سیستمهاي دودويي

نمایش دا ده ها

- انواع داده‌ها
- مکمل‌ها
- نمایش با ممیز ثابت
- نمایش با ممیز شناور
- دیگر کدهای باینری
- کشف خطا

نمایش دا ده ها

اطلاعاتی که یک کامپیوتر با آن سر و کار دارد:

- داده ها
  - داده های عددی (اعداد طبیعی و حقیقی)
  - داده های غیر عددی (حروف ،علائم)
- ارتباط بین عناصر داده ای
  - ساختمان های داده ای(لیست های پیوندی،درخت ها و....)
- برنامه ها (دستورات)

نمایش عددی داده‌ها

- داده‌های عددی.
  - اعداد(طبیعی ،حقیقی)
- سیستم نمایش اعداد.
  - سیستم نمایشی که در آن مکان هر رقم دارای وزن نیست(مثل سیستم اعداد یونانی).
  - سیستم هائی که هر رقم در نمایش یک عدد دارای وزن است.
    - در این سیستم به هر رقم نسبت به جایگاه آن یک وزن اختصاص می دهیم.
    - سیستم های دهدهی ،دودوئی و هگزادسیمال مثالهایی از این سیستم ها هستند.

نمایش عددی دادها

- اگر پایهٔ هر سیستم R باشد
  - باید از R-1 رقم برای نمایش اعداد استفاده کرد.
  - مثال:

$$A_R = a_{n-1}\, a_{n-2} \ldots a_1\, a_0\, .a_{-1} \ldots a_{-m}$$

نقطه در اینجا قسمت طبیعی و کسری عدد را از یکدیگر جدا کرده است.

  - ارزش A_R برابر است با:

$$V(AR\ ) = \sum_{i=-m}^{n-1} a_i R^i$$

R = 10 Decimal number system, R = 2 Binary
R = 8 Octal, R = 16 Hexadecimal

نمایش عددی دادها

- مبنای انتخاب سیستم نمایش اعداد زمان و هزینه میباشد:
  - هزینۀ ساخت سخت افزار(CPU،ALU و کانالهای ارتباطی)
  - زمان لازم برای پردازش داده ها
- جداول لازم برای جمع ریاضی اعداد
  - در سیستم هائی که مکان رقم در آنها دارای وزن نیست:
    - جداول چنین سیستم هایی پایان ناپذیر است و بنابراین غیر قابل ساخت می باشند.
- در سیستم هائی که مکان رقم دارای وزن است:
  - جدول جمع دو رقم برای چنین سیستم‌هائی پایان پذیر است، اما هر چه اندازۀ جدول کوچکتر باشد ،ساخت آن مقرون به صرفه تر است.بنابراین در چنین شرایطی مبنای ۲مرجح تر از مبنای ۱۰است.

نمایش عددی دادها

جداول لازم برای جمع اعداد در دو سیستم دودوئی و ده دهی

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

جدول برای جمع اعداد دهدهی

	0	1
0	0	1
1	1	10

جدول جمع اعداد دودوئی

مقایسهٔ اعداد در چهار مبنا

نمایش ارقام در مبناهای ۱۰،۲،۸،۱۶

Decimal	Binary	Octal	Hexadecimal
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

تبدیل بین مبناهای ۲،۱۶و۸

تبدیل بین مبناهای۲،۸و۱۶به سادگی دسته بندی رقم ها و جایگزینی آنها با رقمهای متناظر در سیستم دیگر می باشد.

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

تبدیل اعداد در مبنای ۱۰به مبناهای دیگر

- تبدیل از مبنای Rبه مبنای ۱۰

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

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

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

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

$(110110)_2 = ... = (54)_{10}$

$(110.111)_2 = ... = (6.785)_{10}$

$(F3)_{16} = ... = (243)_{10}$

$(0.325)_6 = ... = (0.578703703 ....)_{10}$

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 ۱۰ می‌باشد. آنرا به مبنای ۲ تبدیل کنید.

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

اعشار = 0.6875

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

صحیح = 41

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

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

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

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

مکمل اعداد

- دو نوع مکمل برای هر عدد در مبنای Rوجود دارد:
  - مکمل R
  - مکملR-1
- در مکمل R-1از هر رقم مقدار(R-1)راکسر می‌کنیم.
  - مکمل ۹عدد $۸۳۵_{۱۰}$برابر است با $۱۶۴_{۱۰}$
  - مکمل ۱عدد$۱۰۱۰_{۲}$برابر است با$۰۱۰۱_{۲}$
- برای یافتن مکمل Rیک عدد ابتدا مکمل R-1 آن عدد را محاسبه کرده سپس مقدار۱را با آن جمع می کنیم.
  - مکمل ۱۰عدد$۸۳۵_{۱۰}$برابر است با $۱۶۴_{۱۰}+۱ = ۱۶۵_{۱۰}$
  - مکمل ۲عدد $۱۰۱۰_{۲}$برابر است با $۰۱۰۱_{۲}+۱ = ۰۱۱۰_{۲}$

اعداد با ممیز ثابت

- اعداد اعشاری را می توان به دو صورت نمایش داد:
  - اعداد اعشاری با ممیز ثابت.
  - اعداد اعشاری با ممیز شناور.
- نمایش اعداد دودوئی با روش ممیز ثابت:

$$X = x_n x_{n-1} x_{n-2} \ldots x_1 x_0 \bullet x_{-1} x_{-2} \ldots x_{-m}$$

- نحوۀ نمایش اعداد علامتدار به اینصورت است که از باارزش‌ترین بیت برای علامتگذاری استفاده می‌کنیم. اگر $X_n=1$ عدد منفی و اگر $X_n=0$ عدد مثبت است.

اعداد علامت دار

- در حالت طبیعی سیستم اعداد انتخاب شده برای نمایش اعداد باید بتواند هم اعداد بدون علامت و هم اعداد علامت دار را نمایش دهد.
- سه روش زیر برای نمایش اعداد علامت دار وجود دارد:
  - نمایش بصورت اندازه-علامت.
  - نمایش بصورت مکمل یک.
  - نمایش بصورت مکمل دو.

اعداد علامت دار

مثال:اعداد 9+ و 9- را به صورت یک عدد باینری مبنای ۲ با استفاده از ۳روش فوق نمایش دهید:

تنها یک روش برای نمایش عدد 9+ وجود دارد.

+۹⟶۰۰۱۰۰۰۱

برای عدد ۹–سه روش نمایش وجود دارد:

روش اندازه–علامت: 1 001001

روش مکمل یک:1 110110

روش مکمل دو: 1 110111

نمایش اعداد با ممیز ثابت

- بطور کلی در کامپیوتر ها اعداد با ممیز ثابت به دو صورت نمایش داده می شوند:
  - فقط قسمت کسری آن نمایش داده می شود..
  - فقط قسمت صحیح آن.

خصایص سه روش نمایش اعداد علامت دار

- مکمل یک عدد علامت دار در هر یک از ۳سیستم نمایش:
  - اندازه-علامت:برای مکمل کردن کافی است فقط بیت علامت را معکوس کنیم.
  - مکمل یک: تمام بیت ها را که شامل بیت علامت هم می‌شود را مکمل می کنیم.
  - مکمل دو:از عدد مربوطه مکمل دو می گیریم.

$$X = x_n\ x_{n-1} \ldots x_0\ .\ x_{-1} \ldots x_{-m}$$

خصایص سه روش نمایش اعداد علامت دار

– مقایسه بزرگترین و کوچکترین عدد در هر یک از ۳روش و همچنین نمایش ۰در آنها:

علامتدار

Max:	$2^n - 2^{-m}$	011 ... 11.11 ... 1
Min:	$-(2^n - 2^{-m})$	111 ... 11.11 ... 1
Zero:	+0	000 ... 00.00 ... 0
	-0	100 ... 00.00 ... 0

نمایش علامت به کمک مکمل ۱

Max:	$2^n - 2^{-m}$	011 ... 11.11 ... 1
Min:	$-(2^n - 2^{-m})$	100 ... 00.00 ... 0
Zero:	+0	000 ... 00.00 ... 0
	-0	111 ... 11.11 ... 1

نمایش علامت به کمک مکمل ۲

Max:	$2^n - 2^{-m}$	011 ... 11.11 ... 1
Min:	-2^n	100 ... 00.00 ... 0
Zero:	0	000 ... 00.00 ... 0

وزن ارقام در سیستم مکمل دو

- وزن ارقام درنمایش مکمل دو اعداد علامت دار به صورت زیر تعریف می‌شود:
  - بیت علامت دارای وزن منفی است.
  - وزن بقیهٔ بیت ها مشابه حالت اعداد بدون علامت محاسبه می شود:

$$X = x_n\, x_{n-1} \ldots x_0$$

$$\Rightarrow \quad V(X) = -\,x_n \times 2^n + \sum_{i=0}^{n-1} x_i \times 2^i$$

جمع اعداد علامت دار در سیستم اندازه-علامت

۱. علامت دو عدد با یکدگر مقایسه می شوند.
۲. اگر علامتها یکسان بود،مقادیر دو عدد با یکدیگر جمع می‌شوند.و پس از آن وجـود سرریز بررسی می شود.
۳. اگر علامت دو عدد متفاوت بود،عدد کوچکتر از عدد بزرگتر کم می شود.
۴. علامت دو عدد در حالتی که علامت دو عدد با یکدیگر برابرند برابر با علامت یکی از دو عدد است.در حالتی که علامتها متفاوتند علامـت حاصـلجمع علامـت عـدد بزرگتر است.

جمع اعداد علامت دار در سیستم اندازه علامت

چند مثال

6 + 9

```
    6   0110
+)  9   1001
-------------
   15   1111 -> 01111
```

6 + (- 9)

```
    9   1001
-)  6   0110
-------------
  - 3   0011 -> 10011
```

Overflow 9 + 9 or (-9) + (-9)

-6 + 9

```
    9   1001
- ) 6   0110
-------------
    3   0011 -> 00011
```

-6 + (-9)

```
    6   0110
+)  9   1001
-------------
  -15   1111 -> 11111
```

جمع دو عدد علامت دار در نمایش مکمل دو

- دو عدد همراه با بیت علامتشان بایکدیگر جمع می شوند و از رقم نقلی خروجی از پر ارزش ترین بیت صرفه نظر می شود.

مثال

```
   -6   1  1010
+)  9   0  1001
    3   0  0011
```

```
    6   0  0110
+)  9   0  1001
   15   0  1111
```

```
   -9      1  0111
+) -9      1  0111
  -18  (1)0  1110
```

$x'_{n-1}y'_{n-1}s_{n-1}$
$(c_{n-1} \oplus c_n)$

```
    6   0  0110
+) -9   1  0111
   -3   1  1101
```

overflow

```
    9   0  1001
+)  9   0  1001
   18   1  0010
```

$x_{n-1}y_n s'_{n-1}$
$(c_{n-1} \oplus c_n)$

دوعملوند دارای یک علامتند ولی علامت حاصل جمع با آنها متفاوت است.

$$x_{n-1}y_{n-1}s'_{n-1} + x'_{n-1}y'_{n-1}s_{n-1} = c_{n-1} \oplus c_n$$

جمع دو عدد نمایش داده شده در سیستم مکمل یک

دو عدد را همراه با بیت علامتشان جمع می کنیم.اگر یک رقم نقلی از پر ارزش ترین بیت خارج شد حاصل جمع را با ۱جمع می‌کنیم و از رقم نقلی صرفه نظر می کنیم.

مثال **end-around carry**

	6	0	0110
+)	-9	1	0110
	-3	1	1100

+)

+)

not overflow $(c_{n-1} \oplus c_n) = 0$

	-9	1	0110
+)	-9	1	0110
		(1)0	1100
+)			1
		0	1101

	9	0	1001
+)	9	0	1001
		1	(1)0010

overflow
$(c_{n-1} \oplus c_n)$

مقایسۀ نمایش اعداد علامت دار در سه سیستم

- سادگی در منفی کردن

S + M > 1’s Complement > 2’s Complement

- پیاده سازی توسط سخت‌افزار
  - سیستم اندازه علامت به یک تفریق کننده و یک جمع کننده نیاز دارد.
  - در سیستم های مکمل دو و مکمل یک ،فقط به یک جمع کننده نیاز هست.
- سرعت انجام محاسبات

2’s Complement > 1’s Complement(end-around C)

مقایسهٔ نمایش اعداد علامت دار در سه سیستم

- تشخیص عدد صفر
  - سیستم مکمل دو از همه سریعتر است.

تفریق

- تفریق در سیستم مکمل دو
  - مکمل دو، تفریق کننده(همراه با بیت علامت)را می‌گیریم و حاصل را با تفریق شونده جمع می‌کنیم.

$$(\pm A) - (-B) = (\pm A) + B$$

$$(\pm A) - B = (\pm A) + (-B)$$

نمایش اعداد با ممیز شناور

- در این نمایش مکان ممیز قسمت کسری ثابت نیست.
- طول عدد قابل نمایش در این روش وسیع است.

$$F = EM$$

m_n	$e_k e_{k-1} \ldots e_0$	$m_{n-1} m_{n-2} \ldots m_0 . m_{-1} \ldots m_{-m}$
علامت	نما	مانتیس

- مانتیس:عدد علامت دار با ممیز ثابت(خواه عدد صحیح و یا عدد کسری).
- نما:مکان ممیز را مشخص می‌کند.

مقدار معادل ده دهی

$$V(F) = V(M) * RV(E)$$

M: Mantissa
E: Exponent
R: Radix

اعداد با ممیز شناور

مثال:

sign		sign	
0	.1234567	0	04
mantissa		exponent	

$$\Longrightarrow +.1234567 \times 10^{+04}$$

نکته: در نمایش اعداد به صورت شناور تنها مانتیس(M)و نما(E)صریحاًنمایش داده می شوند.مبنا و مکان ممیز به صورت ضمنی استنباط می شوند.

اعداد با ممیز شناور

مثال:عدد ۱۰۰۱,۱۱+را به صورت یک عدد با ممیز شناور ۱۶بیتی (۶بیت نما و ۱۰ بیت مانتیس)نمایش دهید.

0	0 00100	100111000
علامت	نما	مانتیس
0	0 00101	010011100

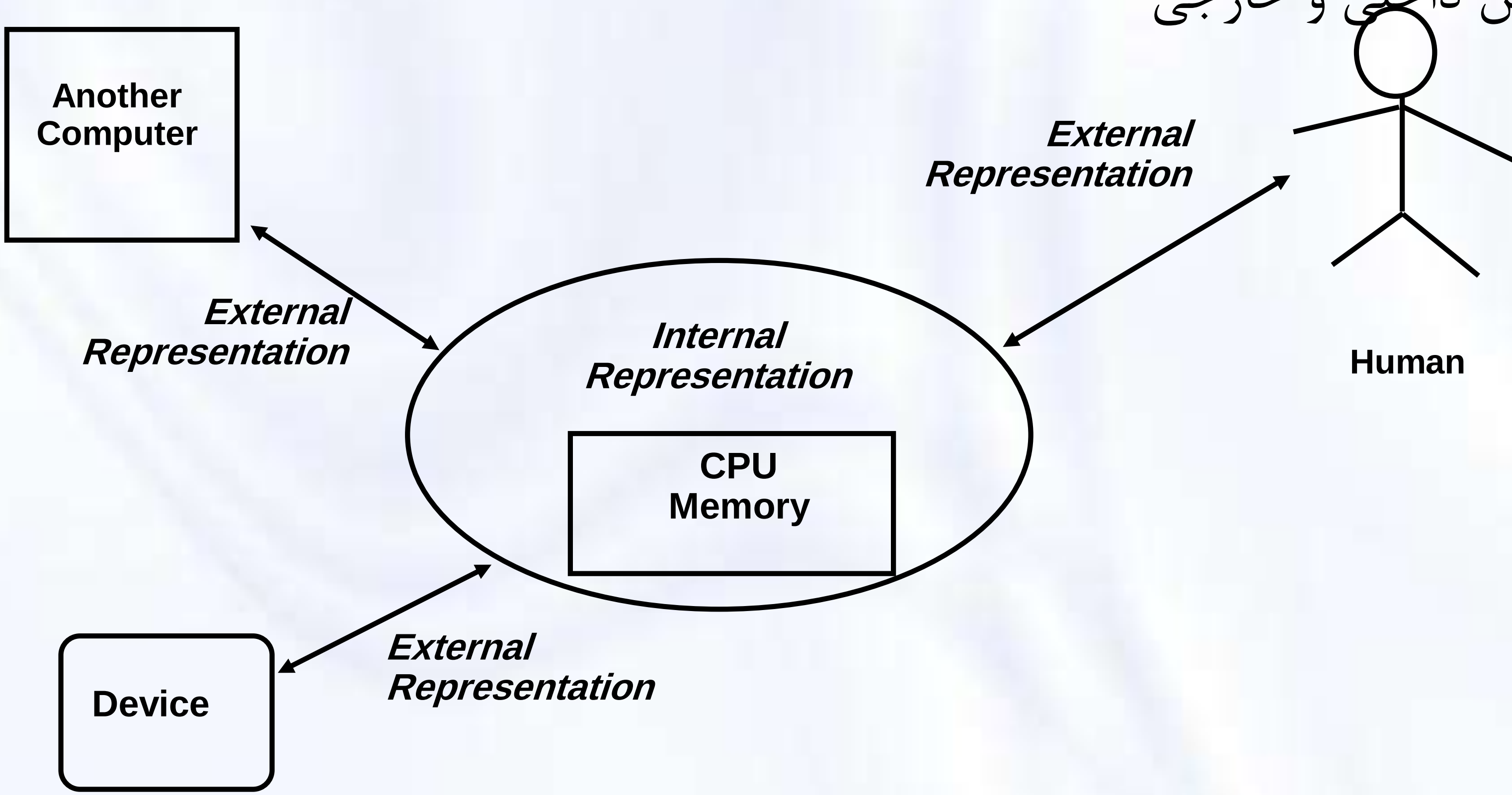
یا

خصوصیات نمایش اعداد بصورت ممیز شناور

- فرم نرمال اعداد با ممیز شناور
  - نمایش‌های متعددی از یک عدد در سیستم نمایش اعداد با ممیز شناور وجود دارد.
  - با ارزش ترین رقم مانتیس هموار باید یک رقم غیر صفر باشد.
- نمایش صفر
  - صفر:نمایش بصورت مانتیس صفر(M=0)
  - صفر حقیقی:
    - مانتیس برابر با صفر.
    - نما برابر است با کوچکترین عدد قابل نمایش که بصورت ۰...۰۰نمایش داده می شود.

نمایش داخلی و خارجی

نمایش داخلی و خارجی



نمایش خارجی

- اعداد
  - اغلب داده هایی که درون کامپیوتر ذخیره می شوند در نهایت توسط محاسباتی که بر روی آنان انجام می گیرد تغییر پیدا می کند.
    - نمایش داخلی برای کارآمدی انجام محاسبات.
    - لازم است نتایج نهائی جهت نمایش به یک فرمت خارجی مناسب تبدیل شوند.
- حروف الفبا،علائم و برخی اعداد
  - عناصر این داده ها توسط هیچ پردازشی تغییر پیدا نمی کنند.
    - نیازی به نمایش داخلی ندارند،چرا که در هیچگونه محاسبه‌ای مورد استفاده قرار نمی‌گیرند.
    - نمایش خارجی برای پردازش و ارائه در فرمت قابل لازم است.

نمایش خارجی

مثال:اعداد دهدهی

- نمایش به صورت یک کد باینری دودوئی.
- نمایش بصورت BCD(Binary Coded Decimal)

Decimal	BCD Code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

انواع کدهای دسیمال

- 8,4,2,-2,1,-1هر کدام وزن هایی هستند که به هر بیت اختصاص پیدا کرده اند.

Decimal	BCD(8421)	2421	84-2-1	Excess-3
0	0000	0000	0000	0011
1	0001	0001	0111	0100
2	0010	0010	0110	0101
3	0011	0011	0101	0110
4	0100	0100	0100	0111
5	0101	1011	1011	1000
6	0110	1100	1010	1001
7	0111	1101	1001	1010
8	1000	1110	1000	1011
9	1001	1111	1111	1100

انواع کدهای دسیمال

بدست آوردن مکمل ۹کد BCDکار مشکلی است.اما از آنجا که سایر کدها خود مکمل هستند این کار برای آنها ساده است.

- نکات دیگر:

d_3 d_2 d_1 d_0: symbol in the codes

BCD: $d_3 \times 8 + d_2 \times 4 + d_1 \times 2 + d_0 \times 1$

$\Rightarrow$ 8421 code.

2421: $d_3 \times 2 + d_2 \times 4 + d_1 \times 2 + d_0 \times 1$

84-2-1: $d_3 \times 8 + d_2 \times 4 + d_1 \times (-2) + d_0 \times (-1)$

Excess-3: BCD + 3

کد گری

- یکی از خصیصه های کد گری اینست که دو کد متوالی آن تنها در یک بیت با یکدیگر تفاوت دارند.
- خصیصهٔ فوق در بعضی مواقع کاربرد دارد.

Decimal number	Gray g_3	g_2	g_1	g_0	Binary b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

کد گری ۴-بیتی

تحلیل کد گری

- فرض کنید بیت های $g_n g_{n-1} \ldots g_1 g_0$ نمایش دهندهٔ یک کد گری برای عدد باینری $b_n b_{n-1} \ldots b_1 b_0$ باشند.

$$g_i = b_i \oplus b_{i+1} \quad , \; 0 \le i \le n\text{-}1$$
$$g_n = b_n$$

and

$$b_{n-i} = g_n \oplus g_{n-1} \oplus \ldots \oplus g_{n-i}$$
$$b_n = g_n$$

Reflection of Gray codes

ε	0	0 0	0 00	0 000
	1	0 1	0 01	0 001
		1 1	0 11	0 011
		1 0	0 10	0 010
			1 10	0 110
			1 11	0 111
			1 01	0 101
			1 00	0 100
				1 100
				1 101
				1 111
				1 010
				1 011
				1 001
				1 101
				1 000

نمایش کاراکترها بوسیلهٔ کد ASCII

<u>ASCII</u> (American Standard Code for Information Interchange) <u>Code</u>

MSB (3 bits)

LSB (4 bits)

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	‘	P
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	I	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	m	n	~
F	SI	US	/	?	O	n	o	DEL

نمایش کاراکترهای کنترلی توسط کد ASCII

NUL	Null	DC1	Device Control 1
SOH	Start of Heading (CC)	DC2	Device Control 2
STX	Start of Text (CC)	DC3	Device Control 3
ETX	End of Text (CC)	DC4	Device Control 4
EOT	End of Transmission (CC)	NAK	Negative Acknowledge (CC)
ENQ	Enquiry (CC)	SYN	Synchronous Idle (CC)
ACK	Acknowledge (CC)	ETB	End of Transmission Block (CC)
BEL	Bell	CAN	Cancel
BS	Backspace (FE)	EM	End of Medium
HT	Horizontal Tab. (FE)	SUB	Substitute
LF	Line Feed (FE)	ESC	Escape
VT	Vertical Tab. (FE)	FS	File Separator (IS)
FF	Form Feed (FE)	GS	Group Separator (IS)
CR	Carriage Return (FE)	RS	Record Separator (IS)
SO	Shift Out	US	Unit Separator (IS)
SI	Shift In	DEL	Delete
DLE	Data Link Escape (CC)		

(CC) Communication Control
(FE) Format Effector
(IS) Information Separator

کدهای تشخیص خطا

- سیستم توازن:
  - ساده‌ترین روش برای تشخیص خطا.
  - یک بیت توازن به اطلاعات اضافه می گردد.
  - دو نوع توازن وجود دارد،توازن زوج و توازن فرد.

- توازن زوج:
  - یک بیت به اطلاعات اضافه می‌شود بنابراین تعداد کل ۱های کد زوج است.

1011001 $\boxed{0}$

1010010 $\boxed{1}$

کدهای تشخیص خطا

- توازن فرد:
  - یک بیت به اطلاعات اضافه می‌شود.بنابراین تعداد کل بیت‌ها فرد است.

1011001 $\boxed{1}$
1010010 $\boxed{0}$

تولید بیت توازن

- تولید بیت توازن:
  - برای عدد باینری $b_6b_5\ldots b_1b_0$ (۷-بیت اطلاعات) و با توازن زوج به صورت زیر تولید می گردد:

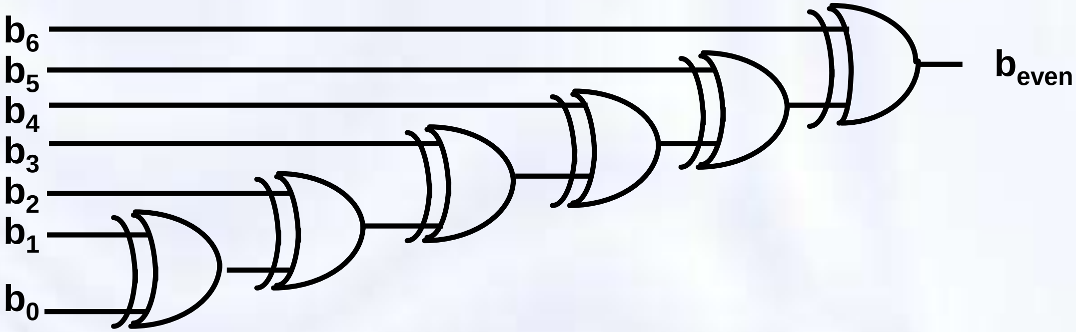
$$b_{even} = b_6 \oplus b_5 \oplus \ldots \oplus b_0$$

  - برای توازن فرد:

$$b_{odd} = b_{even} \oplus 1 = b_{even}$$

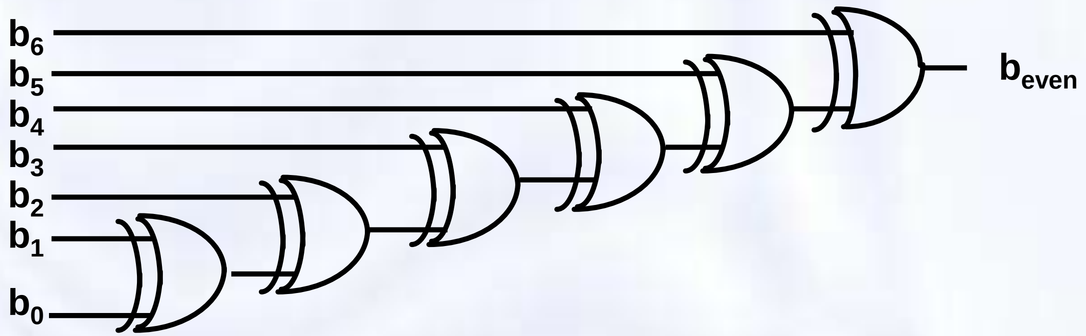
تولید کنندهٔ بیت توازن

- مدار تولید کنندهٔ بیت توازن زوج



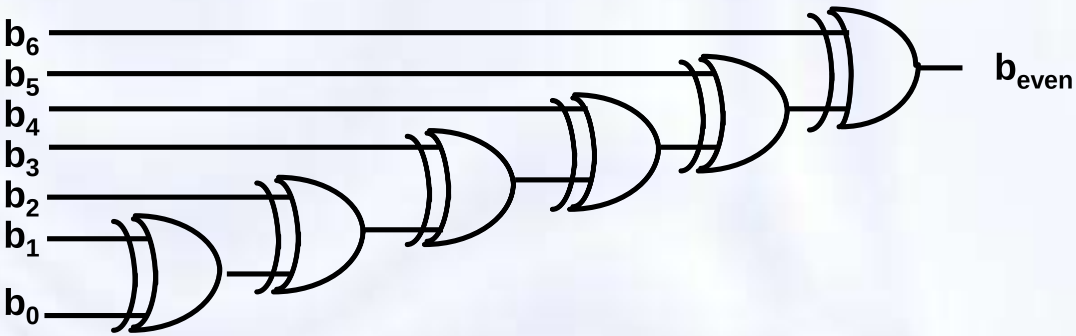
تولید کنندهٔ بیت توازن

- مدار تولید کنندهٔ بیت توازن زوج



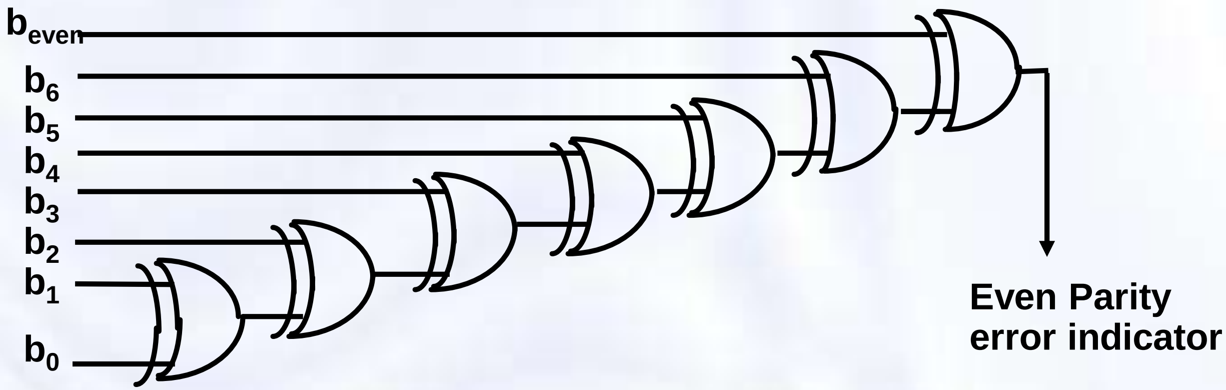
تولید کنندۀ بیت توازن

• مدار تولید کنندۀ بیت توازن زوج



چک کنندۀ توازن

- مدار چک کنندۀ توازن زوج



جبریول

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 می‌شوند.
- ***اصل ۱(P1): عضو خنثی***

(a) $a + 0 = a$ (عضو همانی برای +), (b) $a \bullet 1 = a$ (عضو همانی برای •)

- ***اصل۲(P2): جابجایی***

(a) $a + b = b + a$, (b) $a \bullet b = b \bullet a$

- ***اصل ۳(P3):شرکت پذیری***

(a) $a + (b + c) = (a + b) + c$ (b) $a \bullet (b \bullet c) = (a \bullet b) \bullet c$

- ***اصل۴(P4): توزیع پذیری***

(a) $a + (b \bullet c) = (a + b) \bullet (a + c)$ (b) $a \bullet (b + c) = a \bullet b + a \bullet c$

- ***اصل۵(P5): مکمل***

(a) $a+a'=1$ (b) $a.a'=0$

قضایای جبر بول

- قضیه ۱ *(T1)*:

(a) $a + a = a$ (b) $aa = a$

- قضیه ۲ *(T2)*:

(a) $a + 1 = 1$ (b) $a0 = 0$

- قضیه ۳ *(T3)*:

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

- قضیه ۴ *(T4)*:

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

- قضیه ۵ *(T5)*:

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

بعضی از خواص صفر و یک در جبر بول:

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

- ***مثال:*** با استفاده از قضایا و اصول جبر بول تساوی زیر را ثابت کنید:

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

حل

- $B + AB'C'D = (B + ABC'D) + AB'C'D$
  $= B + (ABC'D + AB'C'D)$
  $= B + AC'D\,(B + B')$
  $= B + AC'D$

- ***مثال:*** رابطه زیر را ثابت کنید:

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

حل:

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

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

$=0 + (X+Y)Z$

$= (X+Y)Z$

- ***قضیه۶ (T6):***

(a) $ab + ab' = a$

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

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 ۷ *(T7)*:

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

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

(b)مثال: عبارت بولی زیر را ساده کنید

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

حل

– $wy' + wx'y + wxyz + wxz'$

$$= wy' + wx'y + wxy + wxz' \quad [T7(a)]$$
$$= wy' + wy + wxz' \quad [T7(a)]$$
$$= w + wxz' \quad [T7(a)]$$
$$= w \quad [T7(a)]$$

قوانین دمورگان

قضیه ۸ (T8) : قانون دمورگان

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

(a)فرم کلی قانون دمورگان

(a) $(a + b + \dots z)' = a'b' \dots z'$ (b) $(ab \dots z)' = a' + b' + \dots z'$

(a)مثال: عبارت بولی زیر را با استفاده از قانون دمورگان ساده کنید:

$$
\begin{aligned}
(a + bc)' &= (a + (bc))' && \\
&= a'(bc)' && \text{[T8(a)]} \\
&= a'(b' + c') && \text{[T8(b)]} \\
&= a'b' + a'c' && \text{[P5(b)]}
\end{aligned}
$$

- ***مثال:*** عبارت زیر را ساده کنید:

$$(a(b+z(x+a')))'$$

حل:

$$
\begin{aligned}
(a(b+z(x+a')))' &= a' + (b+z(x+a'))' && [\text{T8(b)}]\\
&= a' + b'\,(z(x+a'))' && [\text{T8(a)}]\\
&= a' + b'\,(z' + (x+a')') && [\text{T8(b)}]\\
&= a' + b'\,(z' + x'(a')') && [\text{T8(a)}]\\
&= a' + b'\,(z' + x'a) && [\text{T3}]\\
&= a' + b'\,(z' + x') && [\text{T5(a)}]
\end{aligned}
$$

قضیه ۹*(T9):*

- (a) $ab + a'c + bc = ab + a'c$ (b) $(a + b)(a' + c)(b + c) = (a + b)(a' + c)$

مثال : نشان دهید:

$$ABC + A'D + B'D\ ABC + A'D + B'D + CD =$$

حل:

$$\begin{aligned} ABC + A'D + B'D + CD &= ABC + (A' + B')D + CD && \text{[P5(b)]} \\ &= ABC + (AB)'D + CD && \text{[T8(b)]} \\ &= ABC + (AB)'D && \text{[T9(a)]} \\ &= ABC + (A' + B')D && \text{[T8(b)]} \\ &= ABC + A'D + B'D && \text{[P5(b)]} \end{aligned}$$

جدول درستی توابع بولی

- برای نشان دادن درستی قضایا و یا روابط بولی می‌توان از جدول درستی استفاده کرد.
- جدول درستی برای 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 برای 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 بولی

- توابع بولی را می‌توان به چند طریق نمایش داد:
- بصورت ترکیبی نامرتب از عبارات و متغیرهای بولی.
- بصورت مجموع جملات مینترم
- بصورت حاصلضرب جملات ماکسترم

- ***چند جمله‌ای مینترم***: یک عبارت حاصلضرب است که همهٔ متغیرها یا مکمل آنها حتماً وجود دارد.

- برای حالتی که سه متغیر داشته باشیم، جدول جملات مینترم 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

- هر تابع بولی را می‌توان بصورت مجموعی از جملات مینترم نشان داد. بعنوان مثال:

جدول درستی تابع $f_1(A,B,C) = m2 + m3 + m_6 + m_7$ *که بصورت*

$f_1(A,B,C) = S\ m\,(2,3,6,7)$ 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 یک تابع که با جملات مینترم نشان داده شده، کافی است آن دسته از جملات مینترم 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 می‌کنیم:

$$f'(A,B,Q,Z) = m_2 + m_3 + m_4 + m_5 + m_8 + m_9 + m_{10} + m_{11} + m_{12} + m_{13} + m_{14} + m_{15}$$
$$= \Sigma\, m(2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15)$$

- ***چند جمله‌ای ماکسترم:*** یک عبارت حاصلجمع است که همهٔ متغیرها یا مکمل آنها حتماً وجود دارد.
- *هر تابع بولی را می‌توان بصورت حاصلضرب جملات ماکسترم نشان داد.*
- *جدول زیر نشان دهندهٔ لیست جملات ماکسترم برای سه متغیر است:*

Maxterm	Maxterm Code	Maxterm Number
$A+B+C$	000	M_0
$A+B+C'$	001	M_1
$A+B'+C$	010	M_2
$A+B'+C'$	011	M_3
$A'+B+C$	100	M_4
$A'+B+C'$	101	M_5
$A'+B'+C$	110	M_6
$A'+B'+C'$	111	M_7

مثال: تابع زیر بصورت حاصلضرب جملات ماکسترم نشان داده شده است. جدول درستی آنرا بنویسید.

$$f_2(A,B,C) = M_0M_1M_4M_5 = \Pi M(0,1,4,5)$$

حل

(i)	Inputs ABC	M_0 $A+B+C$	M_1 $A+B+C'$	M_4 $A'+B+C$	M_5 $A'+B+C'$	Outputs $f_2(A,B,C)$
0	000	0	1	1	1	0
1	001	1	0	1	1	0
2	010	1	1	1	1	1
3	011	1	1	1	1	1
4	100	1	1	0	1	0
5	101	1	1	1	0	0
6	110	1	1	1	1	1
7	111	1	1	1	1	1

تبدیل بین شکلهای متعارف

هرگاه یک تابع بولی بر حسب مجموع جملات مینترم نوشته شده باشد ، مکمل آن برابر است با حاصلجمع جملات مینترمهایی که از تابع اصلی حذف شده اند.

بطور مشابه می توان هر تابع بصورت مجموع جملات میینترم را بصورت حاصلضرب جملات ماکسترمهایی نوشت که شماره های متناظر آن از جملات مینترم، از تابع اصلی حذف شده‌اند.

مثال:

$$f_1(A,B,C) = \Sigma\, m\,(2,3,6,7)$$
$$= \overline{f_2}(A,B,C)$$
$$= \Pi M(0,1,4,5)$$

تبدیل توابع بولی به مجموع جملات مینترم

- با استفاده از قضایای گفته شده در جبر بول، در هر جمله عبارت بولی تمام متغیرها را وارد می‌کنیم.
- ***مثال:*** تابع بولی زیر را بصورت مجموع جملات مینترم نمایش دهید:
- $f(A,B,C) = AB + AC' + A'C$
- ***حل***

$$
\begin{aligned}
f(A,B,C) &= AB.1 + AC'.1 + A'C.1 \\
&= AB.(C+C') + AC'.(B+B') + A'C.(B+B') \\
&= ABC + \underline{ABC'} + \underline{ABC'} + AB'C' + A'BC + A'B'C \\
&= ABC + ABC' + AB'C' + A'BC + A'B'C \\
&= \Sigma m(1, 3, 4, 6, 7)
\end{aligned}
$$

تبدیل توابع بولی به فرم حاصلضرب جملات ماکسترم

- با استفاده از قضیه گفته شده ($a+a'=0$) تمامی متغیرها را در هر عبارت وارد می کنیم.
- ***مثال:***تابع بولی زیر را بصورت حاصلضرب جملات ماکسترم نمایش دهید:
- $f(A,B,C) = AB + AC' + A'C$
- ***حل***
- $f(A,B,C) =(A+AC+A'C)(B+AC+A'C)$

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

$$=(A+A')(A+C)(B+C+AA')$$

$$=(A+C+BB')(B+C+A)(B+C+A')$$

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

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

$$=\mathbf{\Pi(0,2,4)}$$

روش دیگر

- می توان برای یافتن حاصلضرب جملات ماکسترم از مجموع جملات مینترم نیز استفاده کرد.

مثال: *تابع* زیر را بصورت حاصلضرب جملات ماکسترم نمایش دهید:

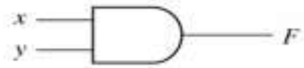
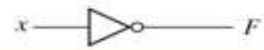
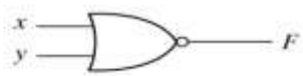
- $f(A,B,C) = AB + AC' + A'C$

حل:

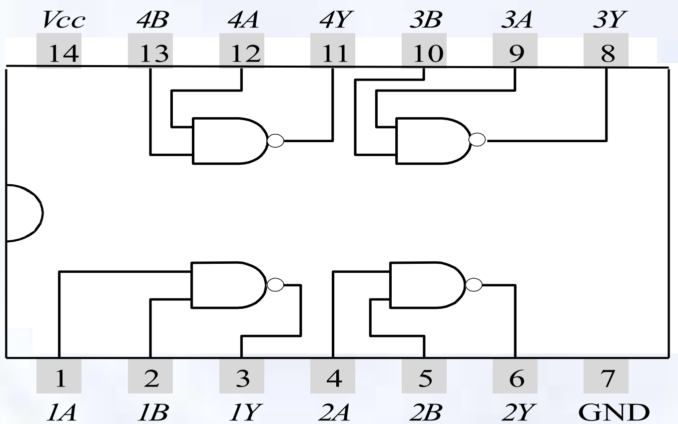
- در مثال قبل این تابع بولی را به صورت مجموع جملات مینترم نمایش دادیم:
- $f(A,B,C) = \Sigma m(1, 3, 4, 6, 7)$

بنابراین می توان نوشت:

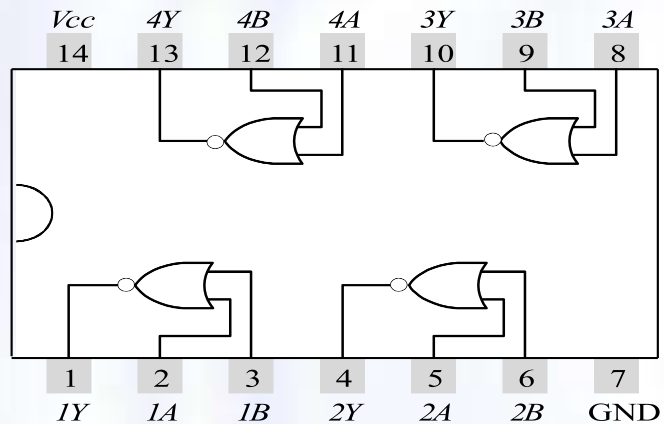
$f(A,B,C) = \Sigma m(1, 3, 4, 6, 7) = \mathbf{\Pi(0,2,4)}$

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

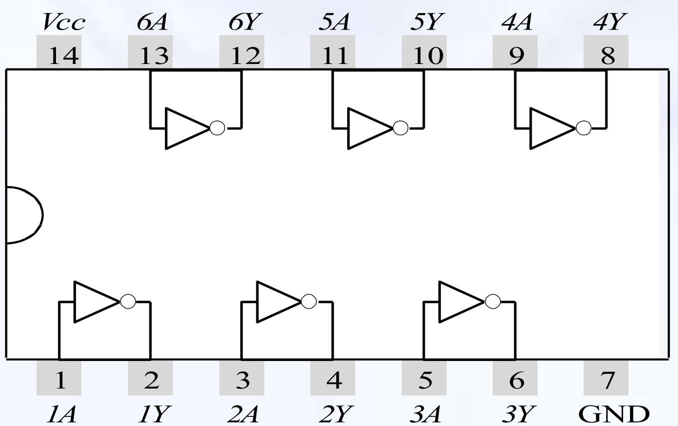
گیت های منطقی در درون ICها



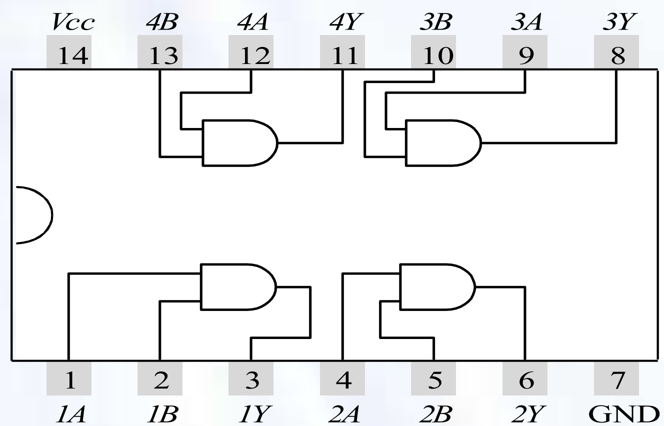
7400:$Y=\overline{AB}$
Quadruple two-input NAND gates



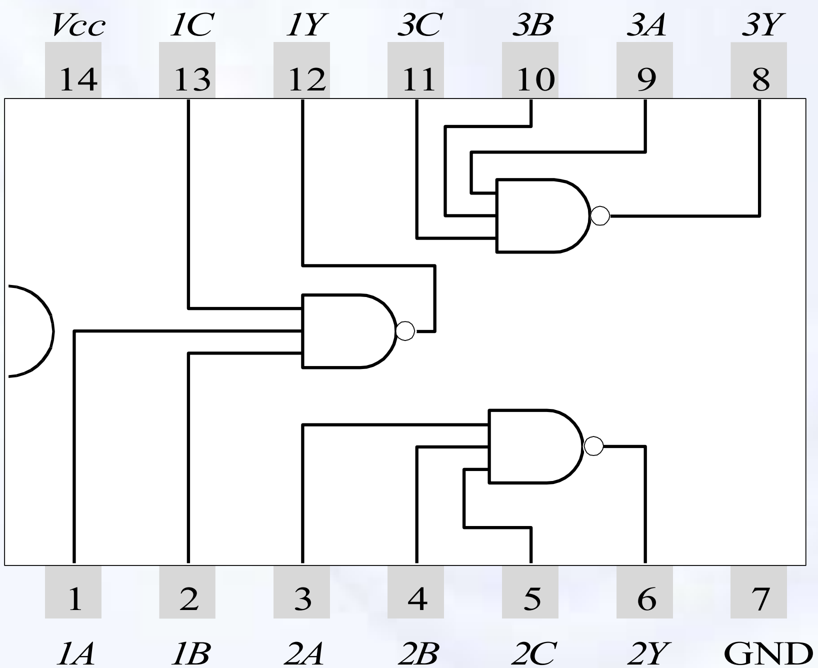
7402:$Y=\overline{A+B}$
Quadruple two-input NOR gates



7404:$Y=\overline{A}$
Hex inverters

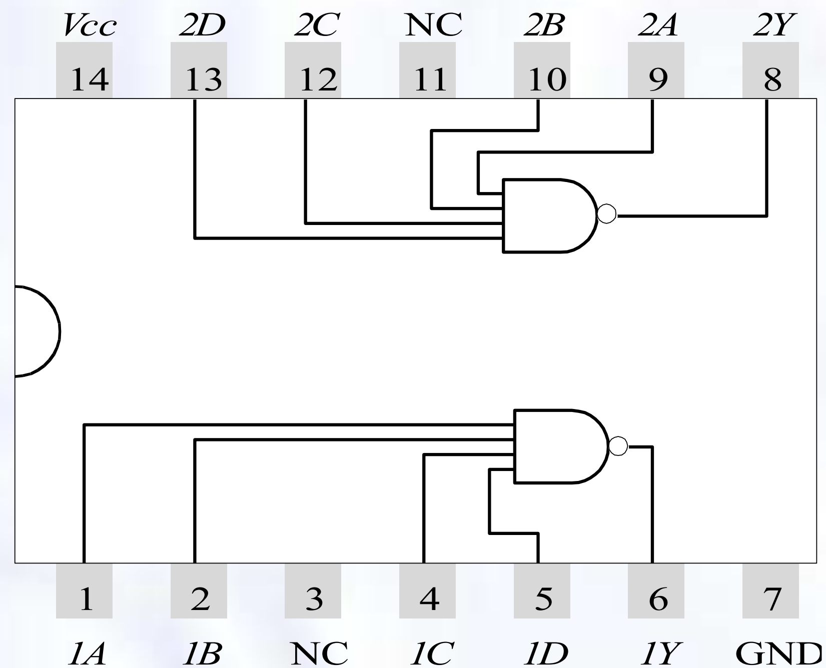


7408:$Y=AB$
Quadruple two-input AND gates



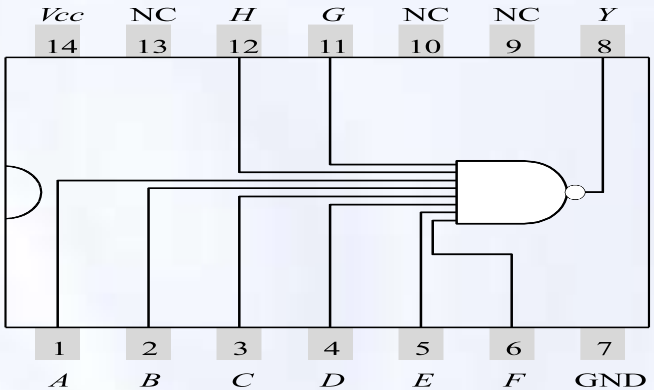
7410: $Y=\overline{ABC}$

Triple three-input NAND gates

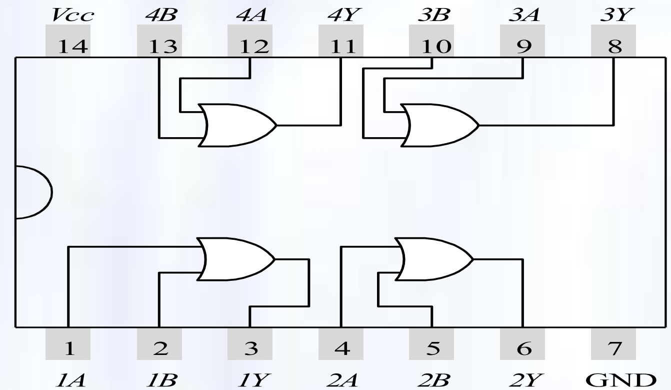


7420: $Y=\overline{ABCD}$

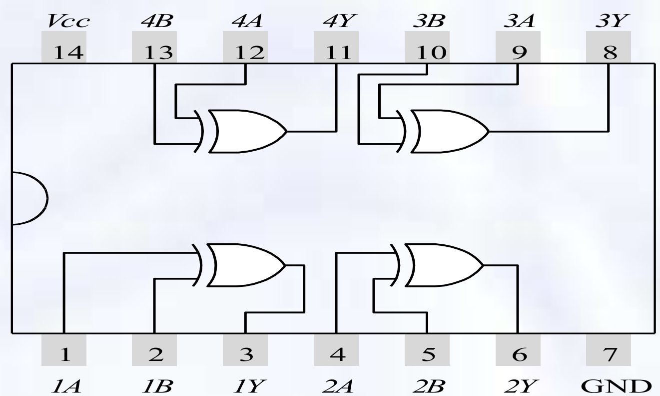
Dual four-input NAND gates



7430: $Y = \overline{ABCDEFGH}$
8-input NAND gate



7432: $Y = A + B$
Quadruple two-input OR gates



7486: Y = AÅB
Quadruple two-input exclusive-OR gates

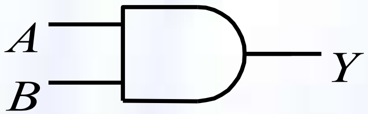
گیت AND

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

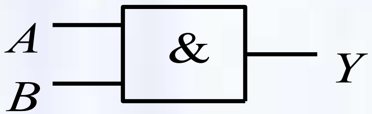
(a)

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

(b)



(c)



(d)

(a) جدول درستی یک AND

(a) استفاده از سطح ولتاژهای Hو L برای نشان دادن صفر و یک منطقی

(b) سمبل استاندارد گیت

(d) سمبل گیت در استاندارد IEEE

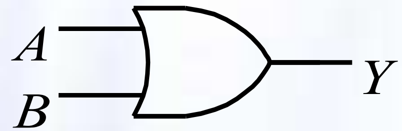
گیت OR

a	b	$f_{OR}(a, b) = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

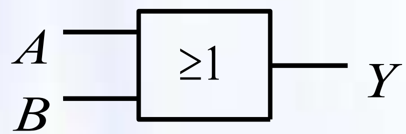
(a)

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

(b)



(c)



(d)

(a) جدول درستی یک OR

(b) استفاده از سطح ولتاژهای Hو L برای نشان دادن صفر و یک منطقی

(c) سمبل استاندارد گیت

(d) سمبل گیت در استاندارد IEEE

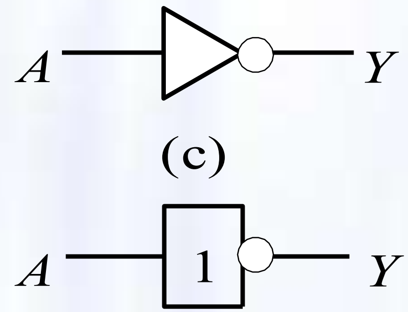
گیت NOT

a	$f_{NOT}(a) = \bar{a}$
0	1
1	0

(a)

A	Y
L	H
H	L

(b)



(d)

(a) جدول درستی یک NOT

(b) استفاده از سطح ولتاژهای H و L برای نشان دادن صفر و یک منطقی

(c) سمبل استاندارد گیت

(d) سمبل گیت در استاندارد IEEE

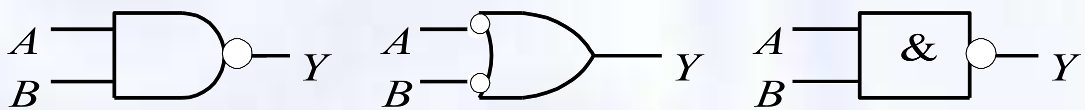
گیت NAND

a	b	$f_{NAND}(a,b) = \overline{ab}$
0	0	1
0	1	1
1	0	1
1	1	0

(a)

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

(b)



(c) (d) (e)

(a)جدول درستی یک NAND

(b) استفاده از سطح ولتاژهای Hو L برای نشان دادن صفر و یک منطقی

(c)سمبل استاندارد گیت

(d) گیت OR معادل با ورودی‌های مناسب

(e) سمبل گیت در استاندارد IEEE

گیت NOR

a	b	$f_{NOR}(a, b) = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

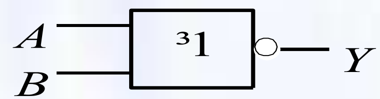
(a)

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

(b)

(c)

(d)



(e)

(a)جدول درستی یک NOR

(b) استفاده از سطح ولتاژهای Hو L برای نشان دادن صفر و یک منطقی

(c)سمبل استاندارد گیت

(d) گیت AND معادل با ورودی‌های مناسب

(e) سمبل گیت در استاندارد IEEE

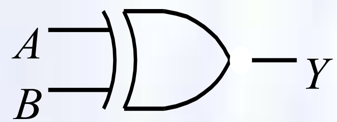
گیت XOR

a	b	$f_{XNOR}(a,b) = a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

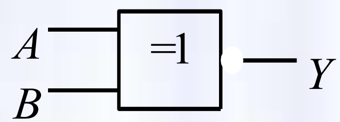
(a)

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

(b)



(c)



(d)

(a)جدول درستی یک XOR

(b) استفاده از سطح ولتاژهای Hو L برای نشان دادن صفر و یک منطقی

(c)سمبل استاندارد گیت

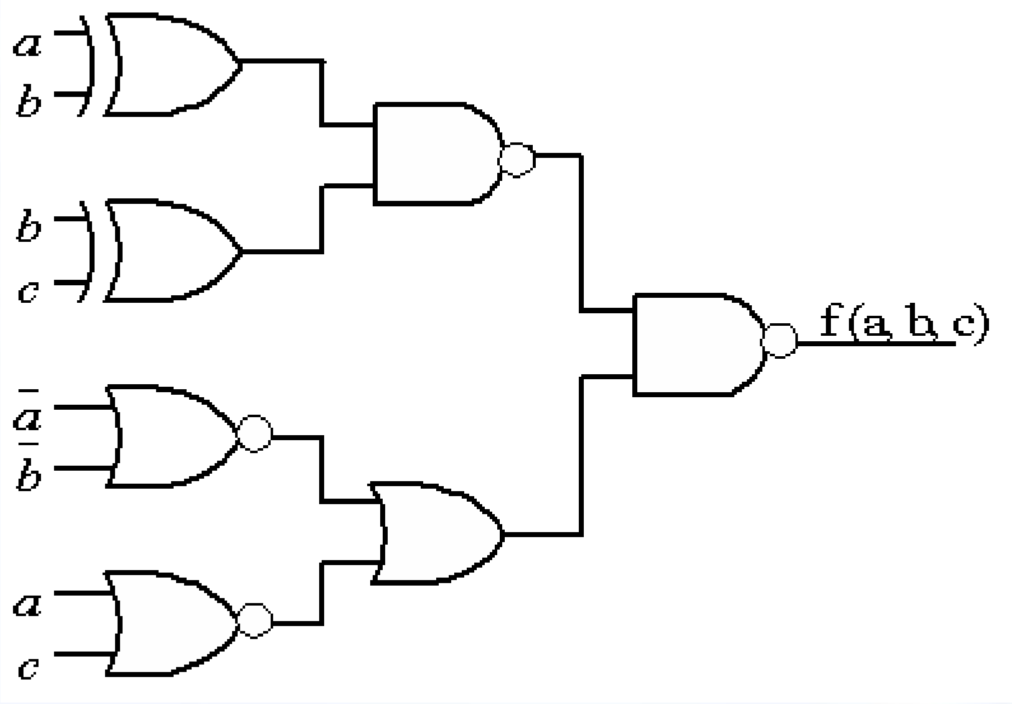
(d) سمبل گیت در استاندارد IEEE

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

طراحی و تحلیل مدارهای ترکیبی

- تحلیل مدار: با استفاده از روابط جبر بول و شکل مدار و روابط گیت‌ها، رابطه‌ای بین ورودی ها و خروجی مدار بدست می‌آید.

- ***مثال:*** تابع بولی مدار زیر را تعیین کنید:



حل

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

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

$$= \overline{\overline{(a\oplus b)(b\oplus c)}} + \overline{\overline{\overline{a}+\overline{b}}+\overline{a+c}}$$

$$= (a\oplus b)(b\oplus c) + (\overline{a}+\overline{b})(a+c)$$

$$= (a\overline{b}+\overline{a}b)(b\overline{c}+\overline{b}c) + (\overline{a}+\overline{b})(a+c)$$

$$= a\overline{b}b\overline{c} + a\overline{b}\overline{b}c + \overline{a}bb\overline{c} + \overline{a}b\overline{b}c + \overline{a}a + \overline{a}c + a\overline{b} + \overline{b}c$$

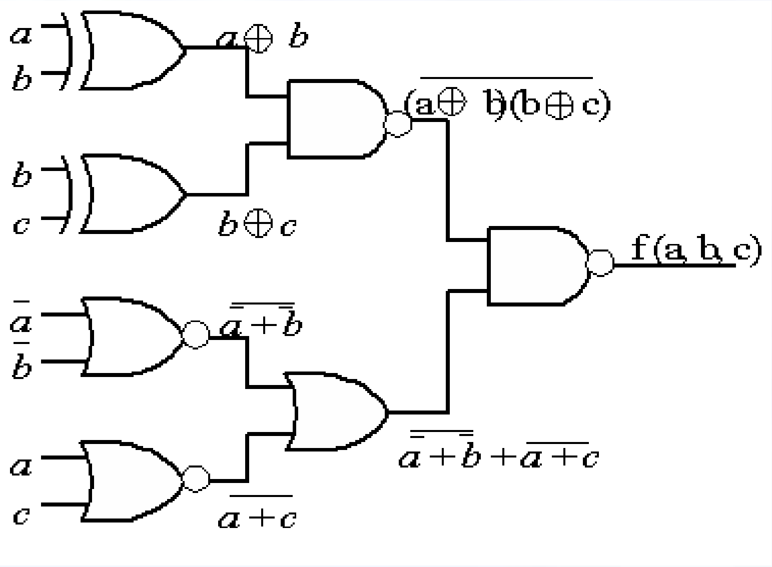
$$= a\overline{b}c + \overline{a}b\overline{c} + \overline{a}c + a\overline{b} + \overline{b}c$$

$$= \overline{a}b\overline{c} + \overline{a}c + a\overline{b} + \overline{b}c$$

$$= \overline{a}b\overline{c} + \overline{a}c + a\overline{b}$$

$$= \overline{a}b + \overline{a}c + a\overline{b}$$

$$= \overline{a}c + a\oplus b$$



تحلیل مدار با استفاده از جدول درستی

- میتوان با مقداردهی ورودی‌های مدار خروجی را بدست آورد و بر اساس مقادیر خروجی، تابع بولی مدار را بدست آورد.

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

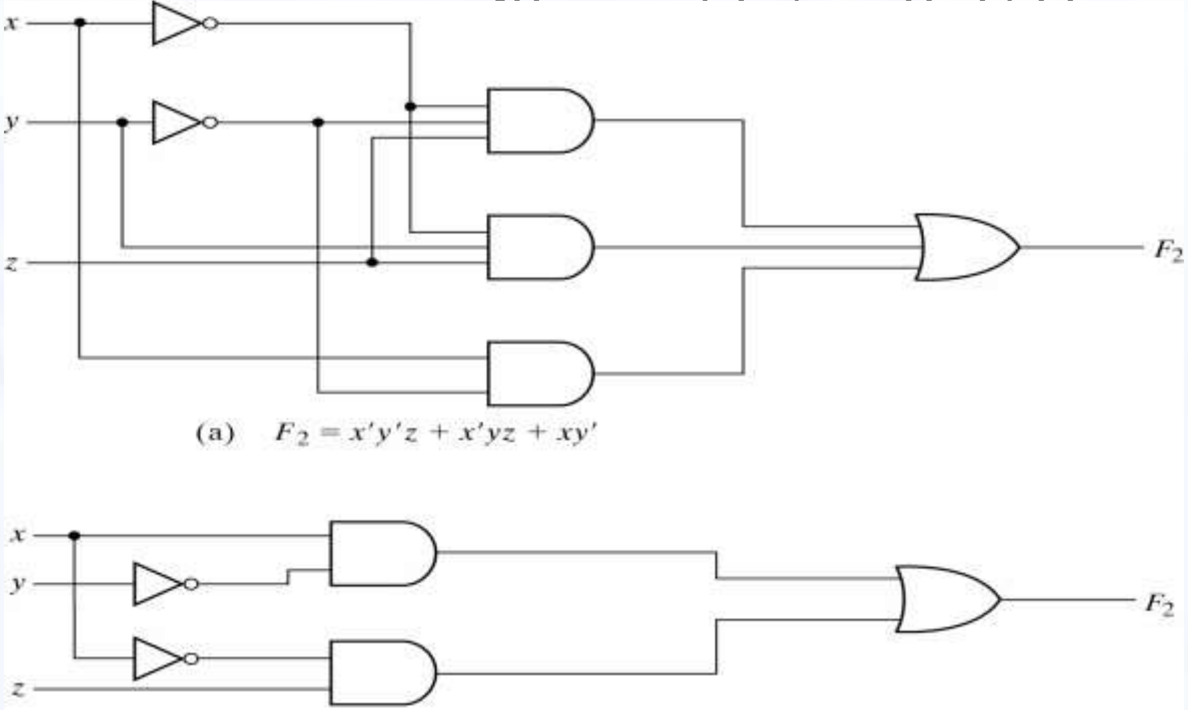
- برای مثال قبلی:

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 مدارهای ترکیبی

یکی از ساده‌ترین روشها برای طراحی و ساخت مدار ترکیبی، 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. معمولا ابتدا از گیتهای AND و سپس از OR استفاده می‌شود.

- ***مثال:*** دو تابع بولی متفاوت توسط گیتهای AND و OR و NOT ساخته شـده‌اند. توجه کنید که توابع بصورت مجموع حاصلضربها هستند.



(a) $F_2 = x'y'z + x'yz + xy'$

(b) $F_2 = xy' + x'z$

تحقق NAND و NOR

می‌توان هر تابع بولی را *فقط* توسط گیتهای NAND یا *فقط* توسط گیتهای NOR ساخت. *مدار بدست آمده نیز شامل دو تراز گیت خواهد بود.*

برای پیاده سازی تابع با گیتهای NAND (NOR)

۱-ابتدا تابع را بصورت مجموع حاصلضربها (حاصلضرب مجموع‌ها) می‌نویسیم.

۲- برای هر جملهٔ حاصلضرب (حاصل جمع) یک گیت NAND (NOR) قرار دهید. ورودیهای هر گیت، حرفهای آن جمله هستند.

۳-یک گیت NAND (NOR) منفرد در دومین تراز قرار دهید بطوریکه ورودی‌های آن خروجی های گیت‌های طبقه اول هستند.

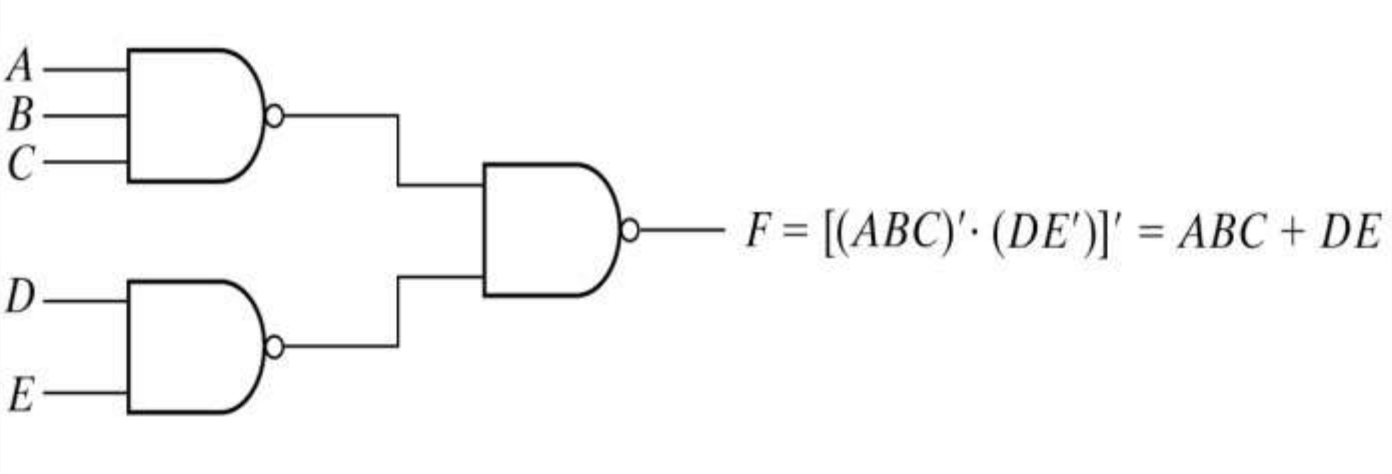
۴-جملهٔ با یک حرف منفرد نیاز به یک معکوس کننده در اولین تراز دارد یا ممکن است که مکمل شده و بصورت یک ورودی برای گیت‌های دومین تراز بکار رود.

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

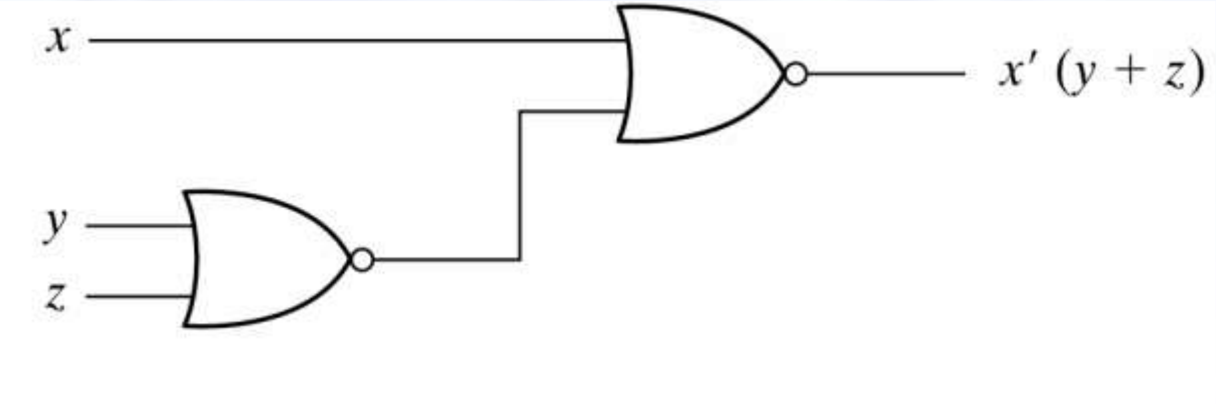
- تابع زیر را توسط گیتهای NOR پیاده سازید:

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

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.



ساده سازي توابع بولي

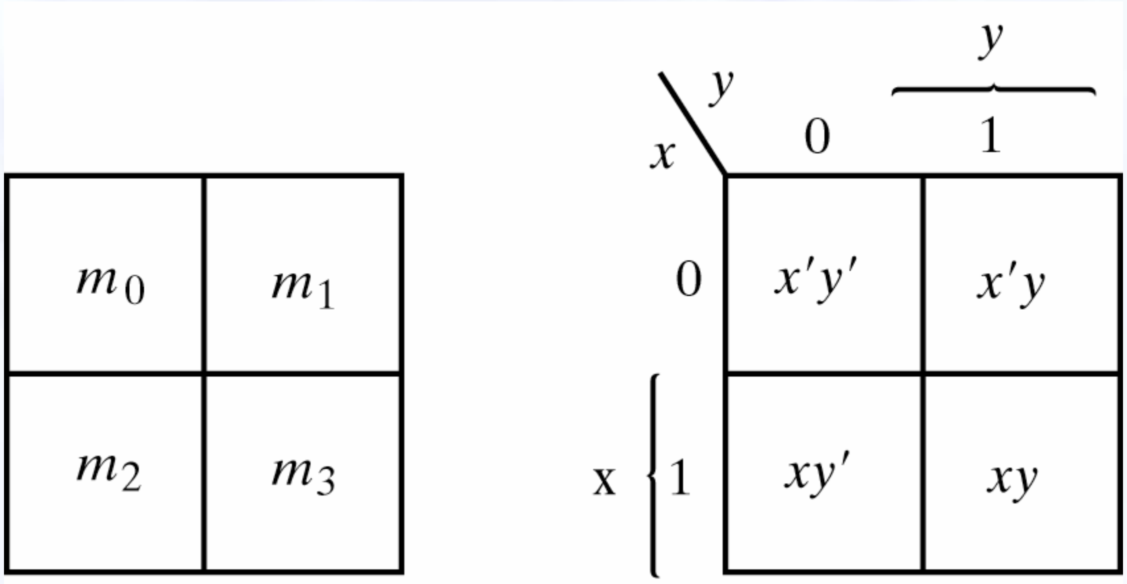
روشهای ساده‌سازی توابع

- ***استفاده از قضایا و اصول جبر بول***
- ***استفاده از جدول کارنو***

- هر تابع بولی را می‌توان بصورت مجموعی از جملات می‌نیمم نشان داد.
- جدول کارنو از مربعهایی تشکیل شده است که هر یک نشان دهندۀ یکـی از جمـلات می‌نیمم 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 می‌نیمم در جدول کارنو
- هر خانهٔ جدول با خانهٔ مجاور خود تنها در یک رقم تفاوت دارد.



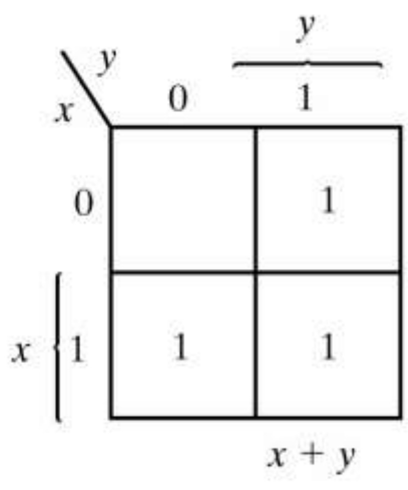
(a) (b)

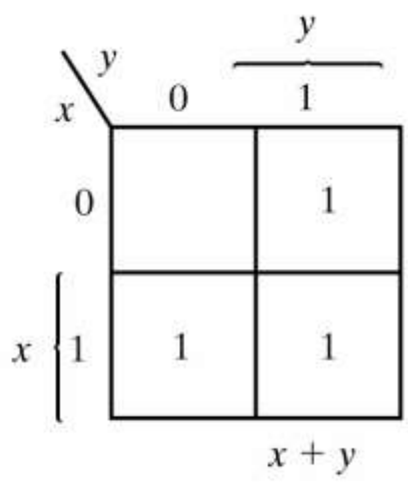
مثال

- 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 صـورت در جـدول ظـاهر شـده انـد. همچنین با استفاده از قوانین جبر بول میتوان f_1 را ساده کرد و به f_2 رسید.





جدول کارنو برای سه متغیر

- برای سه متغیر، هشت عدد جملهٔ می‌نیمم وجود دارد و بنـابراین جـدول کـارنو بایـد هشت خانه داشته باشد.

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

			y	y
$x \backslash xz$	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x 1	$xy'z'$	$xy'z$	xyz	xyz'
		z	z	

(b)

کاربرد جدول کارنو در ساده‌سازی توابع

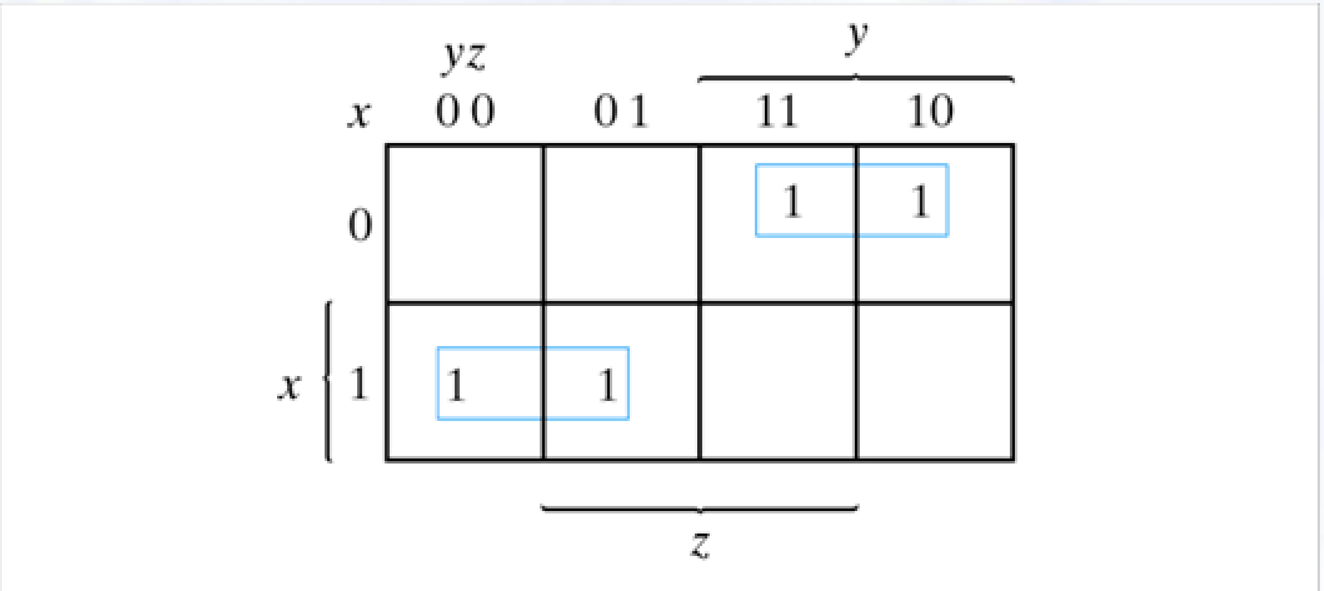
- یکی از بهترین ابزارها برای ساده‌سازی توابع،جدول کارنو است.
- خانه‌هایی از جدول که مقدارتابع در آنها برابر با یک می‌باشد را مشخص می‌کنیم.
- خانه‌هایی که دارای یک هستند و مجاور یکدیگر هم هستند بصورت دوبدو انتخاب کرده و ساده می‌کنیم.
- خانه‌های لبه بالا وپایین هم مجاور یکدیگر هستند، اگرچه در کنار یکدیگر قرار ندارند.

مثال

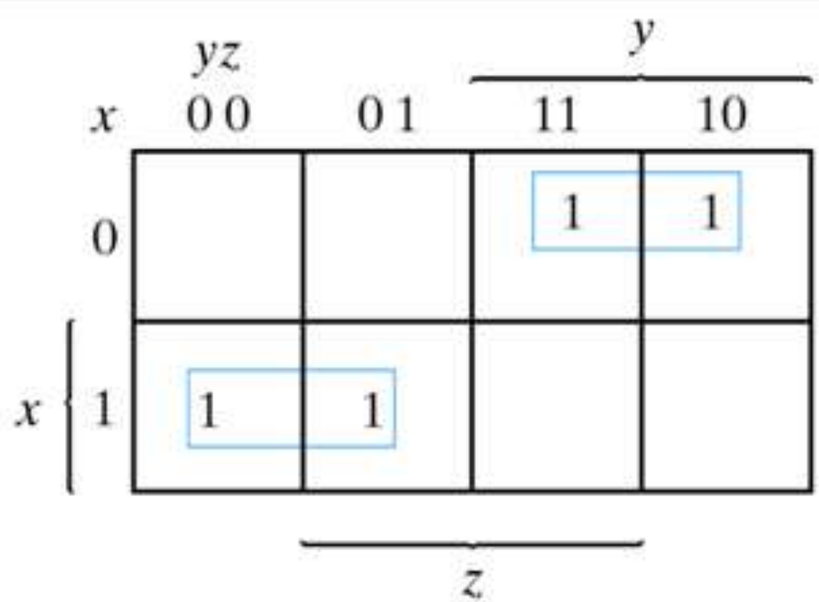
- تابع زیر را ساده کنید:

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

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



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



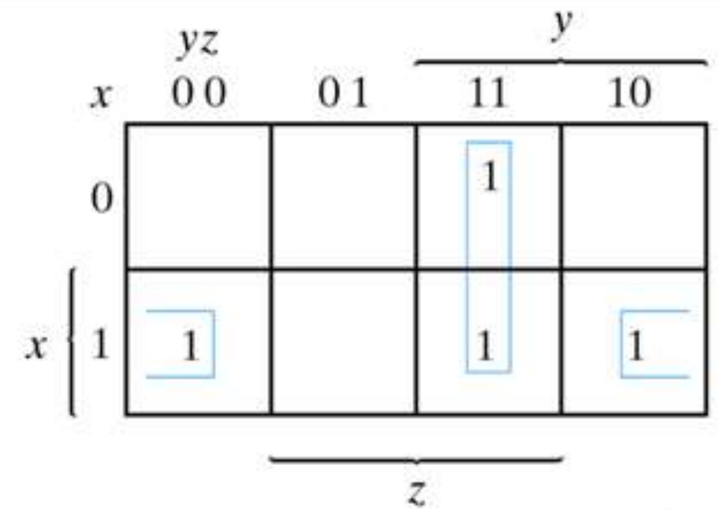
$$F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$$

مثال

- تابع زیر را ساده کنید:

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

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



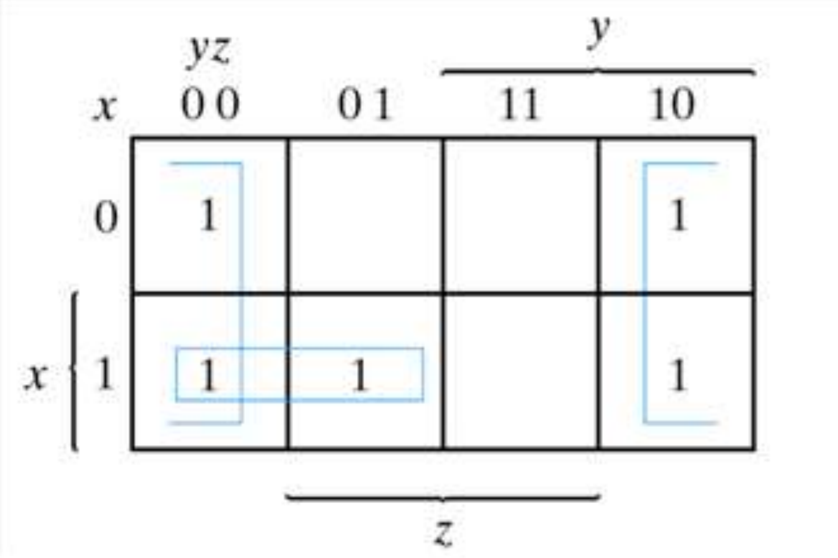
$$F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$$

مثال

- تابع زیر را ساده کنید:

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

حل: با استفاده از جملات می‌نیمم، جدول کارنو را پر می‌کنیم. در ایـن مثـال سـتونهای کناری نیز مجاور یکدیگر هستند:



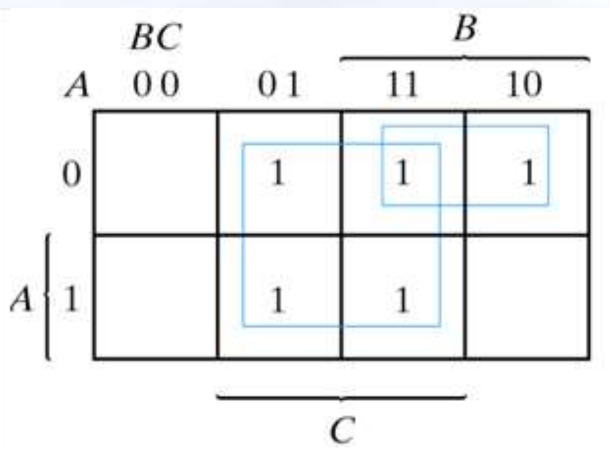
$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$$

مثال

- تابع زیر را ساده کنید:

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

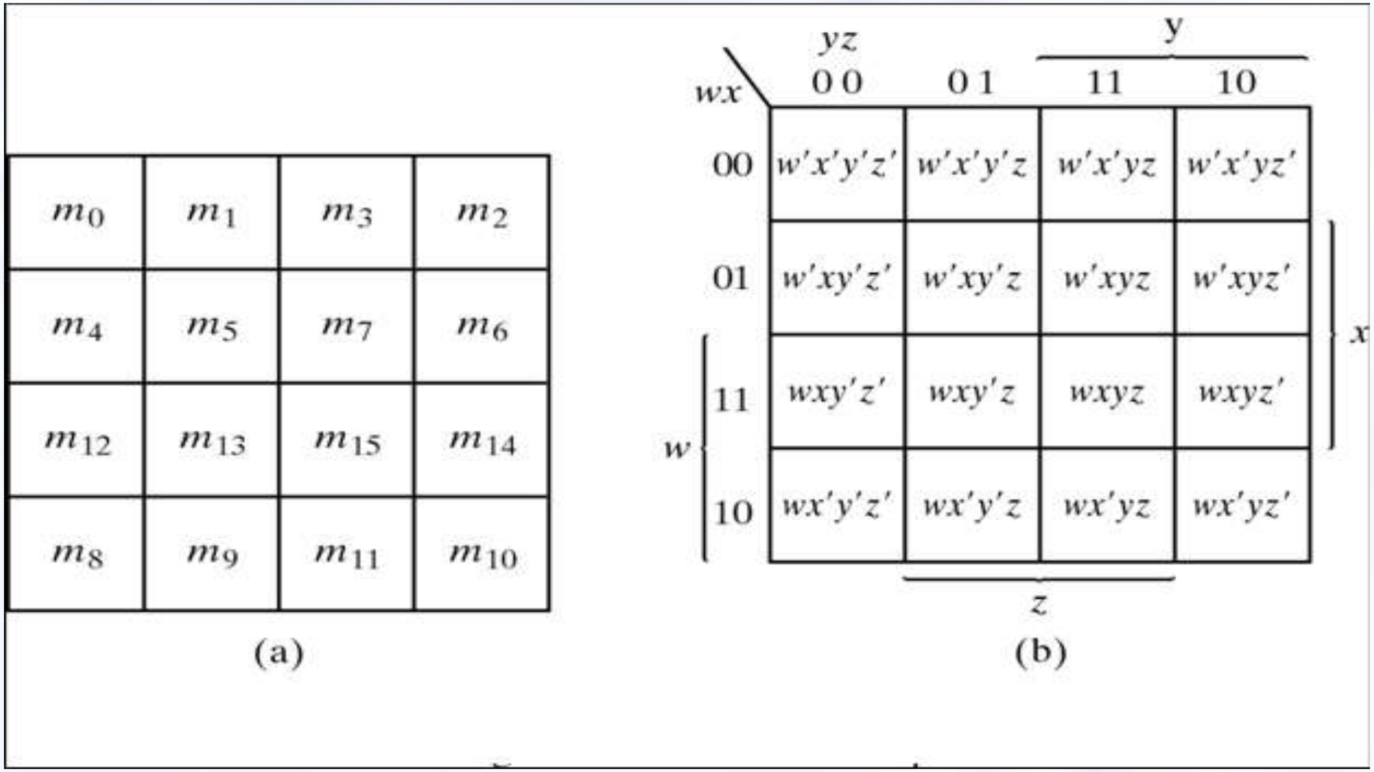
حل: با استفاده از جملات می‌نیمم، جدول کارنو را پر می‌کنیم. در این مثال چهار خانهٔ مجاور انتخاب و ساده شده‌اند. هر چه تعداد بیشتری خانه انتخاب شود بهتر است.



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

جدول کارنو برای چهار متغیر

- برای چهار متغیر، شانزده عدد جملهٔ می‌نیمم وجود دارد و بنابراین جدول کـارنو بایـد شانزده خانه داشته باشد.

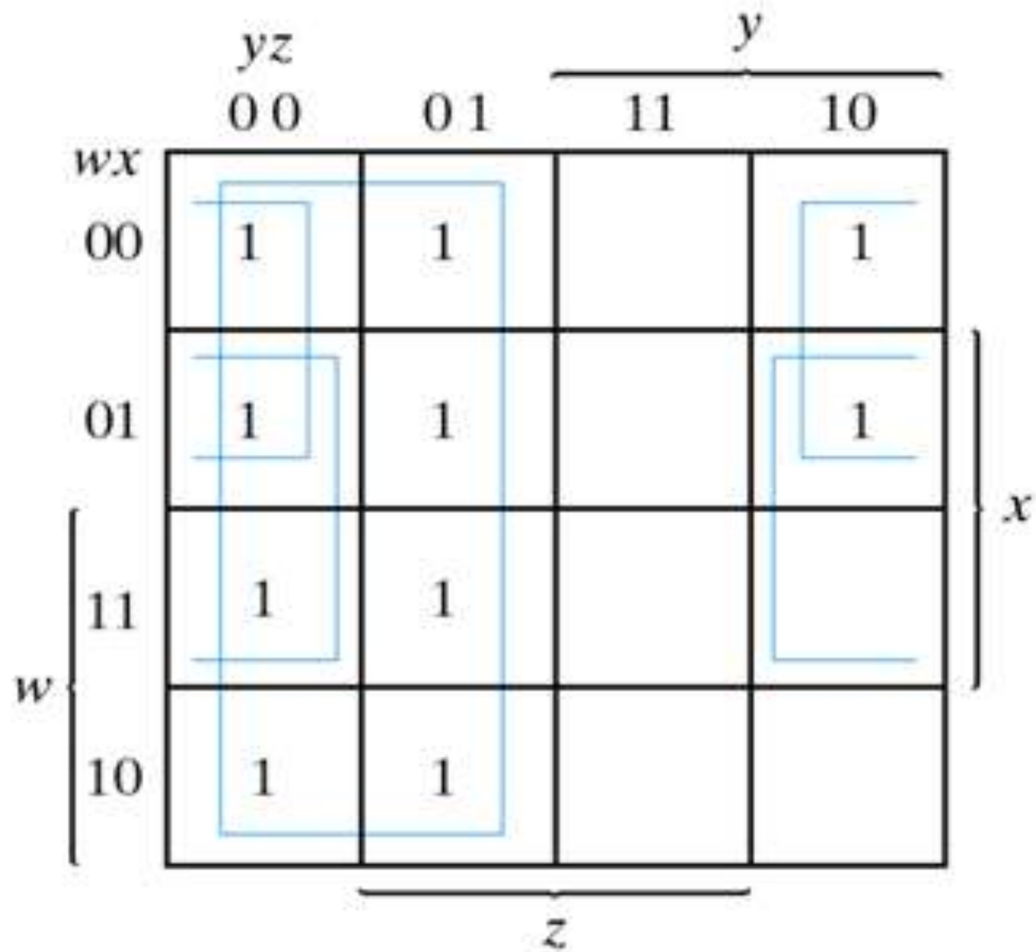


مثال

- تابع زیر را ساده کنید:

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

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



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

$= \Sigma\,(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

$= y' + w'z' + xz'$

جدول کارنو برای پنج متغیر

- برای پنج متغیر، سی و دو عدد جملهٔ می‌نیمم وجود دارد و بنـابراین جـدول کـارنو را بصورت زیر رسم می‌کنیم:

$A = 0$

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

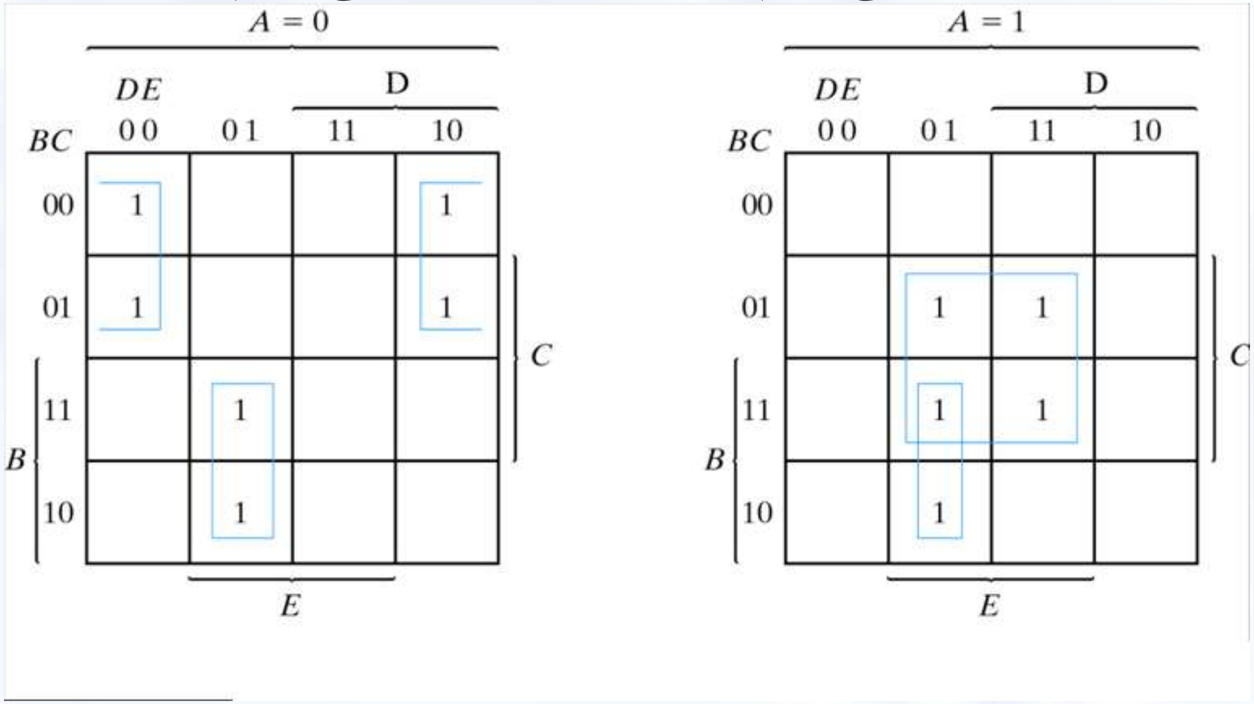
$A = 1$

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

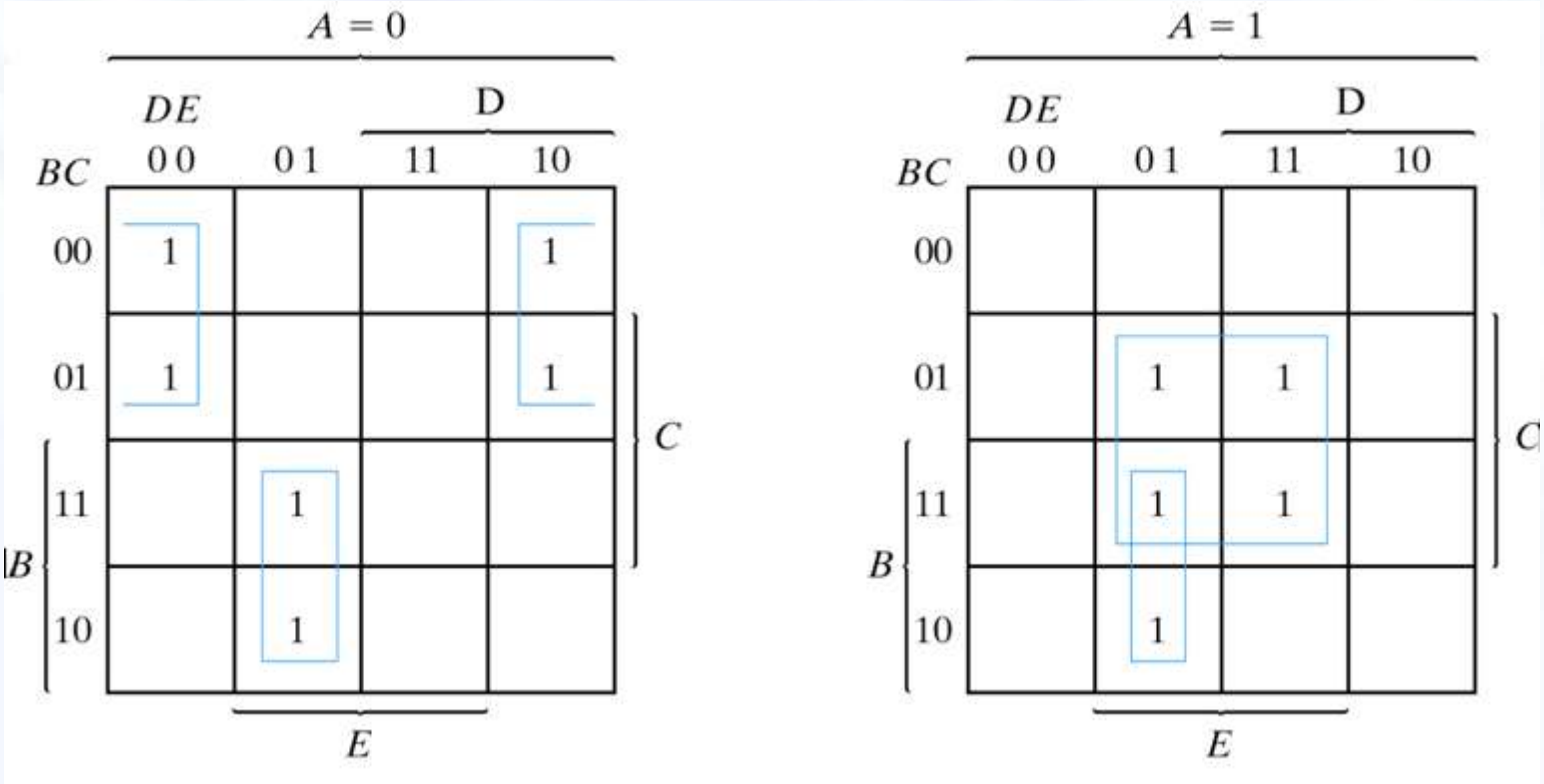
مثال

- تابع زیر را ساده کنید:
- f(A,B,C,D)=∑(0,2,4,6,13,21,23,25,29,31)

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



- توجه:اگر دو جدول زیر را بر هم قرار دهیم، خانه‌هایی که بر روی هم قرار می‌گیرنـد، نیز مجاور هستند.

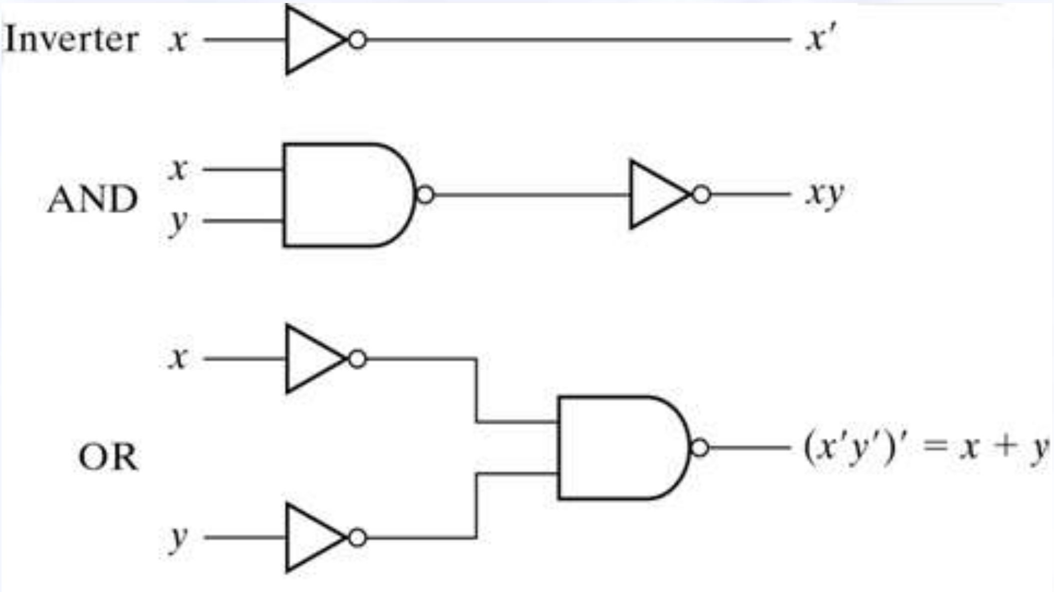


$F = A'B'E' + BD'E + ACE$

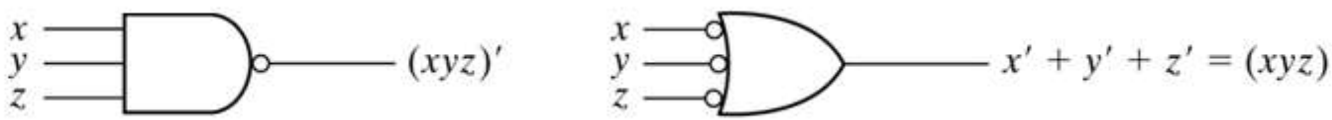
تبديل انواع مختلف
مدارهاي منطقي به يكديگر

مداری تنها با گیتهای NAND

- می‌توان هر مدار منطقی که با هر نوع گیتی ساخته شده باشد را به مداری تبدیل کرد که تنها از گیتهای NANDتشکیل شده باشد. با استفاده از شکلهای زیر، بجای هر گیت معادل آنرا قرار داده و سپس آنرا ساده می‌کنیم.



- در هنگام تبدیل مدارها از گیتهای معادل زیر هم می‌توان کمک گرفت:

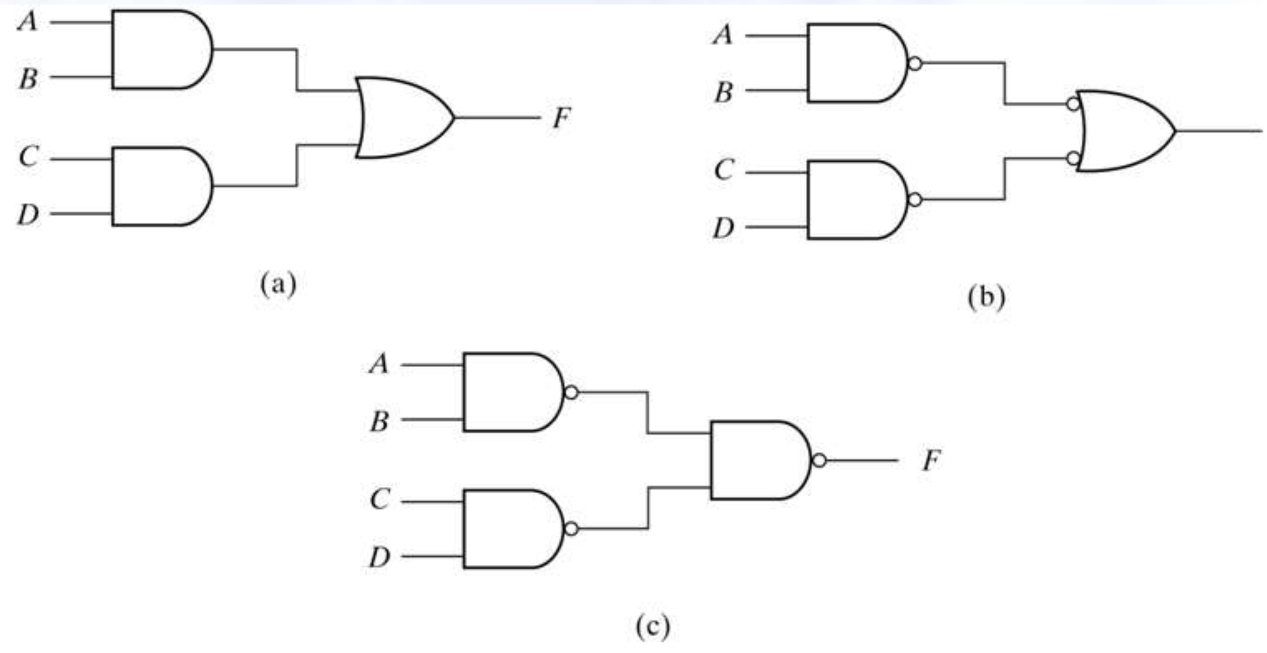


(a) AND–invert

(b) Invert–OR

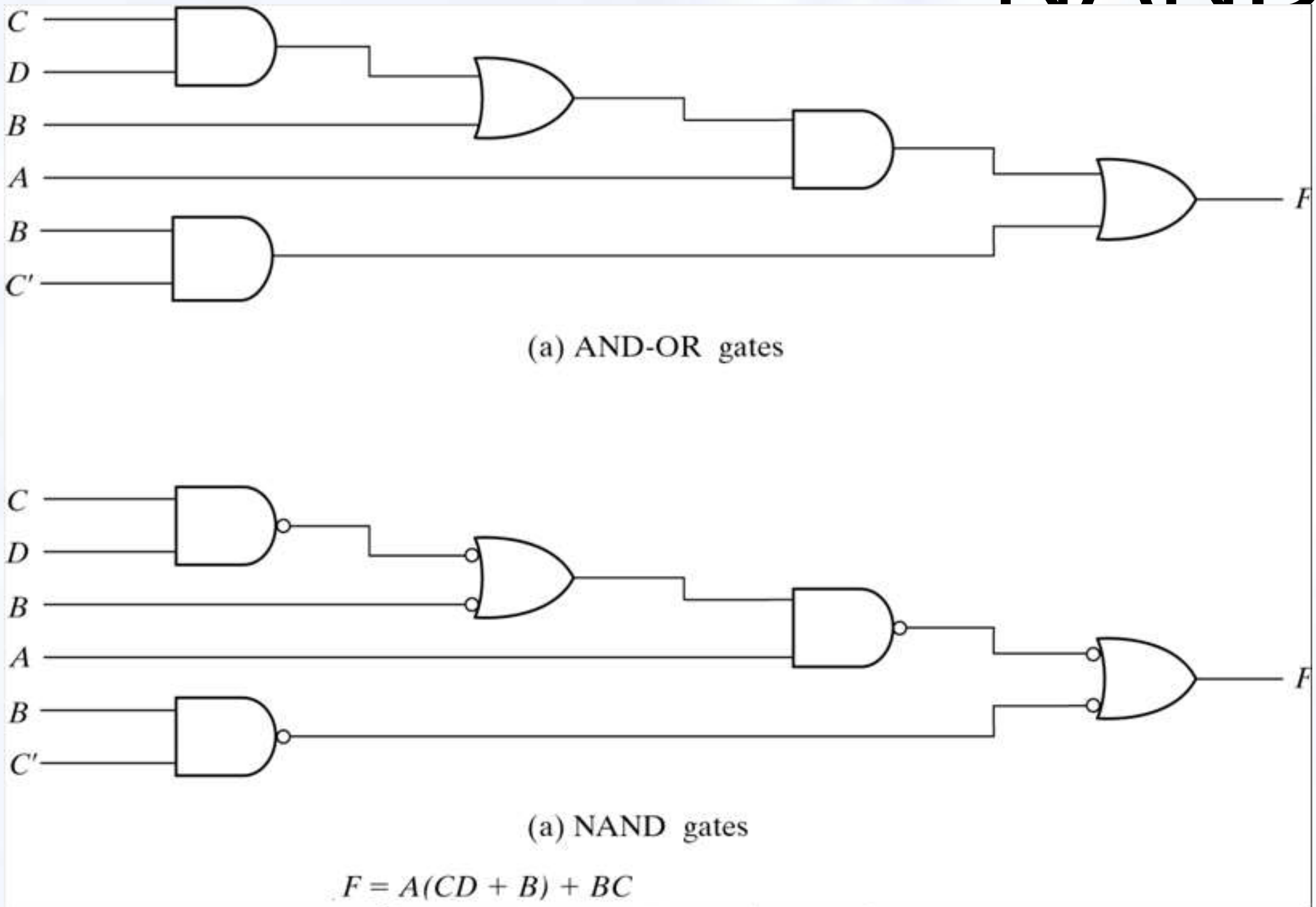
مثال

- تابع بولی زیر به دو صورت پیاده‌سازی شده است:



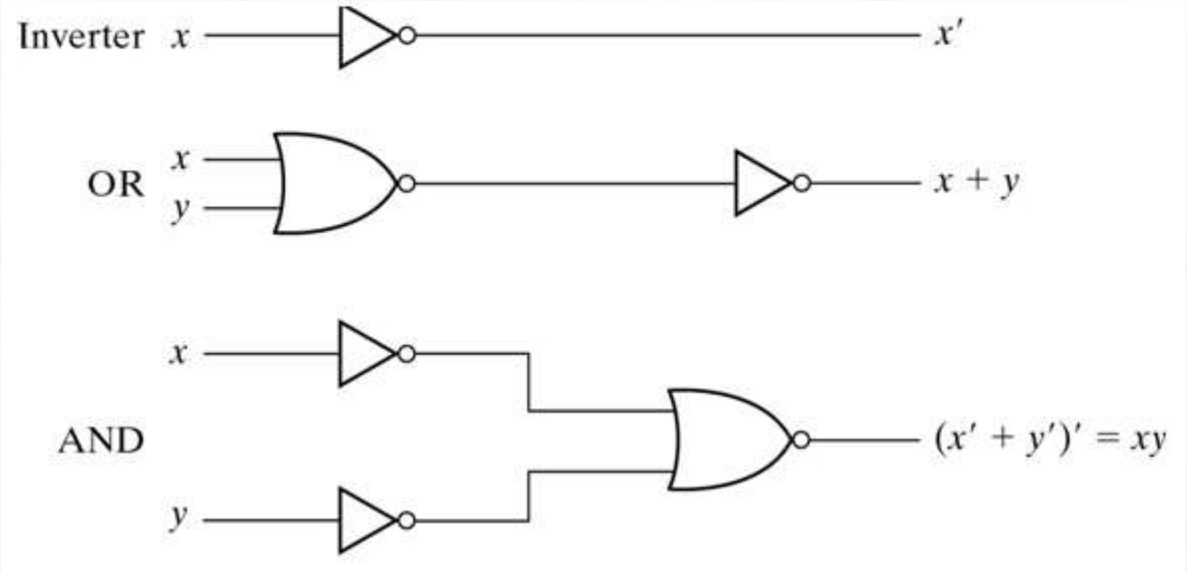
$F = AB + CD$

مثال: تبدیلی دیگر به مداری تنها با NAND

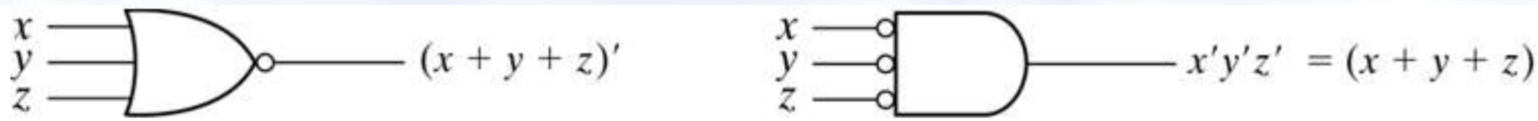


مداری تنها با گیتهای NOR

- می‌توان هر مدار منطقی که با هر نوع گیتی ساخته شده باشد را به مداری تبدیل کرد که تنها از گیتهای NORتشکیل شده باشد. با استفاده از شکلهای زیر، بجای هر گیت معادل آنرا قرار داده و سپس آنرا ساده می‌کنیم.



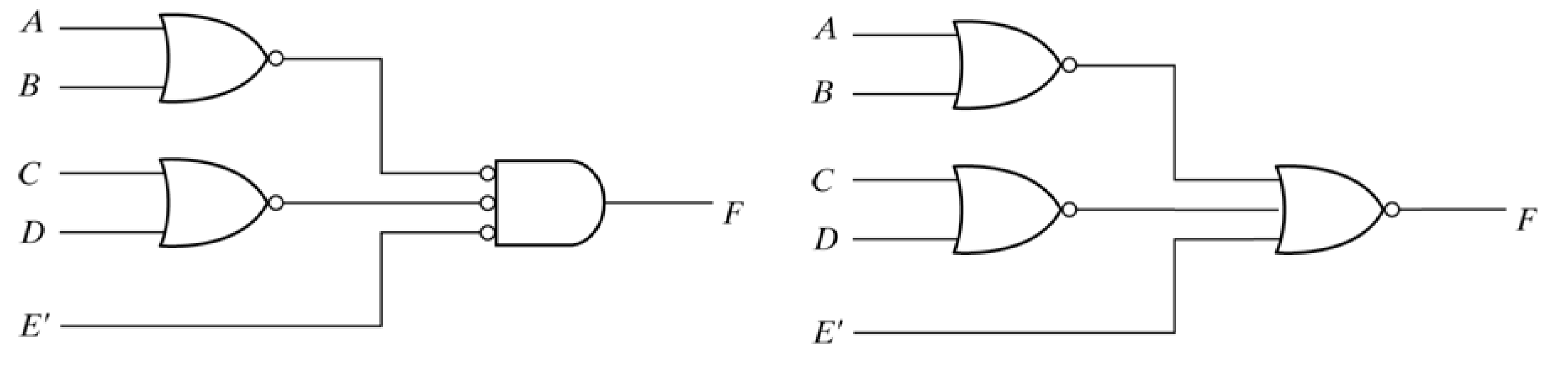
- در هنگام تبدیل مدارها از گیتهای معادل زیر هم می‌توان کمک گرفت:



(a) OR–invert (a) Invert–AND

مثال

- تابع بولی زیر به دو صورت پیاده‌سازی شده است:



$F = (A + B)(C + D)E$

توابع XOR و XNOR

- جداول کارنو برای دو تابع XOR و XNOR در زیر آورده شده‌اند.

A \ BC	00	01	11	10
0		1		1
1	1		1	

توازن فرد

$F = A \oplus B \oplus C$

A \ BC	00	01	11	10
0	1		1	
1		1		1

توازن زوج

$F = (A \oplus B \oplus C)'$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

توازن فرد

$$F = A \oplus B \oplus C \oplus D$$

AB \ CD	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

توازن زوج

$$F = (A \oplus B \oplus C \oplus D)'$$

مدارهاي جمع‌كننده و ضرب كننده

روش جمع کردن دو عدد بصورت دستی

- دو عدد دودوئی همانند اعداد دهدهی از راست بـه چپ رقم به رقم جمع می شوند.
- قواعد جمع اعداد دودوئی به صورت زیر است:

1+1=0

1+0=1

0+0=0

The initial carry
in is implicitly 0

	1	1	1	0		رقم نقلي ورودي
		1	0	1	1	جمع شونده
+		1	1	1	0	جمع كننده
	1	1	0	0	1	حاصل جمع

با ارزش ترين بيت

كم ارزش ترين بيت

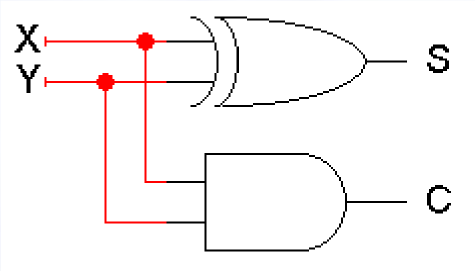
جمع دوبیت

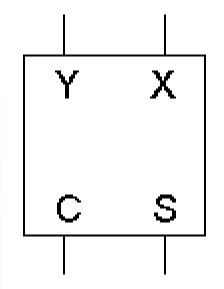
- برای ساخت یک تمام جمع کننده ابتدا بایـد یـک نیم جمع کننده که فقط ۲ بیت را جمع مـی کنـد ساخته شود.در این نیم جمع کننده دو بیت حاصل جمع وجود دارد کـه یکـی sumودیگـری carry out رقم نقلی خروجی نامیده می شود.

- در زیر جـدول صـحت ،دیـاگرام منطقـی و روابـط

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10





$C = XY$

$$S = X'Y + XY' = X \oplus Y$$

جمع سه بیت

- اما چیزی که واقعاً لازم است جمع ۳بیت ؛که یـک بیت آن مربوط به جمـع کننـده ،یـک بیـت جمـع شوند و یک بیت رقم نقلی ورودی است.

X	Y	C_{in}	C_{out}	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

0 + 0 + 0 = 00
0 + 0 + 0 = 01
0 + 1 + 0 = 01
0 + 1 + 1 = 10
1 + 0 + 0 = 01
1 + 0 + 1 = 10
1 + 1 + 0 = 10
1 + 1 + 1 = 11

	1	1	1	0	
		1	0	1	1
+		1	1	1	0
	1	1	0	0	1

رابطهٔ تمام جمع‌کننده

- یک تمام جمع کننده ۳بیت را جمع می کند و دو بیـــت خروجـــی حاصـــل جمـــع(S) و رقـــم نقلـــی خروجی(cout) را تولید می کند

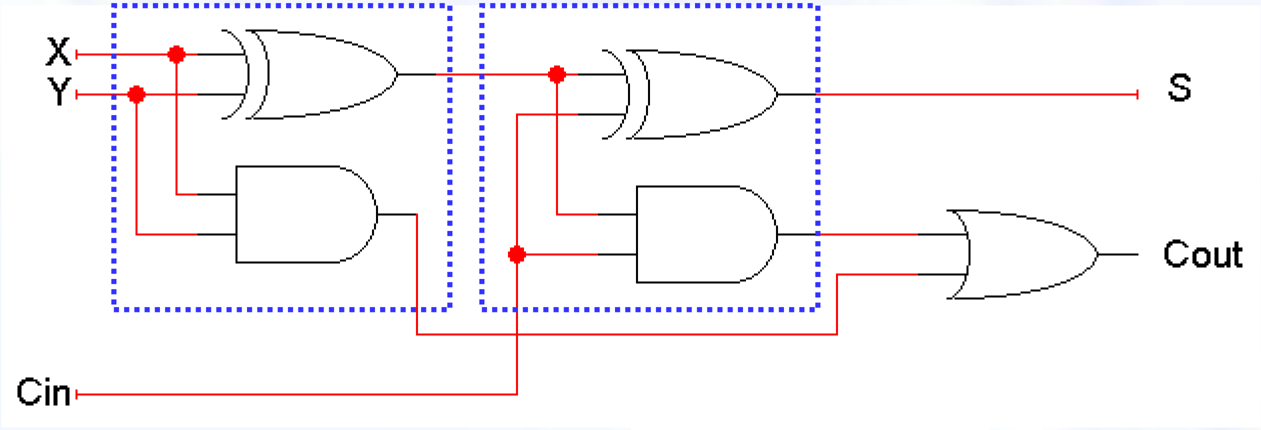
X	Y	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

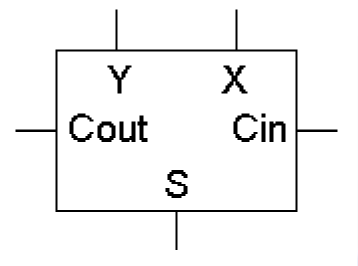
$$\begin{aligned}
S &= \Sigma m(1,2,4,7) \\
&= X' Y' C_{in} + X' Y C_{in}' + X Y' C_{in}' + X Y C_{in} \\
&= X' (Y' C_{in} + Y C_{in}') + X (Y' C_{in}' + Y C_{in}) \\
&= X' (Y \oplus C_{in}) + X (Y \oplus C_{in})' \\
&= X \oplus Y \oplus C_{in}
\end{aligned}$$

$$\begin{aligned}
C_{out} &= \Sigma m(3,5,6,7) \\
&= X' Y C_{in} + X Y' C_{in} + X Y C_{in}' + X Y C_{in} \\
&= (X' Y + X Y') C_{in} + XY(C_{in}' + C_{in}) \\
&= (X \oplus Y) C_{in} + XY
\end{aligned}$$

مدار تمام جمع‌کننده

- همانطور که از عبارت قبلی مشخص است با استفاده از ۲ نیم جمع کننده می توان یک تمام جمع کننده طراحی کرد.





$$S = X \oplus Y \oplus C_{in}$$
$$C_{out} = (X \oplus Y)\, C_{in} + XY$$

یک جمع کنندۀ ۴ بیت

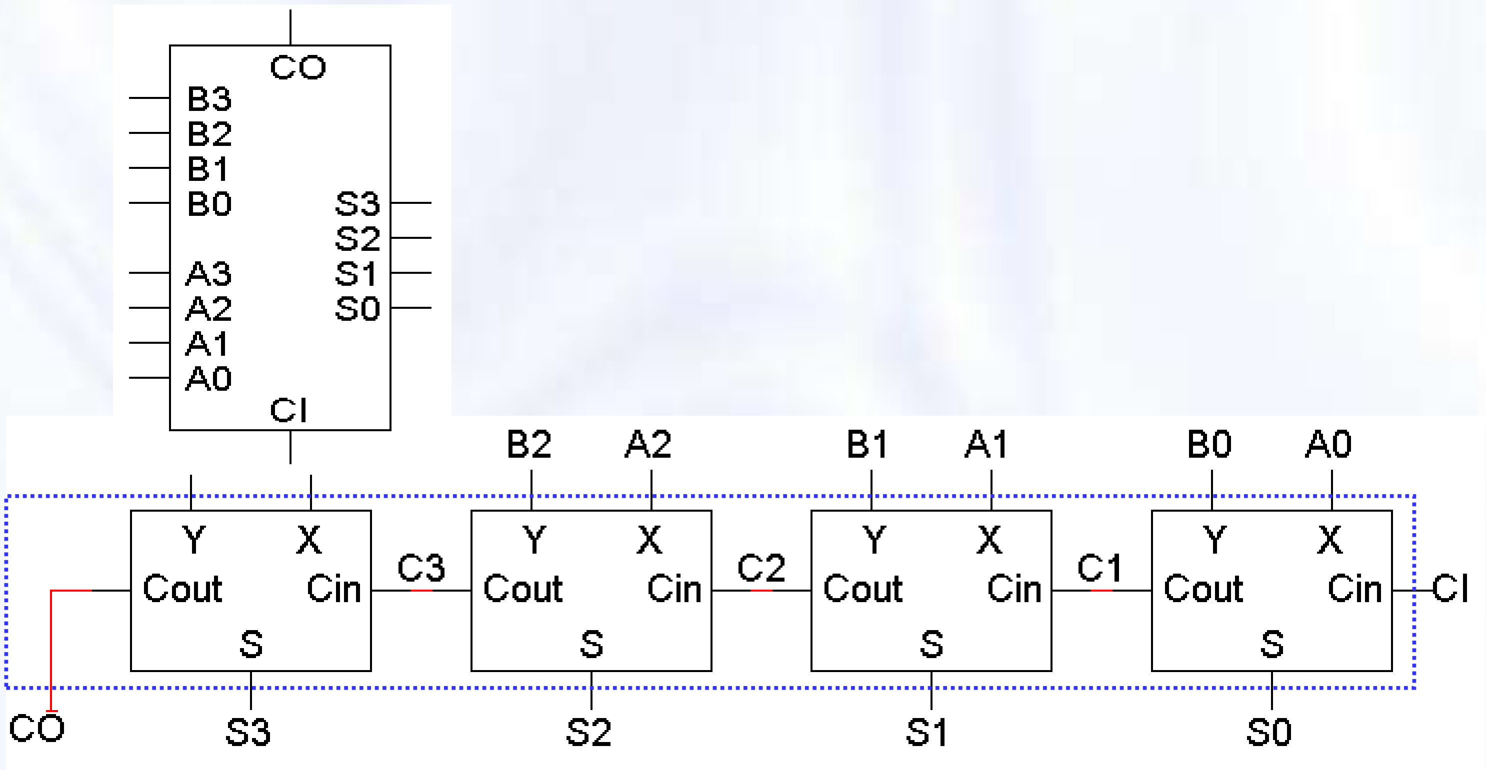
- با استفاده از ۴تمام جمع کننده در کنار یکدیگر ۴ می توان یک جمع کننده ۴بیتی ساخت.

- در یک جمع کننـده ۴ بیتـی در مجمـوع ۹بیـت ورودی وجود دارد:

  –دو عــدد ۴ بیتــی کــه قــرار اســت جمــع شوند($B_3B_2B_1B_0$و$A_3A_2A_1A_0$)

  –یک رقم نقلی ورودی اولیهC_{IN}

یک جمع کننده ۴بیتی



یک مثال از یک جمع کننده ۴بیتی

1 1 1 0 1 1 0 1

- دو عدد ۴بیتی A=1011(۱۱در مبنای ده)وB=1110(۱۴در مبنای ده)

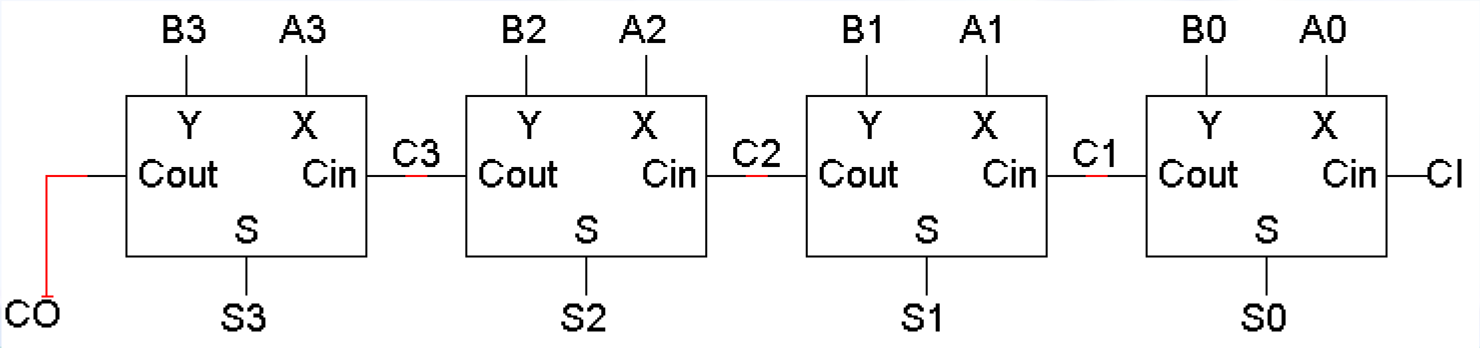
1

0

۱.بیت های دو عدد ۴بیتی به همراه رقم نقلی بر روی ورودی ها قرا ر می گیرند

۲.تمام جمع کننده مرتبه اول با استفاده از A0،B0وCinبیت های s0وc1را تولید می کند

۳.با استفاده از c1می توان s1و c2را بدست آورد با استفاده از c2می توان c3وs3را پیدا کرد



نکات سریزی

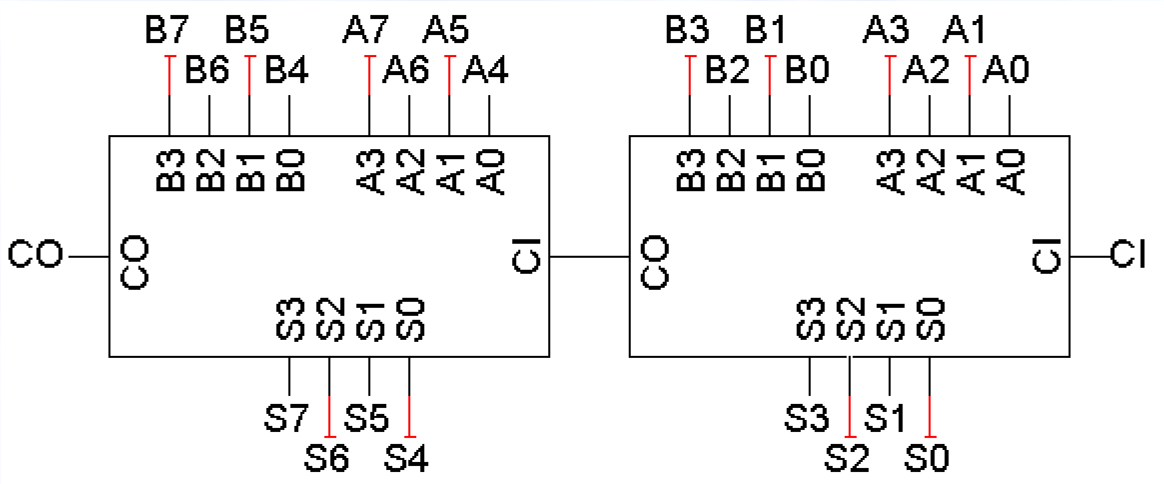
- همانطور که مشاهده می شود . حاصل جمع یک بیت اضافه تر ازجمع کننده و جمع شونده دارد به این پدیده سر ریز (overflow)می گویند.

- به طور کلی سر ریز برای اعداد بدون علامت دودوئی زمانی رخ می دهد که$c_{out}=1$باشد.

طراحی سلسله مراتبی جمع‌کننده

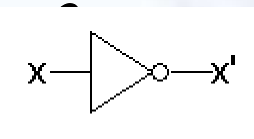
- اگر دقت کنید متوجه می شوید که رقم نقلی ورودی در جمع دو عدد ۴بیتی برابر با ۰می باشد.چه لزومی دارد که رقم نقلی ورودی در جمع کننده ۴بیتی وجود داشته باشد؟
- دلیل استفاده از رقم نقلی ورودی تولید جمع کننده های بزرگتر(۸بیتی،۱۲بیتی و...) با استفاده از جمع کنند ه های ۴بیتی می باشد.
- به عنوان مثال روش ساخت جمع کننده های ۸بیتی با استفاده از جمع کننده های ۴بیتی به

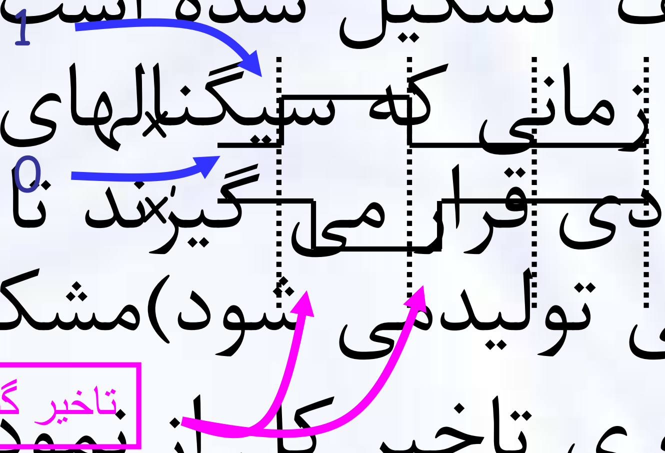
- ساخت یک جمع کننده ۸-بیتی با استفاده جمع کننده های ۴-بیتی



تاخیر گیت ها

- در حالت طبیعی هر گیت برای تولید خروجی در صورت قرار گرفتن خروجی بر روی آن دارای مقداری تاخیر است.
- در یک مدار منطقی که از گیت های متعدد در سطوح مختلف تشکیل شده است محاسبه تاخیر کل (یعنی از زمانی که سیگنالهای ورودی پایه های ورودی قرار می گیرند تا زمانی خروجی نهایی تولیدمی شود)مشکل می باشد.





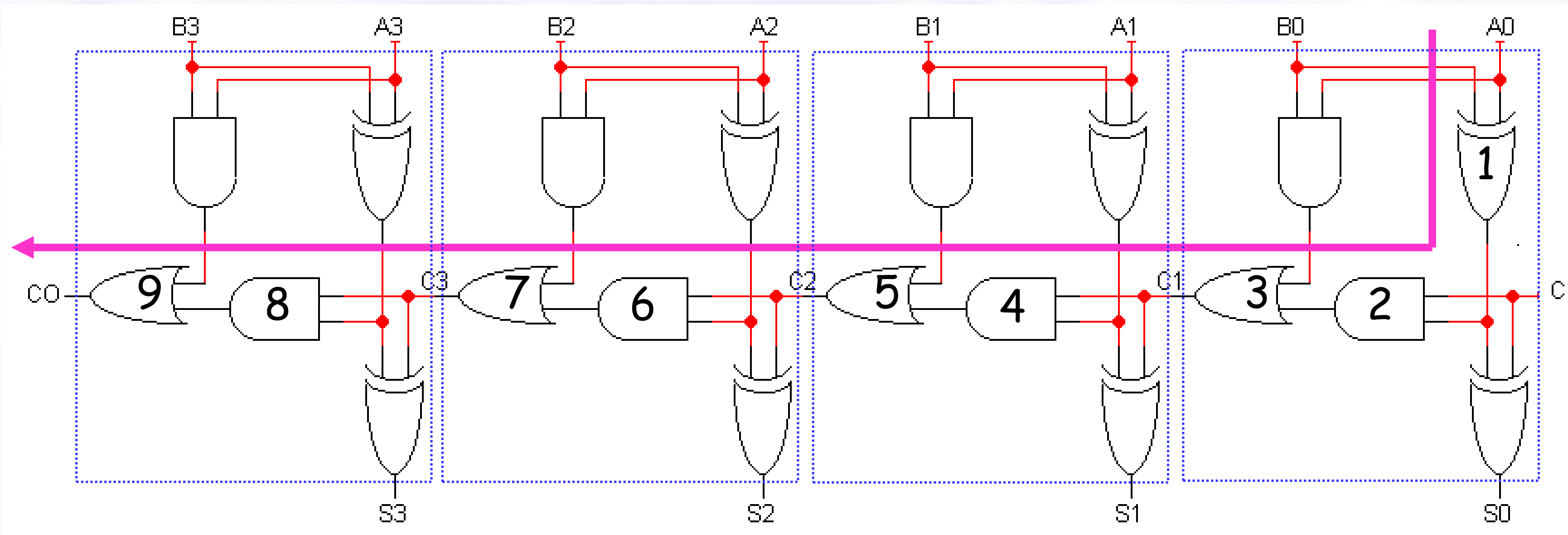
- برای محاسبه ی تاخیر کل از نمودار زمانی استفاده می شود

تأخیر بیت نقلی در جمع کننده

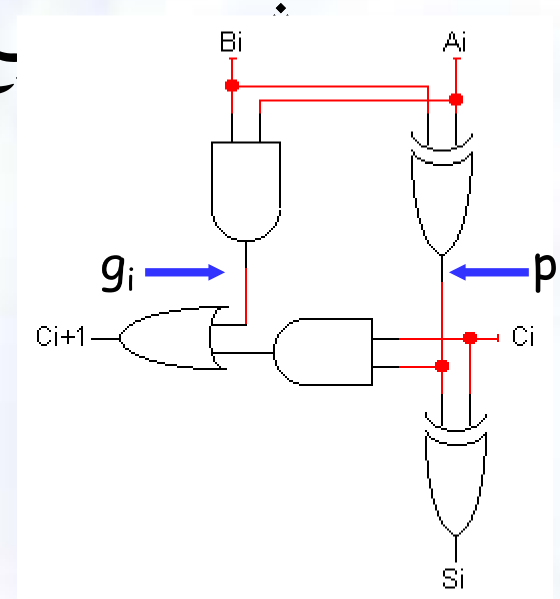
- در دیاگرام زیر یک جمع کننـدهٔ ۴بیتـی بـا گیـت های منطقی داخلی آن نمایش داده شده است.
- به این جمع کننده یک جمع کنندهٔ بـا رقـم نقلـی منتشر شونده می گویند.چرا که بـرای تولیـد رقـم نقلی C_{out}و بیت حاصل جمع S_3با یـد رقـم نقلـی Cin،A_0وB_0در طول مدار منتشر شوند.
- جمع کننده با رقم نقلی منتشر شونده بسـیار کنـد است:
  - برای جمع دو عدد ۴بیتی در مثال فـوق در ۵ مرحله انجام می گیرد.

یک جمع کننده ی ۴-بیتی با رقم نقلی منتشر شونده

تاخیر برای تولید حاصلجمع نهایی در جمع کننده های با رقم نقلی منتشر شونده بسیار زیاد است.



ع برای محاسبه بیت نقلی



- به جای انتظار برای انتشار رقم رنقلی خروجی از تمام سطوح قبلی می توان برای کم کردن تاخیر رقم نقلی را از قبل محاسبه کرد.
- برای انجام اینکار ابتدا ۲ تابع را به صورت زیر محاسبه می کنیم
- تابع g_i دارای مقدار ۱ است هر گاه یک رقم نقلی خروجی باید از تمام جمع کننده سطح

A_i	B_i	C_i	C_{i+1}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- بنابراین رقم نقلی خروجی را می توانیم به صورت زیر بازنویسی کنیم:

$$c_{i+1} = g_i + p_i c_i$$

محاسبهٔ رقم نقلی خروجی نهایی با استفاده از توابع p_i و g_i

برای محاسبه رقم نقلی خروجی بر حسب ورودی های A(0-3) و B(0-3) ورقم نقلی ورودی Cin با استفاده از توابع pi و gi مراحل زیر را دنبال می کنیم:

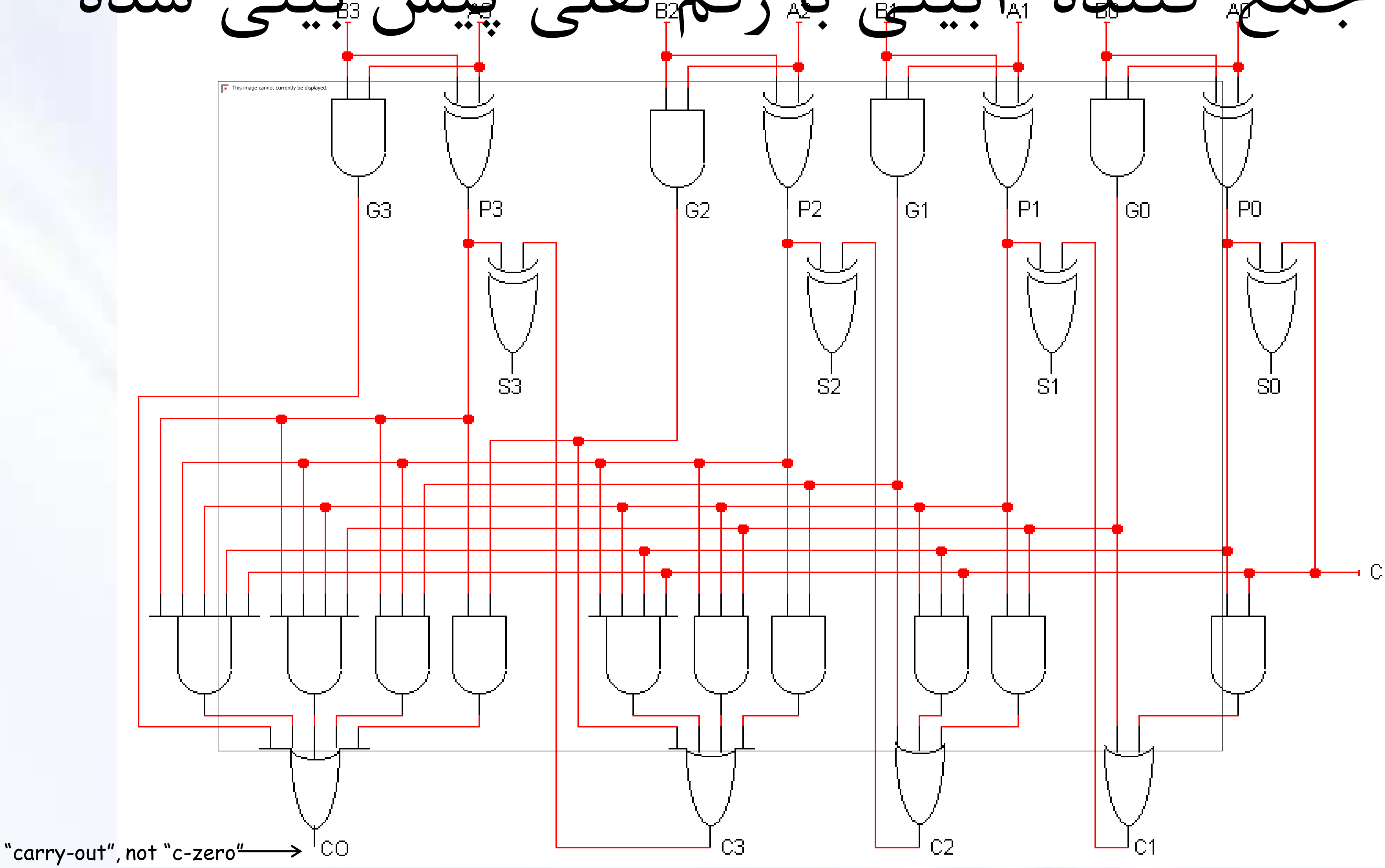
$$c_1 = g_0 + p_0c_0$$

$$\begin{aligned} c_2 &= g_1 + p_1c_1 \\ &= g_1 + p_1(g_0 + p_0c_0) \\ &= g_1 + p_1g_0 + p_1p_0c_0 \end{aligned}$$

$$\begin{aligned} c_3 &= g_2 + p_2c_2 \\ &= g_2 + p_2(g_1 + p_1g_0 + p_1p_0c_0) \\ &= g_2 + p_2g_1 + p_2p_1g_0 + p_2p_1p_0c_0 \end{aligned}$$

$$\begin{aligned} c_4 &= g_3 + p_3c_3 \\ &= g_3 + p_3(g_2 + p_2g_1 + p_2p_1g_0 + p_2p_1p_0c_0) \\ &= g_3 + p_3g_2 + p_3p_2g_1 + p_3p_2p_1g_0 + p_3p_2p_1p_0c_0 \end{aligned}$$

جمع کننده ۴ بیتی با رقم نقلی پیش بینی شده



جمع کننده های با رقم نقلی پیش بینی شده

- به مدار طراحی شدهٔ فوق یک جمع کننده با رقم نقلی پیش بینی شده می گویند.
- با اضافه کردن گیت های اضافی تعداد سطوح مدار را به ۲ کاهش دادیم و سرعت آنرا افزایش دادیم.
- می توان با استفاده از جمع کننده های با رقم نقلی پیش بینی شده جمع کننده‌های بزرگتر با همان روش قبلی تولید کرد.
- جمع کننده های با رقم نقلی پیش بینی شده

جمع کننده های با رقم نقلی پیش بینی شده

- تاخیریک جمع کننده ی با رقم نقلی پیش بینی کننده بصورت لگاریتمی رشد می‌کند در حالی که تاخیر در جمع کننده های با رقم نقلی منتشر شونده به صورت خطی.

- نتیجه اینکه جمع کننده های با رقم نقلی منتشر شونده طراحی ساده ای دارند اما کندترند.درحالی که جمع کننده های بارقم نقلی پیش بینی کننده

ضرب کننده ها

- ضرب کننده ها را می توان از روی جمع کننده ها ساخت.
- ضرب دو عدد یک بیتی همان AND دو بیت است:

a	b	a×b
0	0	0
0	1	0
1	0	0
1	1	1

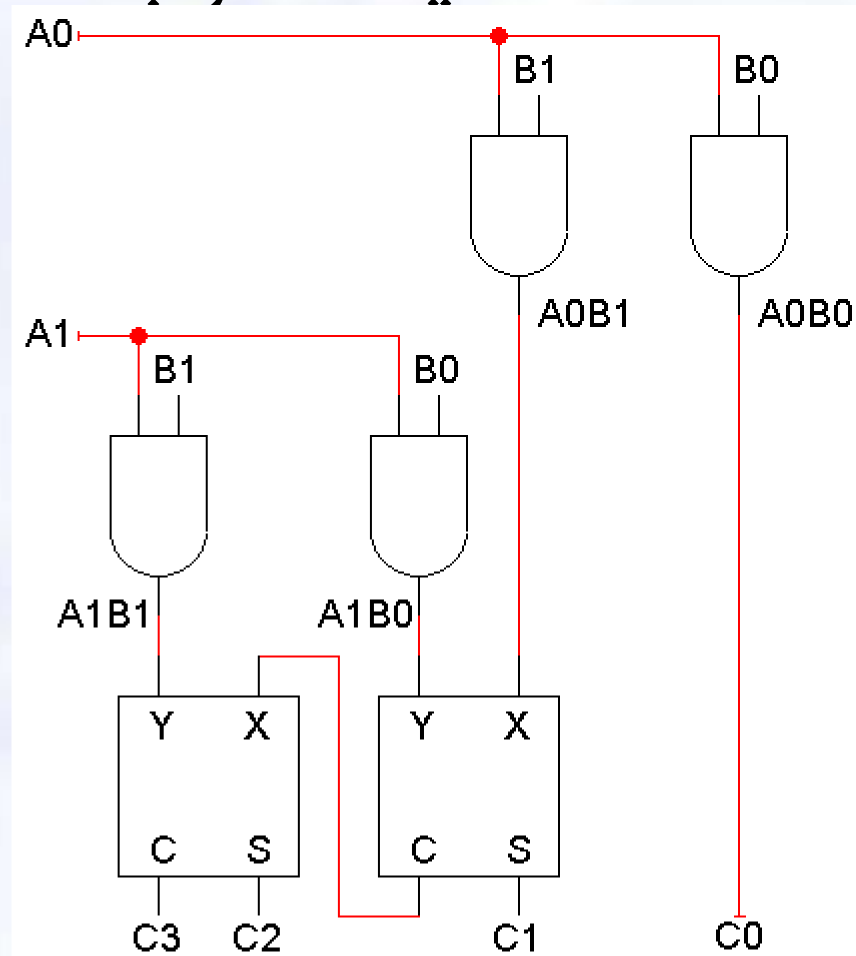
a	b	ab
0	0	0
0	1	0
1	0	0
1	1	1

ضرب‌کننده باینری

				1	1	0	1	مضروب فیه
			×	0	1	1	0	مضروب
				0	0	0	0	partial product
			1	1	0	1		
		1	1	0	1			
+	0	0	0	0				
	1	0	0	1	1	1	0	حاصل ضرب

- چهار مقدار حاصل از هر مرحله، باید با هم جمع شوند تا حاصل ضرب نهائی بدست آید.
  - می توان آنها را جفت جفت با ۳جمع کننده جمع کرد.
  - با استفاده از جمع کننده های ۴بیتی می توان حاصلجمع نهایی را اگرچه حداکثر دارای ۸ بیت است را با شیفت دادن یک بیت **partial product**

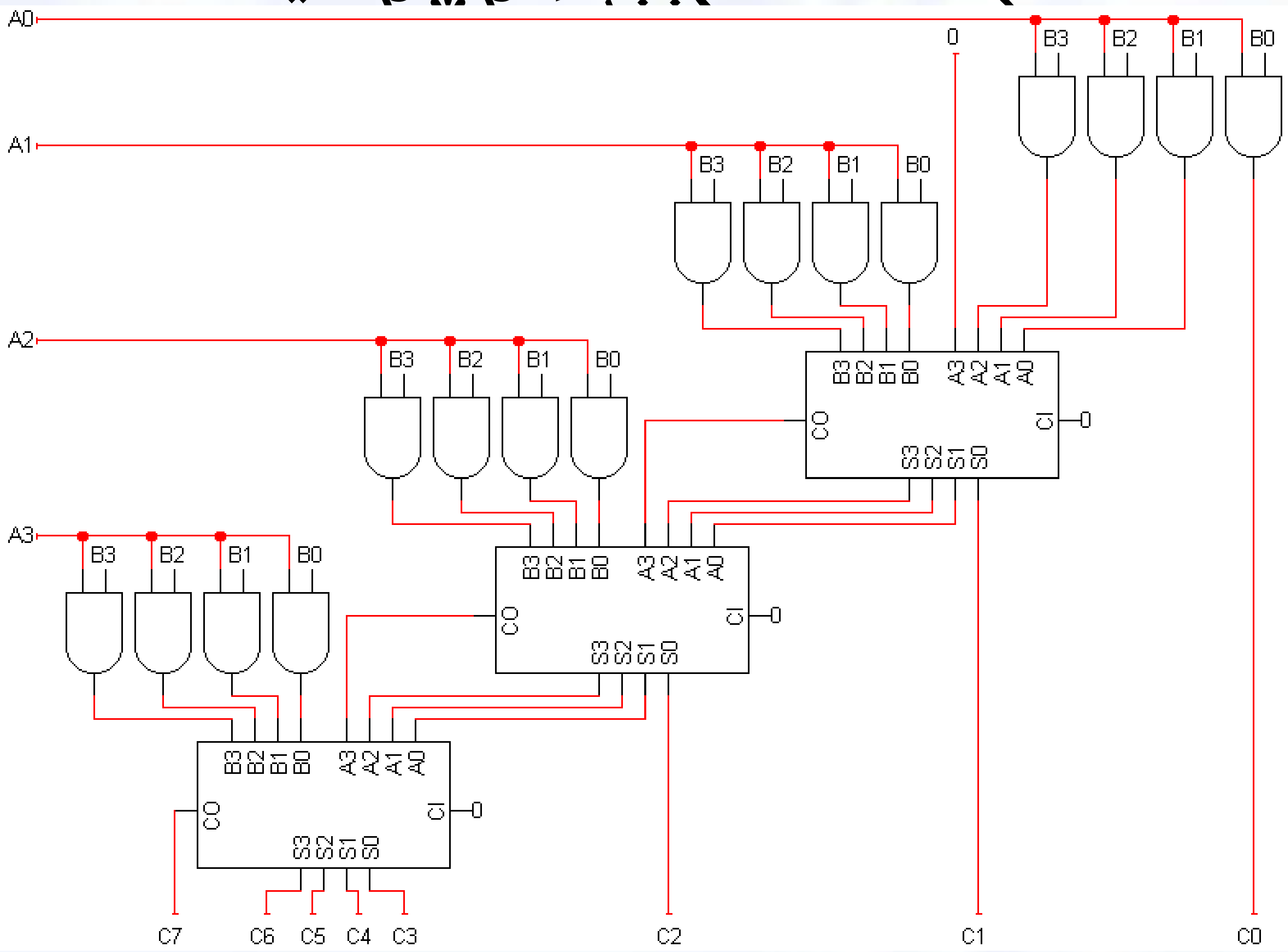
یک ضرب کنندهٔ دو بیتی



- گیتهای andحاصلضرب جزئی(partial product) را تولید می کنند.

$$\begin{array}{rrrrr}
 & & & B_1 & B_0 \\
 & \times & & A_1 & A_0 \\
\hline
 & & & A_0B_1 & A_0B_0 \\
+ & & A_1B_1 & A_1B_0 & \\
\hline
 & C_3 & C_2 & C_1 & C_0
\end{array}$$

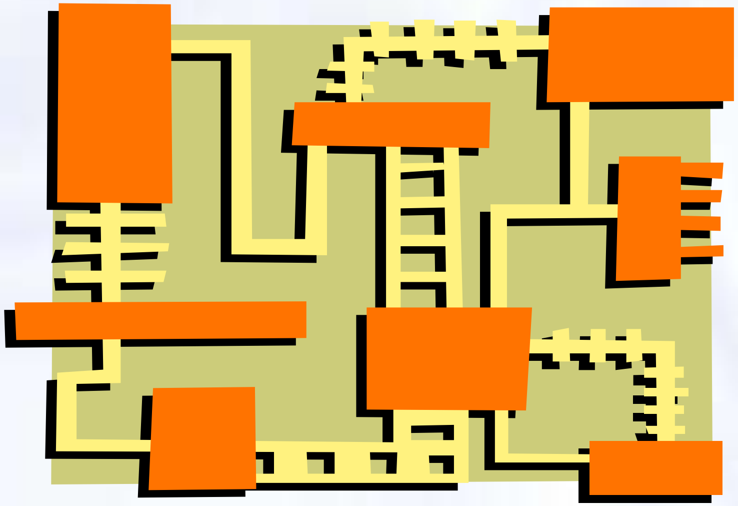
- برای یک ضرب کنندهٔ ۲-بیتی ما باید از ۲نیم جمع کننده


A0
A1
A2
A3
B3 B2 B1 B0
0
B3 B2 B1 B0 A3 A2 A1 A0
CO
CI
S3 S2 S1 S0
C7
C6 C5 C4 C3
C2
C1
C0

نکاتی در مورد ضرب کننده ها

- توجه کنید که یک ضرب کنندهٔ ۴-بیتی حداکثر یک نتیجهٔ ۸-بیتی تولید می کند. بنابراین در مواردی که جواب کوچکتر از ۸-بیت می باشد می توان:
  - ما می توانیم هر ۸-بیت را حفظ کنیم
  - ویا،با دور ریختن بیت های اضافی فقط وجود سرریز را بررسی کنیم.

– مدار لازم برای یک جمع کنندهٔ ۳۲بیتی و یا ۶۴بیتی بسیار بزرگ است.



ضرب کننده برای یک حالت خاص

- در ضرب اعداد هدهی برای ضرب در عدد ۱۰ فقط کافی است مضروب فیه را یک واحد به چپ شیفت دهیم.

128 x 10 = 1280

- اینکار را می توان در ضرب اعداد دودوئی انجام داد،شیفت به سمت چپ معادل ضرب در عدد ۲ می باشد.

- دو بار شیفت به سمت راست معادل ضرب در عدد ۴ می باشد

11 x 100 = 1100 (in decimal, 3 x 4 = 12)

- شیفت به راست معادل تقسیم بر ۲است.

110 ÷ 10 = 11 (in decimal, 6 ÷ 2 =

مدارهاي تركيبي

مدارهای منطقی

- کلاً مدارهای منطقی دو دسته‌اند:
- **مدارهای ترکیبی**: شامل گیتهای منطقی است و خروجیهای آن در هر زمان مستقیماً از روی ترکیب فعلی ورودیها بدون توجه به ورودیهای قبلی تعیین می شوند.

- **مدارهای ترتیبی**: علاوه بر گیتهای منطقی از عناصر حافظه نیز در آنها استفاده می‌شود. خروجیهای آن تابعی از عناصر حافظه و ورودی‌های مدار هستند. همچنین حالت عناصر حافظه نیز تابعی از ورودی‌های قبلی است.

- فرم کلی مدارهای ترکیبی بصورت زیر است. مدار می‌تواند n ورودی و m خروجی داشته باشد.