«بنام خدا»

DATE / / SUBJECT:

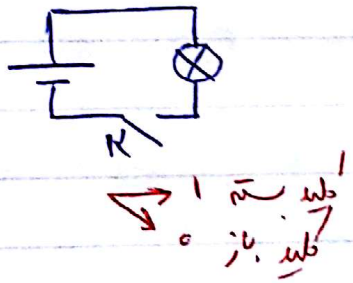
«مدار منطقی»

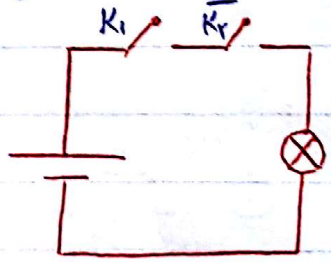
مدار: یک سیکل بسته ای که دارای یک مصرف کننده حداقل و یک منبع تغذیه باشد بگونه ای که جریانی داخل مدار جاری شود و منبع و مصرف کننده در یک محیط واقع گردند. این سیکل بسته به کمک هادی رسانا باشد که جریان از منبع به مصرف کننده جاری شود.

نکته: اگر مصرف کننده و منبع از وسایل الکتریکی تشکیل شده باشند به آن مدار الکتریکی می گویند.

نکته: " " " الکترونیکی " " الکترونیکی می گویند.

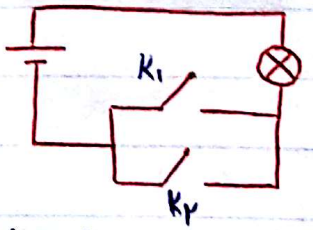
منطقی: عملگر منطقی عملگری است که بین دو عملوند یک کار منطقی انجام می دهد.





عملوند

K۱ and K۲



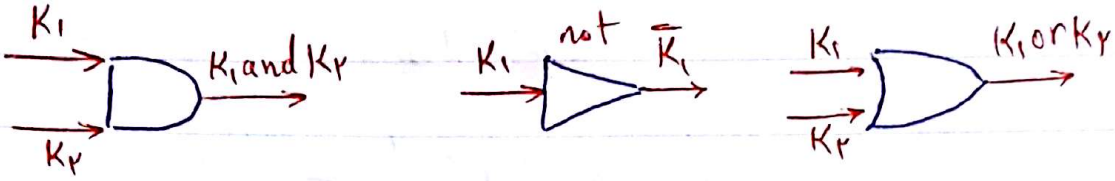
K۱ or K۲

K۱	K۲	K۱ and K۲	K۱ or K۲
۱	۱	۱	۱
۱	۰	۰	۱
۰	۱	۰	۱
۰	۰	۰	۰

Sahand

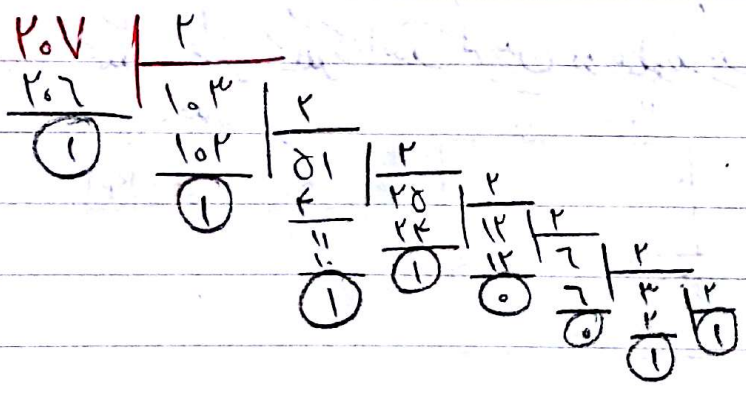
Scanned by CamScanner

DATE / / SUBJECT:



سیستم های عددنویسی در دنیای ریاضی ما بی نهایت است که تعدادی از این سیستم ها بعنوان استاندارد دنیای واقعی کاربرد دارد مهمترین آنها عبارت است از سیستم عددنویسی مبنای ۱۰ (یعنی ده استاندارد بعنوان در مجموعه ای بنام مضروفات

اول از مبنای ۱۰ به ۲ و بعد ۲ به ۱۶



۱۱۰۰۱۱۱۱

C F

Scanned by CamScanner

DATE 95/7/14 SUBJECT:

اعداد

* تبدیل مبناهای دارای اعشاری :

عنوان مثال عدد ده دهی اعشاری برای تبدیل شدن به دودویی ، اول قسمت صحیح عدد تقسیم های متوالی به ۲ و... و قسمت اعشار عدد ضرب های متوالی به دو به اینصورت که عدد اول پس از ضرب عدد یک بصورت صحیح بیرون آمد عدد یک را وارد محاسبه نمیکنیم صفر.

۲۰۷,۷۰۲

↓

۱۱۰۰۱۱۱۱,۱۰۱۱

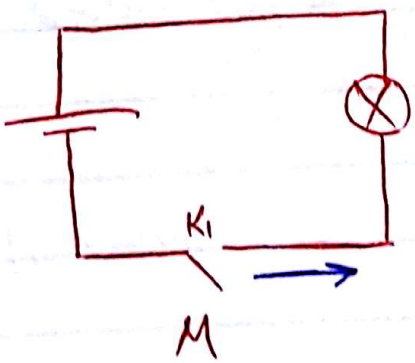
۰,۷۰۲ × ۲ = ①۴۰۴

۰,۴۰۴ × ۲ = ⓪۸۰۸

۰,۸۰۸ × ۲ = ①۶۱۶

۰,۶۱۶ × ۲ = ①۲۳۲

* مدار چند کلیدهای مانند دوم عزضی این کلیدها به سمت مصرف کننده میرود است.

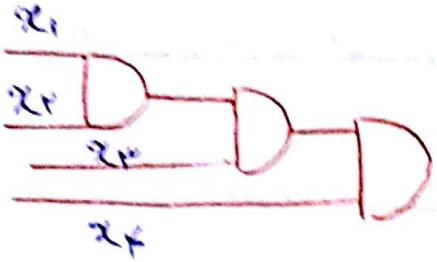


M	K_1	K_1'
1	1	0
0	0	1

Scanned by CamScanner

DATE / / SUBJECT:

and : گیت دو ورودی



جبر بول: سیستم ریاضی‌ای است که بر پایه‌ی مجموعه‌ای از عناصر، عملگرها و تعدادی بدیهیات و مفروضات تعریف می‌شود. استاندارد هانتینگتون مجموعه‌ای بسته‌ای مانند S معرفی می‌شود که در مجموعه S عملگرها و عملوندهای مختلف وجود دارند. عملیات ما در جبر بول در مفروضات تعریف می‌شود در جبر بول اصول اولیه سیستم ریاضی قبل بررسی و اثبات است.

① اصل بسته بودن: مجموعه‌ای مانند S نسبت به عملگر دودویی (+ و ×) یا (∧ و ∨) بسته است به شرط آنکه برای دو عضو از مجموعه S با این عملگر عضو منحصر بفردی را به خود اختصاص دهد.

مثلاً مجموعه اعداد طبیعی مثل $N = \{1, 2, 3, \dots\}$ نسبت به + بسته است چون $1 + 2 = 3$

اما $N = \{1, 2, 5\}$ نسبت به + بسته نیست چون $3 + 2 = 5$

② اصل جابه‌جایی:

Scanned by CamScanner

DATE / / SUBJECT:

③ اصل شرکت پذیری: یعنی اگر در مجموعه ای مانند S سه مجموعه دلخواه x, y, z را برداریم شرط

$\{x, y, z\} \subseteq S = \{+, \cdot, \prime\}$

$\hookrightarrow (x*y)*z = x*(y*z)$

در کلیه ساختارهای جبری مانند اصل بسته بودن و اصل شرکت پذیری و... جزو بدیهیات هستند

جبر بول: جبر بول یک ساختار جبری است که با عناصر مجموعه ای مانند B همراه با دو عمل جمع و ضرب دارای اصول زیر است:

① مجموعه ای نسبت به عمل جمع بسته باشد و همچنین آن مجموعه نسبت به عمل ضرب هم بسته باشد.

$B = \{1, 0\}$

$x \in B \quad 1+0 = 0+1 = 1$ (جابجایی)

$y \in B$

$\forall x, y \in B, \begin{cases} (x+y) = (y+x) \\ (x \cdot y) = (y \cdot x) \end{cases} \in B$

② عضو خنثی در مجموعه برای جمع برابر با صفر و عضو خنثی برای ضرب برابر با یک است.

$x+0 = x \qquad x \cdot 1 = x$

③ نسبت به عمل جمع یا ضرب دارای جابجایی است.

Scanned by CamScanner

DATE / / SUBJECT:

④ عمل ضرب روی جمع کپنی پذیر است.

* خاصیت کپنی با سه متمم قابل بررسی است البته به این معنا نیست که سه متمم در جبر بول نیاز به سه متمم باشد. قانون ما در جبر بول دارای (۱و۰) است و هم

⑤ متمم: اگر x متمم B باشد و x' هم متمم B باشد به عبارتی

$x + x' = 1$ ، $x \cdot x' = 0$

⑥ حداقل دو عضو مانند x و y موجود باشند بطوریکه x و y مخالف هم بوده و این رابطه $(x \neq y)$ برقرار باشد.

قضیه‌های اصلی و خواص جبر بول:

در جبر بول تعدادی (۴ اصل) وجود دارد و تعدادی هم تئوری. (a) در جمع

(b) در ضرب

۲ اصل	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
۵ اصل	(a) $x + x' = 1$	(b) $x \cdot x' = 0$
۱ تئوری	(a) $x + x = x$	(b) $x \cdot x = x$
۲ تئوری	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$

Scanned by CamScanner

DATE / / SUBJECT:

قضیه ۳ (قانون) $(x')' = x$

اصل ۳ جابجایی) (a) $x + y = y + x$ / (b) $xy = yx$

قضیه ۴ شرکت پذیری) (a) $x + (y + z) = (x + y) + z$ / (b) $x(yz) = (xy)z$

اصل ۴ توزیع پذیری یا پخشی) (a) $x(y + z) = xy + xz$ / (b) $x + yz = (x + y)(x + z)$

قضیه ۵ دمورگان) (a) $(x + y)' = x'y'$ / (b) $(xy)' = x' + y'$

قضیه ۶ جذب) (a) $x + xy = x$ / (b) $x(x + y) = x$

نکته: هم اصول و قضیه‌ها را جدول قابل اثبات است مثلاً قضیه ۱. (هر اصل و هر قضیه را دو روش قابل اثبات روش ۱ جدول اندرکی (۰و۱) روش ۲ با کمک اصل‌ها و قضیه‌های دیگر)

مثال) اثبات قضیه ۱ با کمک جدول هم ارزی $(x + y)' = x'y'$

x	y	$x + y$	$(x + y)'$	x'	y'	$x'y'$
۰	۰	۰	۱	۱	۱	۱
۰	۱	۱	۰	۱	۰	۰
۱	۰	۱	۰	۰	۱	۰
۱	۱	۱	۰	۰	۰	۰

برابر

Scanned by CamScanner

DATE / / SUBJECT:

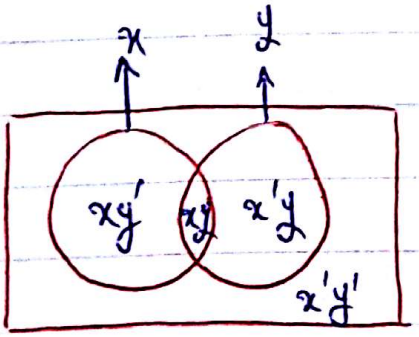
$x(y+z) = xy + xz$ (قانون پخشی)

x	y	z	$(y+z)$	xy	xz

نکته: در جبر بول تقدم اول در انجام پرانتز است. در یک عبارت بولی ابتدا پرانتز بررسی می‌شود. تقدم دوم با not است. تقدم سوم با and و تقدم چهارم با or است.

* یکی از روش‌های ساده برای تفهیم عبارت‌های بولی استفاده از دیاگرام ون است.

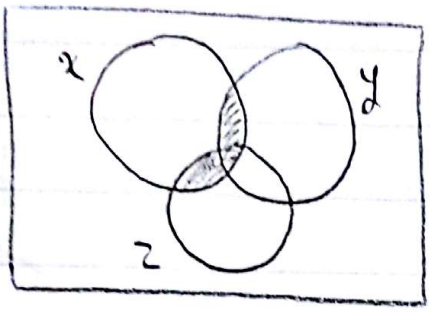
در این روش یک مجموعه‌ی مرجع که با مستطیل ▭ نمایش داده می‌شود یک مجموعه جهانی است و تمام عضوهای موجود در مجموعه آن است.



Scanned by CamScanner

DATE / / SUBJECT:

مثال) $x(y+z)$

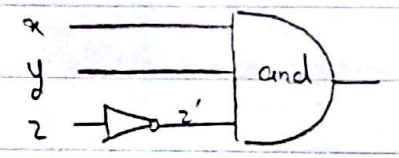


توابع بول: یک متغیر باینری (دودویی) می‌تواند یکی از دو مقدار ۰ یا ۱ را باشد.

تابعی دودویی است که عملگرهای and ، or و not را شامل می‌شود

به‌صورت کلی: ازای مقادیر مختلف از تابع بولی یک یا چند متغیر باینری حتماً باید از حروف بزرگ باشد

و می‌تواند اندیس داشته باشد.



x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	

$F_1 = xyz'$ $2^3 = 8$ تابع F_1 زمانی یک است که متغیرهای بولی و توابع عملیات‌ها (or ، not ، and) یک شوند

$F_2 = xy' + z$ $F_2 = 1 \Rightarrow \begin{cases} xy' = 1 \\ z = \begin{cases} 1 \\ 0 \end{cases} \end{cases}$

$\rightarrow F_2$

$\rightarrow F_1$

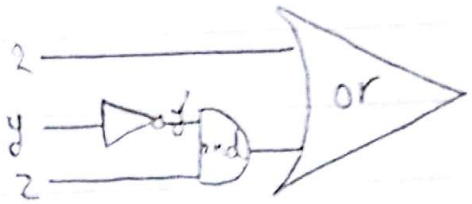
Scanned by CamScanner

DATE / / SUBJECT:

$F_3 = x + y'z$

x	y	z	F_3
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$F_3 = 1 \rightarrow$ ($x = 1$ → $y'z$ هر کدام) ; ($x = 0$ → $y'z = 1$ → $y = 0$, $z = 1$)



$F_4 = x'y'z + x'yz + xy'$

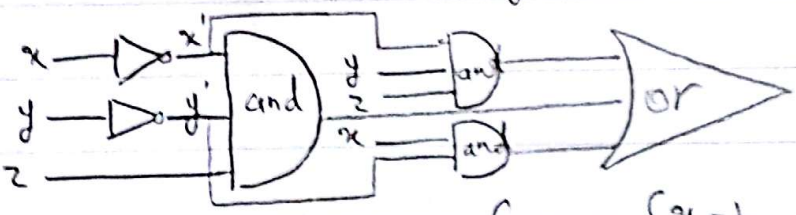
x	y	z	F_4	F_5
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

$F_5 = xy' + x'z$

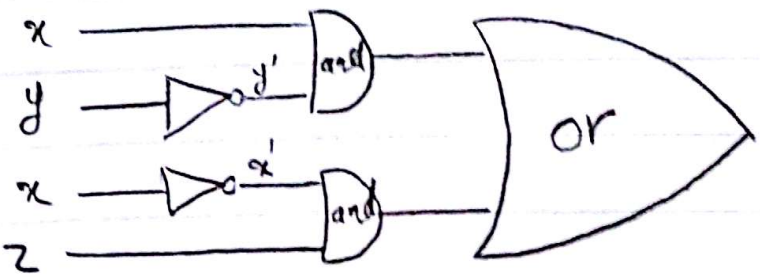
بر اساس جدول ۴ حالت دارد که x صفر است.

حالتی که $x = 0$ هر کدام برای کدام $y'z$ این است.

$F_4 = 1 \rightarrow$
- $x'y'z = 1$ → $z = 1$, $x = 0$, $y = 0$
- $x'yz = 1$ → $x = 0$, $y = 1$, $z = 1$
- $xy' = 1$ → $x = 1$, $y = 0$



$F_5 = 1 \rightarrow$
- $xy' = 1$ → $x = 1$, $y = 0$
- $x'z = 1$ → $x = 0$, $z = 1$



Sahand
Scanned by CamScanner

DATE / / SUBJECT:

* با توجه به اولویت ها عملیات های بولی شامل not ، or ، and و پرانتز درون تابع بولی ابتدا ارزش F مورد بررسی قرار می گیرد. بعنوان مثال در F_1 اگر ارزش تابع F_1 برابر یک گردد یعنی ارزش تابع من True باشد حالت های مختلف این متغیرها بررسی می کنیم یعنی تابع عملیات های تابع F_1 را پیدا کنیم.

* برای اینکه F_{sm} یک شود شرط آنست که x برابر با یک و y و z هرچه می خواهد باشد.

نکته: در تعریف تابع بولی همانطور که شد ابتدا براساس عبارت بولی تعداد متغیرها را در آن مشخص می کنیم مثلاً در F_2 ، 3 متغیر داریم ، در F_{sm} سه متغیر. (در هم آنها جداحساب نمی شود)

جدول ارزش تابع بولی را براساس تعداد متغیرها و استاندارد ها چیده و سپس از آن بررسی می کنیم که تابع بولی کی یک می شود. بدیهی است در تعیین حالت ها تابع صفر خواهد بود.

نکته: در رسم مدارات منطقی باید سعی کنیم که حداقل گیت ها استفاده شود.

Scanned by CamScanner

DATE / / SUBJECT:

مسئله: تابع بولی زیر را به کمک اصول و قضایا مینیم کنید. (مینیم کردن یک تابع بولی = این

مناسب است که از نظر تعداد متغیرها مینیم شود)

$F_1 = x + x'y \xrightarrow{\text{اصل ۴ b}} (x+x')(x+y)$

$1 \times (x+y) = (x+y)$

$F_2 = x(x'+y) \xrightarrow{\text{اصل پخشی}} xx' + xy = 0 + xy = xy$

$F_3 = x'y'z + x'yz + xy' \rightarrow x'z(\underbrace{y+y'}_{1}) + xy' = x'z + xy'$

$F_4 = xy + x'z + yz$

$F_5 = (x+y)(x'+z)(y+z)$

* یک متغیر بولی یا بدون پریم یا با پریم را لیترال گوئیم.

* وقتی که یک تابع بولی به توسط یکی گیت ها منطقی پیاده سازی می شوند هر لیترال در تابع (F) معرف یک ورودی است.

* مینیم کردن تعداد متغیرها و عملات در یک تابع بولی باعث می گردد که ساختمان مدارات با تعداد کمتری گیت ساخته شوند.

* وقتی مدار یک تابع را بخواهیم محاسبه کنیم لیترال ها مدار می شوند و عملیات ها نیز مدار می شوند.

(جمع = ضرب و ضرب = جمع بعنوان دوال پذیر است)

Sahand

Scanned by CamScanner

DATE / / SUBJECT:

مثال) مکمل تابع F_2 را بدست آورید.

$x(x'+y)$

$F_2' = (x(x'+y))'$

$x(y'z'+yz)$

$x'yz'+x'y'z$

مثال) مکمل های توابع F_3 و F_4 را بدست آورید (با توجه به دوگان آن ها و مکمل گیری مستقیم بدست آورید)

* تابع بولی تشکیل شده است از حداقل یک متغیر و می تواند چندین متغیر داشته باشد و چندین عملیات دوتایی با آن

گرفته شده تابع را مکمل نامیم این مکمل برروی تمام متغیرها و عملیات های بین آن ها اعمال میشود.

۱) مکمل جمع ضرب است. ۲) مکمل ضرب جمع است ۳) مکمل مکمل یک متغیر خودش

مثال) $F_1 = xy + x'y' + x(y'z') + yz$

مکمل گیری $\rightarrow$ $F_1' = (xy + x'y' + x(y'z') + yz)'$

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

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

Scanned by CamScanner

DATE / / SUBJECT:

حالت‌های مختلف و استاندارد در جدول بولی (توابع بولی) :

یک متغیر بولی، دو نوع اصلی و مکمل آن نمایش داده می‌شود مانند x و x' . اگر فقط این دو بیش از یک متغیر وجود داشته باشد مانند xy این دو حالت (xy) ، $(x+y)$ ، $(x'y)$ ، $(y'x)$

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

اگر سه متغیر x ، y ، z باشند چند حالت ؟

اگر متغیرهای بولی را (سه متغیر مثلا) حالت‌های ممکن ضرب و یا جمع آن‌ها را در نظر بگیریم

به شرح زیر خواهد بود .

x	y	z	ضرب	جمع	
0	0	0	$x'y'z'=m_0$	$x+y+z=M_0$	
0	0	1	$x'y'z=m_1$	$x+y+z'=M_1$	
0	1	0	$x'yz'=m_2$	$x+y'+z=M_2$	
0	1	1	$x'yz=m_3$	$x+y'+z'=M_3$	
1	0	0	$xy'z'=m_4$	$x'+y+z=M_4$	
1	0	1	$xy'z=m_5$	$x'+y+z'=M_5$	
1	1	0	$xyz'=m_6$	$x'+y'+z=M_6$	
1	1	1	$xyz=m_7$	$x'+y'+z'=M_7$	

M ماکسترم m مینترم

به این حالت‌ها که ممکن ضرب بین سه متغیر x و y و z باشد :

ضرب دارای خاصیت جابجایی است یعنی حالت‌های دیگر وجود ندارد.

Scanned by CamScanner

DATE / / SUBJECT:

* چنانچه عبارت بولی با هم ضرب شوند (همه عملیات‌ها ضرب باشد) به آن مینترم گوییم (m)

* " " جمع " " " " ماکسترم گوییم (M)

* توابع بولی عبارتی است از متغیرها، ثابت‌ها، پرانتزها بولی که حاصل جمع یا ضرب استفاده شده بولی مینترم یا ماکسترم نباشد. مانند $x+y$ یا $xy+x'y'z'$

* تابع بولی که جمع مینترم‌ها باشد

$$F = xyz' + xyz + x'y'z'$$

$$F = m_4 + m_7 + m_0 = \sum m_4 m_7 m_0 = \Sigma(4,7,0)$$

حاصل جمع مینترم‌ها

* تابع بولی که ضرب ماکسترم‌ها باشد

$$F = \underbrace{(x+y+z')}_{\text{جمع}} \cdot \underbrace{(x'+y'+z)}_{\text{جمع}} = M_1 \cdot M_4 = \Pi M_1 M_4 = \Pi(1,4)$$

تبدیل شده به حاصل جمع مینترم‌ها

$$F = xy + x'y'z' = xy(1) + x'y'z' = xy(z+z') + x'y'z' = xyz + xyz' + xy'z' + x'y'z'$$

$$= m_7 + m_4 + m_0$$

* پس تابع بولی که شامل حاصل جمع مینترم‌ها یا ماکسترم‌ها باشد را می‌توانیم براساس جدول مینترم‌ها و ماکسترم‌ها بنویسیم. بعنوان مثال:

$$F_3 = xyz' + xyz + x'y'z'$$

$$F_3 = m_4 + m_7 + m_0$$

حاصل جمع مینترم‌ها

* همچنین تابع بولی مانند F_4 که حاصل ضرب ماکسترم‌ها است را می‌توانید با استفاده از جدول بنویسید.

از دو نماد Σ و Π می‌توانیم برای نمایش حاصل ضرب و جمع استفاده نمایم.

Scanned by CamScanner

DATE / / SUBJECT:

* عامل Σ و π است و برعکس و یکی از راه‌های فشرده‌سازی که در ساده‌سازی توابع بولی داشته باشیم

$$\Sigma(4, V, 0) \longrightarrow \pi(1, 2, 3, 4, 5)$$

* همانطور که یک تابع بولی داشته باشیم که از مجموع مقادیر کاهلی ضرب تشکیل شده باشد و برخی از این حاصلضرب‌ها مینترم نباشد می‌توانیم متغیری را برای مینترم کردن آن حاصلضرب اضافه کنیم (اصل ۵)

$$F = xy + x'y'z' = xy(1) + x'y'z' = xy(z+z') + x'y'z'$$

می‌توانیم آن را به مینترم تبدیل نماییم.

تمرین: حاصل جمع مینترم‌ها و حاصل ضرب ماکسترم‌ها را بدست آورید.

$$F_1 = x'y'z + xy'z' + xyz$$
$$= m_1 + m_4 + m_7$$
$$= \Sigma(1, 4, 7) \longrightarrow \pi(0, 2, 3, 5, 6)$$

$$F_2 = A + B'C = A(B+B')(C+C') + (A+A')B'C$$
$$= (AB + AB')(C+C') + AB'C + A'B'C = ABC + ABC' + AB'C + AB'C' + A'B'C + AB'C$$

$$F_3 = m_5 + m_6 + m_7 + m_4 + m_1 + m_3 = \Sigma(1, 3, 4, 5, 6, 7) \longrightarrow \pi(0, 2)$$

* سایر عملگرها مشترک در جبر بولی:

Scanned by CamScanner

DATE / / SUBJECT:

توابع بول	سمبل عملگر	نام	توضیحات
$F_0 = 0$	—	Null	Binary Constant
$F_1 = xy$	$x.y$	And	x and y
$F_2 = xy'$	x/y	Inhibition	x but not y
$F_3 = x$	—	Transfer	x
$F_4 = x'y$	y/x	Inhibition	y but not x
$F_5 = y$	—	Transfer	y
$F_6 = xy + x'y$	$x \oplus y$	Exclusive-or	x or y but not both
$F_7 = x + y$	$x + y$	Or	x or y
$F_8 = (x+y)'$	$x \downarrow y$	NOR	Not-or
$F_9 = xy + x'y'$	$x \odot y$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x C y$	Implication	If x then y

Scanned by CamScanner

DATE / / SUBJECT:

توضیحات	نام	سمبل عملگر	توابع بول
Not x	Complement	x'	$F_{12} = x'$
If x then y	Implication	$x \supset y$	$F_{13} = x' + y$
Not-And	NAnd	$x \uparrow y$	$F_{14} = (xy)'$
Binary Constant 1	Identity	—	$F_{15} = 1$

گیت‌های منطقی: از آن جایی که توابع بولی برحسب عملگرهای (not ،and ،... و) بیان می‌شوند لذا پیاده‌سازی این توابع (توابع بولی) با استفاده از گیت‌ها ساده است.

نکته: در عمل امکان ساخت گیت‌ها با مدارات الکترونیکی یا المان‌های زیر امکان‌پذیر است:

۱) امکان ساخت اقتصادی بدون روش ساخت و مدار از قطعات الکترونیکی که بخواهیم استفاده کنیم.

۲) امکان تعداد ورودی گیت‌ها بیش از دو مورد

۳) در نظر گرفتن خواص اجزای عملگرها دوتایی مانند خاصیت جابجایی، شرکت‌پذیری و...

۴) توانایی گیت در اجرای توابع به تنهایی و یا همراه با سایر گیت‌ها

نکته: در طراحی گیت‌های منطقی دیجیتال از ۶۴ تابع ایجاد شده توابع معادل با حداقل تعداد

Sahand

Scanned by CamScanner

DATE / / SUBJECT:

چهار تابع دوبادوتکرار و تنها دو تابع برای تهیه ی کلیه های منطقی کافیست و دو تابع را برای خاصیت جابجایی دارا برای بالا بردن کیفیت عنوان کلیه های منطقی نمی توانیم آن ها را پیاده سازی کنیم. ۸ تابع دیگر مانند نظیر: انتقال، And، Or، NAnd، Nor، Xor، Xnor بعنوان کلیه های استاندارد در طراحی سیستم های دیجیتالی بکار می روند لذا کلیه ها منطقی دیجیتال به شرح زیر اند:

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

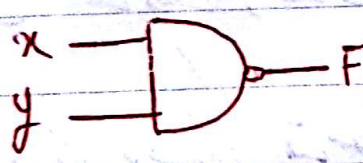
Buffer



$F = x$

x	F
0	0
1	1

NAnd



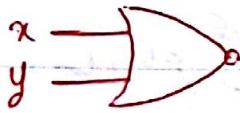
$F = (xy)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

Scanned by CamScanner

DATE / / SUBJECT:

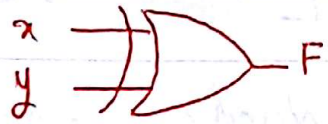
NOr



$F = (x+y)'$

x	y	F
:	:	:

Exclusive-Or (xor)

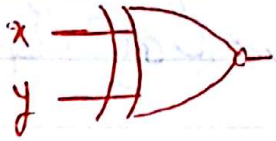


$F = xy' + x'y$

$= x \oplus y$

x	y	F
:	:	:

Exclusive-Nor or equivalence



$F = xy + x'y'$

$x \odot y$

x	y	F
:	:	:

نکته مهم: لیت هایی که نشان داده شده در جدول مذکور به خروجیت معکوس کننده و بافر قابل گسترش

به حالت بیش از دو ورودی هستند به شرط اینکه عمل دوتویی ارائه شده به وسیله آن جابجایی



And جابجایی و شرکت پذیری داشته باشد. مثلا عمل

نکته: مدارهای دیجیتال با استفاده از مدارهای مجتمع ساخته می‌شوند که به آن‌ها IC هم گفته می‌شود

اندازه‌ی IC ها براساس کاربرد آن‌ها متفاوت است. (۱۲ پایه، ۲۴ پایه، ۴۰ پایه و ...)

نکته: IC های دیجیتال اغلب براساس پیچیدگی عملیات داخل آن‌ها و تعداد گیت‌های مختلف آن‌ها مشخص

می‌شوند. به این IC ها اگر تعلقات با فشردگی کم بود آن‌ها را SSI دارای چندین گیت

مستقل. گروه دیگر با فشردگی متوسط به نام MSI (۱۰ تا ۱۰۰) گروه بعدی فشردگی زیاد LSI

بیش از ۱۰۰ تا چند هزار گیت داراست مثل تراشه‌های حافظه. گروه دیگر تعلقات بسیار زیاد شده

Scanned by CamScanner

DATE / / SUBJECT:

بنابراین VLSI دارای چنین مزایایی است.

انواع پردازنده‌ها:

TTL = transistor-transistor logic

ECL = emiter-coupled logic

MOS = metal-oxide semiconductor

CMOS = Complementary metal-oxide semiconductor

Scanned by CamScanner

DATE ۹۵/۹/۷ SUBJECT:

گروه TTL متداول ترین خانواده IC ها بطور گسترده در تولید انواع توابع دیجیتال استفاده می شود. گروه ECL در سیستم هایی که نیاز به سرعت بالا دارند گروه MOS در مدارهایی که نیاز به حجم های متراکم بالایی دارند استفاده می شود. گروه CMOS توان مصرفی سیستم را به بیست پایین تا یک استفاده می شود.

مدارهای کامبنی SSI و MSI و LSI :

پیچیدگی مدارهای مجتمع یا IC ها به ۴ طبقه دسته بندی می شود. SSI مدارهای مجتمع کوچک، MSI مدارهای متوسط مقیاس اند. LSI مدارهای پیچیدگی زیاد و VLSI مدارهای پیچیدگی خیلی زیاد.

نکته: یک مدار ترکیبی طراحی شده با گیت ها (شکل دلخواه) قابل پیاده سازی با مدارهای SSI چندگیره است.

نکته: تعداد گیت ها در یک پایه های بسته آن IC ها محدود می شود.

نکته: روش اقتصادی آن است که گیت ها را حد الامکان از بسته های موجود استفاده کنیم. (حتی اگر با این عمل تعداد کل گیت ها افزایش یابد.

نکته: بجز از اتصالات بین گیت ها (گیت های درداخل IC ها) اتصالات داخلی است یعنی بین پایه های IC و چیست نیست. این اتصالات به کمک مدارهای چاپی صورت می گیرد.

Scanned by CamScanner

DATE / / SUBJECT:

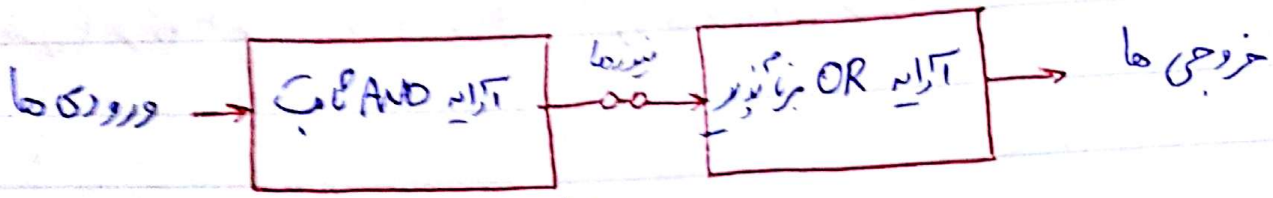
نکته: با مدارهای مجتمع تعداد گیت‌های موجود تعیین کننده قیمت مالی نیست بلکه نوع IC‌ها و مدارهای مجتمع به کار رفته و مقدار تعداد اتصالات بکار رفته به این مدار دیجیتال کاربرد دارد مانند مدار دیجیتال

نکته: هیچ مدار تولیدی وجود دارد که بطور گسترده‌ای در طراحی سیستم‌های دیجیتال بکار می‌رود. این مدارات بصورت مجتمع و بعنوان قطعات MSI طبقه‌بندی می‌شوند. مهمترین قطعات MSI ها عبارتند از جمع کننده‌ها تفریق کننده‌ها، مقایسه کننده‌ها، دیکودرها، انکودرها، موتی پلکسرها همچنین این قطعات (قطعات دیجیتال) بعنوان موجود ها که استاندارد در داخل مدارهای LSI و VLSI در اینها بکار می‌رود.

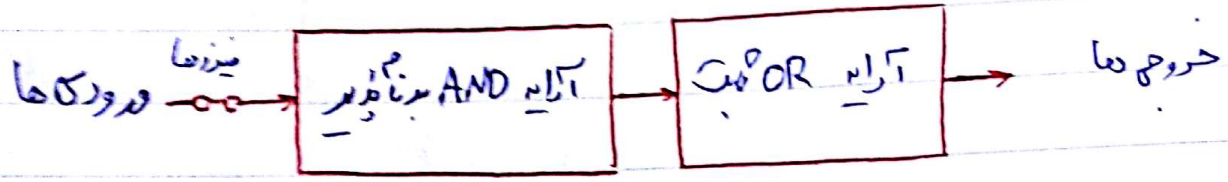
نکته: قطعات MSI از نوع استاندارد بوده و استفاده از آنها کاهش قابل ملاحظه‌ای را در کل هزینه در مقایسه با مدارهای LSI در بر دارد.

Scanned by CamScanner

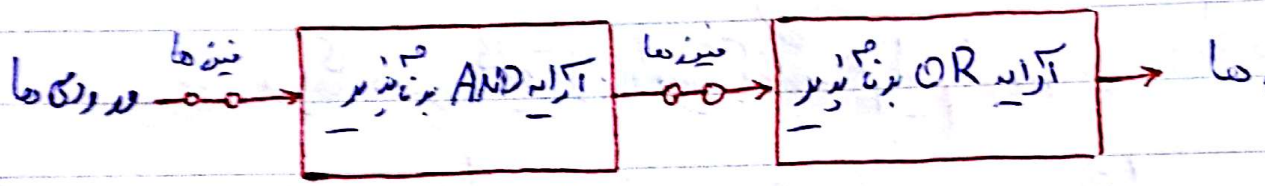
DATE / / SUBJECT:



الف) حافظه فقط خواندنی برنامه پذیر (PROM)



ب) آرایه برنامه پذیر منطقی I/O

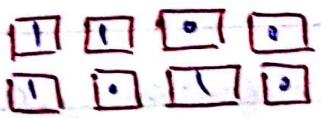


ج) آرایه منطقی برنامه پذیر

پیلر نیدگی باید ای برای 3PLD

نکته: مزیت کاربرد PLD در طراحی مدارهای دیجیتال این است که آن ها قادر به تبدیل برنامه زیر کاربرد لذا در توابع پیچیده ی منطقی و مدار I کاها می توانند شرکت کنند.

جمع کننده های موازی دودویی: تمام جمع کننده ها (ادرها) جمع دو بیت و یک رقم نقلی



دو بیت خواهد با هم جمع شود 4 حالت زیر را دارد:

Scanned by CamScanner

DATE / / SUBJECT:

مثال) برای ۴ بیت به عنوان مثال موارد زیر در نظر گرفته می‌شود:

A = 1011
B = 0011

کاما جمع کننده	۱	۲	۳	۴	برای رقم نام
z	C_1	۰	۱	۰	رقم نقلی ورودی
x	A_1	۱	۱	۰	مضاف
y	B_1	۱	۰	۰	مضاف الیه
S	S_1	۰	۱	۱	جمع
C	C_{i+1}	۱	۰	۰	رقم نقلی خروجی

بسیاری از کاربردهای کامپیوترهای دیجیتال (کامپیوتر آنالوگ هم داریم) نه تنها نیاز به دستکاری این داده‌ها است بلکه روی حروف نیز به نحوی این دستکاری انجام می‌شود به عنوان مثال یک مجموعه مانند دانشگاه باید خیلی جاها پردازش شود مجموعه‌ای از اطلاعات دانشجو، استاد و ... باید به صورت عددی وارد سیستم کامپیوتر می‌شود که پردازش شود. برای هر کدام از کلیدها شامل حروف کوچک و بزرگ و هشتاد علائم جانبی مثل سوال، تعجب، نقطه و... علائم حساب (+ - / ×) و... بعضی از عملیات خاص که مخصوص کامپیوتر تعریف می‌شود مثل Enter و Delete یک کد دیجیتال و عددی به آن‌ها اختصاص داده شد این کدها در یک جدول در کشور به نام American standard code for information interchange معروف است و این کدها را در برنامه‌ها به صورت کدهای باینری پردازش می‌شود اما در مدارات مختلف کدهای خاصی است و... در آینده است این کدها فرضیت ولی کدی به نام کد Gray در سیستم جدید

Sahand

Scanned by CamScanner

DATE / / SUBJECT:

برای کدگذاری داده‌ها کدی بسته طراحی شد. که Grey کد نیز است.

معادل ده دهی	کد گری
۹	۱۱۰۱
۱۰	۱۱۱۱
۱۱	۱۱۱۰
۱۲	۱۰۱۰
۱۳	۱۰۱۱
۱۴	۱۰۰۱
۱۵	۱۰۰۰

معادل ده دهی	کد گری
۰	۰۰۰۰
۱	۰۰۰۱
۲	۰۰۱۱
۳	۰۰۱۰
۴	۰۱۱۰
۵	۰۱۱۱
۶	۰۱۰۱
۷	۰۱۰۰
۸	۱۱۰۰

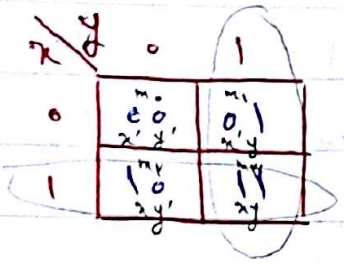
* یکی از روش‌های ساده‌سازی توابع بولی، کمک نقشه یا جدول:

پیچیدگی کیت‌های منطقی دیجیتال که یک تابع بول را پیاده‌سازی کنند مستقیماً به پیچیدگی تابعی که آن را پیاده‌سازی می‌کند بستگی دارد. در مقابل تلاش‌های زیادی برای ساده‌سازی یک تابع بولی از جدول قوانین استفاده می‌کردیم تا یک تابع را ساده‌تر نماییم و بعد از روی تابع ساده شده می‌توانستیم کیت‌های منطقی دیجیتال آن را ترسیم نماییم. (بدیهی است هرچه این کیت دیجیتال ساده‌تر باشد ارزان‌تر تمام می‌شود)

DATE / / SUBJECT:

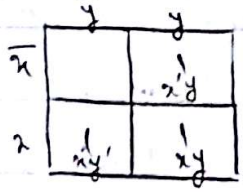
نقشه یا دیاگرام کارنو مشکل از تعدادی مربع است که هر مربع نشان دهنده یک مینترم است.

چون هر تابع بول را می‌توان بصورت مجموع مینترم ها نمایش داد لذا یک تابع بول را می‌توان بصورت مصور (تصویر شده) با در نظر گرفتن نواحی اشغال شده توسط مربع هایی که مینترم آن ها در تابع وجود دارد مشخص نمود. بعنوان مثال اگر یک تابع بولی داشته باشیم در این تابع دو متغیره بود (مثلاً x و y) جدول کارنو آن می‌شود



مثال) $F = x + y$

$$x(y+y') + y(x+x') = xy + xy' + yx + yx' = xy + xy' + x'y = m_3 + m_2 + m_1$$
$$= \Sigma(1, 2, 3)$$



$$m_1 + m_3 = x'y + xy = y(x+x') = y$$
$$m_2 + m_3 = xy' + xy = x(y+y') = x$$

پس از ترسیم جدول کارنو به از تابع استخراج مینترم ها بدست می‌آید می‌توانیم مربع ها را هم جوار را مشخص کنیم ملاحظه می‌شود که تنها اختلاف بین مربع هجوار در نقطه ترتیب متغیر باینری و بدون باینری است مثلاً

DATE / / SUBJECT:

نکته: توجه کنید که مینترم ها براساس ترتیب دودویی مرتب شده اند و ترتیبشان براساس کدگری می باشد.

نکته: در واقع جدول کارنو سه متغیره مانند (x,y,z) جدول کارنو به شرح زیر است:

x \ yz	00	01	11	10
0	$x'y'z'$ (m_0)	$x'y'z$ (m_1)	$x'yz$ (m_3)	$x'yz'$ (m_2)
1	$xy'z'$ (m_4)	$xy'z$ (m_5)	xyz (m_7)	xyz' (m_6)

مثال:

$$F_{xyz} = \Sigma(2,3,4,5) = m_2+m_3+m_4+m_5 = (m_2+m_3)+(m_4+m_5) = x'y+xy'$$

$$m_2+m_3 = x'yz'+x'yz = x'y(z+z') = x'y$$

$$m_4+m_5 = xy'z'+xy'z = xy'(z+z') = xy'$$

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

قدم بعدی یافتن مربع هایی هم جواری است که داخل آن ها یک است. مربع هم جوار مربعی است که اختلاف در مربع زیر یک بیت باشد. مربع هایی هم جواری که یک هستند را مشخص می کنیم.

نکته: در جدول کارنو دو متغیره هم جواری ها دو هم جواری باشند.

* هم جواری ها همواره بر اساس 2^n می باشند.
* یک مربع در جدول سه متغیره شامل سه متغیر است.
* دو مربع هم جوار شامل یک جمله با دو متغیر است.

Scanned by CamScanner

DATE / / SUBJECT:

* ۴ مربع هم جوار نشان دهنده ی یک جمله با یک متغیر است.

* ۸ " " تابع یک را تولید می کند.

مثال ۱) با استفاده از جدول کارنو حل نمایید.

$F(x,y,z) = \Sigma(0,1,2,4,5)$

$= m_0 + m_1 + m_2 + m_4 + m_5 = x'z' + y'$

x \ yz	00	01	11	10
x' 0	m_0 1	m_1 1	m_3	m_2 1
x 1	m_4 1	m_5 1	m_7	m_6

$m_0 + m_1 + m_4 + m_5 = x'y'z' + x'y'z + xy'z' + xy'z = y'$

$m_0 + m_2 = x'y'z' + x'yz' = x'z'(y'+y) = x'z'$

مثال ۲)

$(F)\ A'B + BC' + B'C'$

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

$A'BC + A'BC' + ABC' + A'BC' + AB'C' + A'B'C'$

$m_3 + m_2 + m_6 + m_4 + m_0 = C' + A'B$

A \ BC	00	01	11	10
A' 0	m_0	m_1	m_3	m_2
A 1	m_4	m_5	m_7	m_6

DATE / / SUBJECT:

نکته: در هنگام انتخاب مربع ها کاملا در دقت و درستی مطمئن باشیم که تمام مینترم ها کاملا جزئی از ترکیب پوشش داده شده اند و لازم است که تعداد مینترم ها در عبارت حداقل شده باشد.

wx \ yz	00	01	11	10
00	$x'y'z'w'$ 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 (11, 10) — x (01, 11) — w (11, 10) — z (11, 10)

نقشه کارنو ۴ متغیره: (حداکثر ۱۶ مربع داریم)

در نقشه ۴ متغیره که مربوط به یک تابع بولی است ۴ متغیر داده می شود، یک تابع ۴ متغیره حداکثر ۱۶ جمله مینترم دارد این ۱۶ مینترم بر اساس استاندارد گری در خانه ها به شکل روبرو قرار می گیرد.

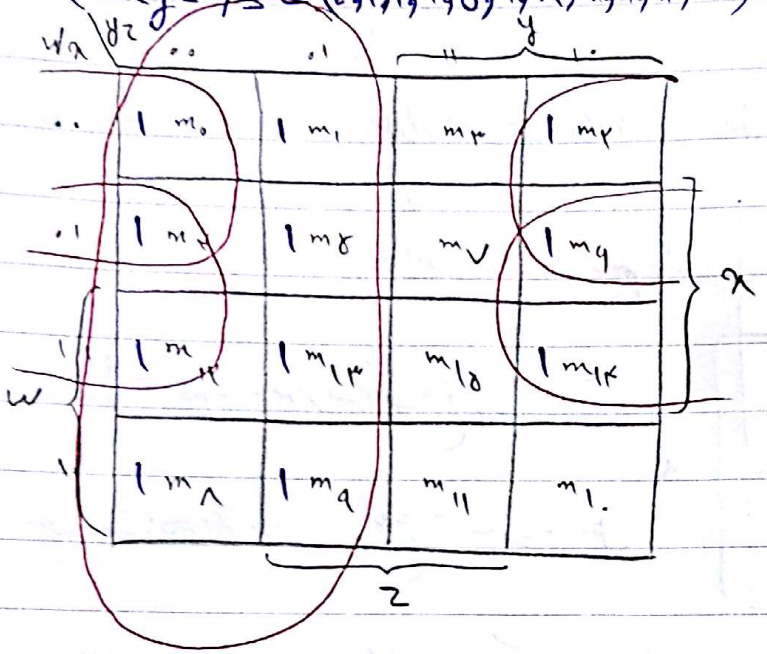
ساده کردن نقشه ۴ متغیره بر این اساس است: یک مربع که یک جمله مینترم را نشان می دهد دارای ۴ متغیر است.

دو مربع هم جوار نشان دهنده ی یک عبارت ۳ متغیره است. ۴ مربع هم جوار نشان دهنده ی یک عبارت ۲ متغیره است و ۸ مربع هم جوار نشان دهنده ی یک عبارت یک متغیره است.

Scanned by CamScanner

DATE / / SUBJECT:

مثال) تابع بولی زیر را ساده کنید. $F(wxyz) = \Sigma(0,1,2,4,5,7,8,9,12,13,14)$



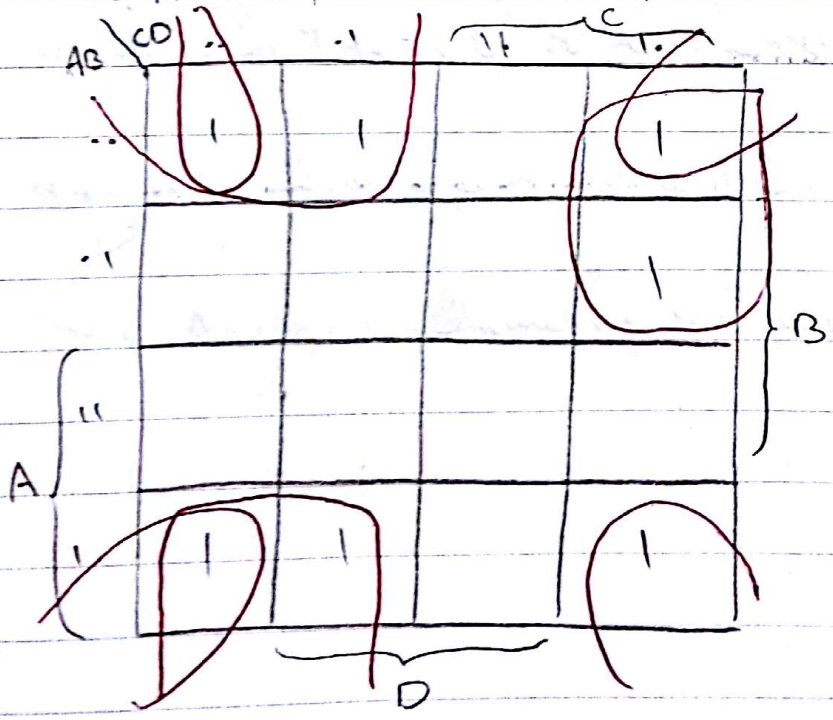
$$m_0 + m_1 + m_2 + m_4 + m_5 + m_7 + m_8 + m_9 + m_{12} + m_{13} + m_{14}$$

$$= y' + z'w' + z'x$$

مثال) $F = A'B'C' + B'CD' + A'BCD' + AB'C'$

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

$$= A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'$$



$$F = C'B' + B'D' + CD'A'$$

Sahand

Scanned by CamScanner

DATE / / SUBJECT:

ساده سازی تابع بولی ۴ متغیره سخت تر از این است که تعداد مینترم ها و متغیرها کم شود.

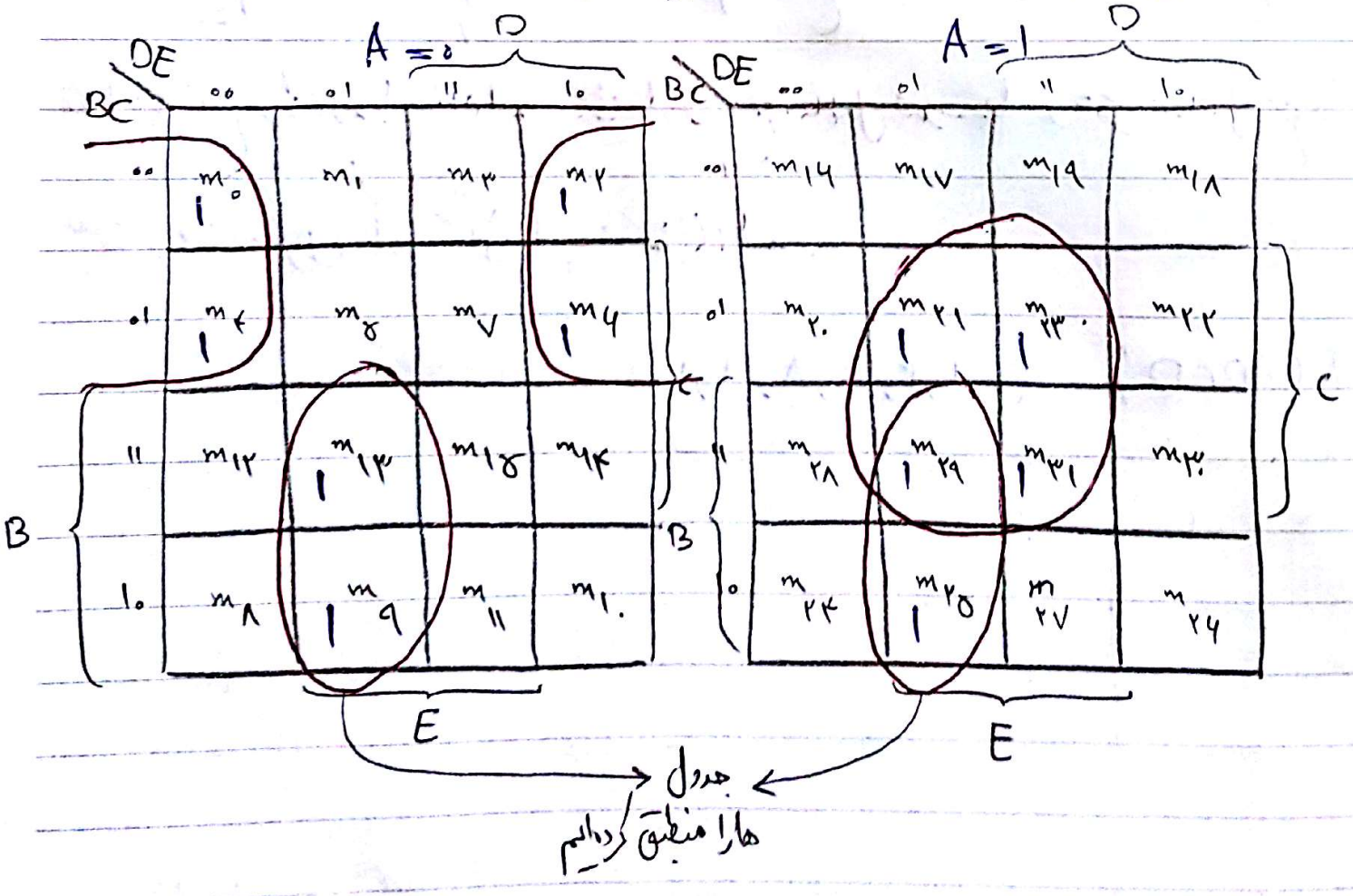
نقشه ۵ متغیره: تابع بولی زیر مفروض است آن را ساده کنید.

(۲ جدول ۱۶ متغیره)

$$F(ABCDE) = (0,2,4,6,9,13,21,23,25,29,31)$$

در کار با نقشه های ۵ متغیره ۳۲ مربع داریم (۶ متغیره ۶۴ مربع) و هرچه تعداد مینترم ها زیاد شود تعداد مربع ها هم افزایش می یابد و با فرض مربع ها هم جوار به شکل هندسی نقشه را بسته می شود.

نقشه ۵ متغیره به صورت ۲ جدول ۱۶ متغیره است.



Scanned by CamScanner

DATE / / SUBJECT:

$m_0 + m_1 + m_2 + m_4 = A'B'E'$

$m_{21} + m_{23} + m_{29} + m_{31} = ACE$

$m_{13} + m_9 + m_{29} + m_{25} = BD'E$

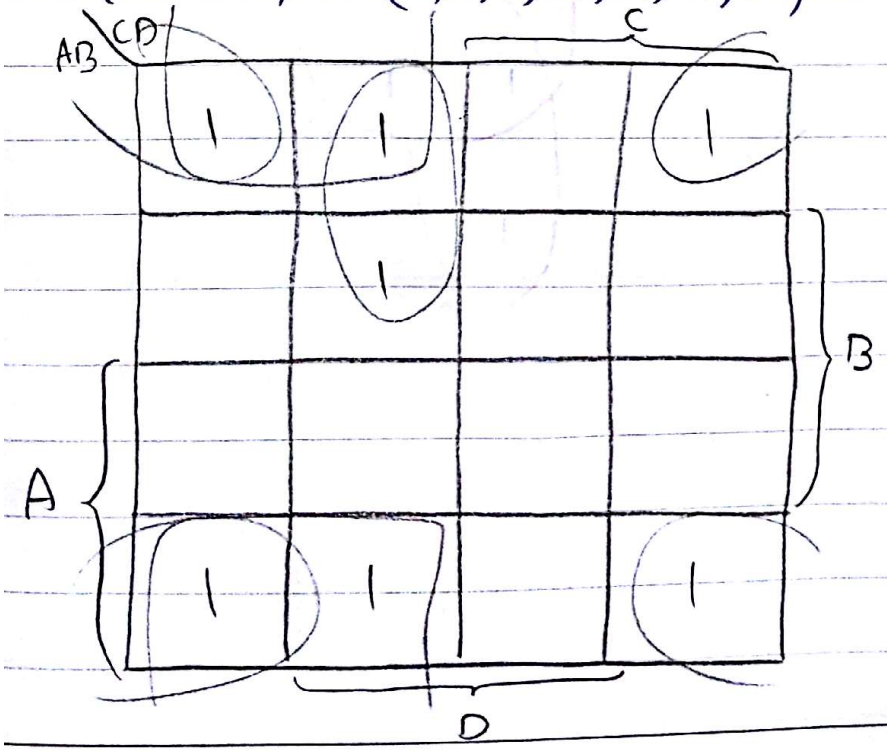
* یک عبارت بولی را می‌توانیم با استفاده از جمع حاصل ضرب‌ها یا ماکسترم‌ها ساده کنیم.

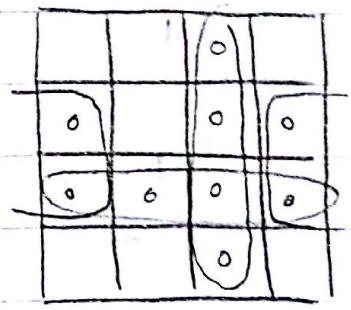
همانگونه که ساده‌سازی با استفاده از ضرب جمع‌ها را می‌توانیم ابتدا طبق روش‌های گفته شده مربع جدول کارنو را بدست آوریم (براساس مینترم‌ها) و برای ماکسترم‌ها مربع‌هایی که صفر است در نظر گیریم.

مثال) تابع بولی زیر را ساده کنید براساس جمع حاصل ضرب‌ها و ضرب حاصل جمع‌ها و پس از ساده کردن کدام یک آن را پیاده‌سازی کنید.

$F(ABCD) = (0, 1, 2, 5, 8, 9, 10)$

$F = B'D' + C'B' + A'C'D$





$F' = CD + AB + BD'$

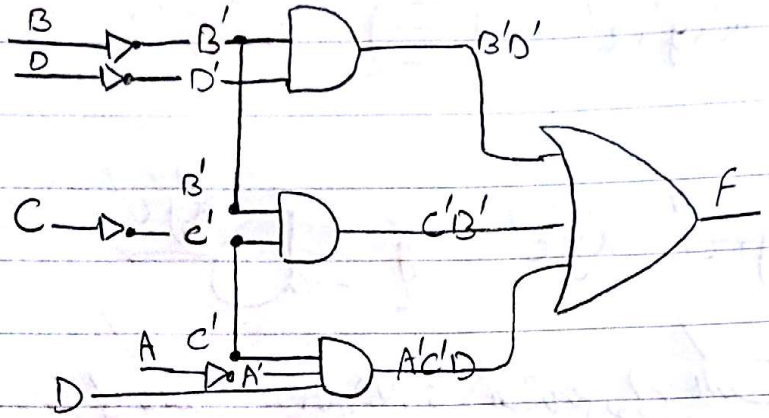
$= (C'+D')(A'+B')(B'+D)$

Sahand

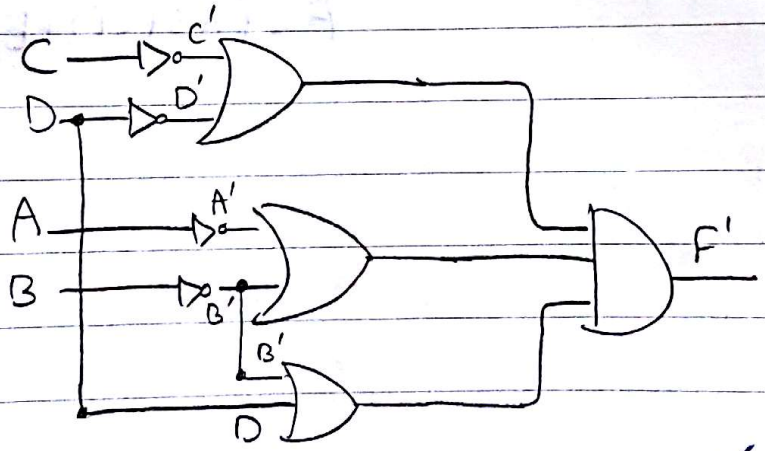
Scanned by CamScanner

DATE / / SUBJECT:

$F = B'D' + C'B' + A'C'D$



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



پیاده سازی به وسیله گیت های NAND و NOR :

همچنان که قبلاً اشاره شد مدارهای دیجیتال با گیت های NAND و NOR ساخته می شوند.

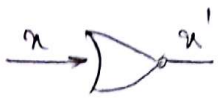
برای این منظور توانایی هر گیت ها در پیاده سازی از NAND و NOR کمتر نیست

NotAnd و Notor است استفاده کرد.

Scanned by CamScanner



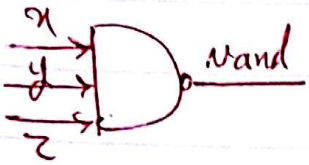
Not(And)



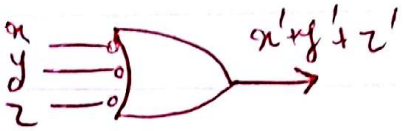
Not(or)

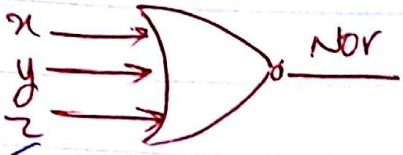
DATE / / SUBJECT:

یک گیت NAND ترکیبی از and و not است.

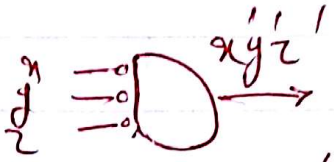


$(xyz)' = x'+y'+z'$





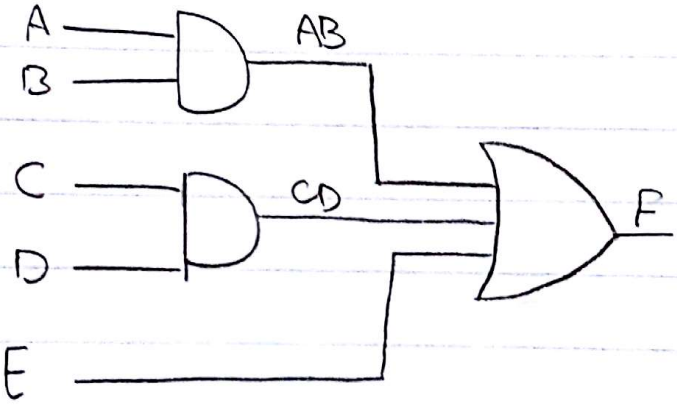
$(x+y+z)' = x'y'z'$

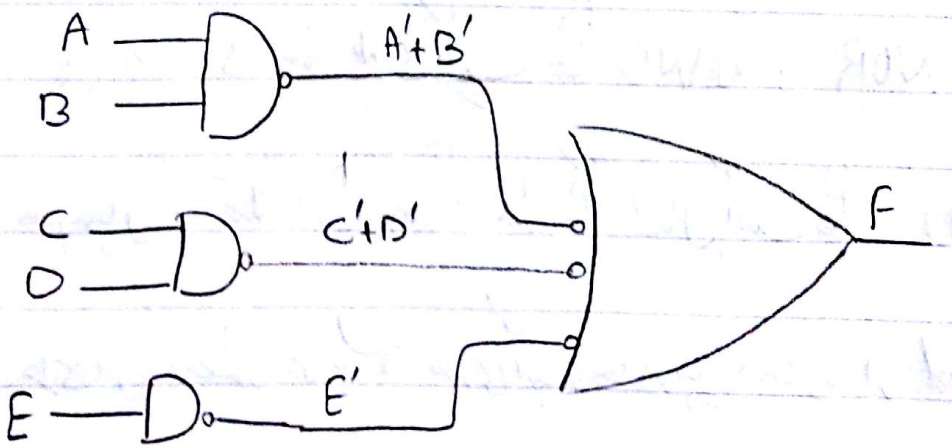


پیاده سازی با گیت NAND: گیت منبع یونی با گیت های Nand مشابه این است که

تابع به فرم جمع حاصل ضرب ها ساده شده باشد. (جمع منترم ها) مثلا

$F = (AB+CD+E)$

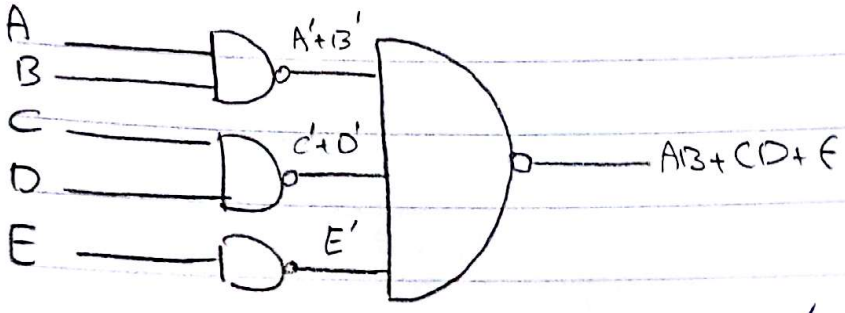




Nand-Nor

Scanned by CamScanner

DATE / / SUBJECT:



Nand - Nand

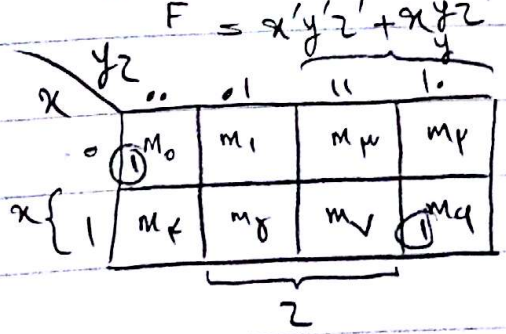
* از تبدیل گام به گام در این مثال نتیجه میگیریم یک تابع بولی مانند F میتواند با دو طبقه یا دو سطح پیاده سازی شود و بطور کلی میتوان گفت ۱) تابع را ساده کرده آن را به فرم جمع حاصل ضرب ها پیاده سازی کردیم ۲) برای هر جمله ضرب موجود در تابع F حداقل دارای دو متغیر است

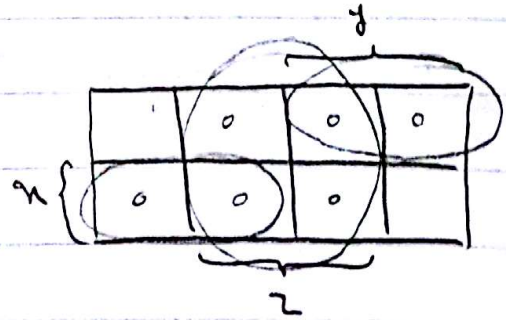
گیت Nand استفاده میشود. میتوانیم با استفاده از invert or And invert تا ورودی ها و خروجی ها را براساس مورد نیاز (F) جایگزین کرد.

مثال) با استفاده از گیت Nand پیاده سازی کنید.

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

$F = x'y'z' + xyz'$





$F' = z + xy' + x'y'$

Scanned by CamScanner