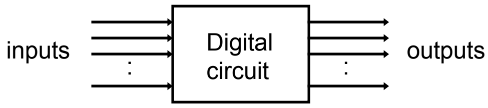
فصل دوم: جبر بول (Boolean Algebra)

- مدار دیجیتالی بصورت یک بلوک با وروودی ها و خروجی ها می تواند نمایش داده شود.



- سیگنالهای استفاده شده در مدارهای دیجیتالی بصورت گسسته/دیجیتال (discrete/digital) می باشند که دارای یکی از دو سطح ولتاژ High یا Low هستند.
در حالیکه سیگنالهای استفاده شده در مدار های آنالوگ بصورت پیوسته می باشند.

- مقایسه مدارهای دیجیتال نسبت به مدارهای آنالوگ:
  - قابلیت اعتماد بالاتر(سادگی مدار- حساسیت کمتر به نویز)
  - دقت قابل تعیین
  - مدل ریاضی (جبر بول)
  - اما : زمان پاسخ کندتر(نرخ نمونه برداری)

سیگنال دیجیتال سیگنال آنالوگ

جبر بول

جبر چیست؟
مجموعه ای از عناصر (e.g. 0,1,2,..)
مجموعه ای از عملگرها (e.g. +, -, *,..)
مجموعه ای از اصول (e.g. 0+x=x,..)

- جبر بول به خاطر George Boole نامگذاری شده است که برای بیان منطق انسان از آن استفاده نموده.
- رویداد: *true* or *false*
- عملگر ها: a *OR* b; a *AND* b, *NOT* a

a	b	a AND b
F	F	F
F	T	F
T	F	F
T	T	T

a	b	a OR b
F	F	F
F	T	T
T	F	T
T	T	T

a	NOT a
F	T
T	F

بعداً **Shannon** جبر کلیدی (Switching Algebra) را برای نمایش مدارهای کلیدی دوحالته معرفی کرد **(جبر بول دو ارزشی)** .

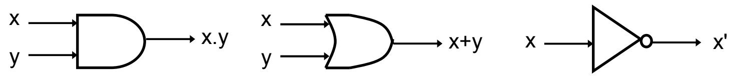
جبر بول دو ارزشی
(Two-valued Boolean Algebra)

- مجموعه ای از عناصر {0,1}
- مجموعه ای از عملگرهای { . , + , ' }

x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0



Signals: High = 5V = 1; Low = 0V = 0

اصول جبر بول
(Boolean Algebra Postulates)

جبر بول شامل مجموعه عناصر B همراه با دو عملگر باینری {+} و {.} و یک عملگر یکتایی {'}
(Unary)است ، طوریکه اصول زیر را بر آورده کند :

- **1- Closure**: For every x, y in *B*,
  - x + y is in *B*
  - x . y is in *B*
- **2- Identities** (0 and 1):
  - 0 + x = x + 0 = x for every x in *B*
  - 1 . x = x . 1 = x for every x in *B*
- **3- Commutative laws**: For every x, y in *B*,
  - x + y = y + x
  - x . y = y . x
- **4- Distributive laws**: For every x, y, z in *B*,
  - x . (y + z) = (x . y) + (x . z)
  - x + (y . z) = (x + y) . (x + z)

اصول جبر بول
(Boolean Algebra Postulates)

- **5- Complement**: For every x in *B*, there exists an element x' in *B* such that
  - $x + x' = 1$
  - $x \cdot x' = 0$
- **6**-The set *B* contains at least two distinct elements x and y.

مجموعه $B = \{0, 1\}$ و عملگرهای منطقی AND ، OR و NOT تمام اصول جبر بول را بر آورده می سازد.

یک تابع بولی **(Boolean function)** یک جمله جبری است که شامل متغیر ها و عملگرهای بولی است و تعدادی ورودی روی مجموعه {0,1}را به مجموعه {0,1} نگاشت می کند.

تقدم عملگرها
(Precedence of Operators)

- به منظور کاهش تعداد پرانتز های استفاده شده در نمایش توابع بولی، می توان از تقدم عملگر ها استفاده کرد.
- Precedence (highest to lowest): () , ' , . , +
- مثال:

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

- با استفاده از پرانتز می توان اولویت را تغییر داد:
- مثال :

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

جدول درستی
(Truth Table)

- جدول درستی لیستی از تمام ترکیبات ممکن در ورودی و خروجی متناظر با هر حالت را نمایش می دهد.

INPUTS	OUTPUTS
…	…
…	…

- مثال:

دو ورودی- دو خروجی

x	y	x . y	x + y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

اثبات با استفاده از جدول درستی

- مثال : ثابت کنید $x . (y + z) = (x . y) + (x . z)$

(i) جدول درستی را برای سمت راست(RHS) و چپ(LHS) معادله تشکیل دهید :

x	y	z	y + z	x.(y + z)	x.y	x.z	(x.y)+(x.z)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

(ii) بررسی کنید آیا RHS=LHS ؟ چون ستون خروجی ۲و ۵ با هم برابرند. پس رابطه برای تمام حالتهای ورودی صحیح بوده و درستی آن اثبات می شود.

تمرین ۱-۲

Q. Use truth table to prove the following.

(a) A'.B' + A.B' + A.B = A + B'

(b) A.B.C + A'.B.C + A'.B.C' = B.C + A'.B

دوگانی
(Duality)

اصل دوگانی: هر عبارت معتبر بولی (معادله) با تغییر زیر معتبر باقی خواهد ماند:

$+ \leftrightarrow .$

$1 \leftrightarrow 0$

- مثال: با داشتن عبارت معتبر زیر:

  a + (b.c) = (a+b).(a+c)

  دوگان آن یعنی رابطه زیر نیز معتبر خواهد بود:

  a . (b+c) = (a.b) + (a.c)

- دوگانی یک قضیه رایگان بدست می دهد "two for the price of one" شما یک رابطه معتبر را اثبات می کنید و رابطه دیگر نیز اثبات می گردد.

- اگررابطه (x+y+z)' = x'.y.'z' معتبر باشد، دوگان آن نیز معتبر خواهد بود. یعنی (x.y.z)' = x'+y'+z'
- اگررابطه x + 1 = 1 معتبر باشد، دوگان آن نیز معتبر خواهد بود. یعنی x . 0 = 0

قضایای اصلی جبربول

- علاوه بر اصول ،قضایای مفیدی نیز می تواند استفاده گردد:

1. **Idempotency**.

   (a) $x + x = x$ (b) $x \,.\, x = x$

   Proof of (a):

   $$\begin{aligned} x + x &= (x + x).1 && \text{(identity)} \\ &= (x + x).(x + x') && \text{(complementarity)} \\ &= x + x.x' && \text{(distributivity)} \\ &= x + 0 && \text{(complementarity)} \\ &= x && \text{(identity)} \end{aligned}$$

قضایای اصلی جبربول

2. **Null elements** for + and . operators.

   (a) $x + 1 = 1$ (b) $x \,.\, 0 = 0$

3. **Involution**. $(x')' = x$

4. **Associative laws**:

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

   (b) $(x \,.\, y) \,.\, z = (x \,.\, y) \,.\, z = x \,.\, y \,.\, z$

5. **DeMorgan**.

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

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

6. **Absorption**.

   (a) $x + x.y = x$ (b) $x.(x + y) = x$

- قضایا را می توان با جدول درستی اثبات نمود.

تمرین:قضیه De-Morgan را با جدول درستی اثبات کنید.

- همچنین می توان آنها را باعملیات جبری (Algebraic Manipulation) وبا استفاده از اصول وقضایای دیگر اثبات نمود

قضایای اصلی جبربول

- اثبات قضیه 4a (absorption) :

$$
\begin{aligned}
x + x.y &= x.1 + x.y && \text{(identity)}\\
&= x.(1 + y) && \text{(distributivity)}\\
&= x.(y + 1) && \text{(commutativity)}\\
&= x.1 && \text{(Theorem 2a)}\\
&= x && \text{(identity)}
\end{aligned}
$$

- با دوگانی قضیه 4b اثبات می گردد.(بعنوان تمرین اثبات شود)

$$x.(x+y) = x$$

توابع بولی(منطقی)
(Boolean Functions)

- تابع بولی عبارتی است شامل متغییر های باینری، عملگر های AND، OR و NOT ،پارانتز و علامت مساوی.
- نتیجه آن نیز مقدار باینری خواهد بود.
- معمولاً **.** برای AND ، **+** برای OR و **'** برای NOT استفاده می گردد.اغلب اگر ابهامی پیش نیاید"**.**" نوشته نمی شود.

- مثال:

F1= xyz'

F2= x + y'z

F3=(x'y'z)+(x'yz)+(xy')

F4=xy'+x'z

x	y	z	F1	F2	F3	F4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

از روی جدول مشاهده می شود که F3=F4 است. با عملیات جبری تساوی را ثابت کنید.

مکمل تابع
(Complement of Functions)

- با داشتن جدول تابع F مکمل آن یعنی F’ با جایگزینی 0 و 1 بدست می آید. با داشتن عبارت منطقی از قضیه دمورگان می توان استفاده کرد.

مثال : $F1 = xyz'$

مکمل تابع:

$$F1' = (xyz')' = x' + y' + (z')' \quad \text{DeMorgan}$$
$$= x' + y' + z \quad \text{Involution}$$

فرم کلی قضیه دمورگان :

$$(A + B + C + \ldots + Z)' = A' . B' . C' \ldots . Z'$$
$$(A . B . C \ldots . Z)' = A' + B' + C' + \ldots + Z'$$

x	y	z	F1	F1'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

مکمل تابع
(Complement of Functions)

- روش دوم: ابتدا دوگان تابع را بدست آورده و با مکمل کردن تمام متغیر های آن مکمل تابع بدست می آید. معمولاً برای توابعی که تعداد متغیر ها وجملات آن زیاد باشد، این روش مفید تر خواهد بود.

مثال :

$$F = x+yz'$$

دوگان تابع:

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

مکمل تابع :

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

فرمهای استاندارد
(Standard Forms)

- از نقطه نظر پیاده سازی (implementation) بعضی از فرمهای توابع بولی مناسبتر هستند.
- دو فرم استاندارد:
  Sum-of-Products and ***Product-of-Sums***
- **Literals:** به یک متغییر در فرم خود یا بفرم مکمل گفته می شود. مثال : x, x' , y, y'
- جمله ضربی (**Product Term**): یک لیترال یا AND چند لیترال
  مثال: x, xyz', A'B, AB
- جمله جمعی (**Sum Term**): یک لیترال یا OR چند لیترال
  مثال: x, x+y+z', A'+B, A+B

فرمهای استاندارد

- **فرم جمع حاصلضرب(SOP):** یک جمله ضربی یا OR چند جمله ضربی . مثال : x, x+yz', xy'+x'yz, AB+A'B'
- **فرم ضرب حاصل جمع(POS):** یک جمله جمعی یا AND چند جمله جمعی . مثال : x, x(y+z'), (x+y')(x'+y+z), (A+B)(A'+B')
- هر عبارت بولی می تواند بفرم SOP و POS نوشته شود.

Examples:

SOP: $x'y + xy' + xyz$

POS: $(x + y')(x' + y)(x' + z')$

both: $x' + y + z$ or xyz'

neither: $x(w' + yz)$ or $z' + wx'y + v(xz + w')$

Minterm & Maxterm

- مینترم عبارتست از AND روی n متغییر ، طوریکه متغییر با مقدار 0 پریم دار و با مقدار 1 بدون پریم می باشد
- ماکسترم عبارتست از OR روی n متغییر ، طوریکه متغییر با مقدار 1 پریم دار و با مقدار 0 بدون پریم می باشد

مثال : مینترم وماکسترمهای دو متغییره :

Minterms				Maxterms	
x	y	term	notation	term	notation
0	0	x'y'	m0	x+y	M0
0	1	x'y	m1	x+y'	M1
1	0	xy'	m2	x'+y	M2
1	1	xy	m3	x'+y'	M3

هر مینترم مکمل ماکسترم متناظرش می باشد.

مثال:

m2 = xy'

m2' = (xy')' = x' + (y')' = x'+y = M2

Canonical Form: Sum of Minterms

- فرم کانونیک(نرمال) چیست؟
  - یک فرم یکه برای نمایش یک عبارت
  - مینترم جمله ضربی می باشد و می توان تابع بولی را بفرم Sum-of-Minterms نمایش داد.

مراحل:

الف- جدول درستی را بدست آورید:

ب- فرم جمع مینترم را با جمع کردن مینترمهایی که تابع در آنها یک است بدست آورید.

مثال:

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

F1 = xyz' = Σm(6)

F2 = x'y'z+xy'z'+xy'z+xyz'+xyz

= m1+m4+m5+m6+m7

= Σm(1,4,5,6,7)

F3 = x'y'z+x'yz+xy'z'+xy'z

= Σm(1,3,4,5)

Canonical Form: Product of Maxterms

- ماکسترمها جملات جمعی (Sum Terms) می باشند.
- ماکسترمهای یک تابع بولی ترمهایی هستند که تابع در آنها **صفر** است.
- توابع بولی را می توان بفرم ضرب ماکسترمها (Products-of- Maxterms) نمایش داد.

E.g.: $F2 = \Pi M(0,2,3) = (x+y+z)(x+y'+z)(x+y'+z')$

$F3 = \Pi M(0,2,6,7) = (x+y+z)(x+y'+z)(x'+y'+z)(x'+y'+z')$

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Canonical Form: Product of Maxterms

می توان درستی این نمایش را بصورت زیر اثبات کرد:

$F2 = \Sigma m(1,4,5,6,7)$

$F2' = \Sigma m(0,2,3)$
$= m0 + m2 + m3$

$F2 = (m0 + m2 + m3)'$
$= m0' . m2' . m3'$
$= M0 . M2 . M3$
$= \Pi M(0,2,3)$

DeMorgan
$(mj)' = Mj$

x	y	z	F2	F2'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

- هر تابع بولی را می توان به هر دوفرم کانونیک تبدیل کرد.

تمرین ۲-۲

1. A Boolean function of 5 variables can have up to ___ minterms.
2. Given a Boolean function $F(A,B,C) = \Sigma\, m(0, 5, 7)$.
   Which of the following is correct?
   a) $F'(A,B,C) = \Sigma m(0,5,7)$ b) $F(A,B,C) = \Pi M(1,2,3,4,6)$
   c) $F(A,B,C) = \Pi M(0,5,7)$ d) $F'(A,B,C) = \Sigma m(1,2,3)$
   e) None above
3. Given a Boolean function $F(x,y,z) = y'.(x + z') + x'.z$.
   Which of the following is correct?
   a) $F(x,y,z) = \Sigma m(0,1)$ b) $F(x,y,z) = \Sigma m(0,1,4,5)$
   c) $F(x,y,z) = \Sigma m(0,1,2,3,4)$ d) $F(x,y,z) = \Sigma m(0,1,3,4,5)$
   e) $F(x,y,z) = \Sigma m(1,2,3,4,5)$

تمرین ۲-۲

4. Given a Boolean function with 6 variables a, b, c, d, e, f.
   What is maxterm $M60$?
   a) $a'.b'.c'.d'.e.f$ b) $a'+b'+c'+d'$ c) $a'+b'+c'+d'+e+f$
   d) $a.b.c.d.e'.f'$ e) $a+b+c+d+e'+f'$
5. Given a Boolean function $F(a,b,c) = \Sigma\, m(0, 2, 5, 6, 7)$.
   If a=0, b=c=1, what is the value of F?
   a) 0 b) 1 c) $b.c$ d) a' e) Unknown
6. Which of the following is NOT a minterm of the Boolean function: $F(w,x,y,z) = w.x.z' + x.y'.z + x.z$
   a) $w.x.y.z'$ b) $w'.x.y'.z$ c) $w.x.y.z$
   d) $w.x.y'.z'$ e) $w.x'.y.z$

تمرین ۲-۲

7. Identify the following function $F(x,y,z)$.

a) $x.y'.z' + x'.y.z + x.z'$
b) $x'.y.z + x.y' + x'.y'.z$
c) $x.y'.z + x'.y'.z + x.y$
d) $x.y.z + x'.y'.z' + x.y'$
e) None above

x	y	z	F
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	0
1	1	1	0

تبدیل فرمهای کانونیک

- Sum-of-Minterms $\Rightarrow$ Product-of-Maxterms

❖ اندیس هایی که در یک فرم نیست در فرم دیگر نوشته می شود

Eg: $F1(A,B,C) = \sum m(3,4,5,6,7) = \prod M(0,1,2)$

- Product-of-Maxterms $\Rightarrow$ Sum-of-Minterms

❖ اندیس هایی که در یک فرم نیست در فرم دیگر نوشته می شود

Eg: $F2(A,B,C) = \prod M(0,3,5,6) = \sum m(1,2,4,7)$

Conversion of Canonical Forms

- Sum-of-Minterms of F $\Rightarrow$ Sum-of-Minterms of F‘
  - اندیس هایی که در یک فرم نیست در فرم دیگر نوشته می شود

  Eg: $F1(A,B,C) = \sum m(3,4,5,6,7)$

  $F1'(A,B,C) = \sum m(0,1,2)$

- Product-of-Maxterms of F $\Rightarrow$ Prod-of-Maxterms of F‘
  - اندیس هایی که در یک فرم نیست در فرم دیگر نوشته می شود

  Eg: $F1(A,B,C) = \prod M(0,1,2)$

  $F1'(A,B,C) = \prod M(3,4,5,6,7)$

Conversion of Canonical Forms

- Sum-of-Minterms of F $\Rightarrow$ Product-of-Maxterms of F'
  - همان اندیس ها تکرار می شود.

  Eg: $F1(A,B,C) = \sum m(3,4,5,6,7)$

  $F1'(A,B,C) = \prod M(3,4,5,6,7)$

- Product-of-Maxterms of F $\Rightarrow$ Sum-of-Minterms of F'
  - همان اندیس ها تکرار می شود.

  Eg: $F1(A,B,C) = \prod M(0,1,2)$

  $F1'(A,B,C) = \sum m(0,1,2)$

تمرین ۳-۲

Q. Consider the function

$$f(A,B,Y,Z) = \Sigma\, m(0, 2, 5, 6, 8, 11, 15).$$

a) Write this as a Boolean expression in canonical SOP form.

b) Rewrite the expression in canonical POS form.

c) Write the complement of *f* in “little m” notation and in canonical SOP form.

d) Write the complement of *f* in “big M” notation and in canonical POS form.

توابع باینری

- برای n متغییر 2^n مینترم وجود خواهد داشت.
- چون هر تابع را می توان بفرم جمع مینترم نمایش داد ، پس 2^{2^n} تابع مختلف خواهیم داشت.
- برای دو متغییر $2^2=4$ مینترم و $2^4=16$ تابع مختلف خواهیم داشت.
- ۱۶ تابع باینری ممکن در جدول زیر نمایش داده شده اند:

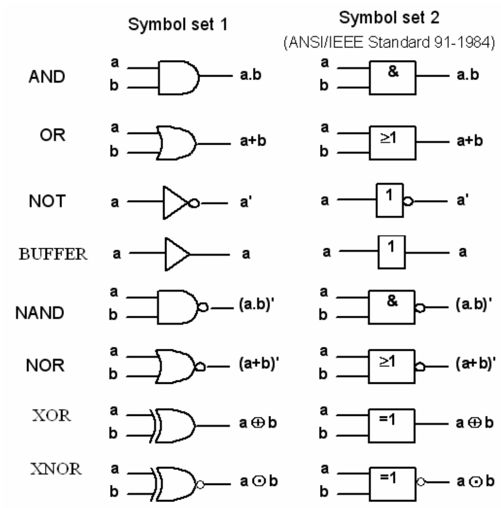
x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Symbol			.	/		/		$\oplus$	+
Name			AND					XOR	OR

x	y	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Symbol		$\downarrow$	$\odot$	'	$\subset$	'	$\supset$	$\uparrow$	
Name		NOR	XNOR					NAND	

- **Inhibition** : x / y :x.y’ , y / x :x’.y
- **Transfer** : F_3=x , F_5=y
- **Complement** : F_{10}=y’ , F_{12}=x’
- **Implication** : x ⊂ y : x+y’
  IF y THEN x
  x ⊃ y : x’+y
  IF x THEN y
- **NULL** : F_0=0
- **Identity** : F_{15}=1

گیت های منطقی
Logic Gates

- **Gate Symbols**



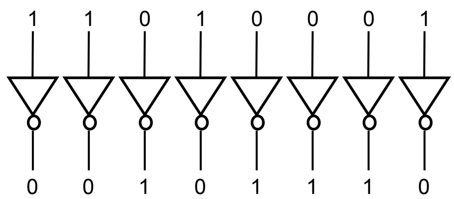
Logic Gates: The Inverter

- The **Inverter**

A	A'
0	1
1	0

- Application of the inverter: complement.

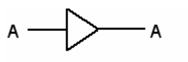
Binary number



1's Complement

Logic Gates: The BUFFER

- The **BUFFER**

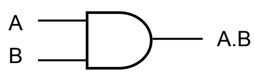


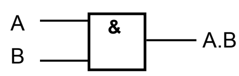
A	A
0	0
1	1

کاربردها:
- تقویت سگنال و گرفتن انشعاب
- ایجاد تاخیر در سیگنال ورودی

Logic Gates: The AND Gate

- The **AND** Gate

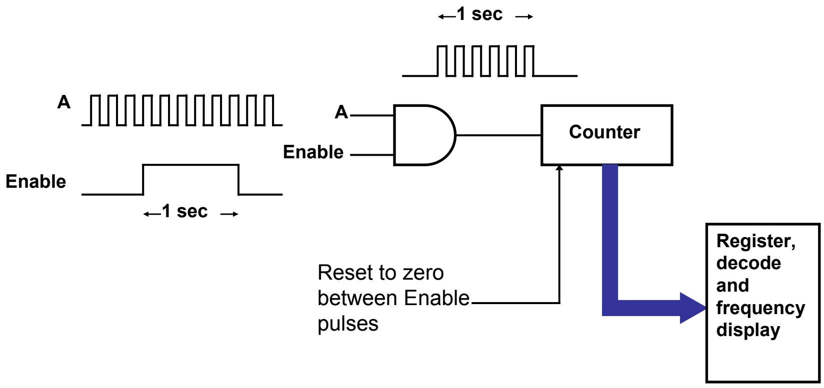




A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gates: The AND Gate

- Application of the AND Gate



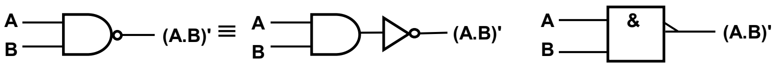
Logic Gates: The OR Gate

- The **OR** Gate

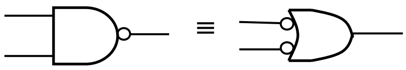
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates: The NAND Gate

- The **NAND** Gate



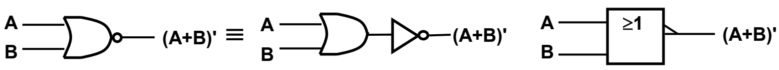
A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0



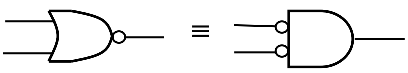
NAND Invert-OR

Logic Gates: The NOR Gate

- The **NOR** Gate



A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0



NOR Invert-AND

Logic Gates: The XOR Gate

- The **XOR** Gate

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Logic Gates: The XNOR Gate

- The **XNOR** Gate

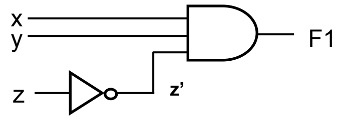
A	B	$A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

رسم(پیاده سازی) مدارهای منطقی

- مثال

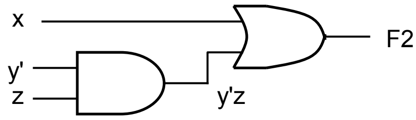
(i) $F1 = xyz'$

دقت: از گیت AND سه ورودی استفاده شده است.

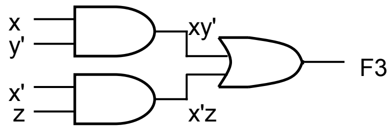


(ii) $F2 = x + y'z$

می توان فرض کرد که متغییر ومکمل آن مستقیماً قابل دسترسی هستند.



(iii) $F3 = xy' + x'z$



تمرین ۴-۲

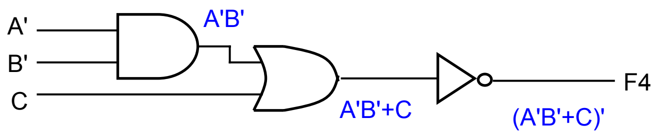
Q1. Draw a logic circuit for BD + BE + D’F

Q2. Draw a logic circuit for

A’BC + B’CD + BC’D + ABD’

تحلیل مدارهای منطقی
Analysing Logic Circuit

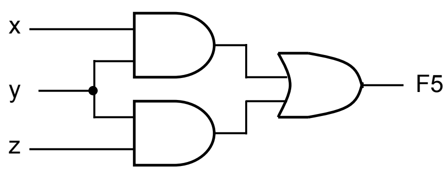
- Example: What is the Boolean expression of F4?



F4 = (A'B'+C)' = (A+B).C'

تمرین ۵-۲

- What is Boolean expression of F5?



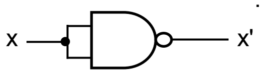
Universal Gates: NAND and NOR

- گیت های AND/OR/NOT برای طراحی تمام توابع منطقی کافی هستند.
- با این وجود گیت های دیگر نیزبدلایل زیر استفاده می گردد:
  - (i) usefulness
  - (ii) economical on transistors
  - (iii) self-sufficient

NAND/NOR: economical, self-sufficient
XOR: useful (e.g. parity bit generation)

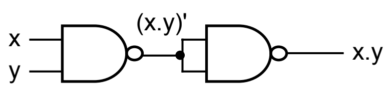
NAND Gate

- گیت NAND حالت self-sufficient دارد یعنی می توان هر مدار منطقی را فقط با آن طراحی کرد (گیت یونیورسال)
- می توان گیتهای AND/OR/NOT را با آن ساخت.
- پیاده سازی گیت NOT با NAND :



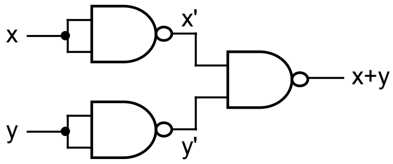
(x.x)' = x'   (T1: idempotency)

- پیاده سازی گیت AND با NAND :



((xy)'(xy)')' = ((xy)')'	idempotency
= (xy)	involution

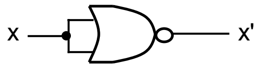
- پیاده سازی گیت OR با NAND :



((xx)'(yy)')' = (x'y')'	idempotency
= x''+y''	DeMorgan
= x+y	involution

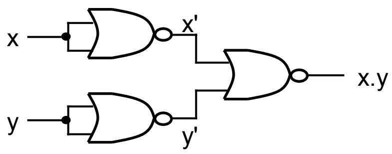
NOR Gate

- گیت NOR نیز حالت self-sufficient دارد (گیت یونیورسال)
- می توان گیتهای AND/OR/NOT را با آن ساخت.
- پیاده سازی گیت NOT با NOR :



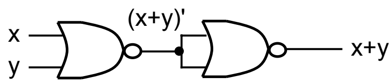
$(x+x)' = x'$ (T1: idempotency)

- پیاده سازی گیت AND با NOR :



$((x+x)'+(y+y)')'=(x'+y')'$	idempotency
$= x''.y''$	DeMorgan
$= x.y$	involution

- پیاده سازی گیت OR با NOR :



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

پیاده سازی با استفاده از گیت NAND

- می توان هر تابع بولی را با استفاده از گیت NAND پیاده سازی کرد:

مراحل :

(i) فرم SOP را بدست آورید:

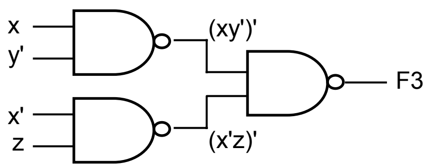
e.g. $F3 = xy'+x'z$

(ii) با استفاده از قضیه DeMorgan فرمی را بدست آورید که بصورت NAND دوطبقه پیاده سازی شود.

e.g. $F3 = xy'+x'z$

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

$= ((xy')' . (x'z)')'$ DeMorgan



پیاده سازی با استفاده از گیت NOR

- می توان هر تابع بولی را با استفاده از گیت NOR پیاده سازی کرد:

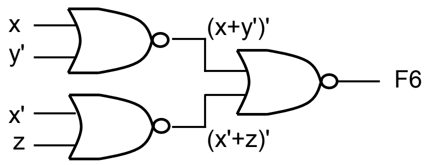
مراحل :

(i) فرم POS را بدست آورید:

$$e.g.\ F6 = (x+y').(x'+z)$$

(ii) با استفاده از قضیه DeMorgan فرمی را بدست آورید که بصورت NOR دوطبقه پیاده سازی شود.

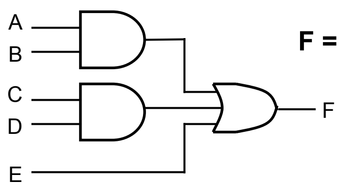
$$e.g.\ F6 = (x+y').(x'+z)$$
$$= ((x+y').(x'+z))'' \quad \text{involution}$$
$$= ((x+y')'+(x'+z)')' \quad \text{DeMorgan}$$



پیاده سازی های دوطبقه فرم SOP

- فرم SOP می تواند به چهارحالت دوطبقه پیاده سازی گردد:
  - 2-level AND-OR logic circuits
  - 2-level NAND-NAND logic circuits
  - 2-level NOR-OR logic circuits
  - 2-level OR-NAND logic circuits

- AND-OR logic circuit



F = AB + CD + E

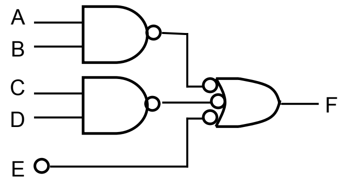
پیاده سازی های دوطبقه فرم SOP

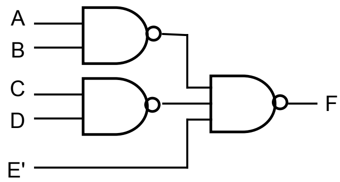
- NAND-NAND circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت OR با ورودی های NOT شده را با گیت NAND جایگزین کنید.

NOT برای متغیر های ورودی با مکمل آن جایگزین می شود.





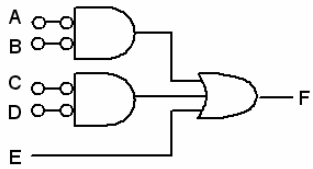
پیاده سازی های دوطبقه فرم SOP

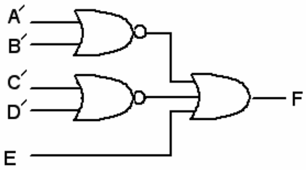
- NOR-OR circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت AND با ورودی های NOT شده را با گیت NOR جایگزین کنید.

NOT برای متغیر های ورودی با مکمل آن جایگزین می شود.





پیاده سازی های دوطبقه فرم SOP

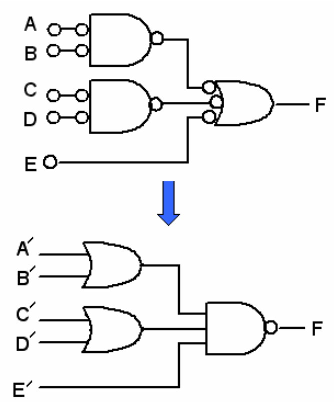
- OR-NAND circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت OR با ورودی های NOT شده را با گیت NAND جایگزین کنید.

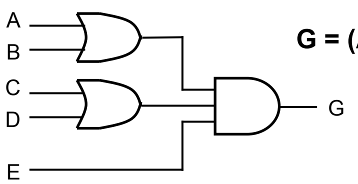
ج- گیت NAND با ورودی های NOT شده را با گیت OR جایگزین کنید.

NOT برای متغیر های ورودی با مکمل آن جایگزین می شود.



پیاده سازی های دوطبقه فرم POS

- فرم POS نیزمی تواند به جهارحالت دوطبقه پیاده سازی گردد:
  - 2-level OR-AND logic circuits
  - 2-level NOR-NOR logic circuits
  - 2-level NAND-AND logic circuits
  - 2-level AND-NOR logic circuits
- OR-AND logic circuit



G = (A+B).(C+D).E

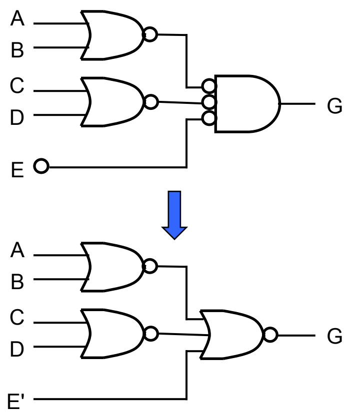
پیاده سازی های دوطبقه فرم POS

- NOR-NOR circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت AND با ورودی های NOT شده را با گیت NOR جایگزین کنید.

NOT برای متغیر های ورودی با مکمل آن جایگزین می شود



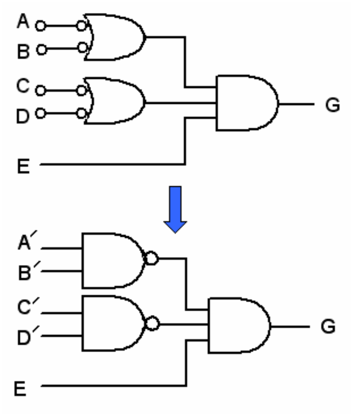
پیاده سازی های دوطبقه فرم POS

- NAND-AND circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت OR با ورودی های NOT شده را با گیت NAND جایگزین کنید.

NOT برای متغیر های ورودی با مکمل آن جایگزین می شود



پیاده سازی های دوطبقه فرم POS

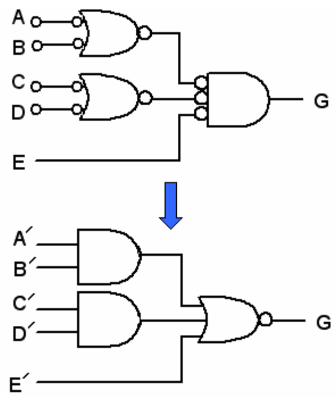
- AND-NOR circuit

الف- در هر مسیر دو عدد NOT قرار دهید.

ب- گیت AND با ورودی های NOT شده را با گیت NOR جایگزین کنید.

ج- گیت NOR با ورودی های NOT شده را با گیت AND جایگزین کنید.

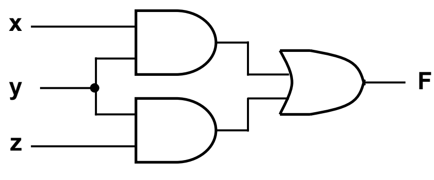
NOT برای متغیر های ورودی با مکمل آن جایگزین می شود



تمرین ۶-۲

Q1. Draw a logic circuit for BD + BE + D’F using only NAND gates. Use both DeMorgan method and SOP method.

Q2. Transform the following AND-OR Circuit to NAND circuit.



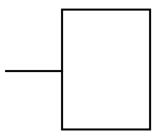
Q3. Using only NOR gates, draw a logic circuit using POS method for (A+B+C’)(B’+C’+D)

منطق مثبت ومنفی
Positive & Negative Logic

- معمولاً در گیت های منطقی داریم:
  - H (high voltage, 5V) = 1
  - L (low voltage, 0V) = 0
- این قرارداد منطق مثبت می باشد.
- با این وجود قرار داد معکوس یعنی منطق مثبت نیز می تواند استفاده گردد.
  - H (high voltage) = 0
  - L (low voltage) = 1
- بسته به نوع منطق استفاده شده یک گیت ممکن است عبارت بولی متفاوتی را نمایش دهد.

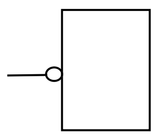
منطق مثبت ومنفی
Positive & Negative Logic

Positive logic:

Enable


Active High:
0: Disabled
1: Enabled

Negative logic:

$\overline{\text{Enable}}$


Active Low:
0: Enabled
1: Disabled

خانواده مدار های مجتمع (IC) دیجیتالی
Integrated Circuit Logic Families

- بعضی خانواده های مدار های منطقی دیجیتالی: TTL, CMOS, ECL
  - TTL: **T**ransistor-**T**ransistor **L**ogic.
    - از ترانزیستور های BJT (bipolar junction transistors) استفاده می کند.
    - شامل مدار های زیر است: standard TTL, low-power TTL, Schottky TTL, low-power Schottky TTL, advanced Schottky TTL, etc.
  - CMOS: **C**omplementary **M**etal-**O**xide **S**emiconductor.
    - از ترانزیستور های FET (field-effect transistors) استفاده می کند
  - ECL: **E**mitter **C**oupled **L**ogic.
    - از تکنولوژی مدار های دوقطبی (bipolar circuit technology) استفاده می کند.
    - دارای سرعت سویچینگ بالایی هستند ولی مصرف توان آنها بالاست.

خانواده مدارهای مجتمع (IC) دیجیتالی
Integrated Circuit Logic Families

TTL Series	Prefix Designation	Example of Device
Standard TTL	54 or 74	7400 (quad NAND gates)
Low-power TTL	54L or 74L	74L00 (quad NAND gates)
Schottky TTL	54S or 74S	74S00 (quad NAND gates)
Low-power Schottky TTL	54LS or 74LS	74LS00 (quad NAND gates)

خانواده مدار های مجتمع (IC) دیجیتالی

- مشخصات ویژه:

  - Fan-Out یا گنجایش خروجی: حداکثر انشعاب هایی که می توان از خروجی گیت گرفت، بدون اینکه تاثیر نامطلوب در عملکرد آن داشته باشد. برای گرفتن انشعاب بیش از مقدار Fan-Out می توان از بافر استفاده کرد.
  - Propagation delay time (nS) یا تاخیر انتشار: زمانی که طول می کشد بعد از اعمال ورودی به گیت اثر آن در خروجی ظاهر گردد.

  تاخیر انتشار کل یک مدار منطقی برابر تعداد طبقات آن ضربدر تاخیر یک طبقه(یک گیت) است. از این لحاظ پیاده سازی های دو طبقه، سرعت بالاتری خواهند داشت و از نقطه نظر پیاده سازی اهمیت زیادی دارند.

  - Power dissipation (mw) یا مصرف توان: مصرف توان هر گیت را مشخص می کند.
  - Speed-power product (SPP): حاصلضرب تاخیر انتشار در مصرف توان می باشد.
  - Noise margin (V) یا محدوده نویز: حد اکثر ولتاژ نویزی را که گیت می تواند بدون تغییر نا مطلوب تحمل کند نشان می دهد.