' + x'y + yy' \\ &= xy' + x'y \\ &= x \oplus y \end{aligned}$$

نشان دهید که شماتیک فوق، عمل گیت

XOR را پیاده‌سازی می‌کند.

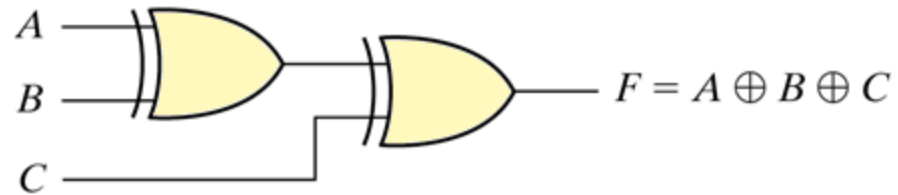


XOR سه متغيره

XOR:

$$\begin{aligned}
 x \oplus y \oplus z &= (x \oplus y) \oplus z = (x'y + xy') \oplus z \\
 &= (x'y + xy')z' + (x'y' + xy)z \\
 &= x'yz' + xy'z' + x'y'z + xyz \\
 &= \sum (1,2,4,7)
 \end{aligned}$$

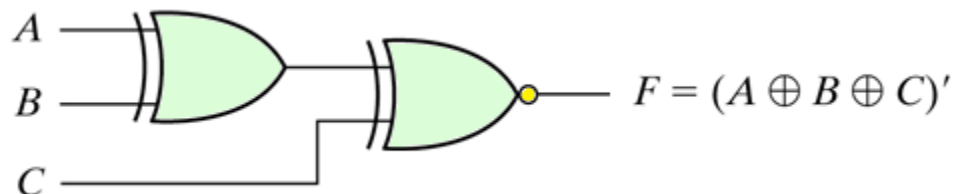
		y			
		00	01	11	10
x	0	0	1	0	1
	1	1	0	1	0



XNOR:

$$(x \oplus y \oplus z)' = \sum (0,3,5,6)$$

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



XOR چهار متغیره

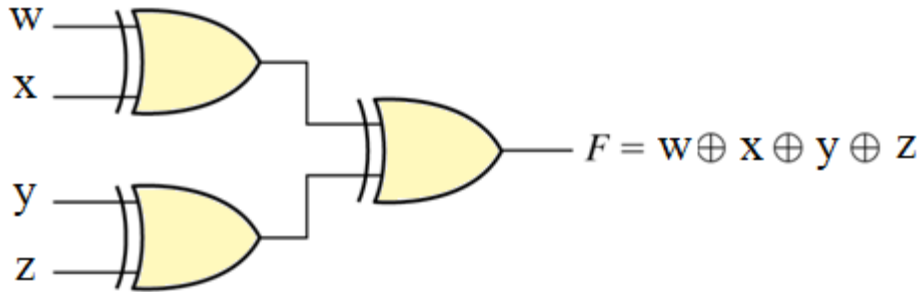
XOR:

$$w \oplus x \oplus y \oplus z = (w \oplus x) \oplus (y \oplus z)$$

$$= (w'x + wx') (y'z' + yz) + (w'x' + wx) (y'z + yz')$$

$$= \sum (4,7,8,11,12,13,14)$$

$$= \sum (1,2,4,7,8,11,13,14)$$



wx \ yz		y			
		00	01	11	10
w	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

Brackets indicate the output is 1 when the number of 1s in the input is odd (1, 3, 5, 7, 9, 11, 13, 15).

تابع زوج و تابع فرد

XOR چند متغیره تنها در صورتی یک می شود که تعداد فردی از متغیرهای آن یک باشد. از این رو، XOR را **تابع فرد** گوییم.

XNOR چند متغیره تنها در صورتی یک می شود که تعداد زوجی از متغیرهای آن یک باشد. از این رو، XNOR را **تابع زوج** گوییم.

کاربردهای XOR

۱- عملیات حسابی

۲- تشخیص خطا

۳- تصحیح خطا

مثال: تشخیص و تصحیح خطا (تولید و بررسی توازن)

می خواهیم برای ارسال یک پیام سه بیتی، به آن بیت توازن زوج اضافه نماییم.

الف: با استفاده از گیت های XOR دو ورودی، مداری برای این منظور طراحی نمایید.

ب: با استفاده از گیت های XOR دو ورودی، مداری جهت تشخیص خطای توازن در گیرنده طراحی کنید.

ج: آیا می توان از یک مدار برای هر دو منظور استفاده نمود؟

پادآوری

بیت توازن زوج: بیتی که به پیام اضافه می گردد تا تعداد کل یک ها زوج شود.

بیت توازن فرد: بیتی که به پیام اضافه می گردد تا تعداد کل یک ها فرد شود.

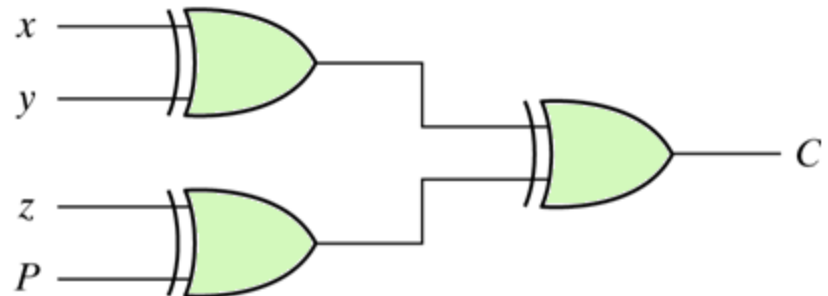
حل الف:

تعداد یک های پیام	بیت توازن زوج
فرد	۱
زوج	۰

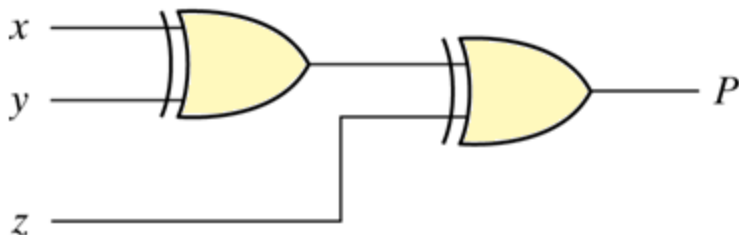
حل ب:

تعداد یک های پیام + توازن	خطا
فرد	۱
زوج	۰

بنابراین نیاز به یک تابع فرد چهار-ورودی داریم.



بنابراین نیاز به یک تابع فرد سه-ورودی داریم.



حل ج: از مدار دوم، در صورتی که $P=0$ شود می توان برای تولید بیت توازن نیز استفاده نمود.

$$\text{for } P=0 \Rightarrow x \oplus y \oplus z \oplus P = x \oplus y \oplus z \oplus 0 = x \oplus y \oplus z$$

مثال: ساده‌سازی با حضور XOR

اگر استفاده از گیت XOR مجاز باشد تابع زیر را با استفاده از روش نقشه و برای پیاده‌سازی با کمترین تعداد گیت ساده نمایید.

$$F(w,x,y,z) = \sum(2,3,5,6,8,9,12,15)$$

wx \ yz		y			
		00	01	11	10
w	00	0	0	1	1
	01	0	1	0	1
	11	1	0	1	0
	10	1	1	0	0

Diagram illustrating the Karnaugh map for the function $F(w,x,y,z) = \sum(2,3,5,6,8,9,12,15)$. The map is a 4x4 grid with variables w and x on the vertical axis, and y and z on the horizontal axis. The cells are grouped into four regions: a green region (top-left and bottom-right), a red region (top-middle and bottom-middle), and two other green regions (top-right and bottom-left). The red region is highlighted with a red border, and the green regions are highlighted with green borders. The output values are 0 or 1.

$$F(w,x,y,z) = z(w \oplus x \oplus y) + z'(w \oplus y)$$

مثال: ساده‌سازی با حضور XOR

اگر استفاده از گیت XOR مجاز باشد تابع زیر را با استفاده از روش نقشه و برای پیاده‌سازی با کمترین تعداد گیت ساده نمایش دهید.

$$F(w,x,y,z) = \sum(1,3,4,6,9,11,12,14)$$

wx \ yz		y			
		00	01	11	10
w	00	0	1	1	0
	01	1	0	0	1
	11	1	0	0	1
	10	0	1	1	0

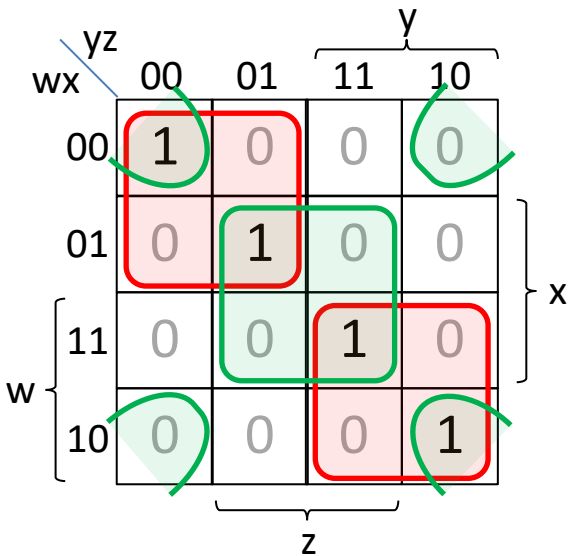
$$F(w,x,y,z) = x \oplus z$$

مثال: ساده‌سازی با حضور XOR

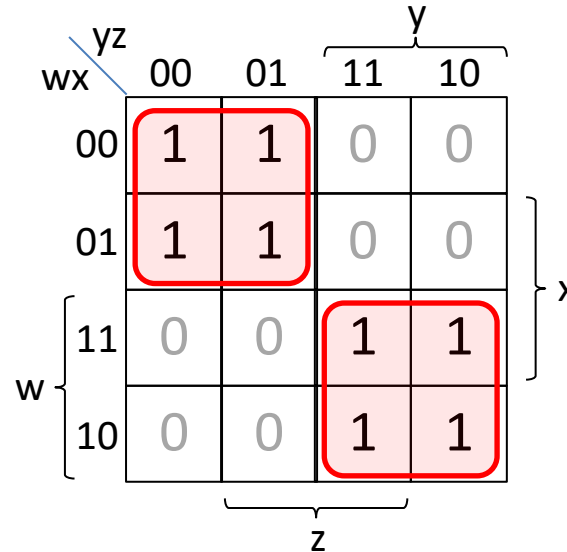
اگر استفاده از گیت XOR مجاز باشد تابع زیر را با استفاده از روش نقشه و برای پیاده‌سازی با کمترین

تعداد گیت ساده نمایید.

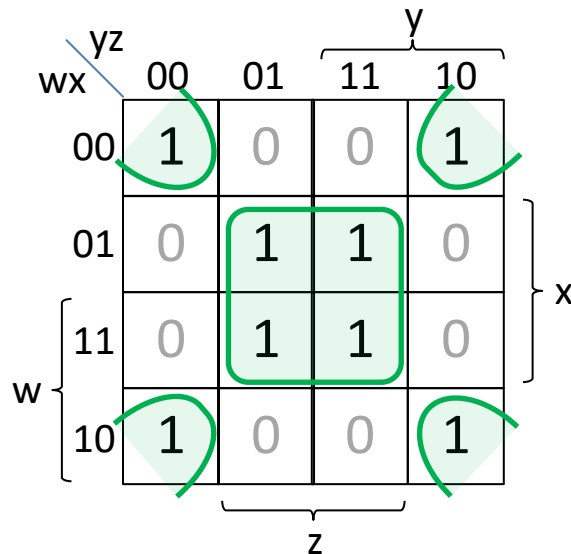
$$F(w,x,y,z) = \sum(0,5,10,15)$$



$$F(w,x,y,z) = (w \oplus y)' (x \oplus z)'$$



$$(w \oplus y)'$$



$$(x \oplus z)'$$

فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog 

زبان توصیف سخت افزار

انواع زبان توصیف
سخت افزار

Verilog

VHDL

ضرورت استفاده
از HDL

مفهوم زبان توصیف
سخت افزار

زبان توصیف سخت افزار Verilog

ویژگی‌ها:

- ❖ از شیوه‌های مختلف بازنمایی طرح دیجیتال (شماتیک، جدول درستی و روابط منطقی) پشتیبانی می‌کند.
- ❖ دارای حدود ۱۰۰ کلیدواژه است.
- ❖ حساس به اندازه است.
- ❖ اجزای اصلی توصیف طرح ماژول است که هر ماژول با کلیدواژه‌ی **module** آغاز و با کلیدواژه‌ی **endmodule** خاتمه می‌یابد.
- ❖ عبارت بعد از **//** یا بین **/*** ***/** به عنوان عبارت توضیحی تلقی می‌گردد.
- ❖ از فاصله چشم‌پوشی می‌شود (جز درون کلیدواژه‌ها، شناسه، عملگر و اعداد)

واژه‌نامه

case sensitive

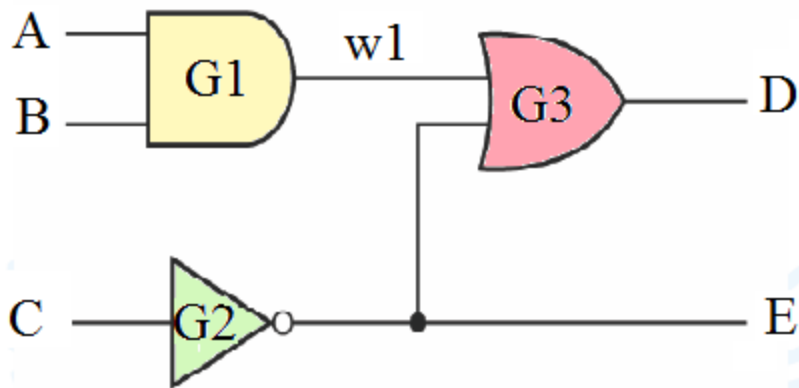
حساس به اندازه

comment

عبارت توضیحی

مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output      D,E;  
input      A,B,C;  
wire       w1;  
  
and        G1(w1,A,B);  
not        G2(E,C);  
or         G3(D,w1,E);  
endmodule
```



مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output      D,E;  
input      A,B,C;  
wire       w1;  
  
and        G1(w1,A,B);  
not        G2(E,C);  
or         G3(D,w1,E);  
endmodule
```

کلیدواژه‌ها:

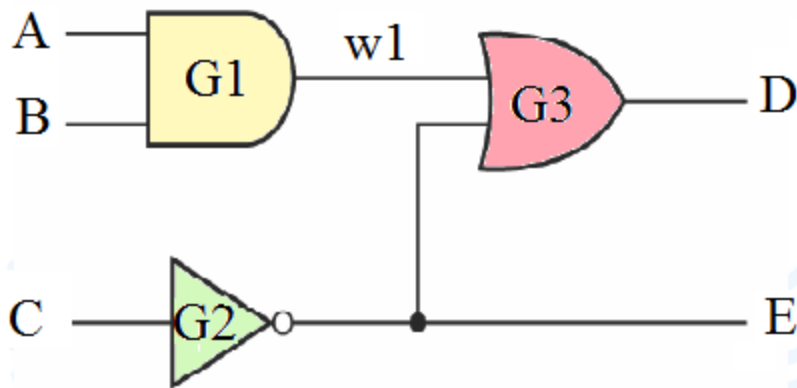
حروف کوچک ✓

از پیش تعریف شده هستند. ✓

حدود ۱۰۰ مورد ✓

اگرچه در متن کتاب با فونت ضخیم چاپ ✓

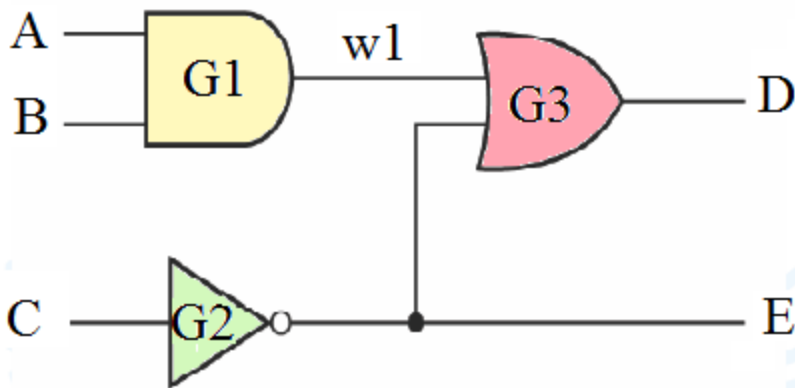
شده است ولی ضرورتی به آن نیست.



مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output          D,E;  
input           A,B,C;  
wire            w1;  
  
and             G1(w1,A,B);  
not            G2(E,C);  
or             G3(D,w1,E);  
endmodule
```

اعلان ماژول:



واژه‌نامه:

declaration

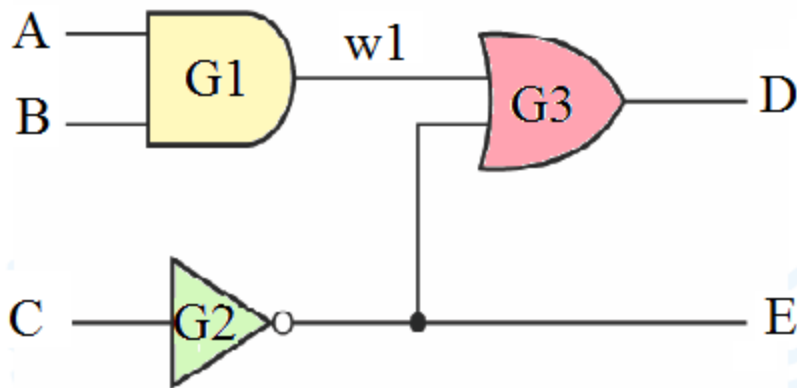
اعلان

مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output D,E;  
input A,B,C;  
wire w1;  
  
and G1(w1,A,B);  
not G2(E,C);  
or G3(D,w1,E);  
endmodule
```

شناسه‌ها:

- ✓ شامل حروف کوچک و بزرگ، اعداد و زیرخط
- ✓ باید با حروف یا زیرخط آغاز گردد نه با عدد.
- ✓ حساس به اندازه (**case-sensitive**)
- ✓ بهتر است نامی بامسمی باشد.



واژه‌نامه:

identifier

شناسه

underscore

زیرخط

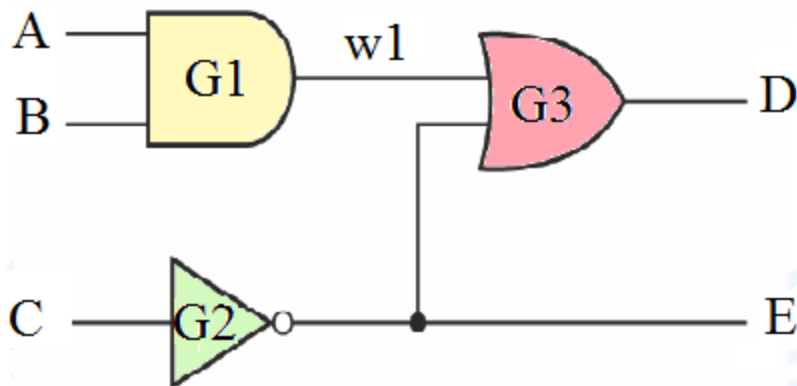
مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output      D,E;  
input      A,B,C;  
wire       w1;  
  
and        G1(w1,A,B);  
not        G2(E,C);  
or         G3(D,w1,E);  
endmodule
```

لیست درگاه‌های ماژول:

درون پراگتزر ✓

با کاما از هم جدا می‌شوند. ✓



واژه‌نامه:

port

درگاه

مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);
```

```
output D,E;
```

```
input A,B,C;
```

```
wire w1;
```

```
and G1(w1,A,B);
```

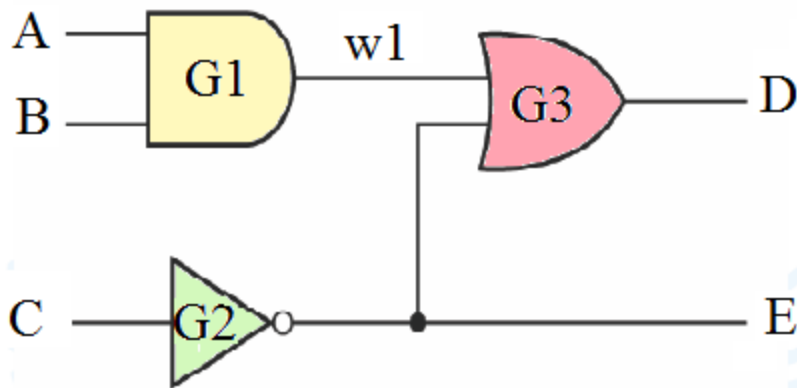
```
not G2(E,C);
```

```
or G3(D,w1,E);
```

```
endmodule
```

: Semi-colon

✓ آخرین کاراکتر تمام خطوط جز خط **.endmodule**



مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);
```

```
output D,E;
```

```
input A,B,C;
```

```
wire w1;
```

```
and G1(w1,A,B);
```

```
not G2(E,C);
```

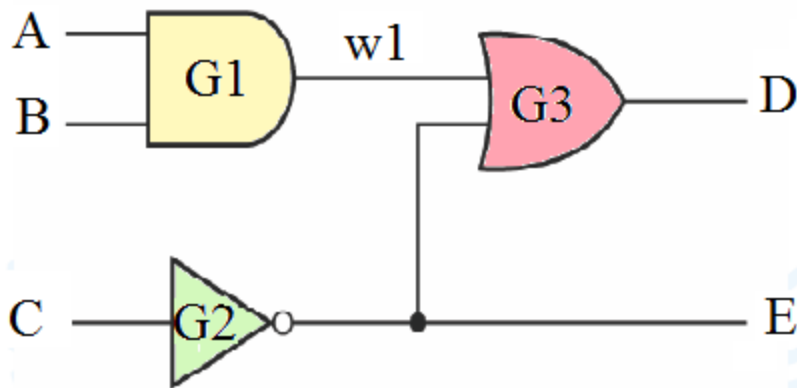
```
or G3(D,w1,E);
```

```
endmodule
```

لیست درگاه‌های خروجی:

مشخص کردن درگاه‌های خروجی از لیست ✓

درگاه‌ها.

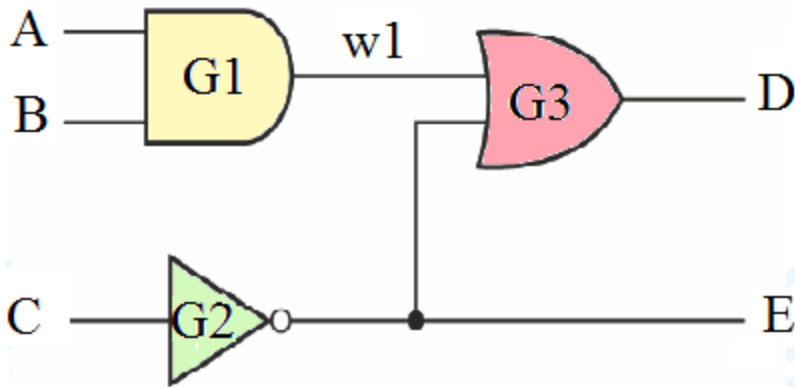


مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output          D,E;  
input          A,B,C;  
wire           w1;  
  
and            G1(w1,A,B);  
not           G2(E,C);  
or            G3(D,w1,E);  
endmodule
```

لیست درگاه‌های ورودی:

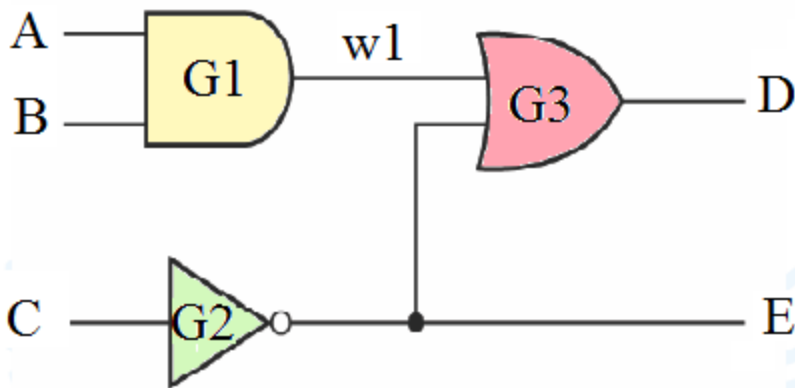
مشخص کردن درگاه‌های ورودی از لیست ✓
درگاه‌ها.



مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output          D,E;  
input           A,B,C;  
wire            w1;  
  
and             G1(w1,A,B);  
not            G2(E,C);  
or             G3(D,w1,E);  
endmodule
```

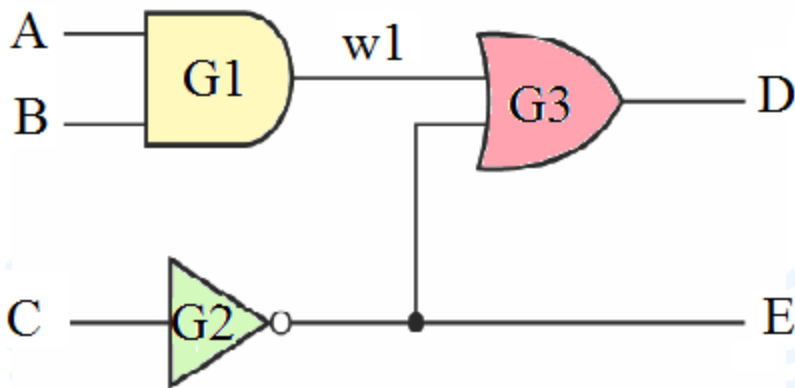
لیست اتصالات درونی:



مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
  output      D,E;  
  input       A,B,C;  
  wire        w1;  
  
  and         G1(w1,A,B);  
  not         G2(E,C);  
  or          G3(D,w1,E);  
endmodule
```

اعلان، نمونه‌سازی و درگاه‌های گیت: باید ابتدا درگاه خروجی و بعد درگاه‌های ورودی ذکر گردد. (برای لیست درگاه‌های ماژول چنین الزامی نیست).



واژه‌نامه:

declaration

instantiation

اعلان

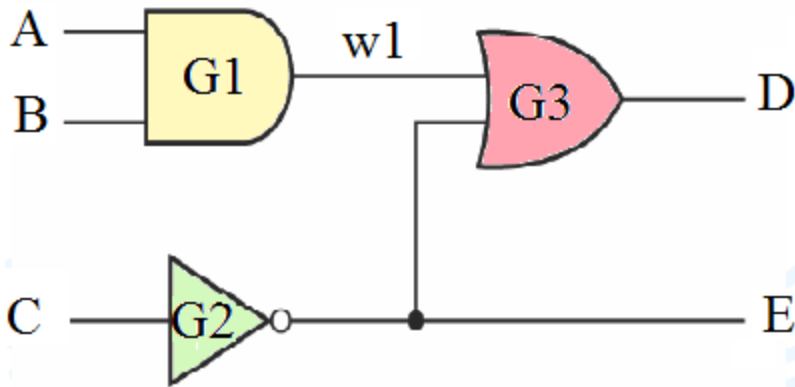
نمونه‌سازی

مثال ساده‌ای از Verilog HDL

```
module Simple_Circuit(A, B, C, D, E);  
output          D,E;  
input           A,B,C;  
wire            w1;  
  
and             G1(w1,A,B);  
not            G2(E,C);  
or             G3(D,w1,E);  
endmodule
```

کلیدواژه‌ی خاتمه‌ی ماژول:

بدون semi-colon در انتهای عبارت. ✓



فهرست مطالب

- مقدمه‌ای بر حداقل سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم ضرب حاصل جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog

