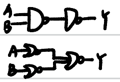
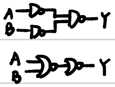
4 Two-level Implementation

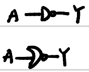
AND $Y=AB$



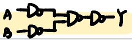
OR $Y=A+B$



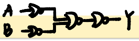
NOT $Y=\overline{A}$



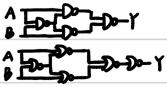
NOR $Y=\overline{A+B}$



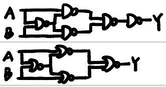
NAND $Y=\overline{A\cdot B}$



XOR $Y=A\oplus B$



XNOR $Y=A\odot B$



NAND & NOR implementation

- NAND and NOR gates are easier to fabricate with electronic components.
- They are the basic gates used in all IC digital logic families.

NAND Gates.

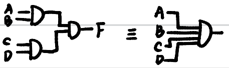
①简化 & express in sum of product form.

② Draw a NAND Gate for each product term that has a least two literals.

③ Draw a single gate using the AND–invert or the invert–OR graphic symbol in the second level, with inputs coming from outputs of first–level gates.

④A term with a single literal requires an inverter in the first level. However, if the single literal is complemented, it can be connected directly to an input of the second level NAND gate.

Non-degenerate forms.



two level using AND, OR, NAND, NOR共有16种.
其中有8种 said to be degenerate form. 可以变为一层

degenerate   Non-degenerate

AND-OR-INVET implementation. ↗其中一种

The two forms NAND-AND ≡ AND-NOR

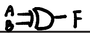
Exclusive-OR function.

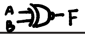
$x\oplus y = xy' + x'y$.

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

$x\oplus 0 = x$ $x\oplus 1 = x'$ $x\oplus x = 0$ $x\oplus x' = 1$ $x\oplus y' = x'\oplus y = (x\oplus y)'$

XOR

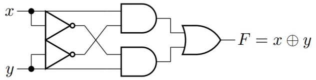




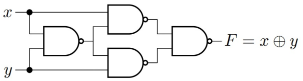
XNOR or equivalence

XOR is hard to fabric ⇒ constructed by other gates.

- $(x'+y')x+(x'+y')y = xy' + xy' = x\oplus y$.



Or use NAND gates:



Odd function.

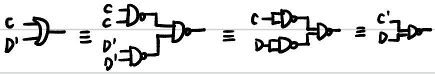
$A \oplus B \oplus C = \Sigma(1,2,4,7)$

An odd number of variables are equal to 1 ⇒ the three-variable function =1.

Parity generation and checking. (one use of XOR gate)

The circuit that generates the parity bit in the transmitter is called a parity generator. The circuit that checks the parity in the receiver is called a parity checker.

$F = C+D' = CC+DD'$ For other degenerate form, we can degenerate them.


C
D'
≡
C
C
D'
D'
≡
C
D
≡
C'
D