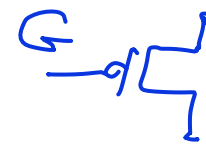


a	b	out
L	L	H
L	H	L
H	L	L
H	H	L

Positive Logic



$G=L \rightarrow$ وصل

$G=H \rightarrow$ وصل

$G=H \rightarrow$ قطع

$G=k \rightarrow$ قطع

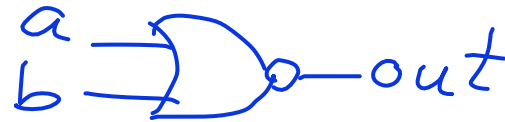
قرارداد اول (منطق مثبت)

$L \leftrightarrow 0$

$H \leftrightarrow 1$

a	b	out
0	0	1
0	1	0
1	0	0
1	1	0

\rightarrow NOR



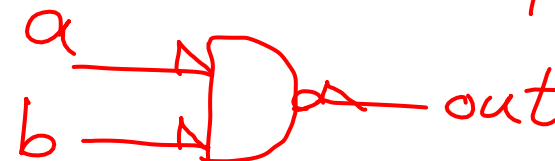
قرارداد دوم (منطق منفی) Negative Logic

$L \leftrightarrow 1$

$H \leftrightarrow 0$

a	b	out
1	1	0
1	0	1
0	1	1
0	0	1

\rightarrow NAND



منطق مثبت

AND

OR

NAND

NOR

XOR

منطق منفی

OR

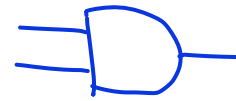
AND

NOR

NAND

XNOR

منطق مثبت

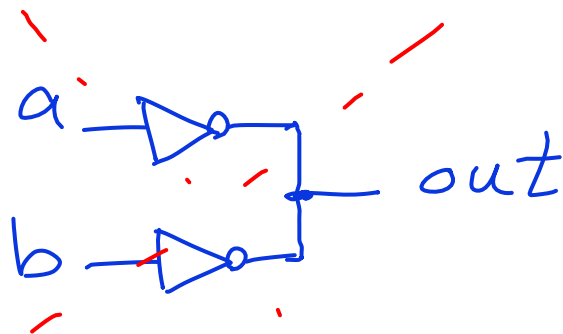


≡

منطق منفی



≡



$a=0 \rightarrow out=1$
 $b=0 \rightarrow out=1$

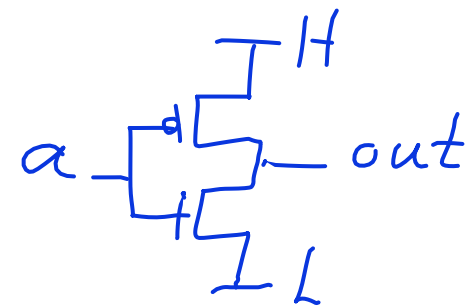
a	b	out
0	0	1
0	1	X
1	0	X
1	1	0

مقدار
نامشخص

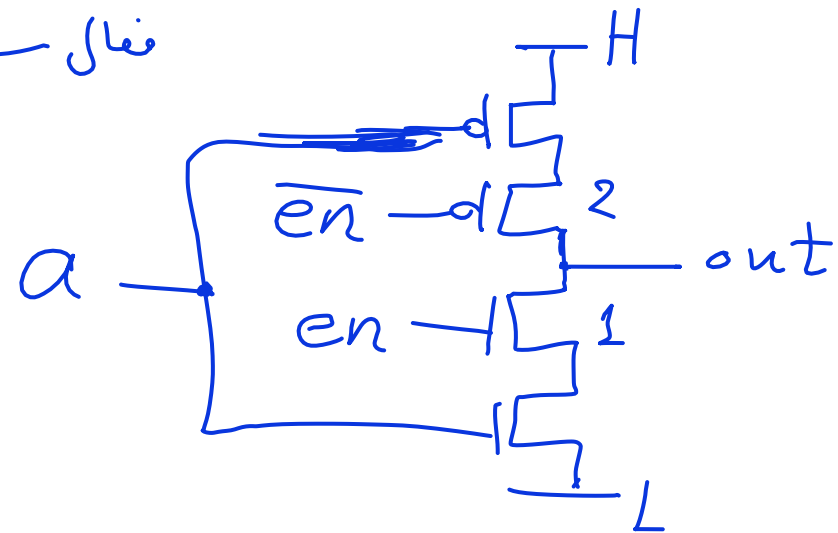
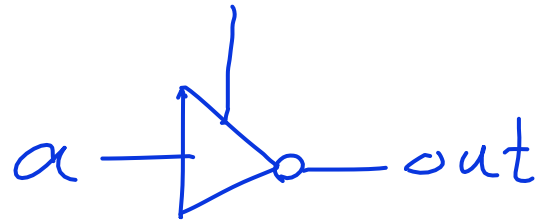
Three-state
Tri-state

گیت سه حالتی

$$a \rightarrow \text{out} = \begin{cases} 0 & \text{منقطع} \\ 1 & \text{موصول} \end{cases}$$



فعال : $en \rightarrow enable =$

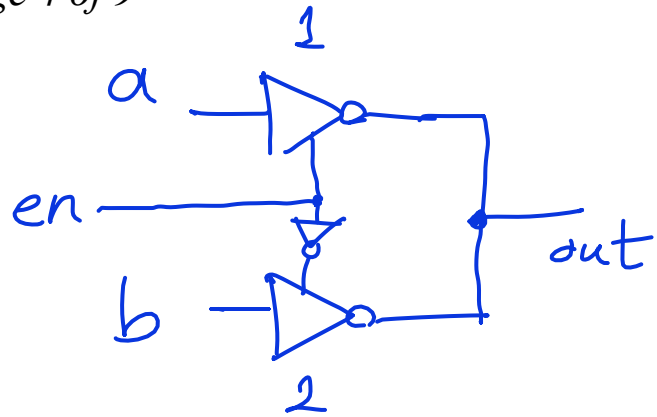


$$en=1 \rightarrow T_1 = \text{موصول}, T_2 = \text{موصول} \Rightarrow out = a'$$

$$en=0 \rightarrow T_1: \text{مقطع}, T_2: \text{مقطع} \Rightarrow out = \text{فلوت} \text{ float}$$

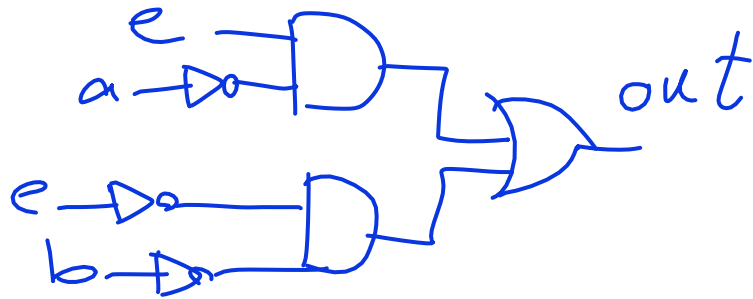
$$out = \begin{cases} 0 \\ 1 \\ Z \end{cases} \text{ منقطع}$$

High-Impedance = High-Z



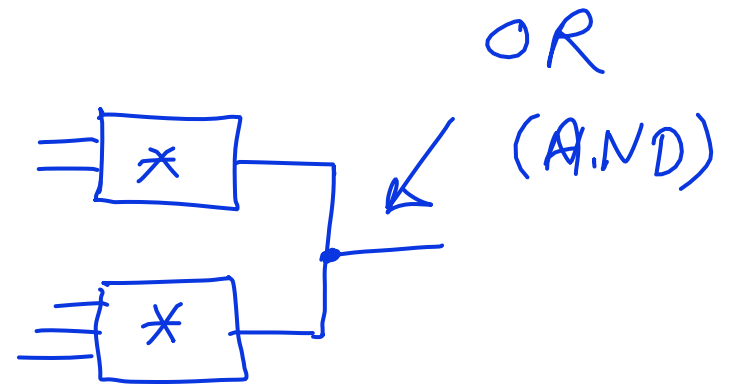
$en = 0 \rightarrow \begin{cases} \text{INV 1: قطع (Z)} \\ \text{INV 2: } \text{yes} \rightarrow \text{out} = b' \end{cases}$
 $en = 1 \rightarrow \begin{cases} \text{INV 1: } \text{yes} \rightarrow \text{out} = a' \\ \text{INV 2: قطع} \end{cases}$

$$\text{out} = e \cdot a' + e' \cdot b'$$



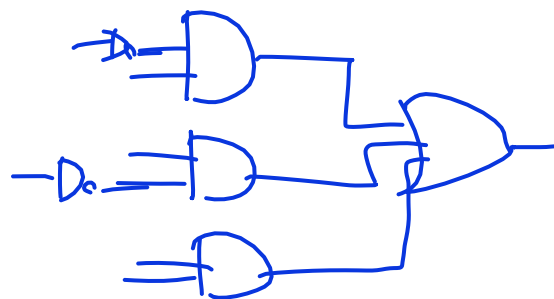
wired Logic $\begin{cases} \text{wired-OR} \\ \text{wired-AND} \end{cases}$

مخرج ~



منطق کامل: مجموعه‌ای از گیت‌ها، که با آنها بتوان هر تابعی را ساخت.
complete

$$f = \{+, \cdot, \sim\}$$



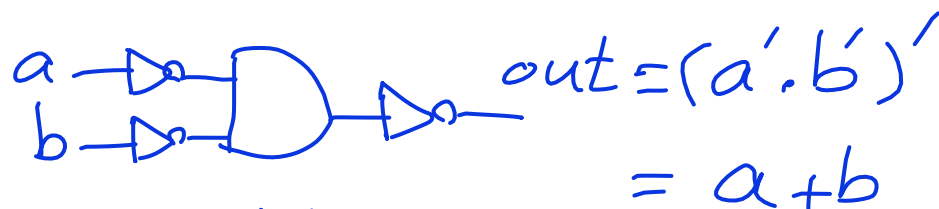
$\{AND, OR, NOT\}$

منطق کامل

منطق کامل نیست: $\{AND, OR\}$

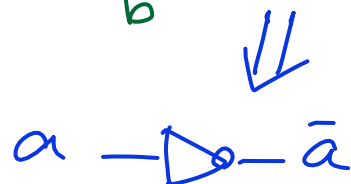
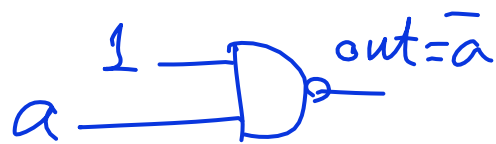
منطق کامل $\{AND, NOT\}$

منطق کامل $\{OR, NOT\}$



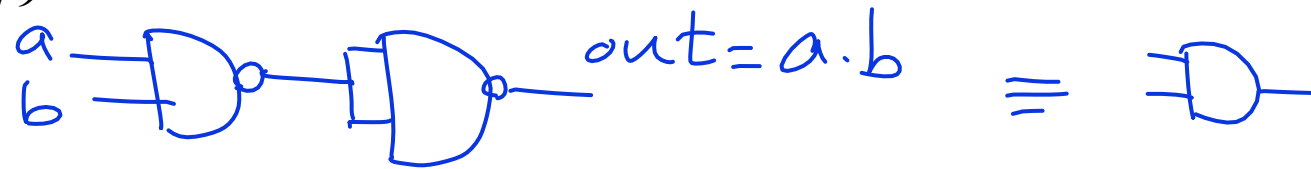
$$a \oplus b = ab' + a'b$$

|||

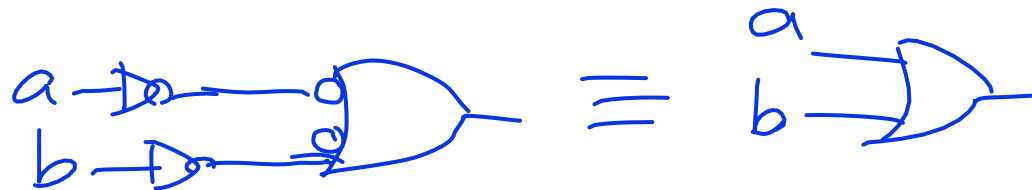
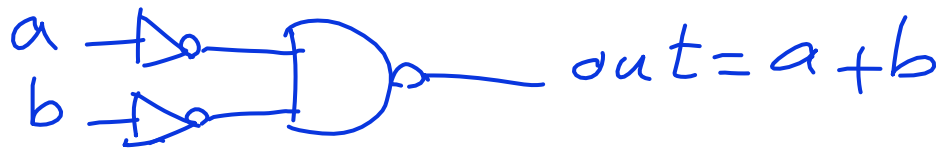


a	b	a . b	{NAND}
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

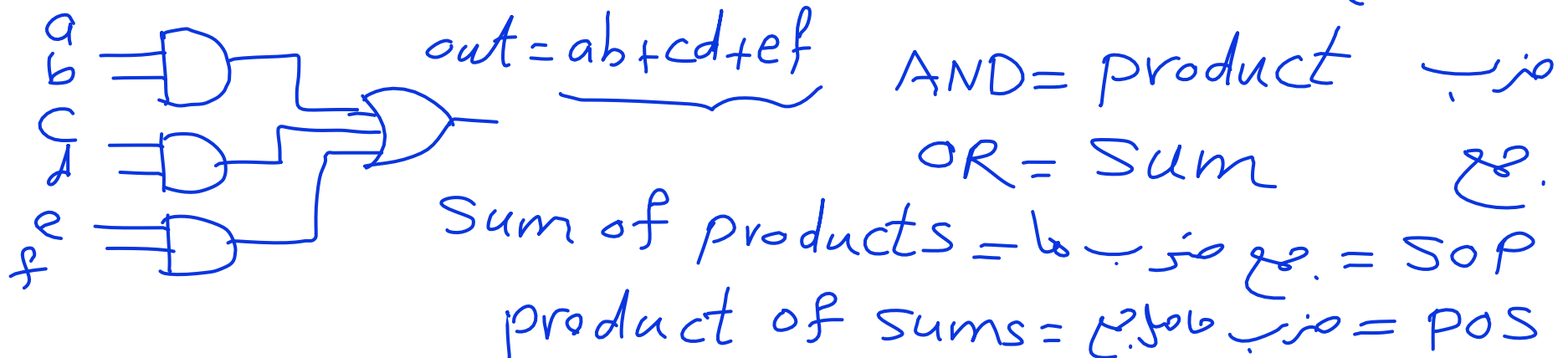
a	out
0	1
1	0

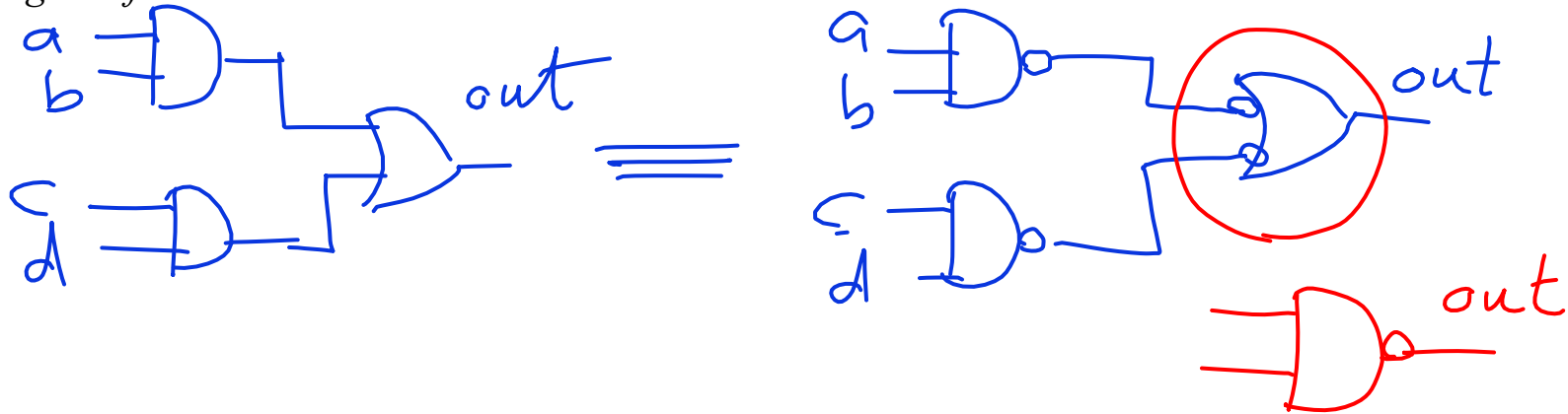


$$(ab)' = a' + b'$$



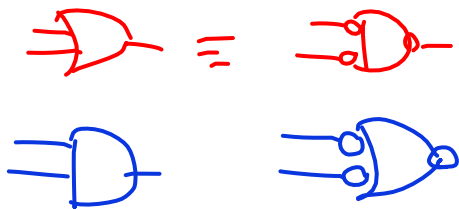
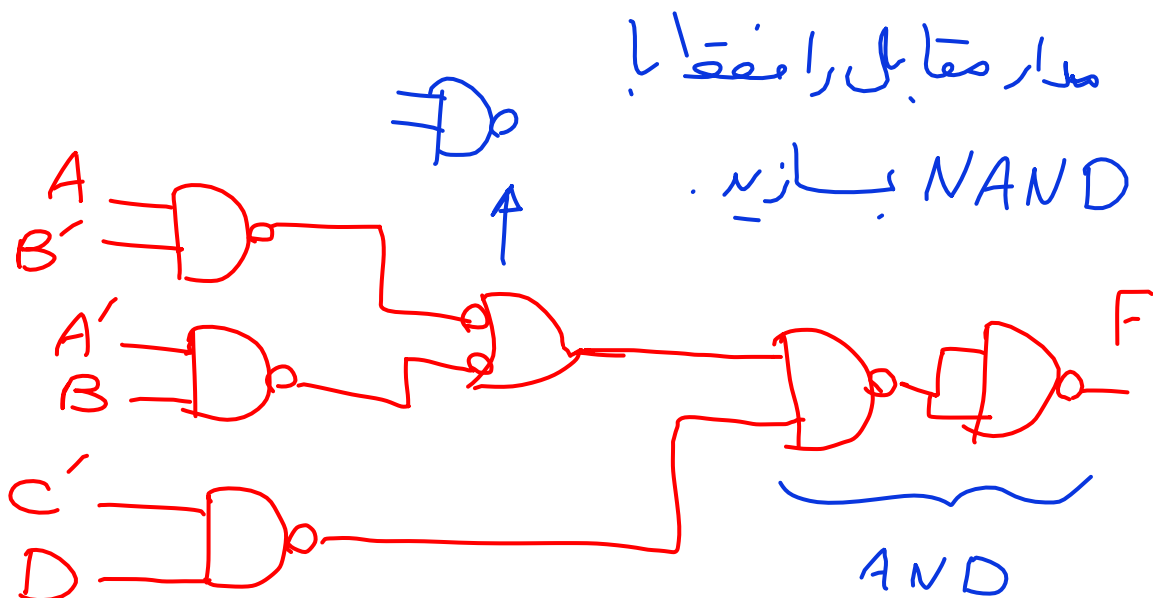
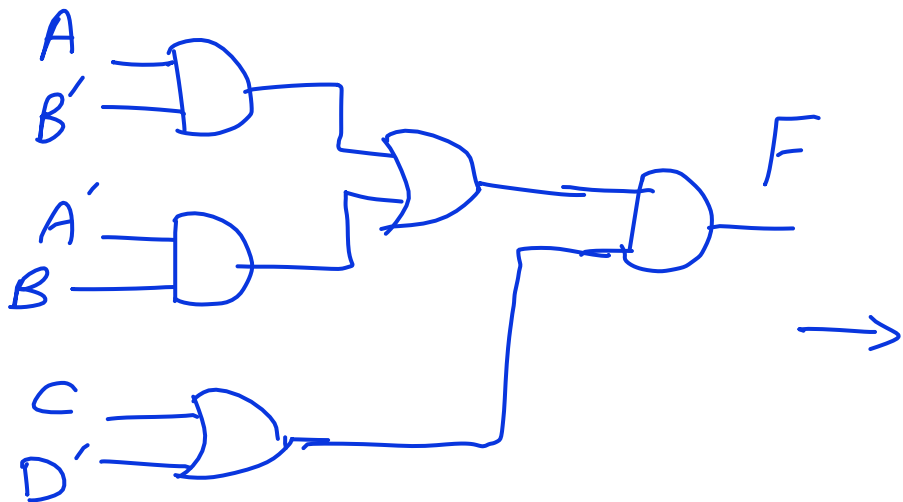
جواب {NOR}

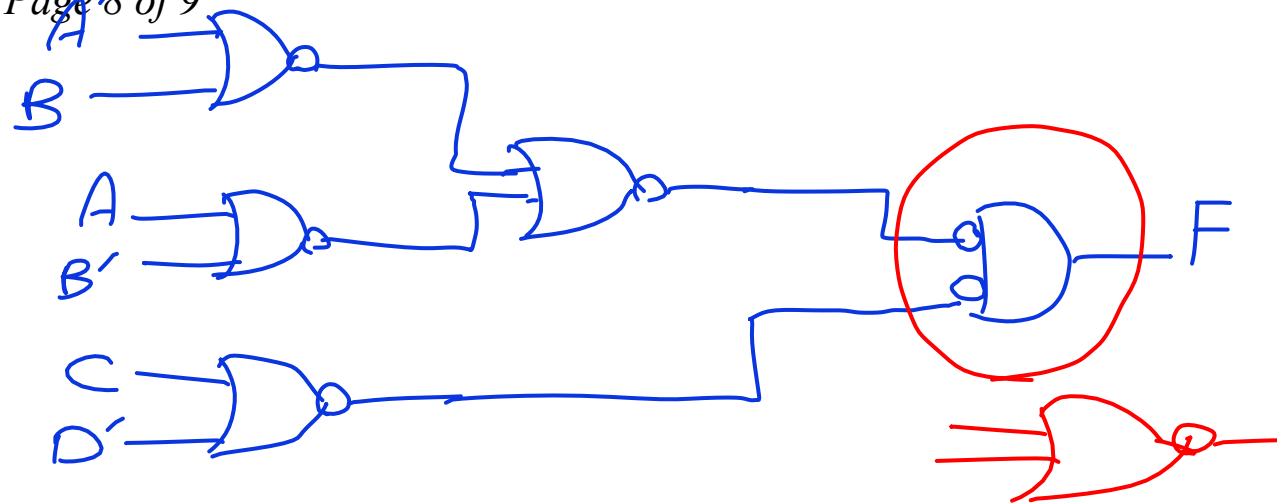




هر تابع SOP (AND-OR) مستقیماً با فضا NAND قابل پیاده سازی است.

$\Rightarrow \quad \sim \quad \sim \quad \text{NOR} \quad \sim \quad \sim \quad \sim \quad (\text{OR-AND}) \quad \text{POS} \quad \sim \quad \sim$





مدار بالا را فقط با
NOR بسازید.

به دست آوردن تابع از روی جدول درستی

$Literal =$ حرف : عبارتی که تنها از یک متغیر یا تنها از نفیض یک متغیر تشکیل شده.

a و b' ~~ab~~

$f(a, b, c)$

عبارت ضربی (PT = product Term): یک حرف تنها، و یا ضرب (AND) تعدادی

حرف مثال: a' , ab , $a'b'c'$, ~~$ab+c$~~

عبارت جمع (ST = Sum Term): یک حرف تنها، و یا جمع (OR) تعدادی حرف

مثال: a' , $a+b+c'$, ~~$a'+bc'$~~

$$f = a + bc' \cdot (d+g)$$

$$\begin{aligned} f' &= [a + bc' \cdot (d+g)]' = a' \cdot (bc' \cdot (d+g))' \\ &= a' \cdot (\overline{bc'} + (d+g)') \\ &= a' \cdot [(b'+c) + d'g'] \\ &= a'b' + a'c + a'd'g' \end{aligned}$$