مبحث سوم

حداقل‌سازی سطح گیت

Presented by A. Maleki Fall Semester, 2021

فهرست مطالب

- مقدمه‌ای بر حداقل‌سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- 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؟

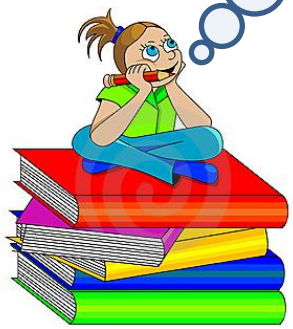
مفهوم و معیارهای حداقل‌سازی؟

اهمیت روش نقشه برای حداقل‌سازی؟

ساده‌ترین عبارت منحصر به فرد نیست.

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

- حداقل جملات
- کمترین لیترال
- کمترین سطوح



فهرست مطالب

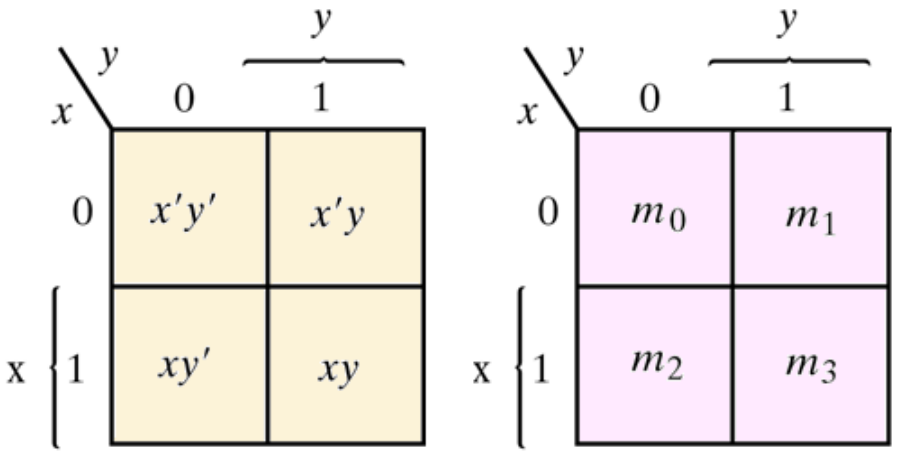
- ☑ مقدمه‌ای بر حداقل‌سازی
- ❑ روش نقشه
- ❑ سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ❑ ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- ❑ حالت‌های بی‌اهمیت
- ❑ پیاده‌سازی با NAND و NOR
- ❑ دیگر پیاده‌سازی‌های دو سطحی
- ❑ تابع OR انحصاری
- ❑ زبان توصیف سخت‌افزار Verilog

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



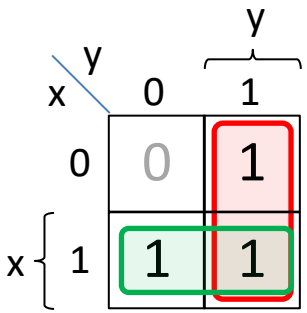
در سطرها و ستون‌های با برچسب صفر، متغیر به صورت مکمل ظاهر می گردد و . . .

مثال: ساده‌سازی با نقشه دو متغیره

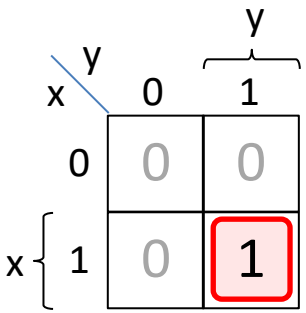
تابع های منطقی زیر را با استفاده از نقشه ساده کنید.

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

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

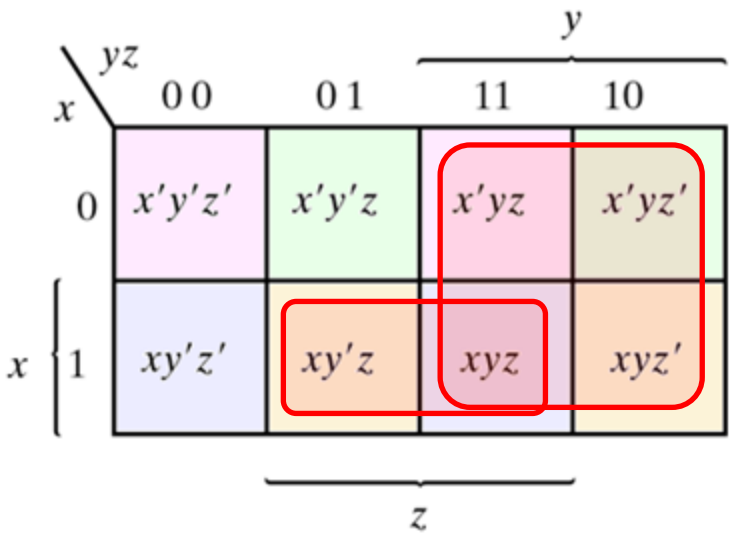


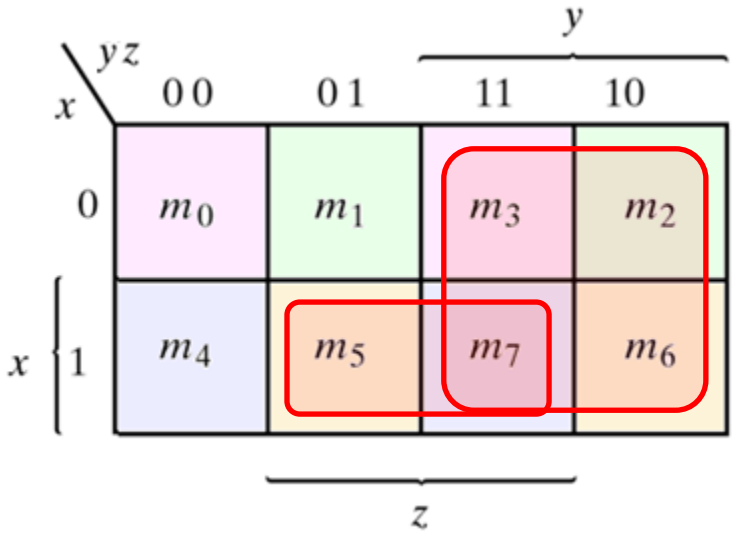
$F_1(x,y) = y + x$



$F_2(x,y) = xy$

نقشه‌ی سه متغیره





Gray Code
00
01
11
10

چرا چیدمان خانه ها اینچنین انتخاب شده است؟

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

مثال:

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

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

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

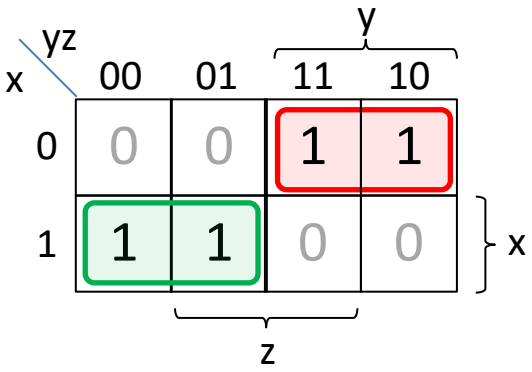
تابع های زیر را ساده کنید.

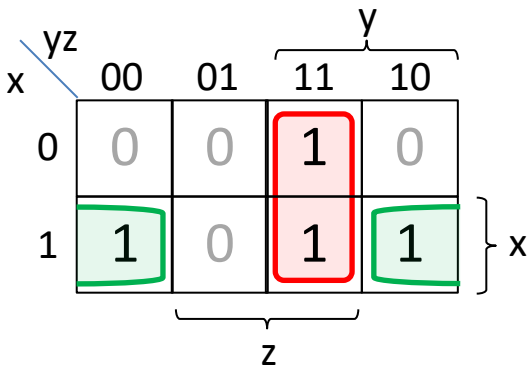
$F_1(x,y,z) = \sum (2,3,4,5)$

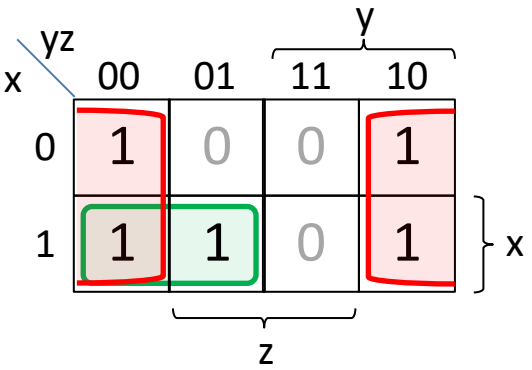
$F_2(x,y,z) = \sum (3,4,6,7)$

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

اطلاعات را وارد نقشه نمایید.







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

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

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

درستی ساده‌سازی خانه‌های سبز را بررسی کنید.

درستی ساده‌سازی خانه‌های قرمز را بررسی کنید.

مثال: نقشه‌ی سه‌متغیره

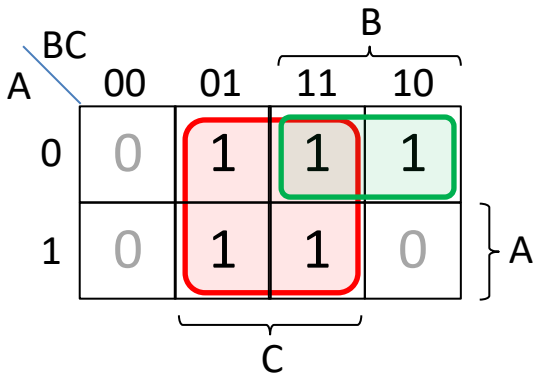
تابع روبرو را در نظر بگیرید.

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

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

ب: تابع را ساده نمایید.

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



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

نقشه‌ی چهار متغیره

wx \ yz	00	01	11	10
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	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

y, x, w, z

wx \ yz	00	01	11	10
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}

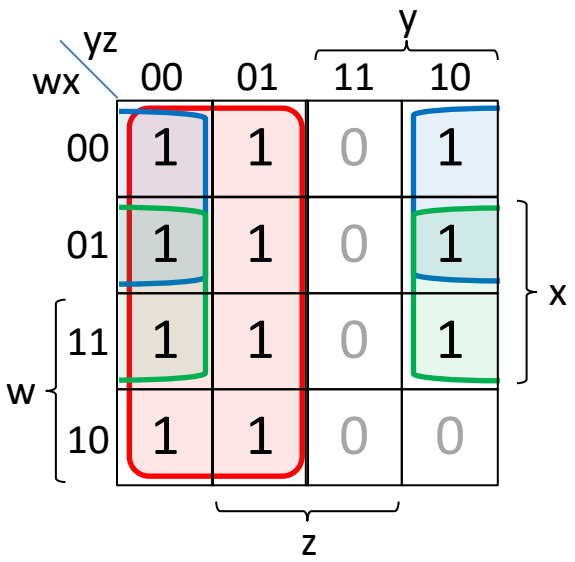
y, x, w, z

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.

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



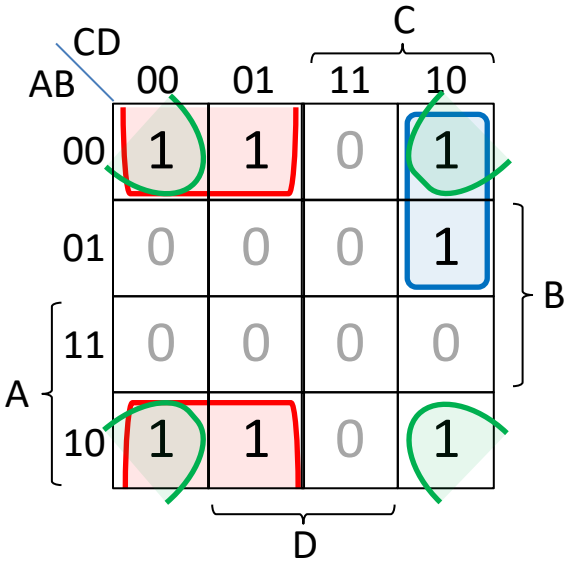
$F(w,x,y,z) = y' + w'z' + xz'$

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.



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

نقشه‌ی پنج متغیره

$A = 0$

BC \ DE	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

B (11, 10) — C (01, 00) — D (11, 10) — E (00, 01)

$A = 1$

BC \ DE	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

B (11, 10) — C (01, 00) — D (11, 10) — E (00, 01)

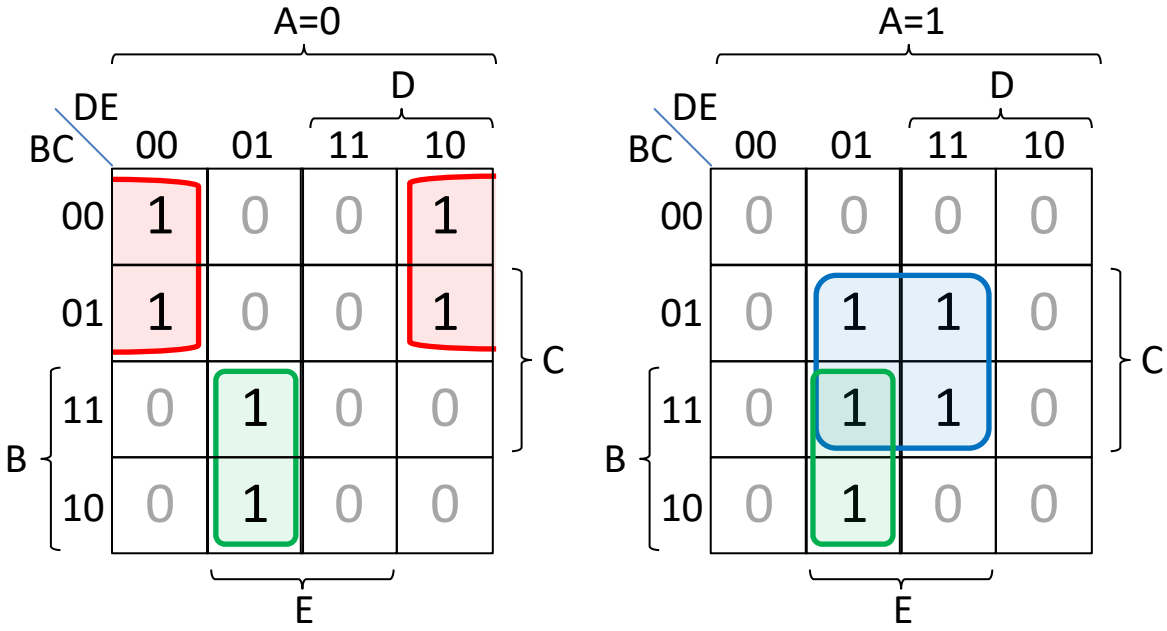
توجه کنید که خانه های m_0 ، m_{16} و . . . مجاور هم هستند.

مثال: ساده‌سازی با نقشه‌ی پنج متغیره

تابع روبرو را ساده کنید.

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

اطلاعات را وارد نقشه نمایید.



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

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: $K(w,x,y,z) = F + G$

اطلاعات تابع‌های F و G را وارد نقشه نمایید.

نقشه‌ی تابع‌های H و K چه ارتباطی با نقشه‌ی تابع‌های F و G دارد؟

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

x: 01, 11 — w: 11, 10 — z: 11, 10 — y: 11, 10

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

x: 01, 11 — w: 11, 10 — z: 11, 10 — y: 11, 10

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

x: 01, 11 — w: 11, 10 — z: 11, 10 — y: 11, 10

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

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

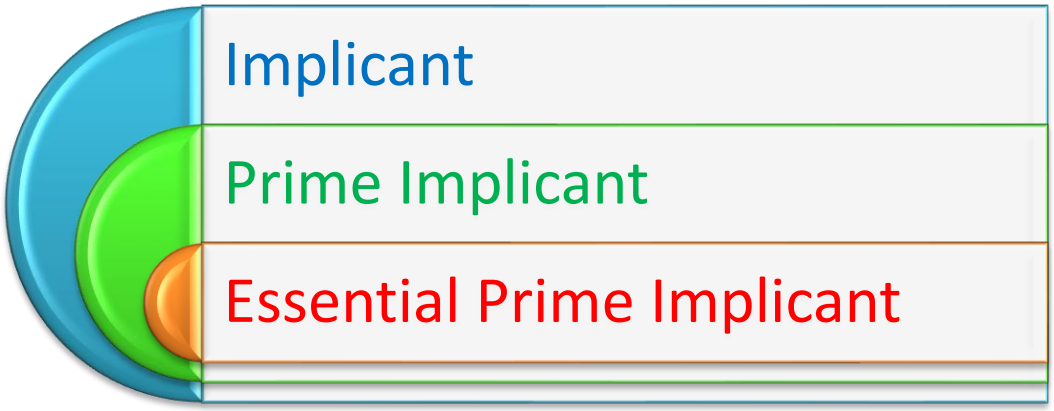
x: 01, 11 — w: 11, 10 — z: 11, 10 — y: 11, 10

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

فهرست مطالب

- ☑ مقدمه‌ای بر حداقل‌سازی
- ☑ روش نقشه
- ☐ سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ☐ ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- ☐ حالت‌های بی‌اهمیت
- ☐ پیاده‌سازی با NAND و NOR
- ☐ دیگر پیاده‌سازی‌های دو سطحی
- ☐ تابع OR انحصاری
- ☐ زبان توصیف سخت‌افزار Verilog

سیستماتیک کردن انتخاب ترکیب‌ها در نقشه



جمله ی حاصلضربی که از ترکیب بیشترین خانه های مجاور هم در نقشه حاصل می گردد.

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

مثال: مفهوم موجب اصلی ضروری

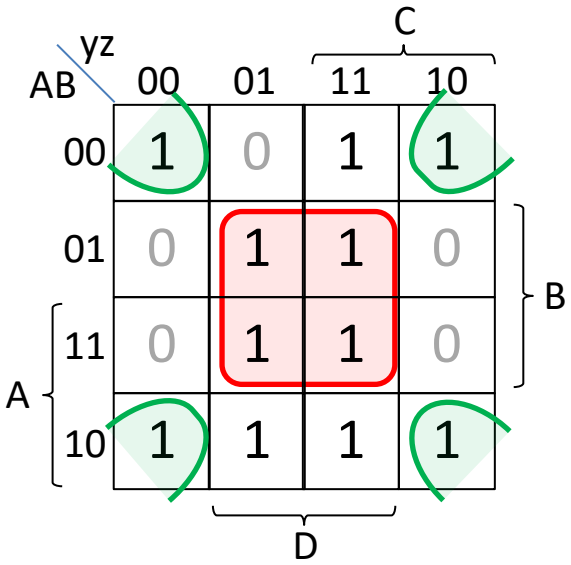
$F(A,B,C,D)=\sum(0,2,3,5,7,8,9,10,11,13,15)$

تابع روبرو را در نظر بگیرید.

الف: موجب های اصلی ضروری تابع را مشخص نمایید.

ب: گزینه های مختلف موجب های اصلی برای پوشش دادن دیگر minterm‌ها را ذکر کنید.

ج: گزینه های مختلف ساده سازی را لیست نمایید.



Essential Prime Implicant: B'D' , BD

چرا این دو ترکیب، ترکیب اصلی ضروری است؟

Prime Implicant: m3 : CD , B'C

m9 : AD , AB'

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

منظورم این بود که گفتم:
«ساده‌ترین عبارت منحصر به فرد نیست».

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

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

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

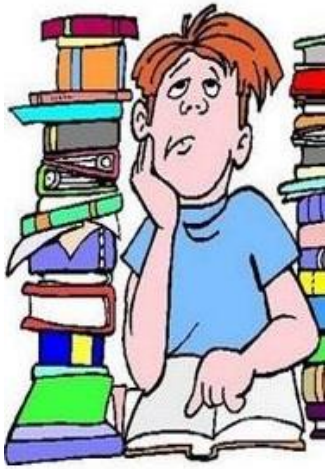
$= B'D' + BD + B'C + AB'$

فهرست مطالب

- ☑ مقدمه‌ای بر حداقل‌سازی
- ☑ روش نقشه
- ☑ سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ☐ ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- ☐ حالت‌های بی‌اهمیت
- ☐ پیاده‌سازی با NAND و NOR
- ☐ دیگر پیاده‌سازی‌های دو سطحی
- ☐ تابع OR انحصاری
- ☐ زبان توصیف سخت‌افزار Verilog

ساده‌سازی به فرم ضرب حاصل‌جمع‌ها

برای ساده‌سازی به فرم ضرب حاصل‌جمع‌ها 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 حاصل‌جمع‌ها بازنویسی می‌کنیم.

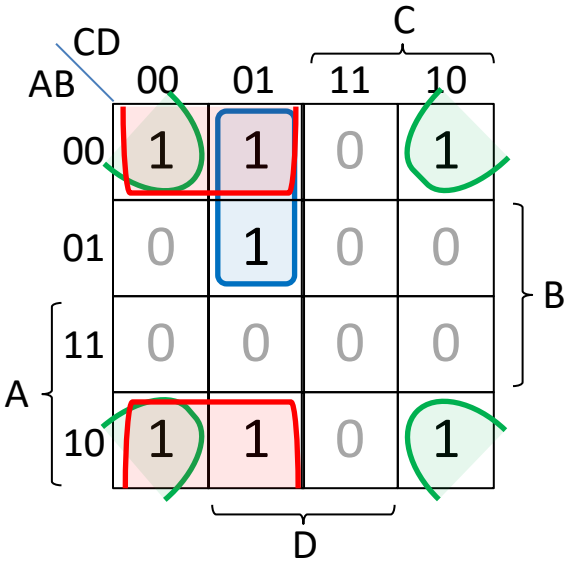
مثال: ساده‌سازی به فرم ضرب حاصل‌جمع‌ها

تابع روبرو را به صورت‌های خواسته شده ساده کنید.

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

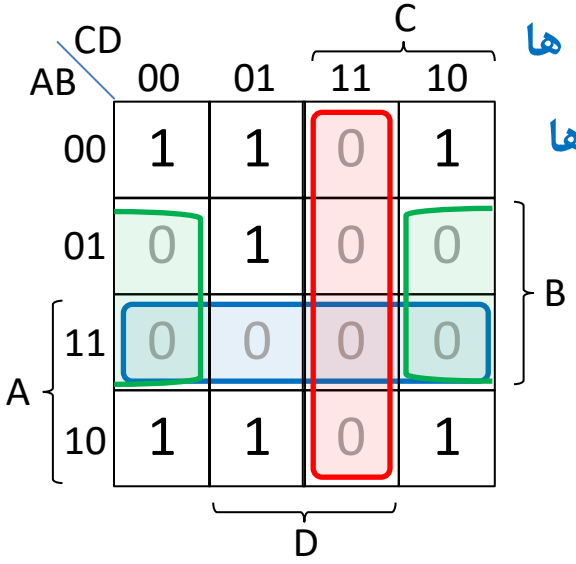
شماتیک مداری ساده شده را نیز برای هر حالت ترسیم نمایید.

الف: به صورت جمع حاصل ضرب ها



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

ب: به صورت ضرب حاصل جمع ها



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

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

مثال: ساده‌سازی به فرم ضرب حاصل‌جمع‌ها

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

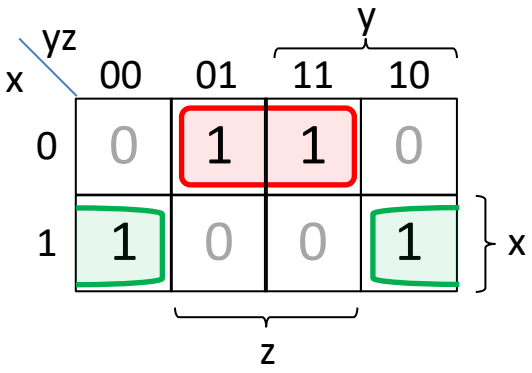
تابع روبرو را به صورت‌های خواسته شده ساده کنید.

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

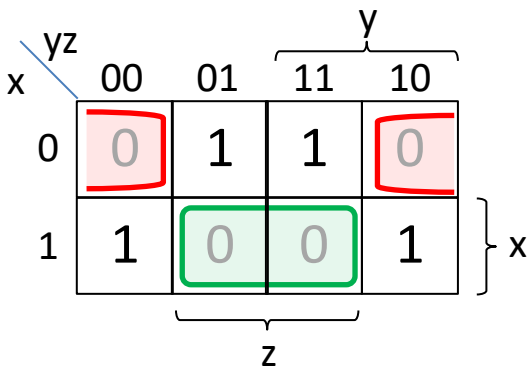
الف: به صورت جمع حاصل‌ضرب‌ها

اطلاعات را وارد نقشه نمایید.

ب: به صورت ضرب حاصل‌جمع‌ها



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



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

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

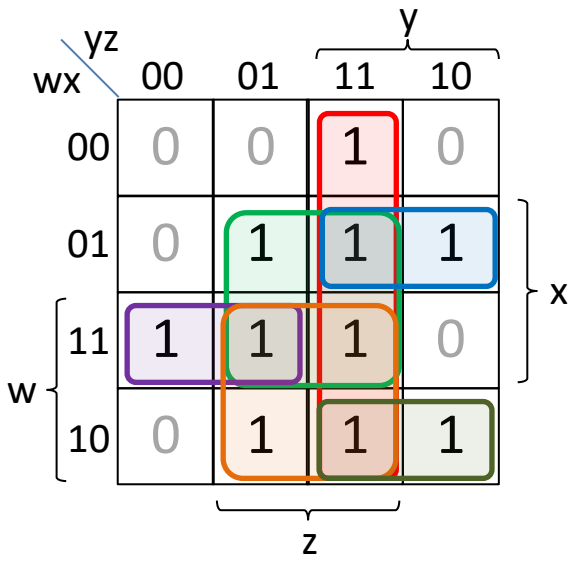
مثال: وارد کردن تابعی به فرم ضرب حاصل‌جمع‌ها در نقشه

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.

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



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

فهرست مطالب

- [x] مقدمه‌ای بر حداقل‌سازی
- [x] روش نقشه
- [x] سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- [x] ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- [ ] حالت‌های بی‌اهمیت
- [ ] پیاده‌سازی با NAND و NOR
- [ ] دیگر پیاده‌سازی‌های دو سطحی
- [ ] تابع OR انحصاری
- [ ] زبان توصیف سخت‌افزار Verilog

حالت‌های بی‌اهمیت

تابع غیرکامل (incompletely specified function)

تابعی که خروجی آن به ازای برخی‌ترکیب های ورودی نامشخص باشد.

حالت‌های بی‌اهمیت

mintermهای مشخص نشده در تابع غیرکامل

مثال: کاربرد حالت‌های بی‌اهمیت در ساده‌سازی

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

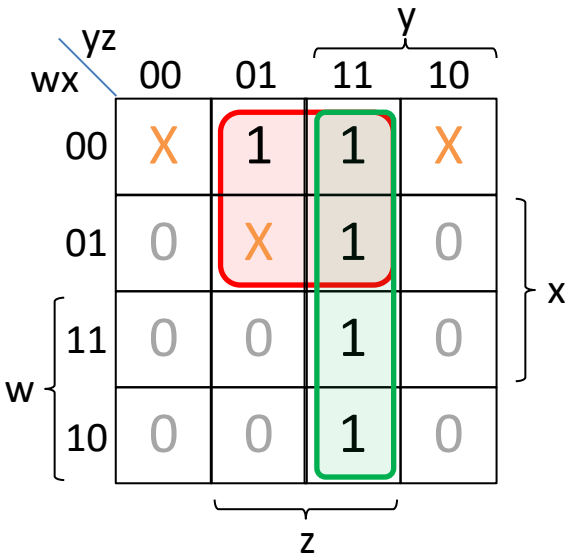
$F(w,x,y,z)=\sum(1,3,7,11,15)$

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

الف: به صورت جمع حاصل‌ضرب‌ها

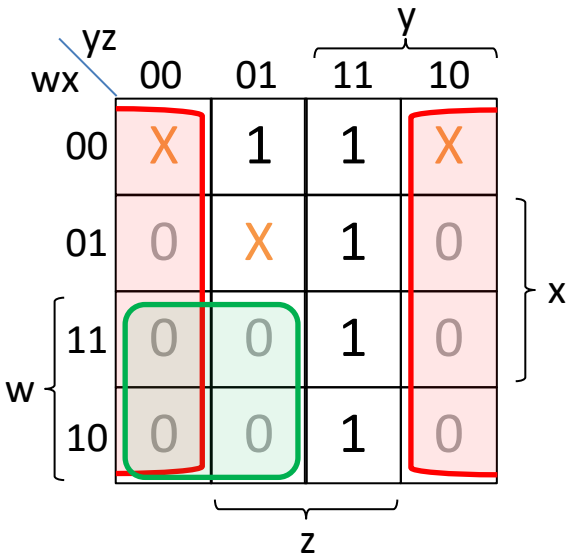
ب: به صورت ضرب حاصل‌جمع‌ها

اطلاعات را وارد نقشه نمایید.



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

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



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

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

استفاده از حالت‌های بی‌اهمیت چه فایده‌ای دارد؟

فهرست مطالب

- ☑ مقدمه‌ای بر حداقل‌سازی
- ☑ روش نقشه
- ☑ سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ☑ ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- ☑ حالت‌های بی‌اهمیت
- ☐ پیاده‌سازی با NAND و NOR
- ☐ دیگر پیاده‌سازی‌های دو سطحی
- ☐ تابع OR انحصاری
- ☐ زبان توصیف سخت‌افزار Verilog

عناوین این بخش:

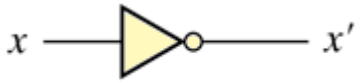
- معرفی گیت یونیورسال
- تعبیری جدید از قضیه‌ی دمورگان
- پیاده‌سازی با گیت‌های NAND
- پیاده‌سازی با گیت‌های NOR
- مدارهای NAND و NOR چندسطحی

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.

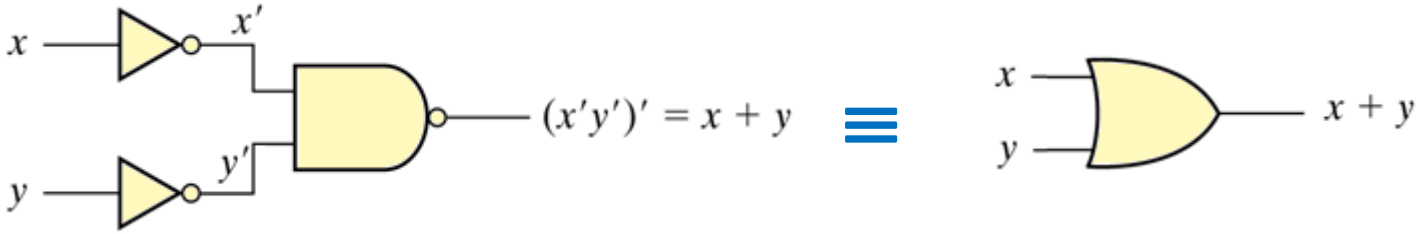
NOT:



گیت NAND یک ورودی مانند NOT عمل می‌کند.

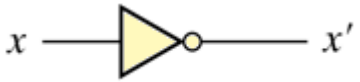
AND:

OR:



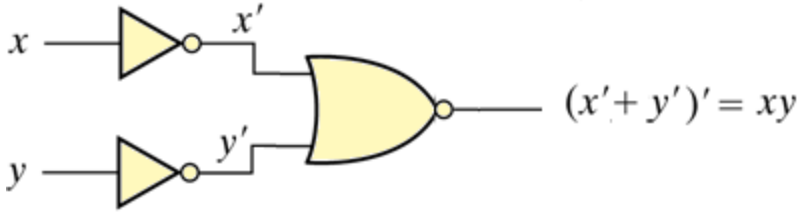
نشان دهید گیت NOR یک گیت یونیورسال است.

NOT:



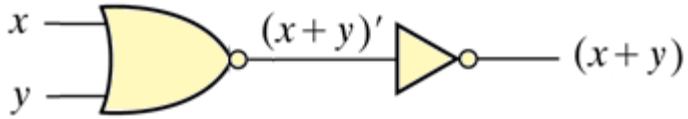
گیت NOR یک ورودی مانند NOT عمل می کند.

AND:

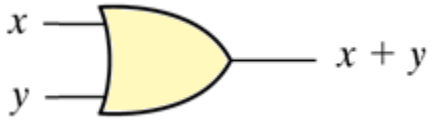


≡

OR:



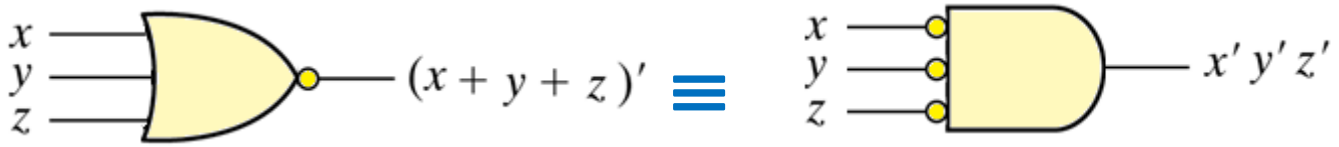
≡



تعبیری دیگر از قضیه‌ی DeMorgan

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

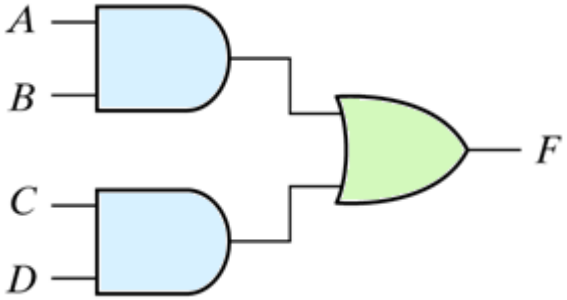
(x+y+z)' = x'y'z'



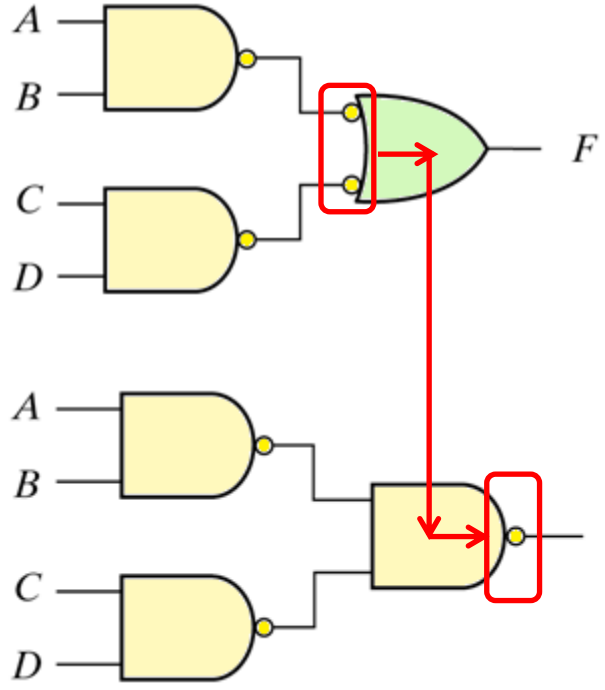
حباب NOT را می‌توان با تغییر عمل گیت، از گیت عبور داد!

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 حاصل‌ضرب‌ها (آرایه‌ی AND-OR) پیاده‌سازی می‌کنیم.



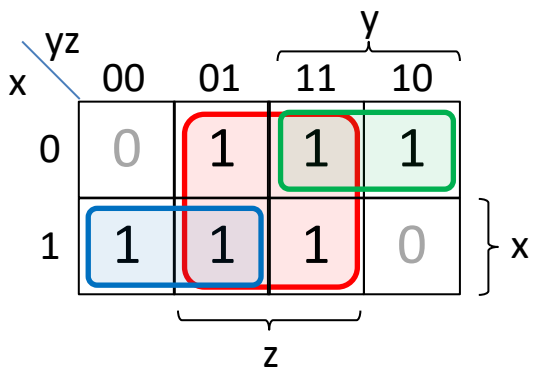
در خروجی گیت‌های AND و ورودی‌های گیت‌های OR حباب اضافه می‌کنیم.

حباب 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.

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

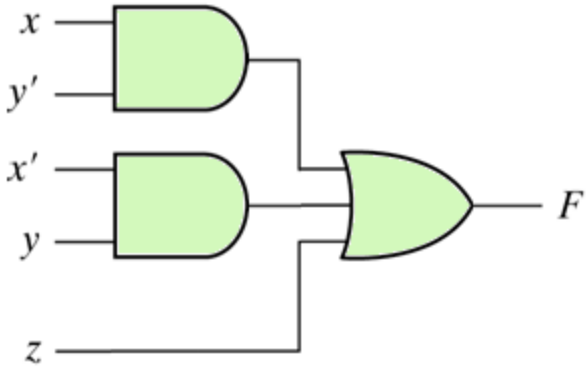


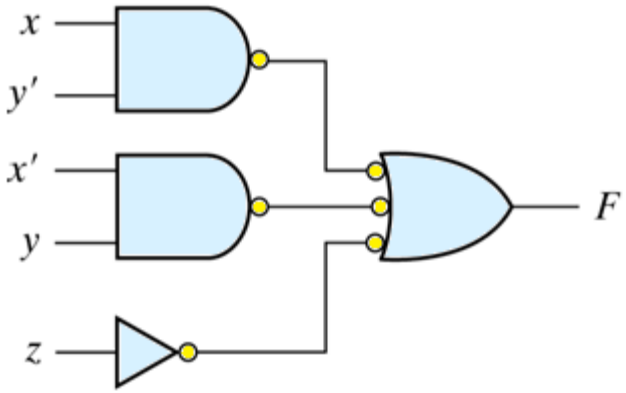
$F(x,y,z) = z + x'y + xy'$

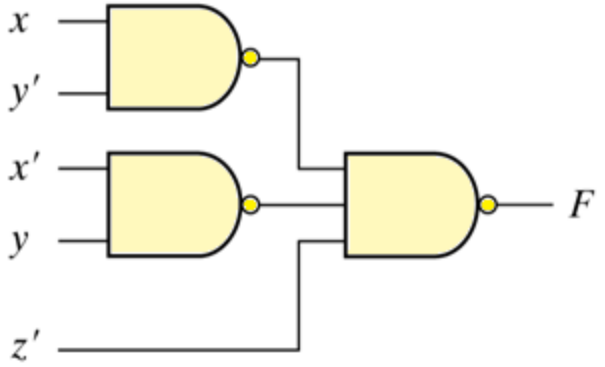
$= ((z + x'y + xy')')'$

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

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







پیاده‌سازی تابع با NOR

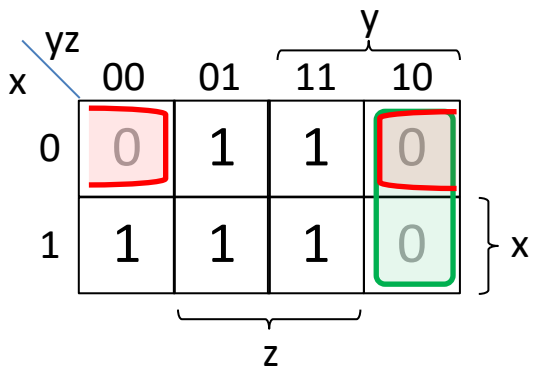
برای پیاده‌سازی تابع با گیت‌های NOR (دستیابی به ساختار NOR-NOR) :

1) با استفاده از روش نقشه، تابع را به فرم ضرب حاصل‌جمع‌ها (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.

مثال:

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

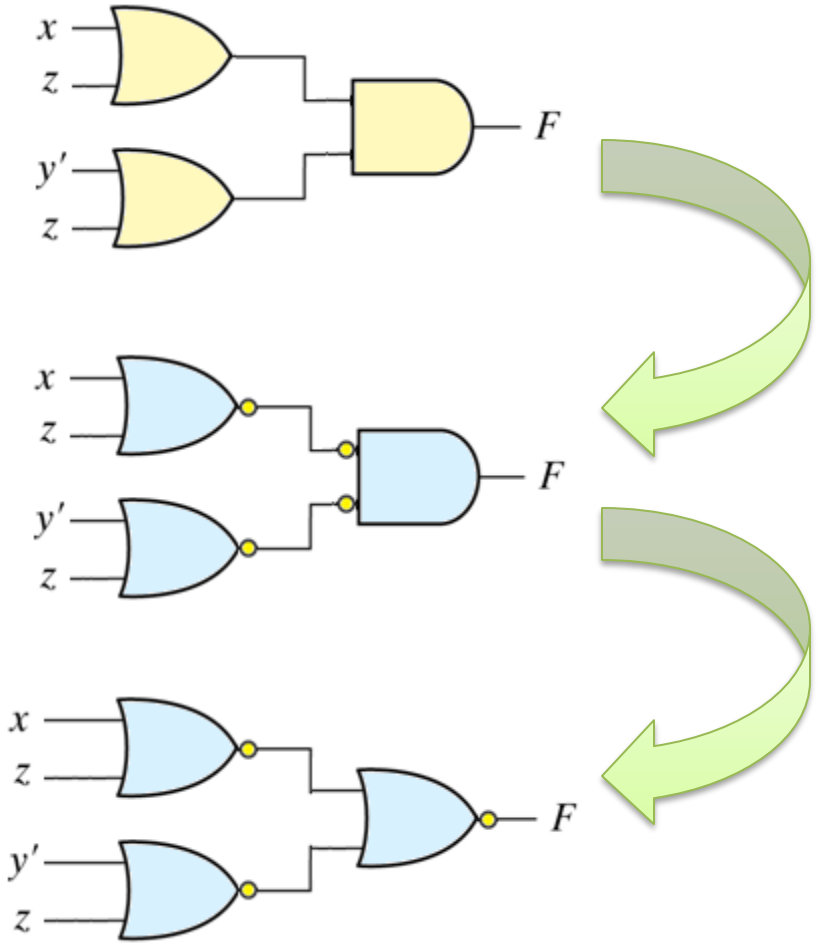
$F(x,y,z)=\sum(1,3,4,5,7)$



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

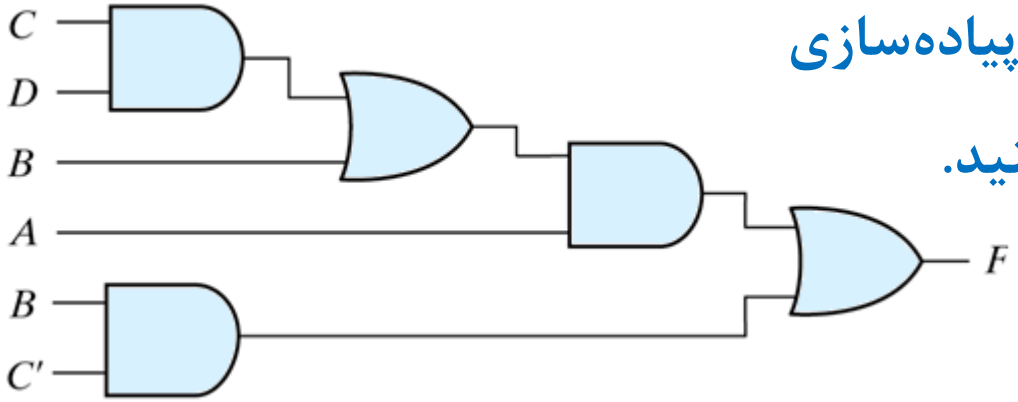
$F(x,y,z) = (x'z' + yz')'$

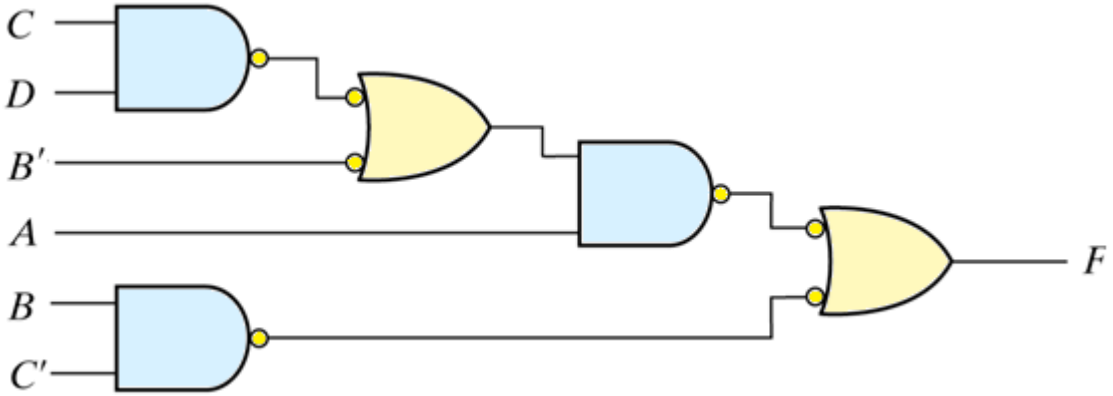
$= (x+z)(y' + z)$

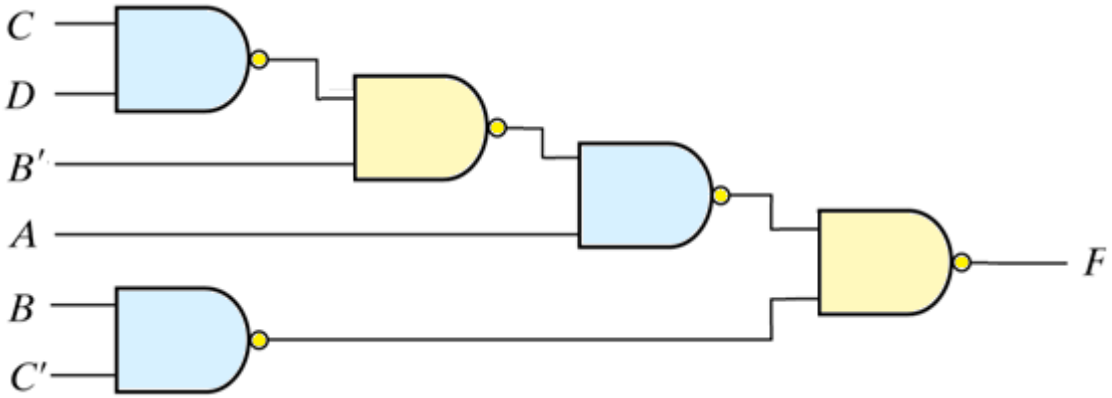


مدارهای NAND چندسطحی

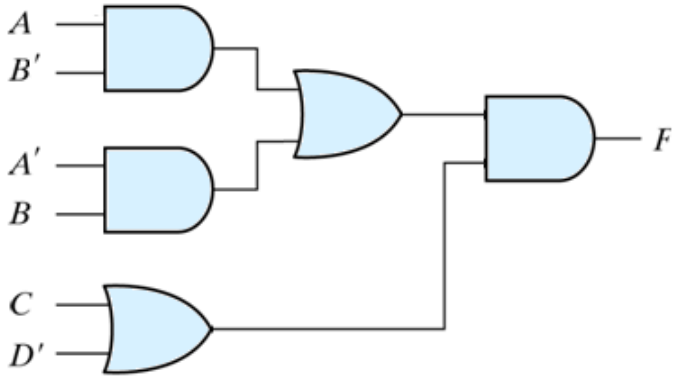
مثال: شماتیک مدار شکل روبرو را برای پیاده‌سازی فقط با گیت‌های NAND بازنمایی کنید.







مدارهای NAND و NOR چندسطحی

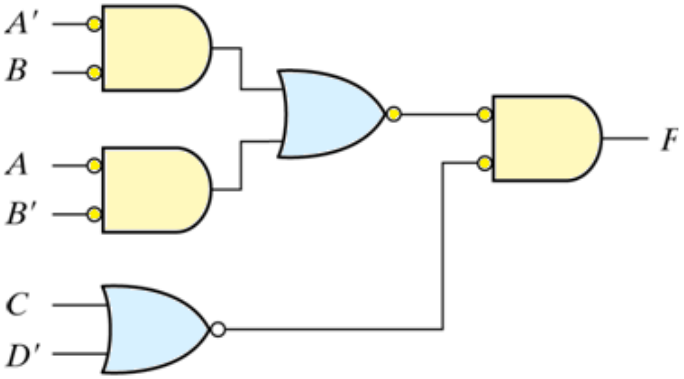


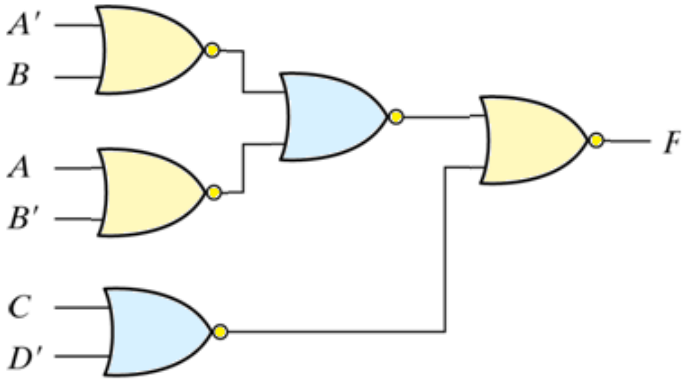
مثال: شماتیک مدار شکل روبرو را به پیاده‌سازی‌های خواسته شده تبدیل نمایید.

الف:پیاده‌سازی با NOR

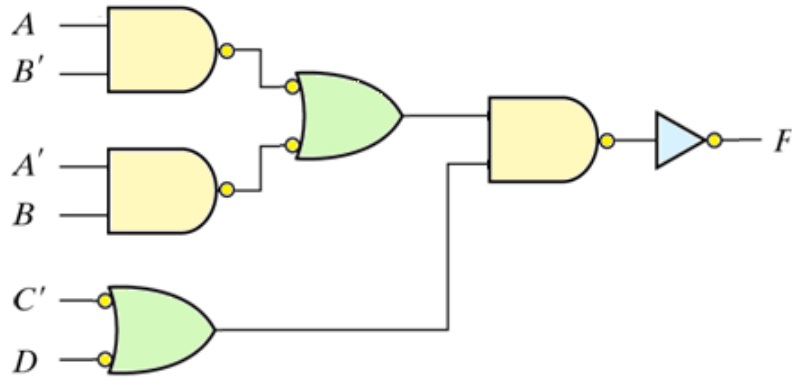
ب: پیاده‌سازی با NAND

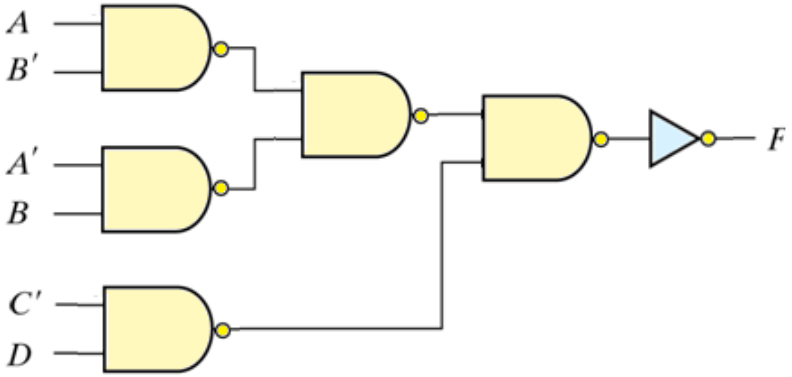
الف:





ب:





فهرست مطالب

- [x] مقدمه‌ای بر حداقل‌سازی
- [x] روش نقشه
- [x] سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- [x] ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- [x] حالت‌های بی‌اهمیت
- [x] 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~~

AND – OR 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.

≡ AND – OR – INVERT

≡ OR – AND – INVERT

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

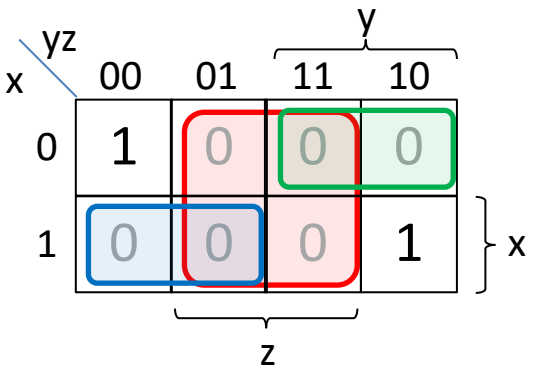
$F(x,y,z)=\sum(0,6)$

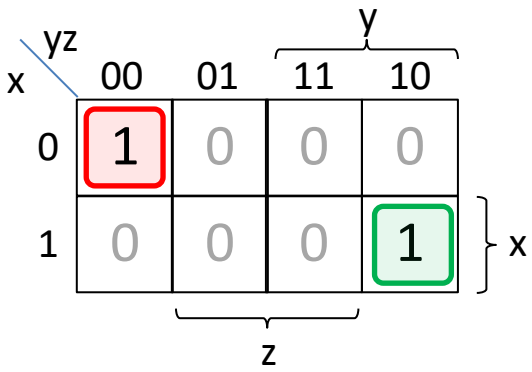
الف: AND – NOR

ب: NAND – AND

ج: OR – NAND

د: NOR – OR





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

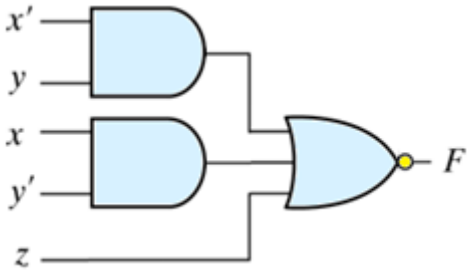
$F(x,y,z) = (z + x'y + xy')'$

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

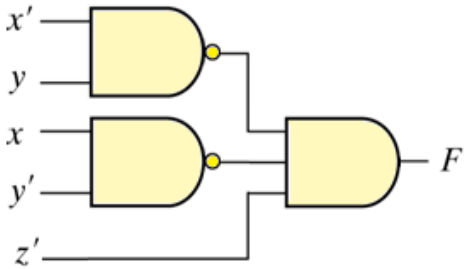
$F(x,y,z) = x'y'z' + xyz'$

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

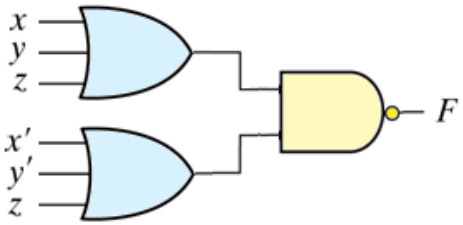
$F(x,y,z) = (x+y+z)' + (x'+y'+z)'$



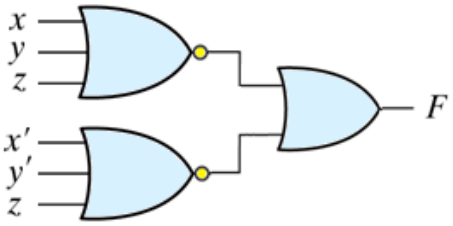
الف:



ب:



ج:



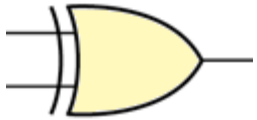
د:

فهرست مطالب

- [x] مقدمه‌ای بر حداقل‌سازی
- [x] روش نقشه
- [x] سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- [x] ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- [x] حالت‌های بی‌اهمیت
- [x] پیاده‌سازی با NAND و NOR
- [x] دیگر پیاده‌سازی‌های دو سطحی
- [ ] تابع OR انحصاری
- [ ] زبان توصیف سخت‌افزار Verilog

OR انحصاری (Exclusive OR - XOR)

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



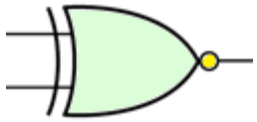
x \ y	0	1
0	0	1
1	1	0

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

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

$= xx' + xy + x'y' + yy'$

$= xy + x'y'$



x \ y	0	1
0	1	0
1	0	1

OR انحصاری (Exclusive OR - XOR)

ویژگی‌ها:

$x \oplus 0 = x$

$x \oplus 1 = x'$

از XOR می‌توان به عنوان NOT کنترل شده استفاده نمود.

$x \oplus x = 0$

$x \oplus x' = 1$

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

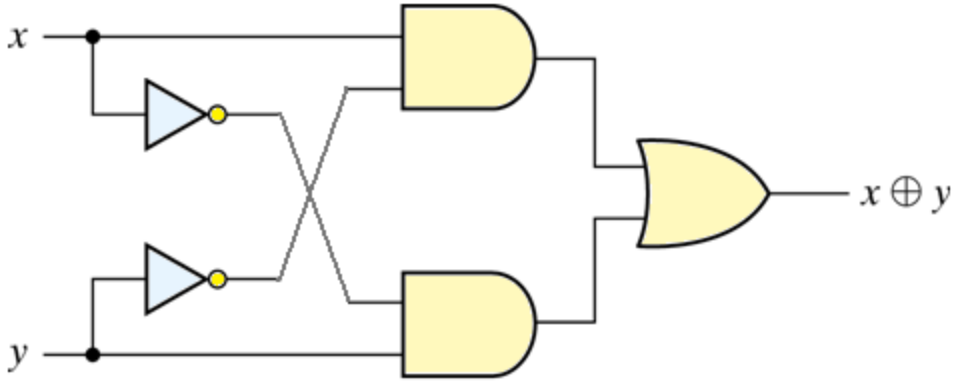
Commutative: $x \oplus y = y \oplus x$

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

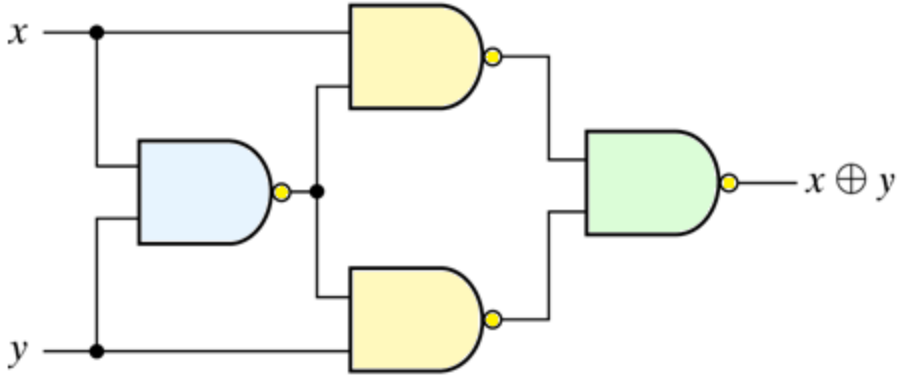
بنابراین شرط‌های گسترش ورودی‌های گیت برای XOR برقرار است.

پیاده‌سازی XOR

پیاده‌سازی با گیت‌های AND ، OR ، NOT :



پیاده‌سازی با گیت‌های NAND :



نشان دهید که شماتیک فوق، عمل گیت XOR را پیاده‌سازی می‌کند.

F = { [x (xy)']' [y (xy)']' }'

= [x (xy)'] + [y (xy)']

= [x (x' + y‘)] + [y (x' + y')]

= xx' + xy' + x'y + yy'

= xy' + x'y

= x ⊕ y

XOR سه‌متغیره

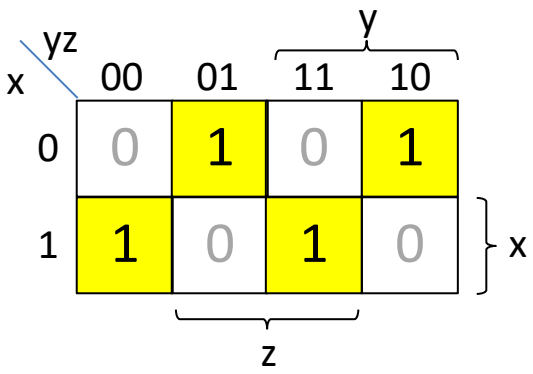
XOR:

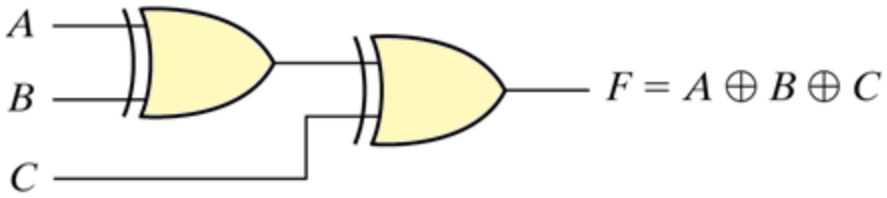
$$x \oplus y \oplus z = (x \oplus y) \oplus z = (x'y + xy') \oplus z$$

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

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

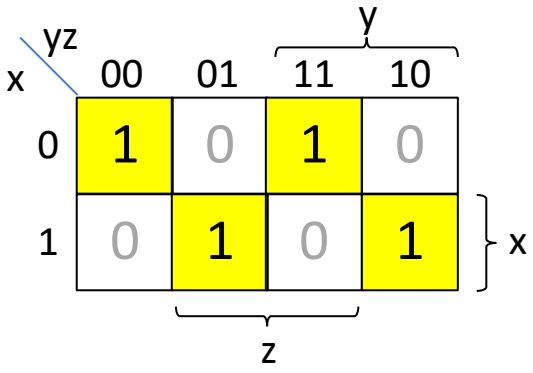
$$= \sum (1,2,4,7)$$

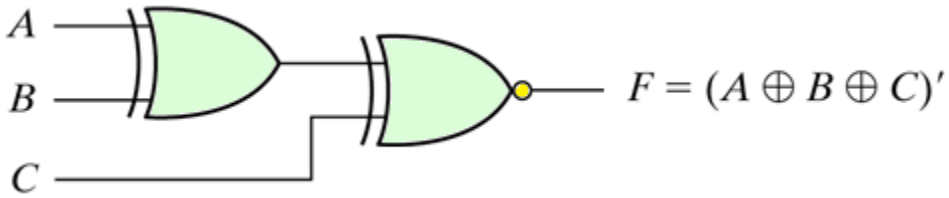




XNOR:

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





چهارمتغیره XOR

XOR:

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

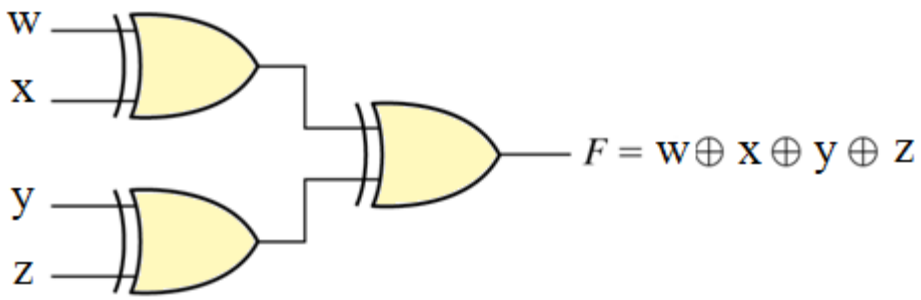
$$= (w'x + wx')(y'z' + yz) + (w'x' + wx)(y'z + yz')$$

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

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

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

y (11, 10) — x (01, 11) — w (11, 10) — z (01, 11)



تابع زوج و تابع فرد

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

XNOR چند متغیره تنها در صورتی یک می‌شود 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؟

یادآوری

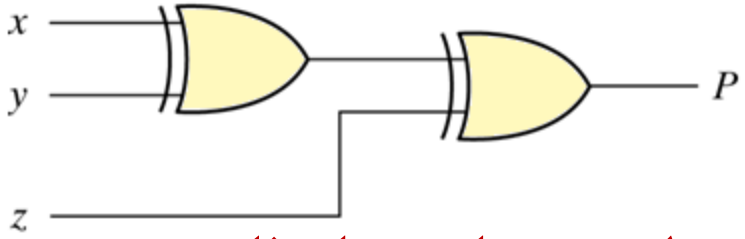
بیت توازن زوج: بیتی که به پیام اضافه می گردد تا تعداد کل یک ها زوج شود.

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

حل الف:

بیت توازن زوج	تعداد یک های پیام
۱	فرد
۰	زوج

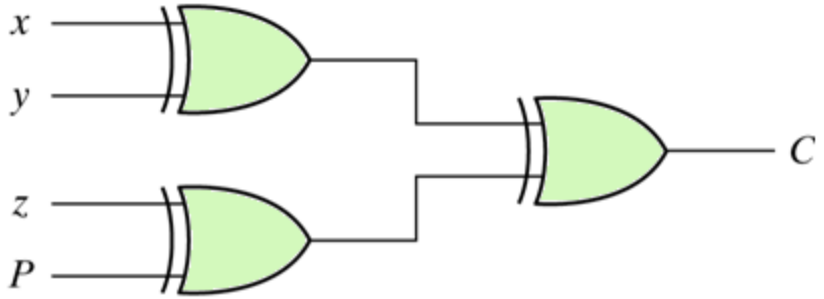
بنابراین نیاز به یک تابع فرد سه-ورودی داریم.



حل ب:

خطا	تعداد یک های پیام + توازن
۱	فرد
۰	زوج

بنابراین نیاز به یک تابع فرد چهار-ورودی داریم.



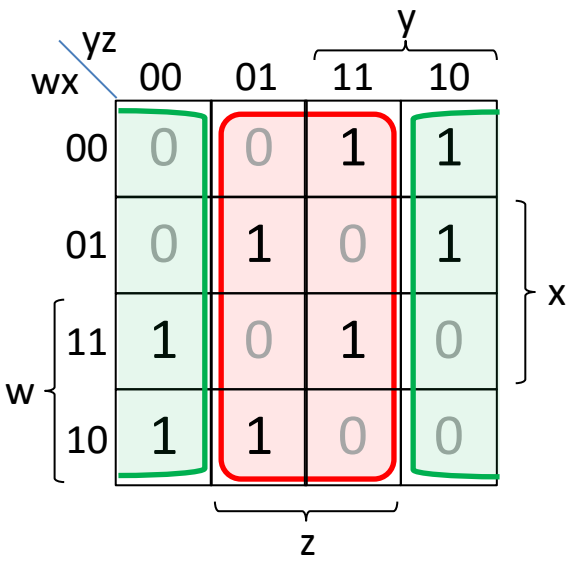
حل ج: از مدار دوم، در صورتی که P=0 شود می توان برای تولید بیت توازن نیز استفاده نمود.

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

مثال: ساده‌سازی با حضور XOR

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

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

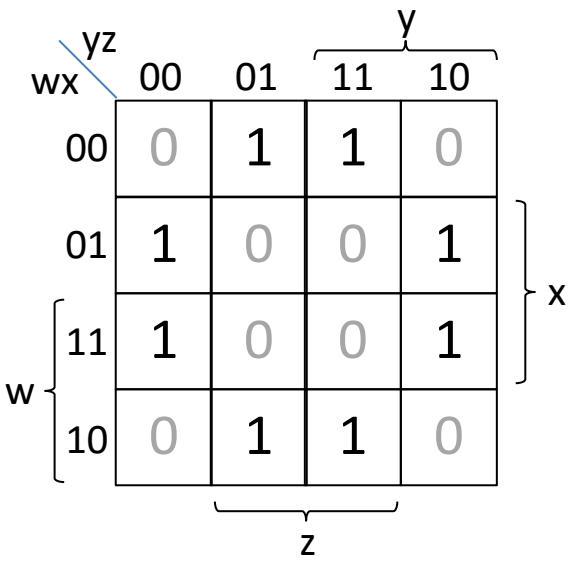


$F(w,x,y,z) = z(w \oplus x \oplus y) + z'(w \oplus y)$

مثال: ساده‌سازی با حضور XOR

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

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

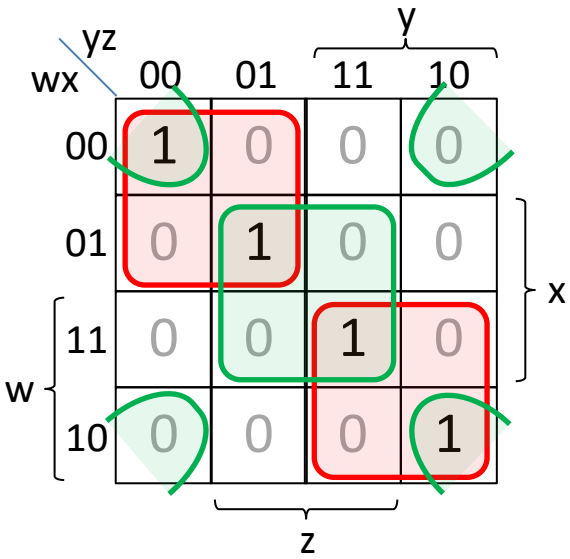


$F(w,x,y,z) = x \oplus z$

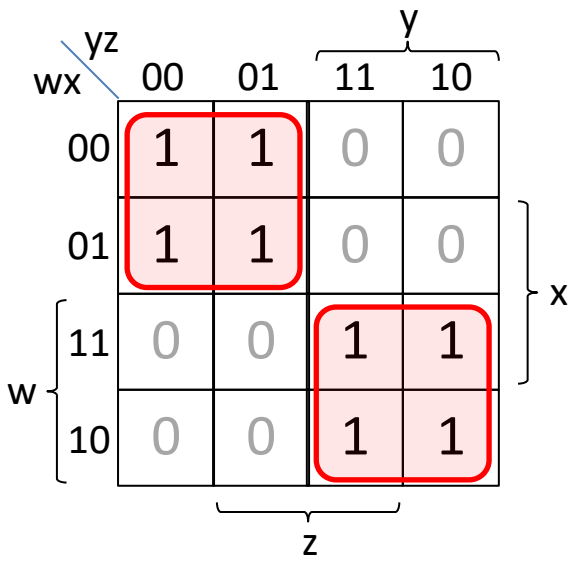
مثال: ساده‌سازی با حضور XOR

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

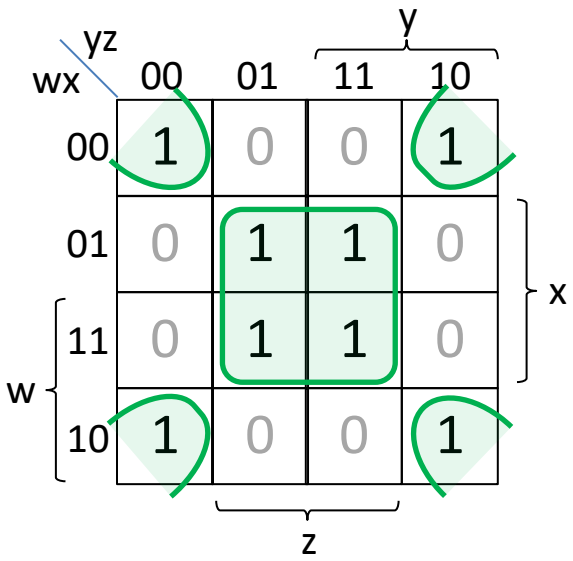
$F(w,x,y,z)=\sum(0,5,10,15)$



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



$(w \oplus y)'$



$(x \oplus z)'$

فهرست مطالب

- [x] مقدمه‌ای بر حداقل‌سازی
- [x] روش نقشه
- [x] سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- [x] ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- [x] حالت‌های بی‌اهمیت
- [x] پیاده‌سازی با NAND و NOR
- [x] دیگر پیاده‌سازی‌های دو سطحی
- [x] تابع OR انحصاری
- [ ] زبان توصیف سخت‌افزار Verilog

زبان توصیف سخت‌افزار

زبان توصیف سخت افزار Verilog

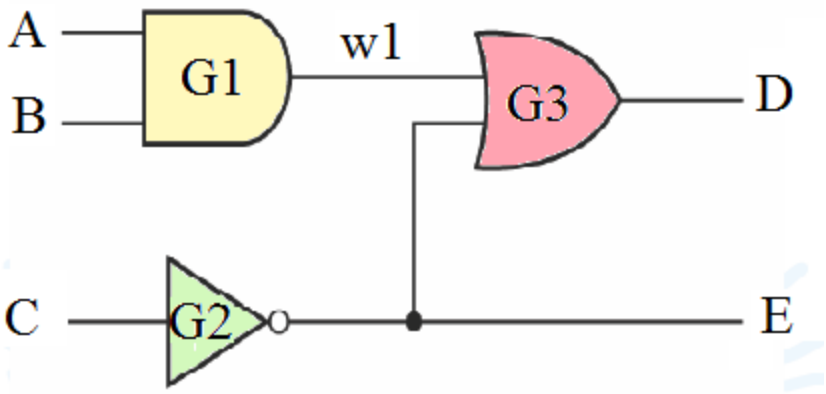
ویژگی‌ها:

- 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

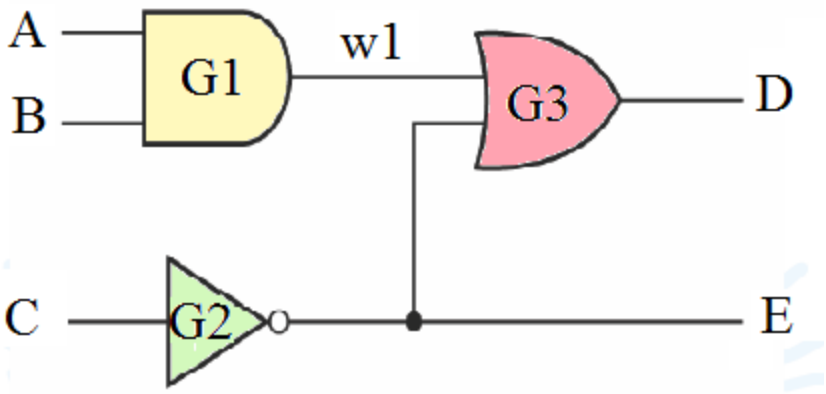
مثال ساده‌ای از Verilog HDL

```
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
```



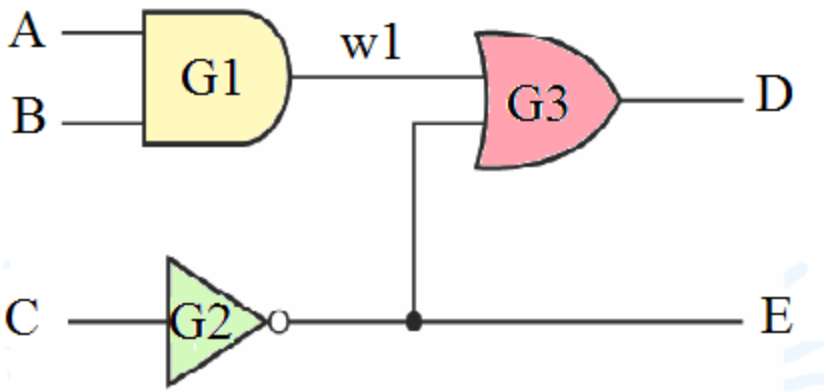
کلیدواژه‌ها:

- ✓ 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

اعلان ماژول:

```
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
```



واژه‌نامه:

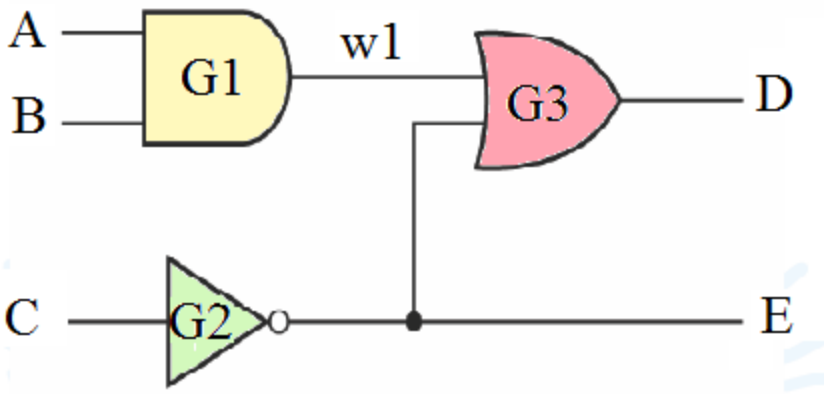
declaration	اعلان

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:

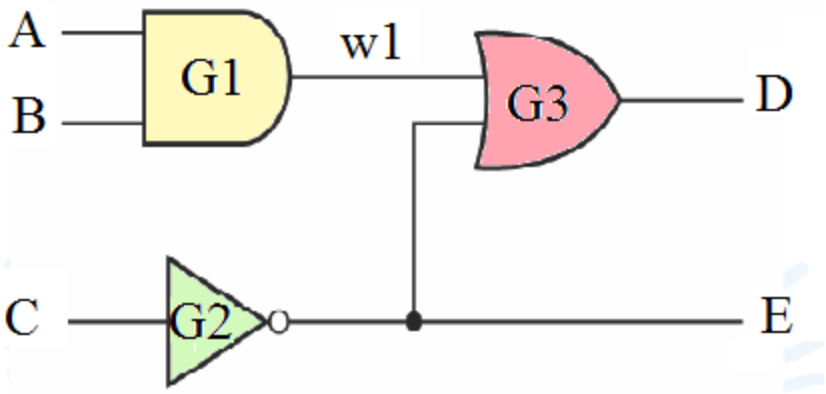
شناسه	identifier
زیرخط	underscore

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 پرانتز
- ✓ با کاما از هم جدا می شوند.



واژه‌نامه:

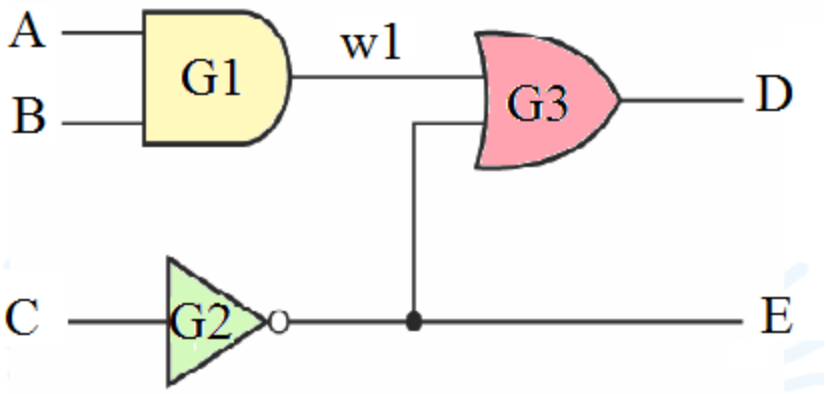
port	درگاه

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.

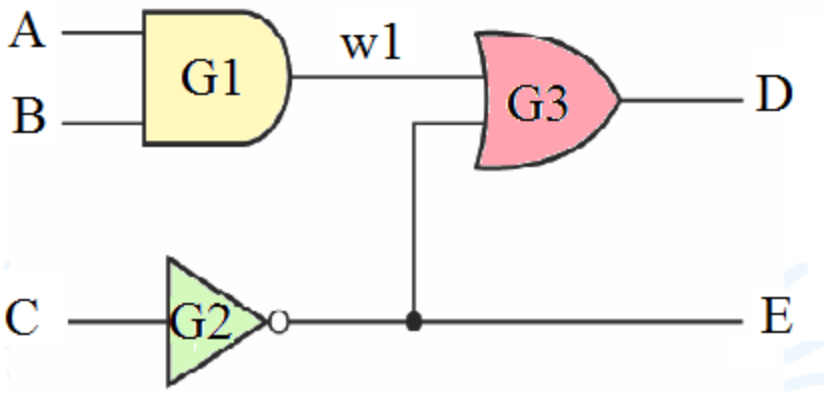


مثال ساده‌ای از Verilog HDL

```
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 درگاه های خروجی:

- ✓ مشخص‌کردن درگاه‌های خروجی از لیست درگاه‌ها.

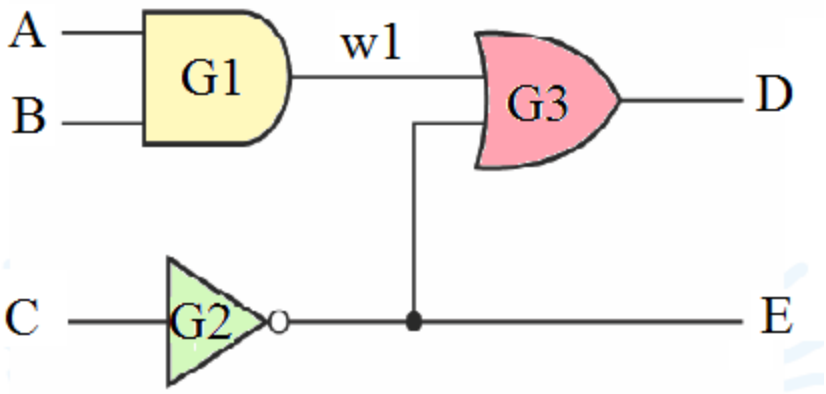


مثال ساده‌ای از Verilog HDL

```
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 درگاه های ورودی:

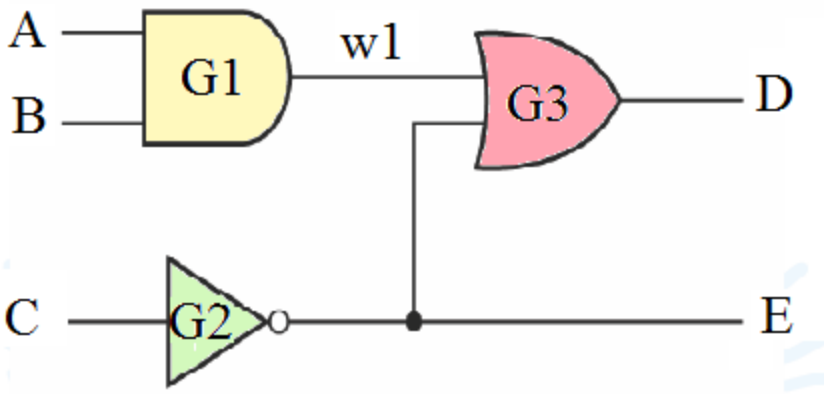
- ✓ مشخص‌کردن درگاه‌های ورودی از لیست درگاه‌ها.



مثال ساده‌ای از Verilog HDL

لیست اتصالات درونی:

```
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
```

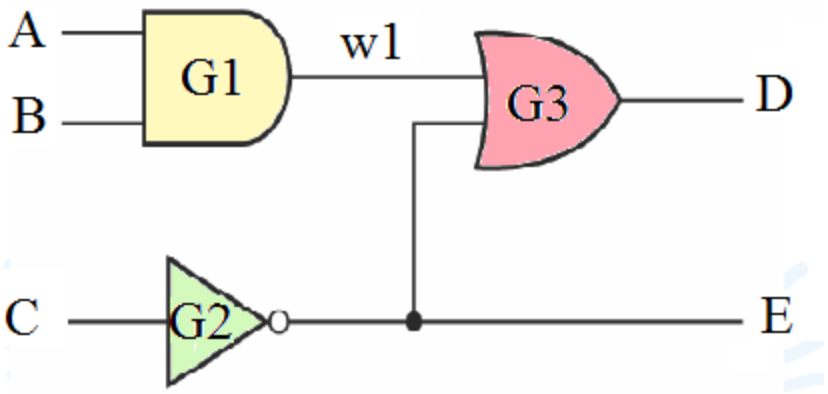


مثال ساده‌ای از Verilog HDL

اعلان، نمونه‌سازی و درگاه‌های گیت:

- باید ابتدا درگاه خروجی و بعد درگاه‌های ورودی ذکر گردد. (برای لیست درگاه‌های ماژول چنین الزامی نیست.)

```
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
```

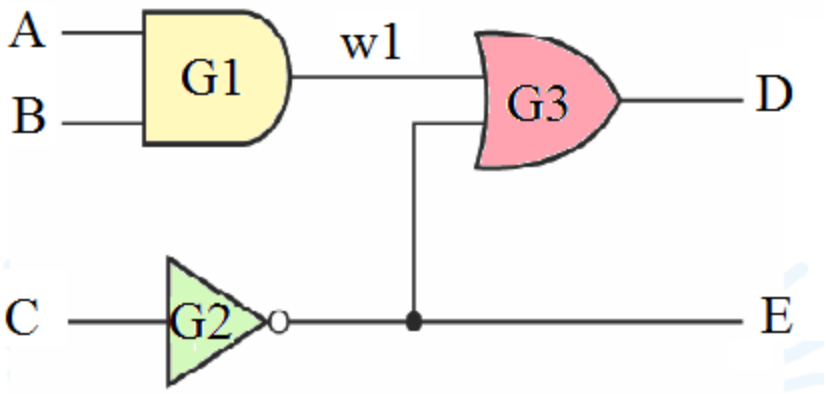


واژه‌نامه:

declaration	اعلان
instantiation	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 حداقل‌سازی
- روش نقشه
- سیستماتیک کردن انتخاب ترکیب‌ها در نقشه
- ساده‌سازی به فرم‌ضرب حاصل‌جمع‌ها
- حالت‌های بی‌اهمیت
- پیاده‌سازی با NAND و NOR
- دیگر پیاده‌سازی‌های دو سطحی
- تابع OR انحصاری
- زبان توصیف سخت‌افزار Verilog