COE 202- Digital Logic

Standard & Canonical Forms

Dr. Abdulaziz Y. Barnawi
COE Department
KFUPM

Outline

- Minterms and Maxterms
- From truth table to Boolean expression
  - Sum of minterms
  - Product of Maxterms
- Standard and Canonical Forms
- Implementation of Standard Forms
- Practical Aspects of Logic Gates

MinTerms

- A **product term** is a term where literals are ANDed.
  - Example: x'y', xz, xyz, …
- A **minterm** is a product term in which all variables appear exactly once, in normal or complemented form
  - Example: F(x,y,z) has 8 minterms: $\bar{x}\bar{y}\bar{z}, \bar{x}\bar{y}z, \bar{x}y\bar{z}, \bar{x}yz, x\bar{y}\bar{z}, x\bar{y}z, xy\bar{z}$ & xyz
- Each minterm equals 1 at exactly one particular input combination and is equal to 0 at all other combinations
- Thus, for example, $\bar{x}\,\bar{y}\,\bar{z}$ is always equal to 0 except for the input combination xyz = 000, where it is equal to 1.

MinTerms

- In general, minterms are designated m_i, where i corresponds the input combination at which this minterm is equal to 1.
- Accordingly, the minterm $\bar{x}\,\bar{y}\,\bar{z}$ is referred to as m_0.

Minterms for Three Variables *Src: Mano's book*

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\bar{X}\bar{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\bar{X}Y\bar{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\bar{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\bar{Y}\bar{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\bar{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\bar{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

MinTerms

- In general, for n-input variables, the number of minterms = the total number of possible input combinations = 2^n.
- Question: What is the number of minterms for a function with 5 input variables?
  - Number of minterms = 2^5 =32 minterms.

MaxTerms

- A **sum term** is a term where literals are ORed.
  - Example: x’+y’, x+z, x+y+z, ...
- A **maxterm** is a sum term in which all variables appear exactly once, in normal or complemented form
  - Example: F(x,y,z) has 8 Maxterms: (x+y+z), (x+y+z’), (x+y’+z), ...
- Each Maxterm equals 0 at exactly one of the 8 possible input combinations and is equal to 1 at all other combinations.
- Thus, for example, (x + y + z) equals 1 at all input combinations except for the combination xyz = 000, where it is equal to 0.

MaxTerms

- In general, Maxterms are designated M_i, where i corresponds the input combination at which this Maxterm is equal to 0.

- Accordingly, the minterm (x + y + z) is referred to as M_0.

Maxterms for Three Variables *Src: Mano's book*

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\overline{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7	1	1	1	1	1	1	1	0

MaxTerms

- For n-input variables, the number of Maxterms = the total number of possible input combinations = 2^n.
- Question: What is the number of Maxterms for a function with 5 input variables?
  - Number of Maxterms = 2^5 = 32 Maxterms.
- Using De-Morgan's theorem, or truth tables, it can be easily shown that minterms and Maxterms are the complement of each other!

$$M_i = \overline{m_i} \qquad \forall i = 0,1,2,\ldots,(2^n - 1)$$

Expressing Functions as a Sum of Minterms

- A Boolean function can be expressed algebraically from a give truth table by forming the logical sum (OR) of ALL the minterms that produce 1 in the function
- Example: Consider the function defined by the truth table
- $F(X,Y,Z) = X'Y'Z' + X'YZ' + XY'Z + XYZ$

$$= m_0 + m_2 + m_5 + m_7$$

$$= \Sigma m(0,2,5,7)$$

X	Y	Z	m	F
0	0	0	m_0	1
0	0	1	m_1	0
0	1	0	m_2	1
0	1	1	m_3	0
1	0	0	m_4	0
1	0	1	m_5	1
1	1	0	m_6	0
1	1	1	m_7	1

Expressing Functions as a Product of Sums

- A Boolean function can be expressed algebraically from a give truth table by forming the logical product (AND) of ALL the Maxterms that produce 0 in the function
- Example: Consider the function defined by the truth table
- $F(X,Y,Z) = \Pi M(1,3,4,6)$
- Applying DeMorgan
  $F' = m_1 + m_3 + m_4 + m_6$
  $= \Sigma m(1,3,4,6)$
  $F = F'' = [m_1 + m_3 + m_4 + m_6]'$
  $= m_1'.m_3'.m_4'.m_6'$
  $= M_1.M_3.M_4.M_6$
  $= \Pi M(1,3,4,6)$

X	Y	Z	M	F	F'
0	0	0	M_0	1	0
0	0	1	M_1	0	1
0	1	0	M_2	1	0
0	1	1	M_3	0	1
1	0	0	M4	0	1
1	0	1	M_5	1	0
1	1	0	M_6	0	1
1	1	1	M_7	1	0

Expressing Functions as Sum of Minterms or Product of Maxterms

- Any function can be expressed both as a sum of minterms (Σm_i) and as a product of Maxterms (ΠM_j)
- The product of Maxterms expression (ΠMj) of F contains all Maxterms M_j ($\forall$ $j \neq i$) that do not appear in the sum of minterms expression of F
- The sum of minterms expression of F` contains all minterms that do not appear in the sum of minterms expression of F
- This is true for all complementary functions. Thus, each of the 2^n minterms will appear either in the sum of minterms expression of F or the sum of minterms expression of F` but not both.

Expressing Functions as Sum of Minterms or Product of Maxterms

- The product of Maxterms expression of F` contains all Maxterms that do not appear in the product of Maxterms expression of F
- This is true for all complementary functions. Thus, each of the 2^n Maxterms will appear either in the product of Maxterms expression of F or the product of Maxterms expression of F` but not both

Expressing Functions as Sum of Minterms or Product of Maxterms

- Example: Given that F (a, b, c, d) = Σ (0, 1, 2, 4, 5, 7), derive the product of Maxterms expression of F and the two standard form expressions of F`
- Since the system has 4 input variables (a, b, c & d), the number of minterms and Maxterms = 2^4= 16
- F (a, b, c, d) = Σ(0, 1, 2, 4, 5, 7)
- F (a, b, c, d) = Π (3, 6, 8, 9, 10, 11, 12, 13, 14, 15)
- F` (a, b, c, d) = Σ (3, 6, 8, 9, 10, 11, 12, 13, 14, 15).
- F` (a, b, c, d) = Π (0, 1, 2, 4, 5, 7)

Expressing Functions as Sum of Minterms or Product of Maxterms

- Example: Let F(X,Y,Z)= Y' + X'Z' , express F as a sum of minterms and product of Maxterms

- F = Y' + X'Z'
  = Y'(X+X')(Z+Z') + X'Z'(Y+Y')
  = (XY'+X'Y')(Z+Z') + X'YZ'+X'Z'Y'
  = XY'Z+X'Y'Z+XY'Z'+X'Y'Z'+ X'YZ'+X'Z'Y'
  = $m_5 + m_1 + m_4 + m_0 + m_2 + m_0$
  = $m_0 + m_1 + m_2 + m_4 + m_5$
  = $\Sigma m(0,1,2,4,5)$

- To find the form Π M, consider the remaining indices

- F = Π M(3,6,7)

- What about F'?

Expressing Functions as Sum of Minterms or Product of Maxterms

- Question: F (a,b,c,d) = Σ m(0,1,2,4,5,7), What are the minterms and Maxterms of F and and its complement F?
- Solution:

Expressing Functions as Sum of Minterms or Product of Maxterms

- Question: $F(a,b,c,d) = \Sigma m(0,1,2,4,5,7)$, What are the minterms and Maxterms of F and and its complement F?
- Solution:
- F has 4 variables; $2^4 = 16$ possible minterms/Maxterms

$$F(a,b,c,d) = \Sigma m(0,1,2,4,5,7)$$
$$= \Pi M(3,6,8,9,10,11,12,13,14,15)$$

$$\overline{F}(a,b,c,d) = \Sigma m(3,6,8,9,10,11,12,13,14,15)$$
$$= \Pi M(0,1,2,4,5,7)$$

Operations on Functions

- The AND operation on two functions corresponds to the intersection of the two sets of minterms of the functions

- The OR operation on two functions corresponds to the union of the two sets of minterms of the functions

- Example
  - Let F(A,B,C)= Σ m(1, 3, 6, 7) and G(A,B,C)= Σ m(0,1, 2, 4,6, 7)
  - F . G = Σ m(1, 6, 7)
  - F + G = Σ m(0,1, 2, 3, 4,6, 7)
  - F’ . G = ?
  - F’ = Σ m(0, 2, 4, 5)
  - F’ . G = Σ m(0, 2, 4)

Canonical Forms

- The sum of minterms and the product of Maxterms forms of Boolean expressions are known as canonical forms.
- Canonical form means that all equivalent functions will have a unique and equal representation.
- Two functions are equal if and only if they have the same sum of minterms and the same product of Maxterms.
- Example:
  - Are the functions F1 = a' b' + a c + b c ' and F2 = a' c' + a b + b' c Equal?
  - F1 = a' b' + a c + b c ' = Σm(0, 1, 2 , 5, 6, 7)
  - F2 = a' c' + a b + b' c = Σm(0, 1, 2 , 5, 6, 7)
  - They are equal as they have the same set of minterms.

Standard Forms

- Remember: a product term is a term with ANDed literals. Thus, AB, A'B, A'CD are all product terms
- A minterm is a special case of a product term where all input variables appear in the product term either in the true or complement form
- Remember: a sum term is a term with ORed literals. Thus, (A+B), (A'+B), (A'+C+D) are all sum terms
- A maxterm is a special case of a sum term where all input variables, either in the true or complement form, are ORed together

Standard Forms

- Boolean functions can generally be expressed in the form of a Sum of Products (SOP) or in the form of a Product of Sums (POS)
- The sum of minterms form is a special case of the SOP form where all product terms are minterms.
- The product of Maxterms form is a special case of the POS form where all sum terms are Maxterms.
- The SOP and POS forms are Standard forms for representing Boolean functions.

SOP and POS Conversion

SOP → POS

$F = AB + CD$

$= (AB+C)(AB+D)$

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

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

POS → SOP

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

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

$= A'C+A'D+BCC+BCD$

$= A'C+A'D+BC+BCD$

$= A'C+A'D+BC$

Question1: How to convert SOP to sum of minterms?
Question2: How to convert POS to product of Maxterms?

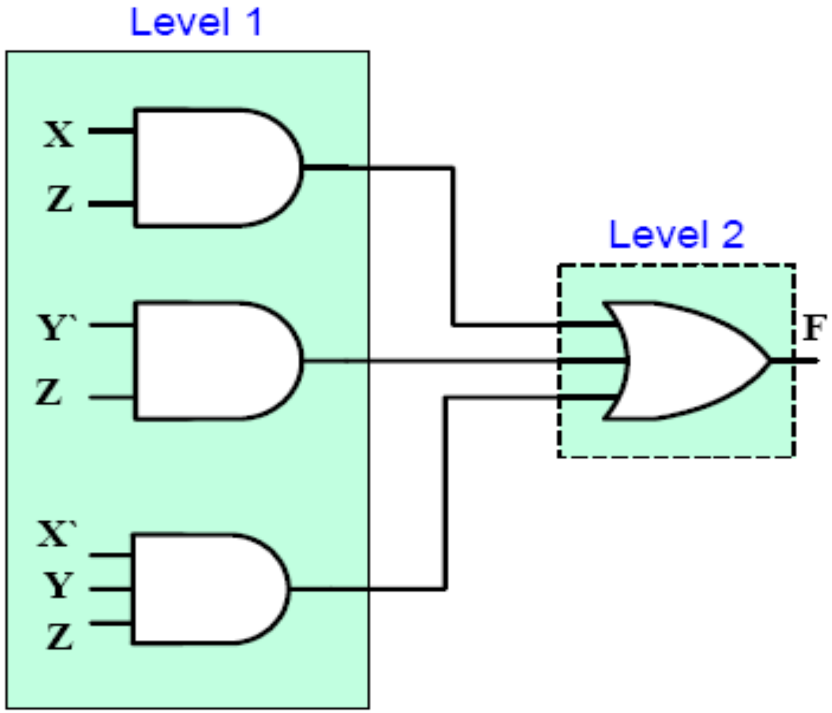
Two-Level Implementations of Standard Forms

- **Sum of Products Expressions (SOP):**
- Any SOP expression can be implemented in 2-levels of gates.
- The first level consists of a number of AND gates which equals the number of product terms in the expression.
- Each AND gate implements one of the product terms in the expression.
- The second level consists of a single OR gate whose number of inputs equals the number of product terms in the expression.

Two-Level Implementations of Standard Forms

- Example: Implement the following SOP function

F = XZ + Y`Z + X`YZ



Two-Level Implementation (F = XZ + Y`Z + X`YZ)
Level-1: AND-Gates ; Level-2: One OR-Gate

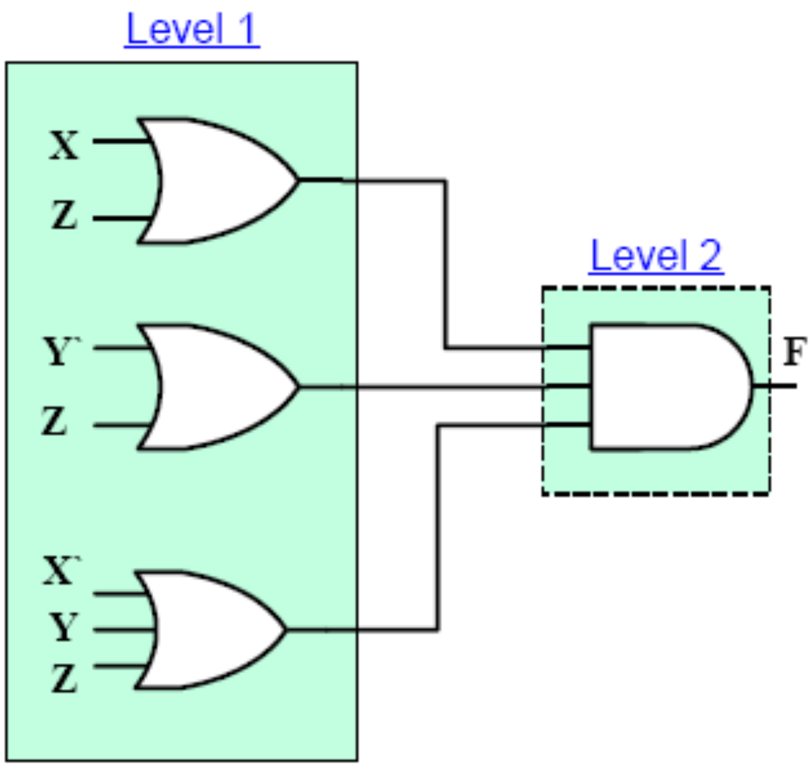
Two-Level Implementations of Standard Forms

- **Product of Sums Expression (POS):**
- Any POS expression can be implemented in 2-levels of gates.
- The first level consists of a number of OR gates which equals the number of sum terms in the expression.
- Each gate implements one of the sum terms in the expression.
- The second level consists of a single AND gate whose number of inputs equals the number of sum terms.

Two-Level Implementations of Standard Forms

- Example: Implement the following POS function

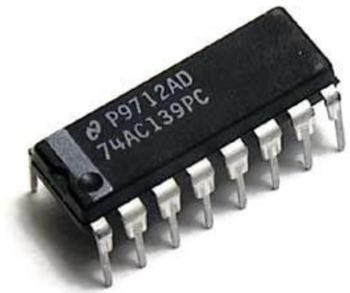
F = (X+Z)(Y`+Z)(X`+Y+Z)

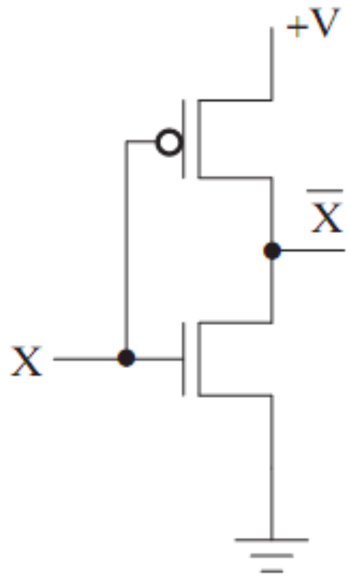


Two-Level Implementation {F = (X+Z)(Y`+Z)(X`+Y+Z)}
Level-1: OR-Gates ; Level-2: One AND-Gate

Practical Aspects of Logic Gates

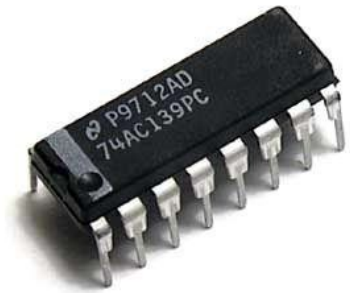
- Logic gates are built with transistors as integrated circuits (IC) or chips.
- ICs are digital devices built using various technologies.
- Complementary metal oxide semiconductor (CMOS) technology
- Level s of Integration:
  - Small Scale Integrated (SSI) < 10 gates
  - Medium Scale Integrated (MSI) < 100 gates
  - Large Scale Integrated (LSI) < 1000 gates
  - Very Large Scale Integrated (VLSI) < 106 gates

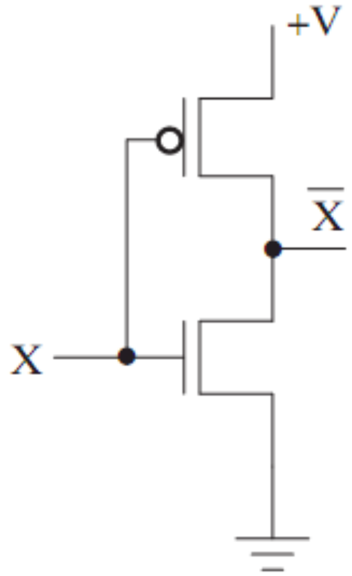




Practical Aspects of Logic Gates

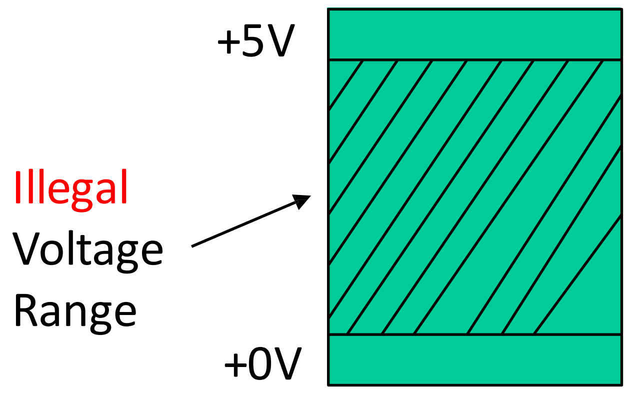
- Key characteristics of ICs are:
  - Voltages ranges
  - Noise Margin
  - Gate propagation delay/speed
  - Fan-in and Fan-out
  - Buffers
  - Tri-state Gates



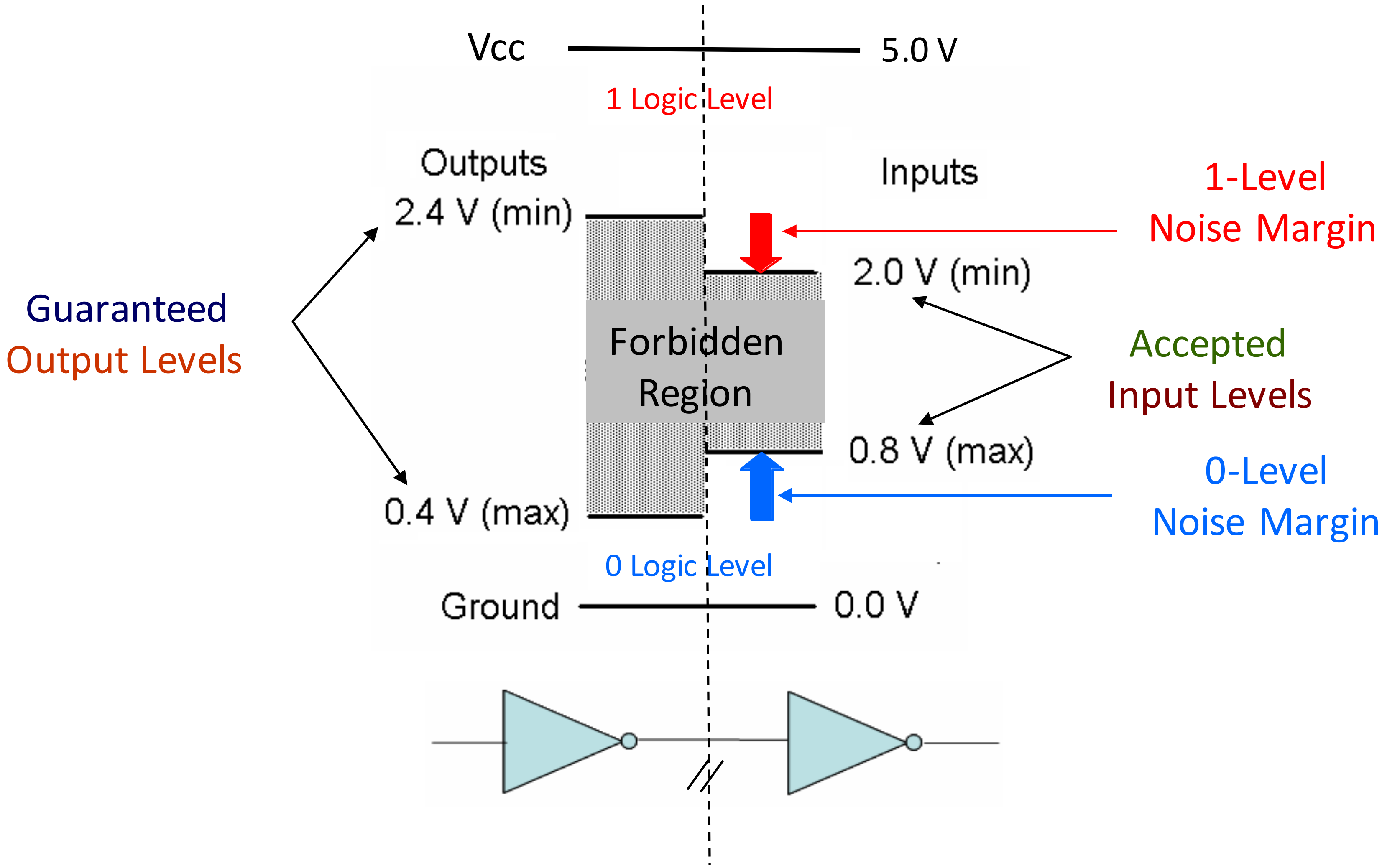


Voltage Levels

- Logic values of 0 & 1 corresponds to voltage level
- A range of voltage defines logic 0 and logic 1
- Any value outside this range is invalid

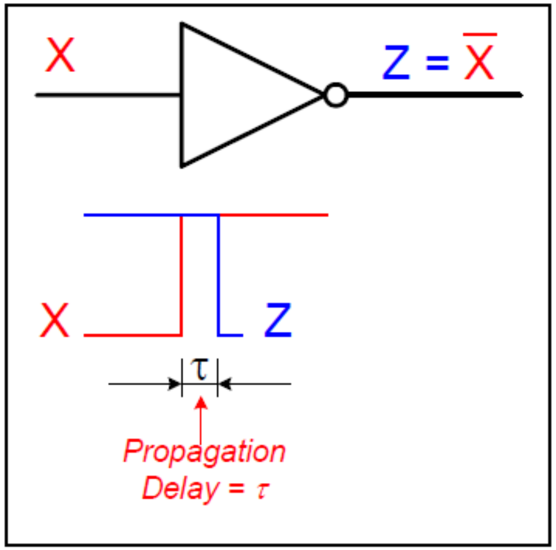


Noise Margins



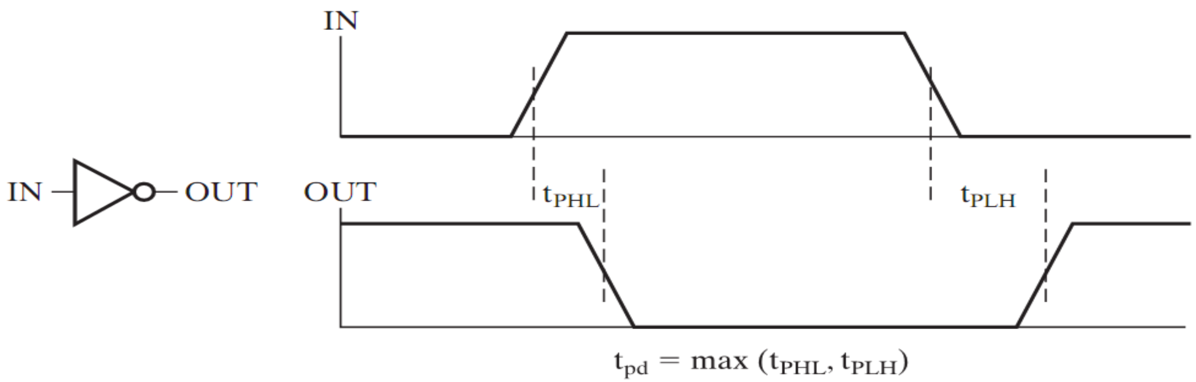
Propagation Delay

- Consider the shown inverter with input X and output Z.
  - A change in the input (X) from 0 to 1 causes the inverter output (Z) to change from 1 to 0.
  - The change in the output (Z), however is not instantaneous. Rather, it occurs slightly after the input change.
  - This delay between an input signal change and the corresponding output signal change is what is known as the **propagation delay**



Propagation Delay

- Formally, the propagation delay (t_{pd}) is the time for a change in the input of a gate to propagate to the output
  - High-to-low (t_{phl}) and low-to-high (t_{plh}) output signal changes may have different propagation delays
  - $t_{pd} = \max \{t_{phl}, t_{plh}\}$

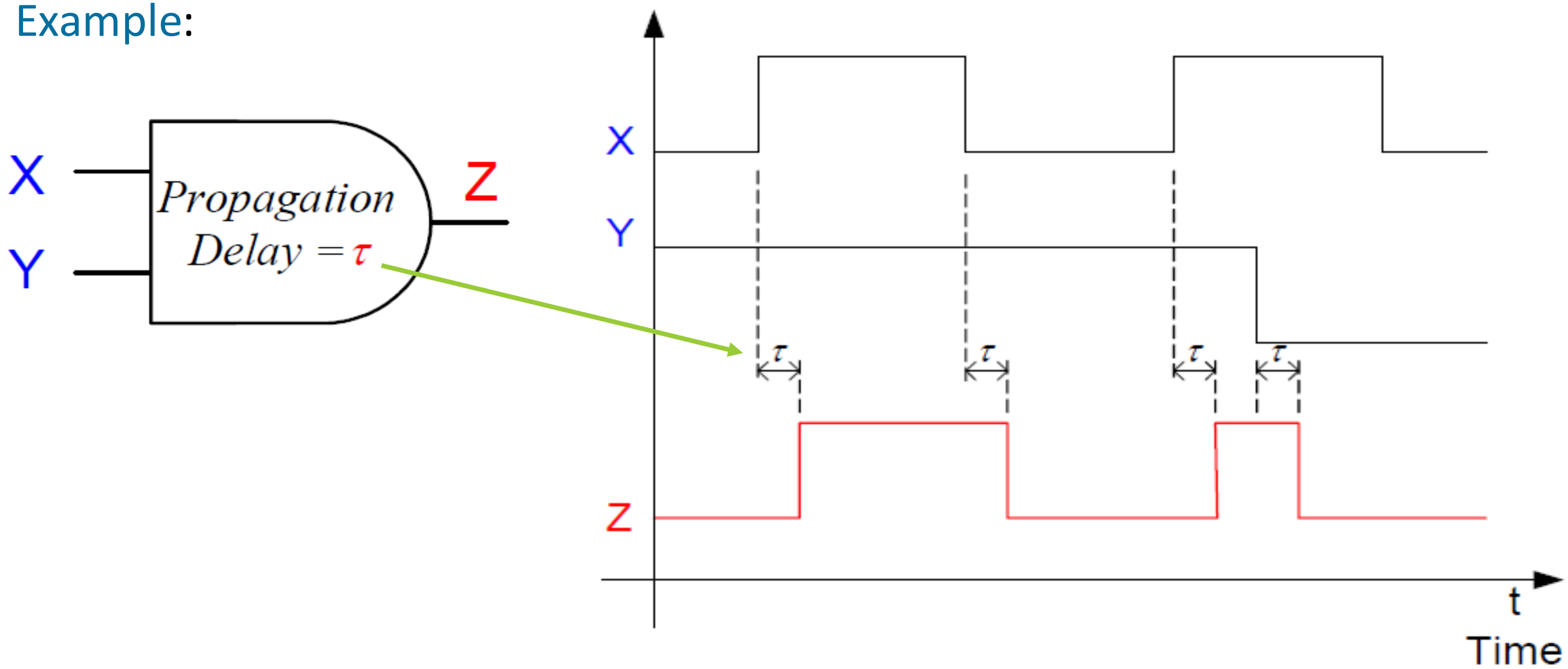


- Faster circuits are characterized by smaller propagation delays

- Higher performance systems require higher speeds, i.e. smaller propagation delays

Propagation Delay

- A timing diagram shows the logic values of signals in a circuit versus time.
- A signal shape versus time is typically referred to as Waveform.
- Example:

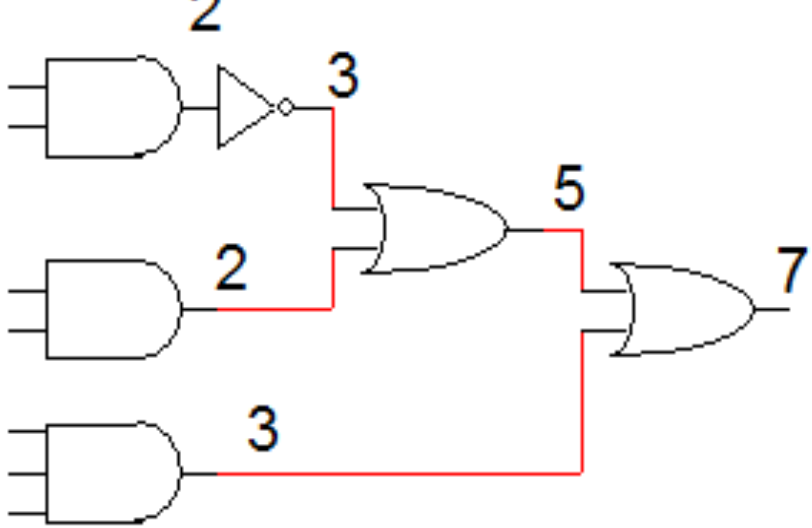


Computing Longest Delay

- Each gate has a given propagation delay
- We start at the inputs and compute the delay at the output of each gate as follows:
  - The delay at the output of a gate = gate propagation delay + maximum delay at its inputs
- Maximum propagation delay from any input to any output is called the Critical Path
- The critical path determines the minimum clock period (T) and the maximum clock frequency (f)
- Clock frequency (f) = 1 / T

Computing Longest Delay

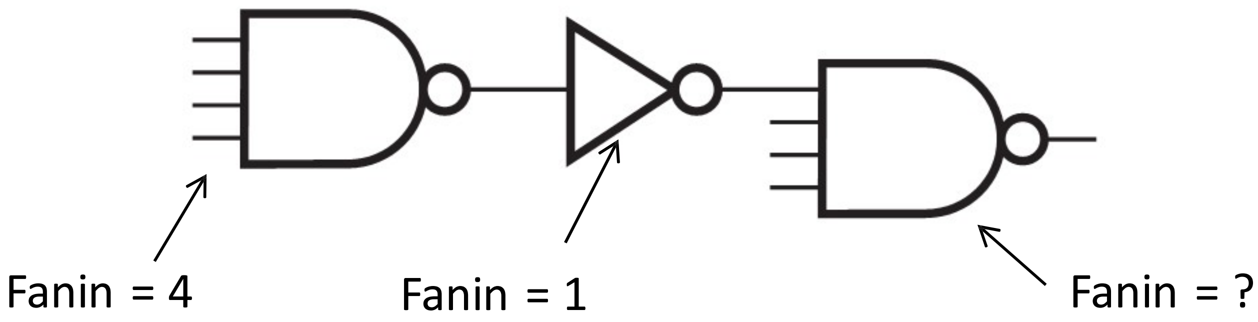
- Example: Assume that delay of each gate is related to number of its inputs i.e. delay of 1 input gate is 1 ns, delay of 2-input gate is 2 ns. Compute longest propagation delay and maximum frequency.



- Longest propagation delay = 7 ns
- Maximum frequency = 1 / 7ns= 143 MHZ

Fanin

- Fan in of a gate is the number of inputs to the gate
  - A 3-input OR gate has a fanin = 3
- There is a limitation on the fanin
- Larger fanin generally implies slower gates (higher propagation delays)



Fanout

- Fan out of a gate is the number of gates that it can drive
  - The driven gate is called a load
- Fan out is limited due to
  - Current in TTL
  - Propagation delays in CMOS