Lecture 2-1
Boolean Algebra

Inversion

DeMorgan's Laws

$(X1 + X2 + X3 + \ldots)'$

$= X1' \cdot X2' \cdot X3' \ldots..$

- Example

[(AB' + C)D' + E]'

$= [(AB' + C)D']' E'$

$= [(AB' + C)' + D]E'$

$= [(AB')'C' + D] E'$

= [(A' + B) C' + D] E'

Duality

- Duality
  - Dual is formed by replacing AND with OR, OR with AND, 0 with 1, and 1 with 0.
  - Variable and complements are left unchanged.

Example

$F = ab' + c + 0 \cdot d'(1 + e)$

$F^D = (a + b')c(1 + d' + 0 \cdot e)$

If $F = G$, then $F' = G'$, $F^D = G^D$.

- This means that if a theorem is true, so is its dual.
- $(X+Y')Y = XY$
- $(XY') + Y = X + Y$

Duality (Cont.)

- Another way of doing duality.
  - Dual of a function is formed by complementing a function and then replacing all variables by their complements.

$F(a,b) = ab'$

$F^D = a+b'$

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

$F^D = a+b'$

Multiplying Out and Factoring

- Multiplying out: Change a POS form into the corresponding SOP form.
- Factoring: SOP to POS

$$X(Y+Z) = XY + XZ$$

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

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

- Example:
  - $(Q + AB')(C'D + Q') = QC'D + Q'AB$

Exclusive OR

- Truth table

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

- XOR: Count No. of 1s, true if this is an odd number.
- XOR = Modulo2 operation (add without carry)

Exclusive OR

- XOR theorems

$X \oplus 0 = X$

$X \oplus 1 = X'$

$X \oplus X = 0$

$X \oplus X' = 1$

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

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

$= X'Y + XY'.$

$(X \oplus Y) \oplus Z$

$= X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$

$X(Y \oplus Z) = XY \oplus XZ$

$(X \oplus Y)' = X \oplus Y' = X' \oplus Y = XY + X'Y'$

Exclusive OR: XOR

- XOR logic

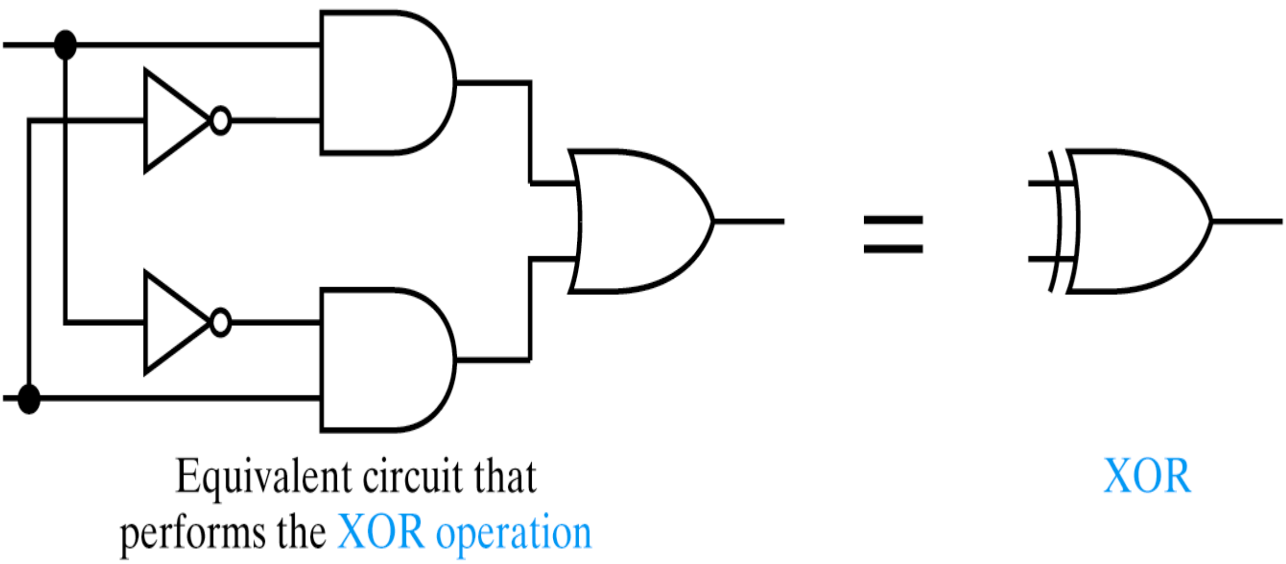


Figure 2.2.8 Equivalent circuit and graphic symbol for the XOR operation.

Equivalence

- $X \equiv Y = XY + X'Y'$
  - Equivalence is the complement of the XOR.
  - Equivalence = XOR'
    - $XY + X'Y' = (X'Y + XY')'$
  - Exclusive-NOR gate.

X Y	$X \equiv Y$
0 0	1
0 1	0
1 0	0
1 1	1

Equivalence: XNOR

- $X \equiv Y = XY + X'Y'$

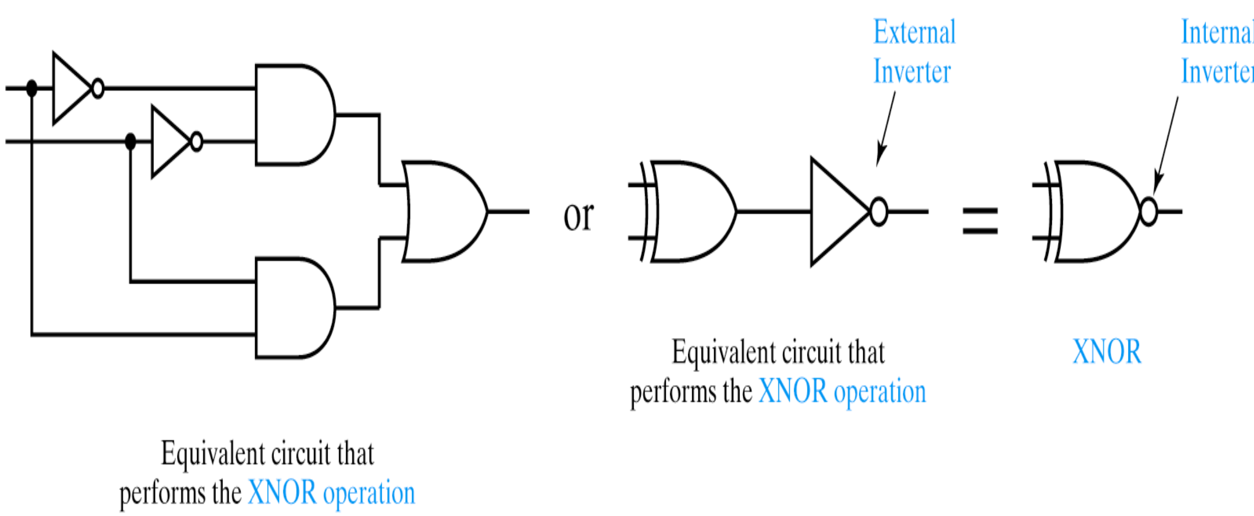


Figure 2.2.9 Equivalent circuits and graphic symbol for the XNOR operation.

Positive Logic and Negative Logic

- Positive Logic:
  - Higher voltage level represents a logic 1, and lower level represents a logic 0.
- The function realized for negative logic is the dual of the function for positive logic.
  - For a given set of input voltages, the output voltage is the same in both cases (positive and negative logic).

Positive Logic and Negative Logic

- If we use positive logic to determine the logic function, it is an OR gate.
- If we use negative logic to determine the logic function, it is an AND gate.

(a) voltages

e_1	e_2	e_3	e_o
0	0	0	0
0	0	+V	+V
0	+V	0	+V
0	+V	+V	+V
+V	0	0	+V
+V	0	+V	+V
+V	+V	0	+V
+V	+V	+V	+V

(b) positive logic

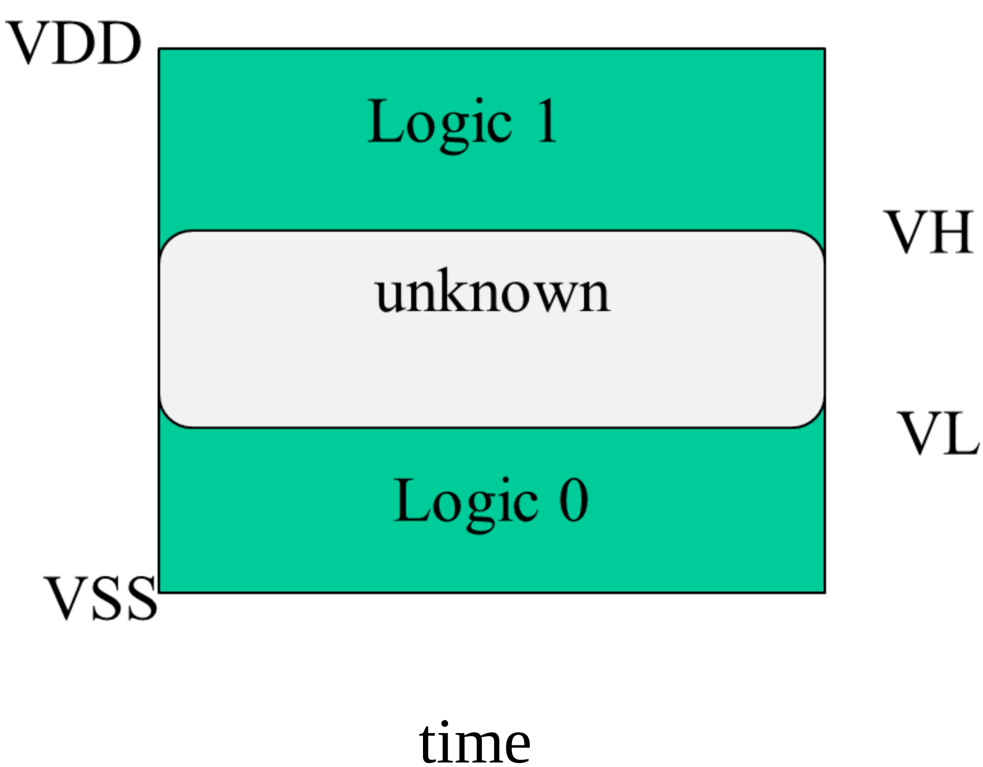
e_1	e_2	e_3	e_o
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(c) negative logic

e_1	e_2	e_3	e_o
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

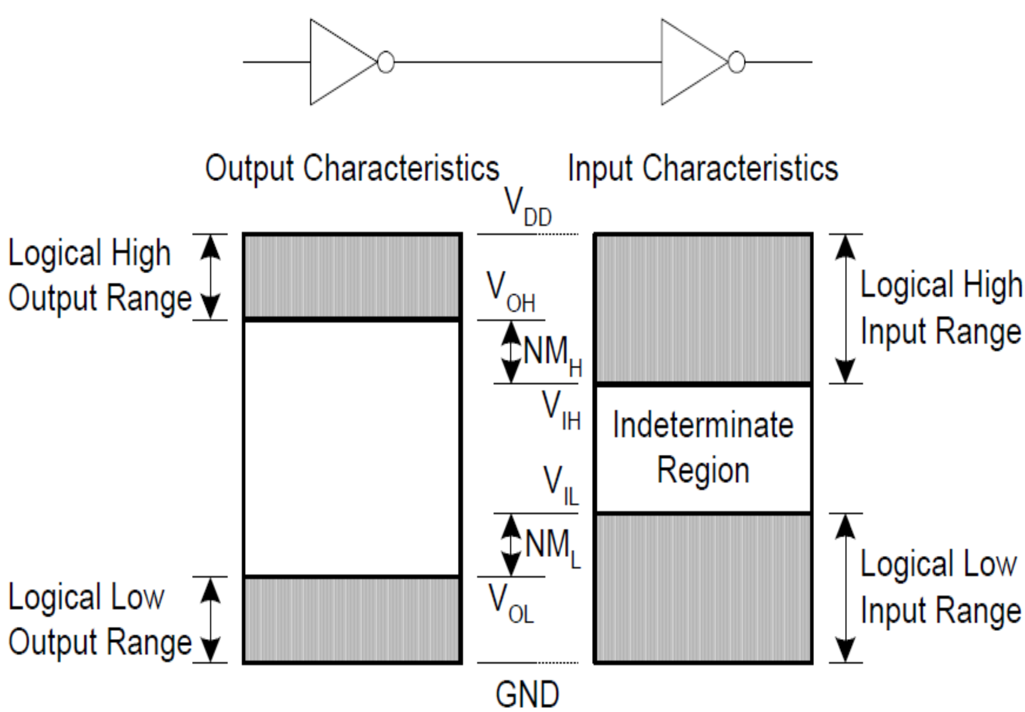
Why 1 and 0?

- **Noise margin:** A digital circuit might be designed to swing between 0.0 and 1.2 volts, with anything below 0.2 volts (VL) considered a '0', and anything above 1.0 volts considered a '1'. Then the noise margin for a '0' would be the amount that a signal is below 0.2 volts, and the noise margin for a '1' would be the amount by which a signal exceeds 1.0 volt (VH).



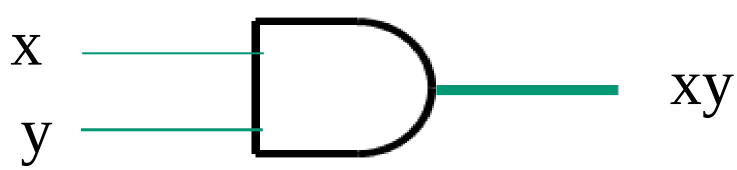
Noise margin

- How much noise can a gate input see before it does not recognize the input ?
- Logic level matching: levels at the output of one gate must be sufficient to drive next gate



Noise Immunity

- The higher the noise margin, the ______ the noise immunity



time