ECE 3401

Lecture 2

Digital Design Fundamentals

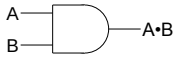
Digital Design Fundamentals

- Combinational circuit basics
  - Basic operators
  - Boolean algebra
  - Complex gates
  - Logic minimization
- Sequential circuit basics
  - Latches and flip-flops
  - Finite state machines

Basic Operators

- AND

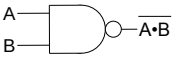
$f(A,B) = A \cdot B = A \cap B$



A	B	A•B
0	0	0
0	1	0
1	0	0
1	1	1

- NAND

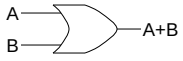
$f(A,B) = \overline{A \cdot B} = \overline{A \cap B}$



A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

- OR

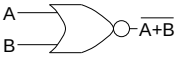
$f(A,B) = A + B = A \cup B$



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

- NOR

$f(A,B) = \overline{A + B} = \overline{A \cup B}$



A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Algebra

- Unity operators

$$A + 0 = A$$
$$A \cdot 1 = A$$

- Complement

$$A + \overline{A} = 1$$
$$A \cdot \overline{A} = 0$$

- Commutativity

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

- Associativity

$$A + (B + C) = (A + B) + C$$
$$A \cdot (BC) = (AB) \cdot C$$

- Distributed Law

$$A \cdot (B + C) = AB + AC$$
$$A + BC = (A + B) \cdot (A + C)$$

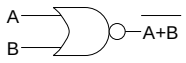
Boolean Algebra

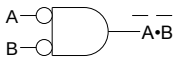
- Duality $f(A,B,1,0,\cdot,+) = \overline{f(\overline{A},\overline{B},0,1,+,\cdot)}$

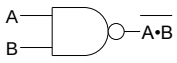
$$A + A = A \qquad A \cdot A = A$$
$$1 + A = 1 \qquad 0 \cdot A = 0$$
$$A + AB = A \qquad A \cdot (A + B) = A$$
$$A + \overline{A}B = A + B \qquad A \cdot (\overline{A} + B) = A \cdot B$$

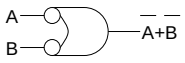
  - DeMorgan's Theorem

$$\overline{A + B} = \overline{A}\,\overline{B}$$
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$





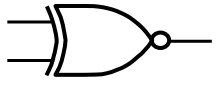




Complex Gates

- Exclusive OR (XOR)/Exclusive NOR (XNOR)

$X \oplus Y = X\,\overline{Y} + \overline{X}\,Y$ $\qquad$ $\overline{X \oplus Y} = X\,Y + \overline{X}\,\overline{Y}$

- Uses for the XOR and XNOR gate include:
  - Adders/subtractors/multipliers
  - Counters/incrementers/decrementers
  - Parity generators/checkers

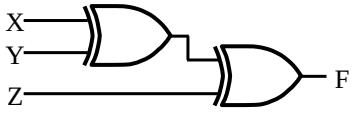
X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

- The XOR function may be implemented
  - directly as an electronic circuit (truly a gate)
  - interconnecting other gate types (used as a convenient representation)

Odd/Even Function

- The XOR function of ≥ 3 variables is called an *odd function* or *modulo 2 sum* (*Mod 2 sum*)

$$X \oplus Y \oplus Z = \overline{X}\,\overline{Y}\,Z + \overline{X}\,Y\,\overline{Z} + X\,\overline{Y}\,\overline{Z} + X\,Y\,Z$$



$X \oplus 0 = X$ $\quad X \oplus 1 = \overline{X}$

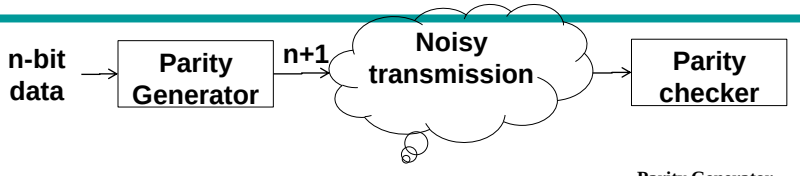
$X \oplus X = 0$ $\quad X \oplus \overline{X} = 1$

$X \oplus Y = Y \oplus X$

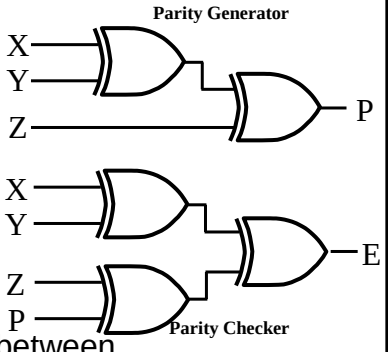
$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$

- The XNOR function of >3 variables is the *even function*

Parity Generators and Checkers

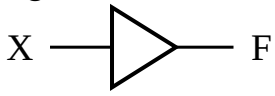


- Example: n = 3
  - Generate a parity code word of length 4 with odd parity generator
  - Check the parity code word of length 4 with odd parity checker:
  - P=X⊕Y⊕Z, E=X⊕Y⊕Z⊕P
  - If Y changes during transmission (between generator and checker), then E = 1 indicates an error.

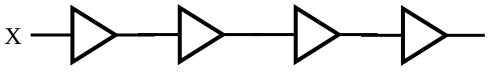


Buffer

- A buffer is a gate with the function F = X:



- In terms of Boolean function, a buffer is the same as a connection!
- So why use it?
  - A buffer is an electronic amplifier used to improve circuit voltage levels and increase the speed of circuit operation.



Hi-Impedance Outputs

- Logic gates introduced thus far
  - have 1 and 0 output values
  - cannot have their outputs connected together
  - transmit signals on connections in only one direction
- Three-state logic adds a third logic value: Hi-Impedance (Hi-Z)
- The presence of a Hi-Z state makes a gate output as described above behave quite differently:
  - "1 and 0" → "1, 0, and Hi-Z"
  - "cannot" → "can"
  - "only one" → "two"

Hi-Impedance Outputs (Contd.)

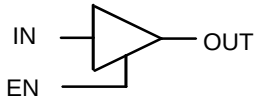
- The Hi-Z value behaves as an open circuit
  - looking back into the circuit, the output appears to be disconnected
- Hi-Z may appear on the output of any gate, but we restrict to gates:
  - a 3-state buffer
  - a transmission gate

  each of which has one data input and one control input

The 3-State Buffer

- For EN = 0, the OUT is Hi-Z regardless of the value on IN
- For EN = 1, the OUT follows the input value

Symbol

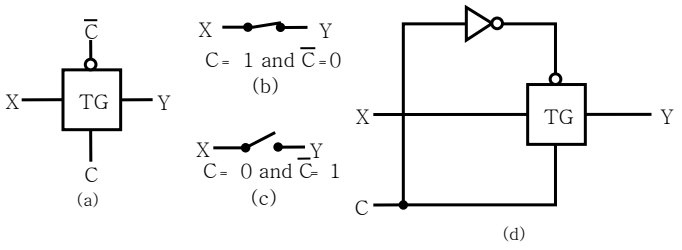


Truth Table

EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

Transmission Gates

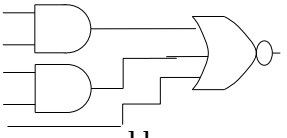
- The transmission gate is an electronic switch for connecting and disconnecting two points in a circuit
  - C = 1, Y = X (X = 0 or 1)
  - C = 0, Y = Hi-Z



- Since X and Y as input and output are interchangeable, and signals can pass in both directions

More Complex Gates

- SOP or POS structures with and without an output inverter.
  - SOP: $F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + ABC$
  - POS: $F = (A + B + C)\cdot(A + \overline{B} + \overline{C})\cdot(\overline{A} + \overline{B} + \overline{C})$
- Naming:
  - A – AND, O – OR, I - Inverter
  - Numbers of inputs on first-level "gates" or directly to second-level "gates" – AOI 221



- These gate types are used because:
  - The number of transistors needed is fewer than required by connecting together primitive gates
  - Potentially, the circuit delay is smaller, increasing the circuit operating speed

Logic Minimization

- Minimizing SOP representation to MSP
- Using Karnaugh Map

$$F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C}$$

C \ AB	00	01	11	10
0	0	1	1	1
1	1	0	0	1

$$F = B\overline{C} + A\overline{B} + \overline{B}C$$

K-Map Example

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

(a) Location of minterms

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	X	1
10	1	1	X	1

Four corner terms combine to give $B'D'$

$A'BD$

$F = \Sigma m\,(0, 2, 3, 5, 6, 7, 8, 10, 11) + \Sigma d\,(14, 15)$
$= C + B'D' + A'BD$

(b) Looping terms

Logic Minimization

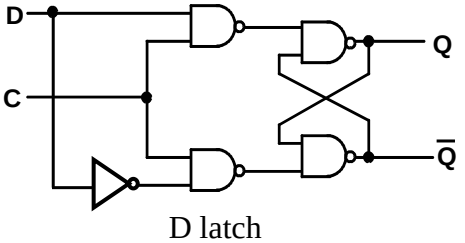
- Minimizing POS representation to MPS
- Using Karnaugh Map

$$F = (A + B + C) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C})$$

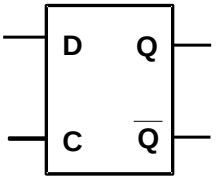
C \ AB	00	01	11	10
0	0	1	1	1
1	1	0	0	1

$$F = (A + B + C) \cdot (\overline{B} + \overline{C})$$

D-Latch



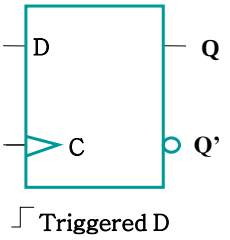
D latch



"Latch" is an important notion: its input is controlled by a gate input C, when C goes from high to low, the state of the device holds.

D flip-flop

- Rising edge triggered D FF
- Timing parameters:
  - Setup time t_{su}: input must be stable before the clock edge
  - Hold time t_h: input must stay stable after the clock edge
  - Clock to Q $t_{c\text{-}q}$: maximum time for output to be stable after the clock edge



Triggered D