ME 3109: Measurement & Instrumentation

Azmir Hasan Mojumder
Lecturer

Department of Mechanical and Production Engineering
Ahsanullah University of Science and Technology

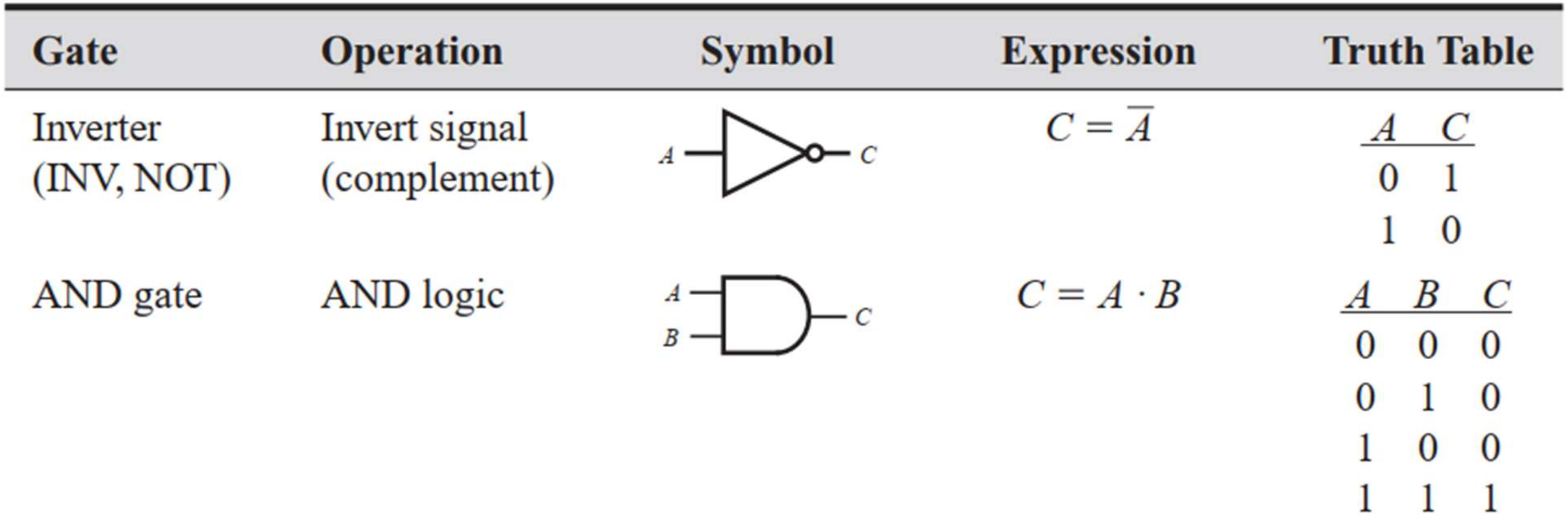
ME 3109: Measurement & Instrumentation

Topic_02: Digital Electronics[2]	
1	Combinational Logic Device
2	Boolean Algebra
3	Combinational Logic Design
4	Sequential Logic Device

[2] Histand, ***"Introduction to Mechatronics & Measurement Systems"***, 4th Edition

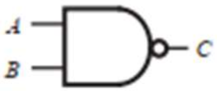
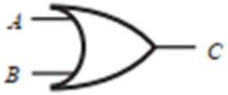
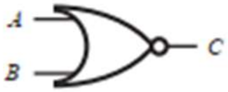
1. Combinational Logic Devices

- Combinational logic devices generate an output based on the input values, independent of the input timing.
- They convert digital inputs into binary outputs based on the rules of mathematical logic. They are also called **gates.**



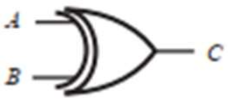
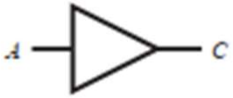
Gate	Operation	Symbol	Expression	Truth Table
Inverter (INV, NOT)	Invert signal (complement)	A C	$C = \overline{A}$	A C 0 1 1 0
AND gate	AND logic	A B C	$C = A \cdot B$	A B C 0 0 0 0 1 0 1 0 0 1 1 1

1. Combinational Logic Devices

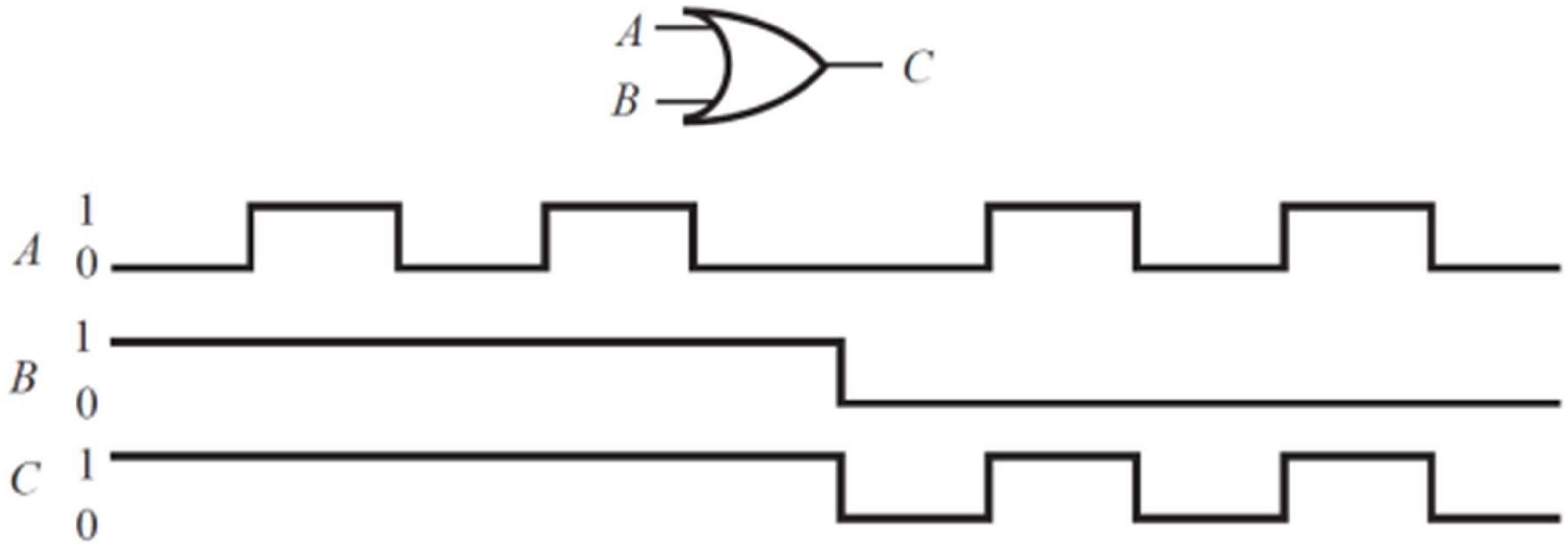
Gate	Operation	Symbol	Expression	Truth Table
NAND gate	Inverted AND logic		$C = \overline{A \cdot B}$	A B C 0 0 1 0 1 1 1 0 1 1 1 0
OR gate	OR logic		$C = A + B$	A B C 0 0 0 0 1 1 1 0 1 1 1 1
NOR gate	Inverted OR logic		$C = \overline{A + B}$	A B C 0 0 1 0 1 0 1 0 0 1 1 0

Histand, “*Introduction to Mechatronics & Measurement Systems*”

1. Combinational Logic Devices

Gate	Operation	Symbol	Expression	Truth Table
XOR gate	Exclusive OR logic		$C = A \oplus B$ $= A \cdot \overline{B} + \overline{A} \cdot B$	A B C 0 0 0 0 1 1 1 0 1 1 1 0
Buffer	Increase output signal current		$C = A$	A C 0 0 1 1

1. Timing Diagram



2. Boolean Algebra

OR → A OR 0 → $\boxed{A + 0 \rightarrow A}$

$A + 1 \rightarrow 1 \rightarrow \boxed{?}$

$A + A \rightarrow A \rightarrow \boxed{?}$

$\boxed{A + \overline{A} \rightarrow 1}$

Anything 'OR'ed with zero is equal to itself

Anything 'ORed with it's own compliment equals 1

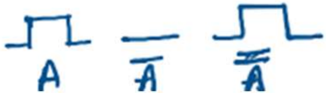
AND $A \cdot 0 = 0$

$A \cdot 1 = A$

$A \cdot A = A$

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

Also, $\overline{\overline{A}} = A$ →



2. Boolean Algebra

- The order of ORing and ANDing does not matter (Commutative law)

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

- The grouping of several variables ORed or ANDed together does not matter (Associative law)

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

- Distributive law $A(B+C) = AB + BC$

$(A+B)(C+D) = AC + AD + BC + BD$

2. Boolean Algebra

Useful Identities

$$A + (A \cdot B) = A$$

$$A \cdot (A + B) = A$$

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

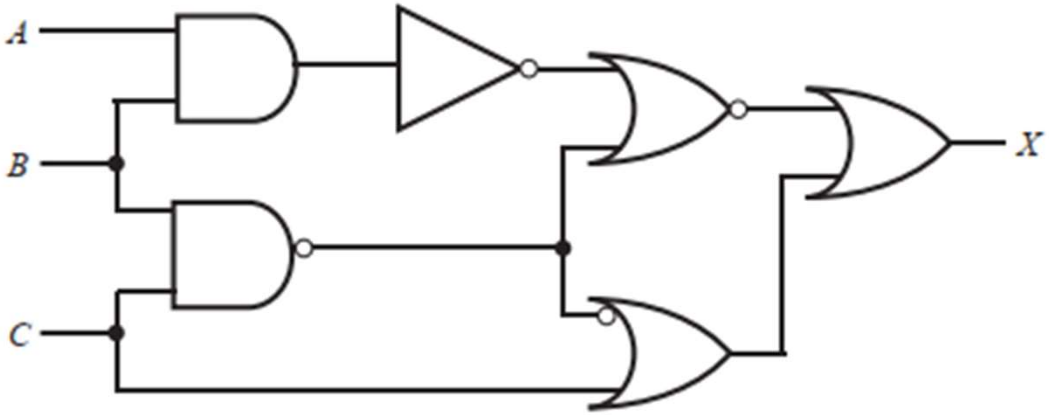
$$\overline{A} + AB = \overline{A} + B$$

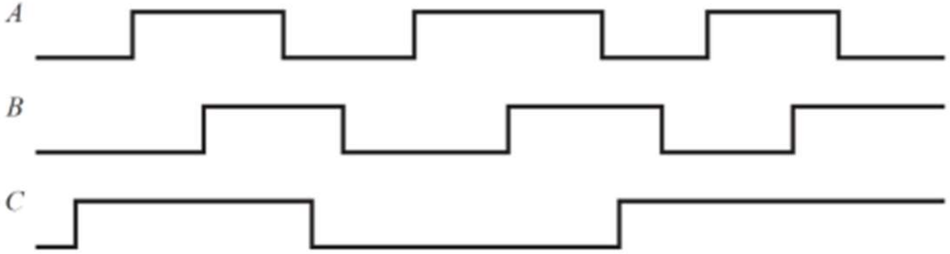
$$\left.\begin{aligned} \overline{A.B} &= \overline{A} + \overline{B} \\ \overline{A+B} &= \overline{A} \cdot \overline{B} \end{aligned}\right\} \text{De Morgan's Theorem}$$

→ Prove by using Truth table

Problems

1. Determine a simplified Boolean expression for X in the combinational logic circuit that follows. Also, complete the timing diagram.





Practice Problems

2. Draw the schematics of logic circuits that produce the following logic expressions:

   a. $\overline{A} + \overline{B}$

   b. $\overline{A} \cdot \overline{B}$

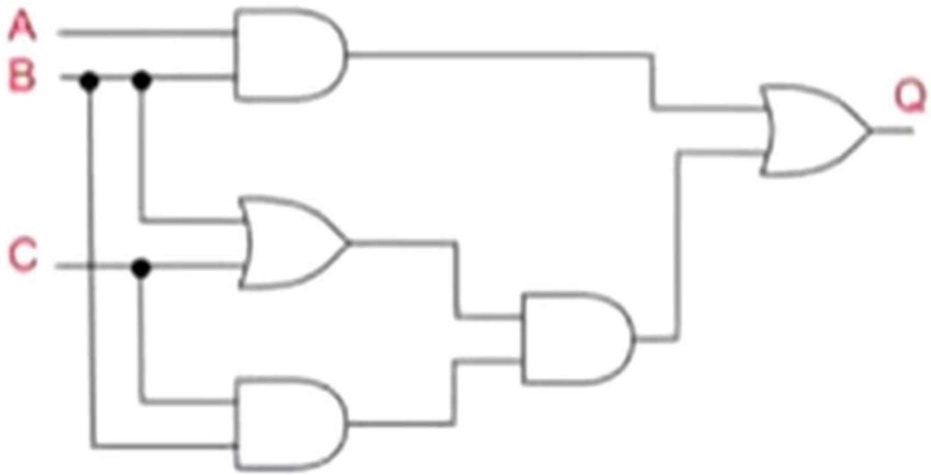
3. Create an inverter using a single NAND gate. Draw the schematic.

4. Show the universality of NAND and NOR gates

5. Use a truth table to prove the validity of De Morgan's laws for two signals (A and B).

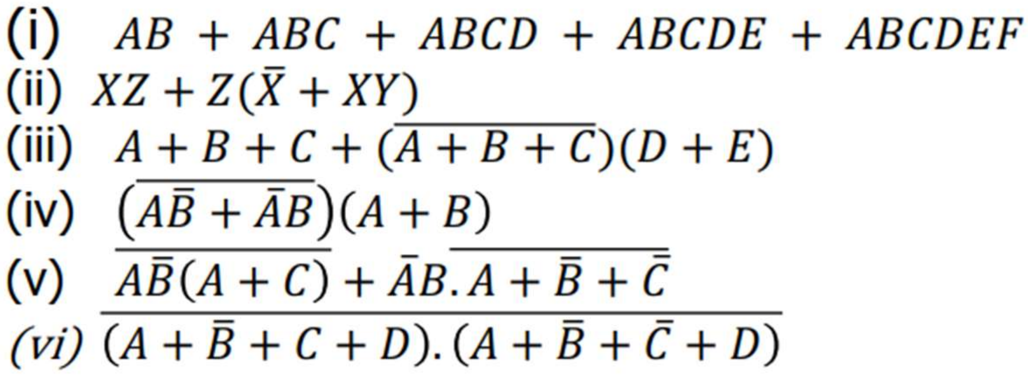
Practice Problems

6. Draw the more simplified logic circuit of the following.



Practice Problems

7. Simplify the following Boolean Expressions:



(i) $AB + ABC + ABCD + ABCDE + ABCDEF$

(ii) $XZ + Z(\bar{X} + XY)$

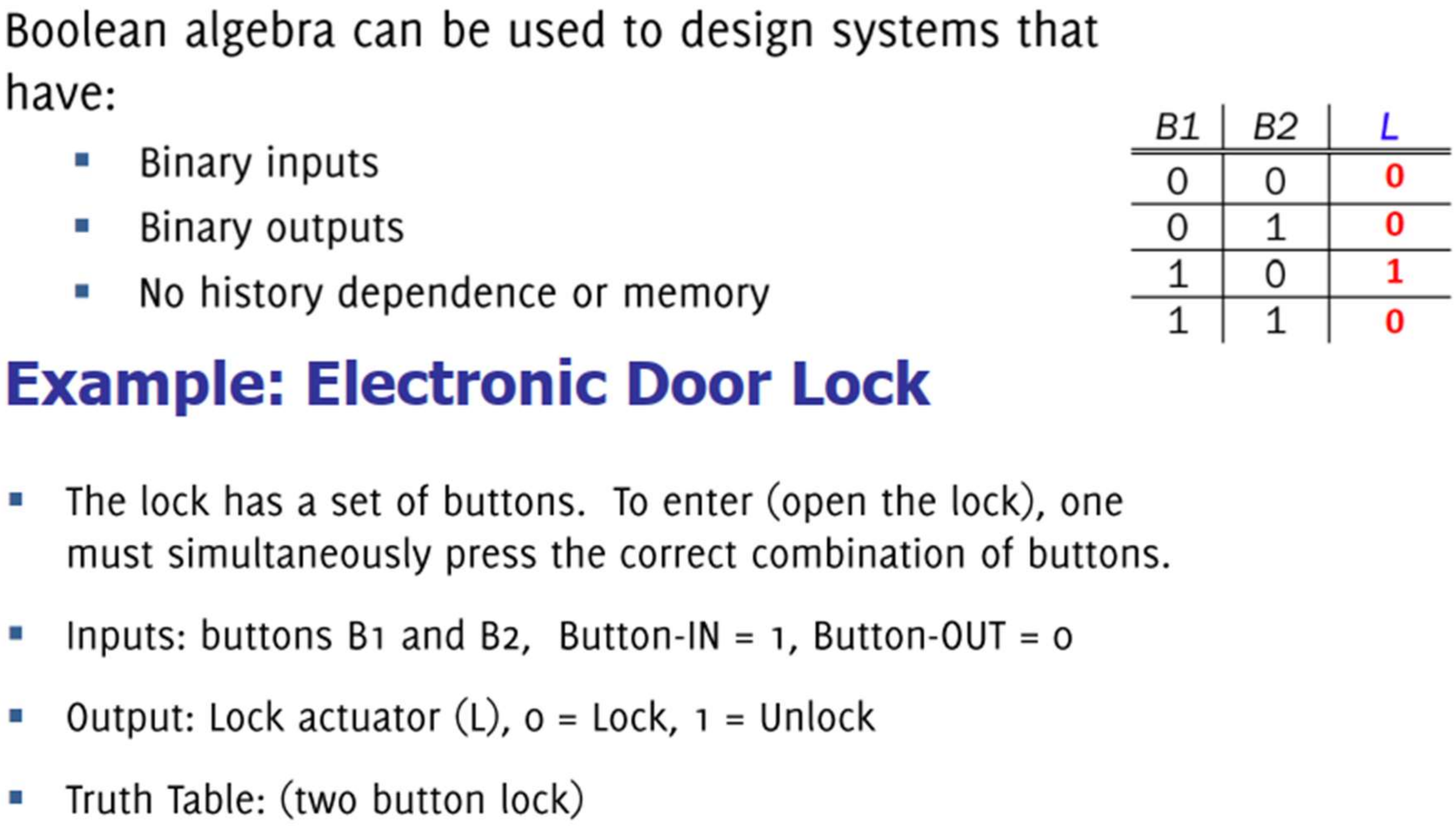
(iii) $A + B + C + (\overline{A + B + C})(D + E)$

(iv) $(\overline{A\bar{B} + \bar{A}B})(A + B)$

(v) $\overline{A\bar{B}(A + C)} + \bar{A}B.\overline{A + \bar{B} + \bar{C}}$

(vi) $\overline{(A + \bar{B} + C + D).(A + \bar{B} + \bar{C} + D)}$

3. Combinational Logic Design



Boolean algebra can be used to design systems that have:

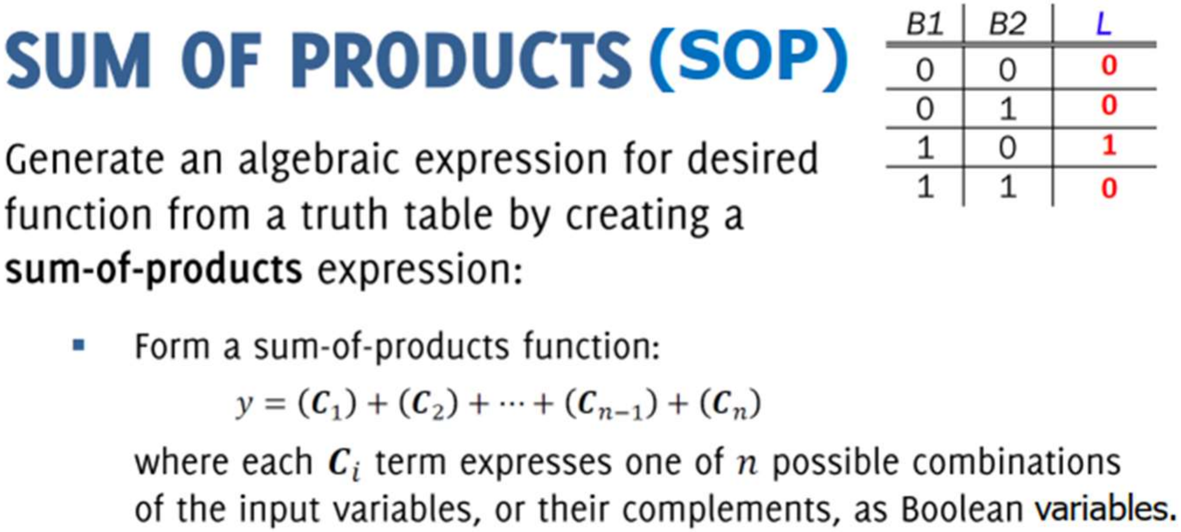
- Binary inputs
- Binary outputs
- No history dependence or memory

B1	*B2*	*L*
0	0	**0**
0	1	**0**
1	0	**1**
1	1	**0**

Example: Electronic Door Lock

- The lock has a set of buttons. To enter (open the lock), one must simultaneously press the correct combination of buttons.
- Inputs: buttons B1 and B2, Button-IN = 1, Button-OUT = 0
- Output: Lock actuator (L), 0 = Lock, 1 = Unlock
- Truth Table: (two button lock)

3. Combinational Logic Design



SUM OF PRODUCTS (SOP)

$B1$	$B2$	L
0	0	0
0	1	0
1	0	1
1	1	0

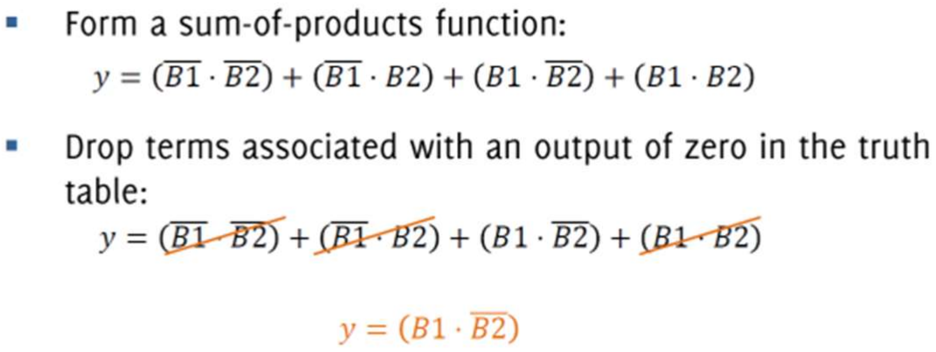
Generate an algebraic expression for desired function from a truth table by creating a **sum-of-products** expression:

- Form a sum-of-products function:

  $$y = (\boldsymbol{C}_1) + (\boldsymbol{C}_2) + \cdots + (\boldsymbol{C}_{n-1}) + (\boldsymbol{C}_n)$$

  where each $\boldsymbol{C}_i$ term expresses one of n possible combinations of the input variables, or their complements, as Boolean variables.

- Treat ones in the truth table as uncomplemented variables, and zeros as complemented variables.
- Drop terms corresponding to an output of zero in the truth table



- Form a sum-of-products function:

  $$y = (\overline{B1} \cdot \overline{B2}) + (\overline{B1} \cdot B2) + (B1 \cdot \overline{B2}) + (B1 \cdot B2)$$

- Drop terms associated with an output of zero in the truth table:

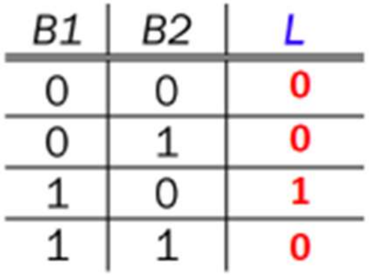
  $$y = \cancel{(\overline{B1} \cdot \overline{B2})} + \cancel{(\overline{B1} \cdot B2)} + (B1 \cdot \overline{B2}) + \cancel{(B1 \cdot B2)}$$

  $$y = (B1 \cdot \overline{B2})$$

3. Combinational Logic Design

PRODUCT OF SUMS (POS)

- Treat zeros in the truth table as uncomplemented variables, and ones as complemented variables.
- Drop terms corresponding to an output of one in the truth table



B1	*B2*	*L*
0	0	0
0	1	0
1	0	1
1	1	0

$$L = (B1+B2).\,(B1+B2').\,(B1'+B2).\,(B1'+B2')$$

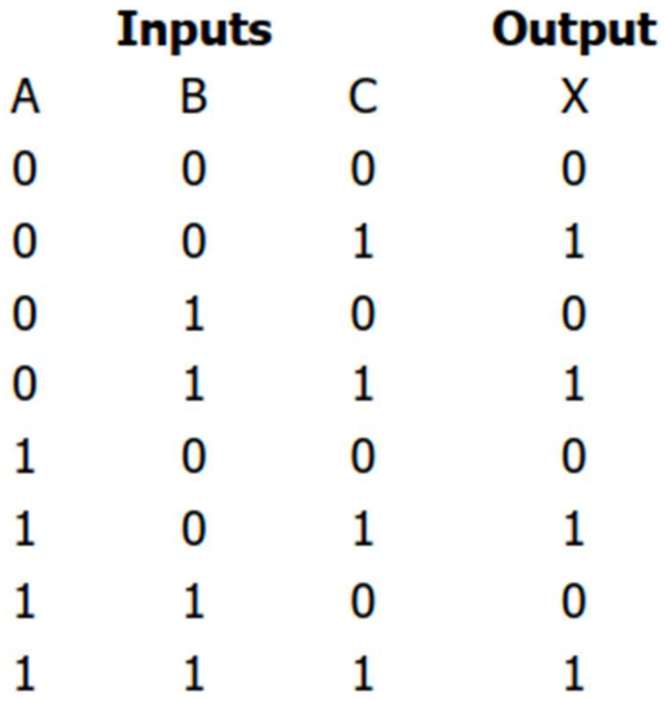
$$= (B1+B2).\,(B1+B2').\,(B1'+B2')$$

After Simplifying

$$\mathrm{L} = B1.\overline{B2}$$

Practice Problems

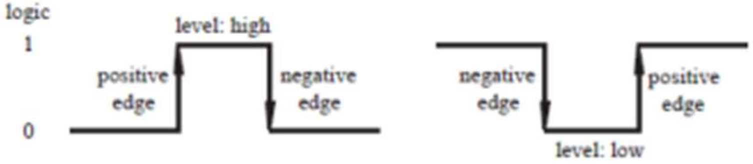
8. Using SOP and/or POS, find the simplified Boolean expression for the following control circuit:



Inputs			Output
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

4. Sequential Logic Devices

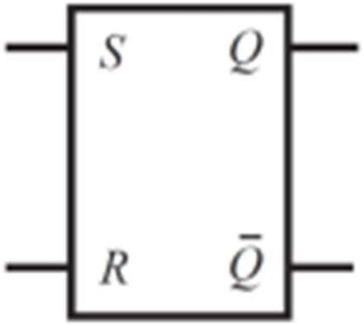
- The timing or sequencing of the input signals is important. **Example:** flip-flops, counters, monostables, latches, microprocessors etc.
- Sequential logic devices usually respond to inputs when a separate trigger signal transitions from one level to another. The trigger signal is usually referred to as the clock (CK) signal
- The clock signal can be a periodic square wave or an aperiodic collection of pulses



4. Flip-Flops

- Flip-Flops enable storing and switching between the two binary states
- A flip-flop is a sequential logic device
- It has two and only two possible stable output states: 1 (high) and 0 (low). Bistable device
- It has the capability to remain in a particular output state (i.e., storing a bit) until input signals cause it to change state
- This is the basis of all semiconductor information storage and processing in digital computers. Flip-flops perform many of the basic functions critical to the operation of almost all digital devices

4. RS Flip-Flops



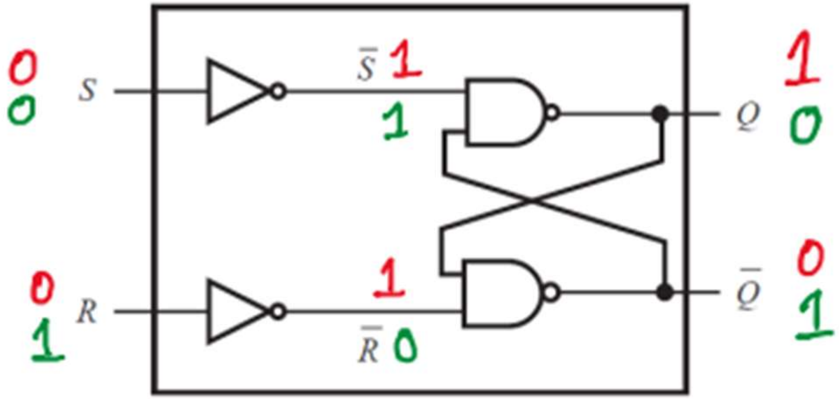
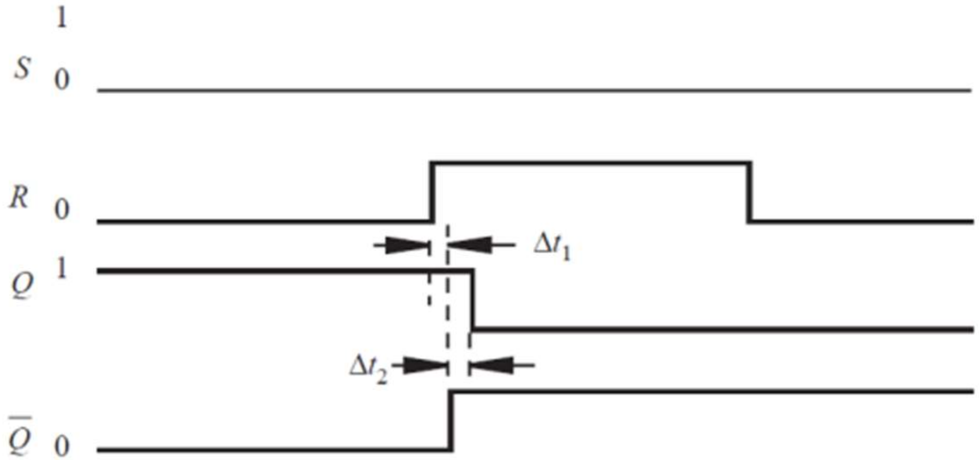
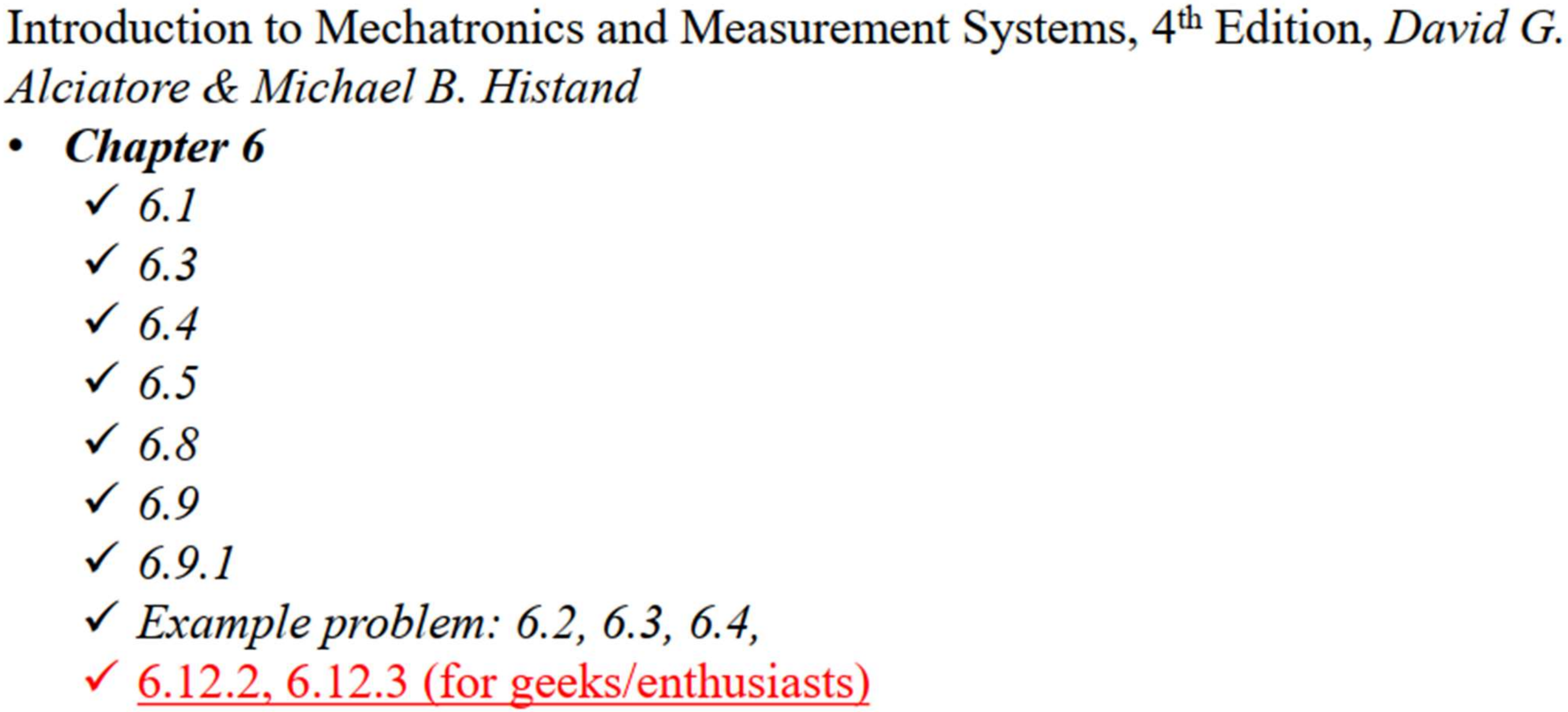


Table 6.4 Truth table for the RS flip-flop

Inputs		Outputs	
S	***R***	***Q***	$\overline{Q}$
0	0	Q_0	$\overline{Q}_0$
1	0	1	0
0	1	0	1
1	1	NA	



Reading



Introduction to Mechatronics and Measurement Systems, 4th Edition, *David G. Alciatore & Michael B. Histand*

- ***Chapter 6***
  - ✓ *6.1*
  - ✓ *6.3*
  - ✓ *6.4*
  - ✓ *6.5*
  - ✓ *6.8*
  - ✓ *6.9*
  - ✓ *6.9.1*
  - ✓ *Example problem: 6.2, 6.3, 6.4,*
  - ✓ 6.12.2, 6.12.3 (for geeks/enthusiasts)

Thank you