Real Time Systems Design

Lecture (9): Introduction to digital systems

Dr. Rasha Thabit

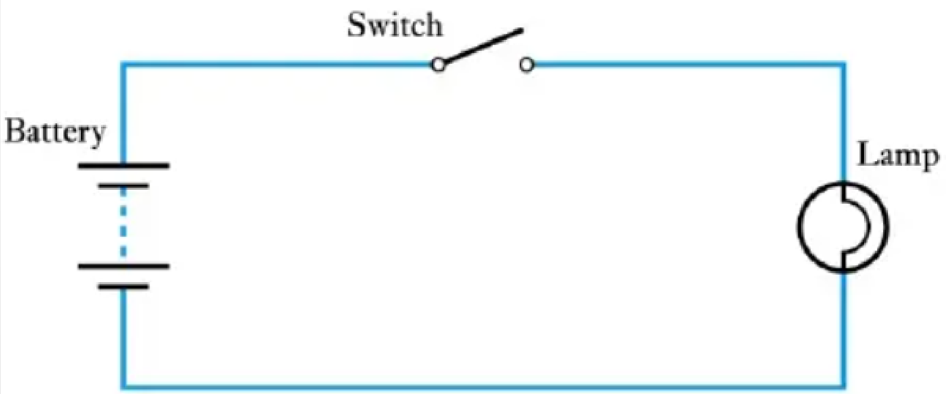
Department of Computer Techniques Engineering

Introduction

- Digital systems are concerned with digital signals
- Digital signals can take many forms
- Here we will concentrate on **binary signals** since these are the most common form of digital signals
  - can be used individually
    - perhaps to represent a single binary quantity or the state of a single switch
  - can be used in combination
    - to represent more complex quantities

Binary quantities and variables

- A **binary quantity** is one that can take only 2 states



A simple binary arrangement

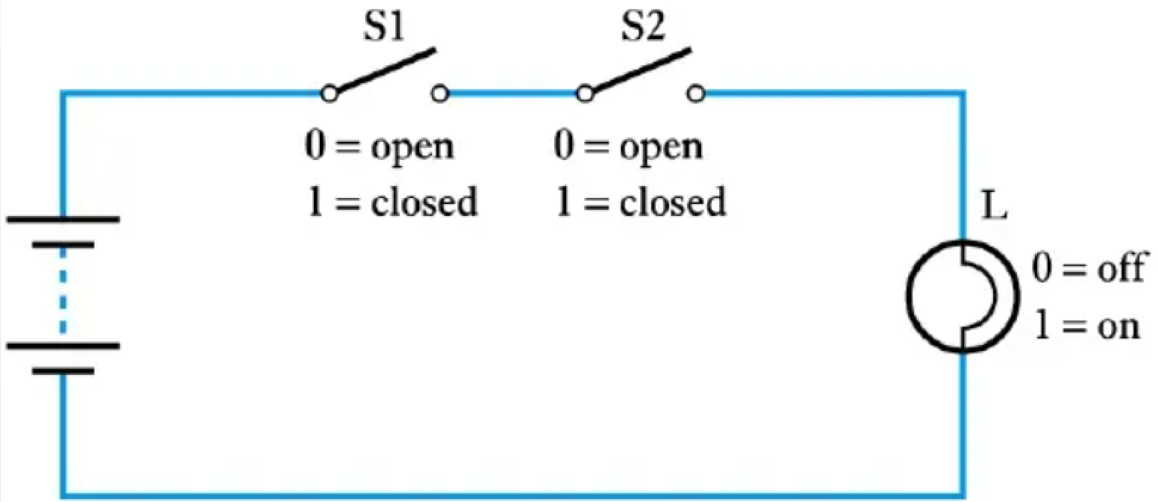
S	L
OPEN	OFF
CLOSED	ON

S	L
0	0
1	1

A truth table

Binary quantities and variables

- A binary arrangement with two switches in series



(a) Circuit

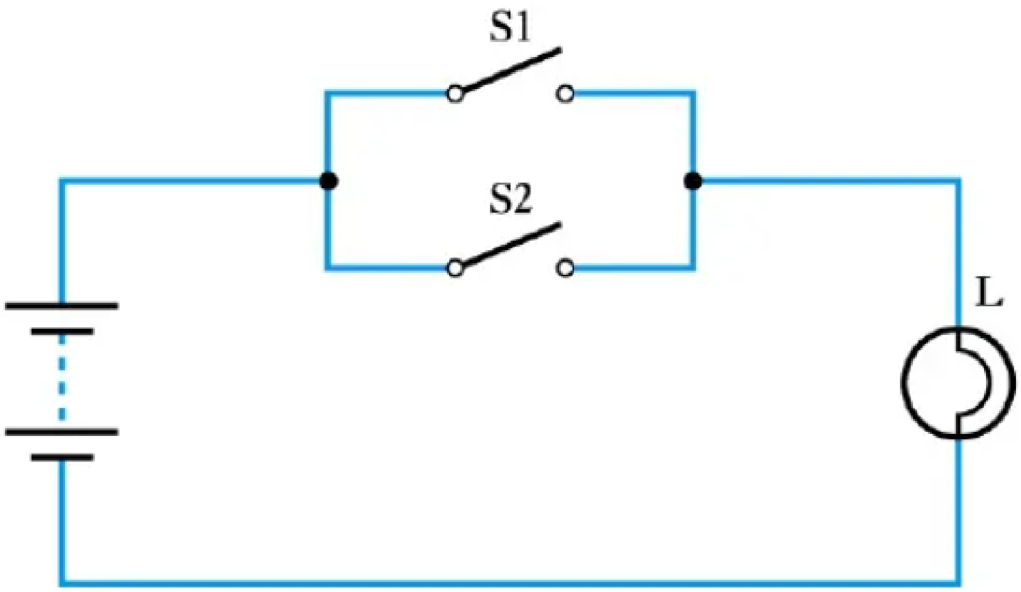
S1	S2	L
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth table

$L = S1 \text{ AND } S2$

Binary quantities and variables

- A binary arrangement with two switches in parallel



(a) Circuit

S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	1

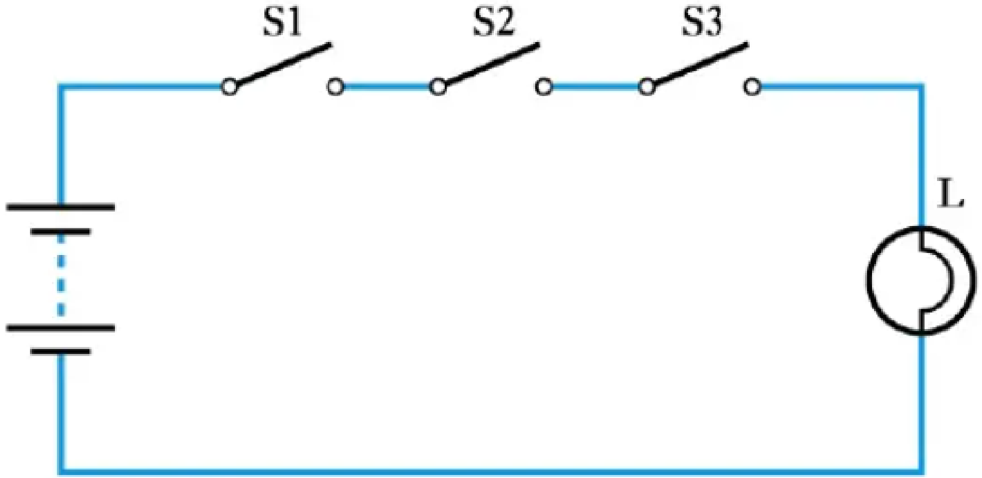
(b) Truth table

$L = S1 \text{ OR } S2$

Dr. Rasha Thabit

Binary quantities and variables

- Three switches in series

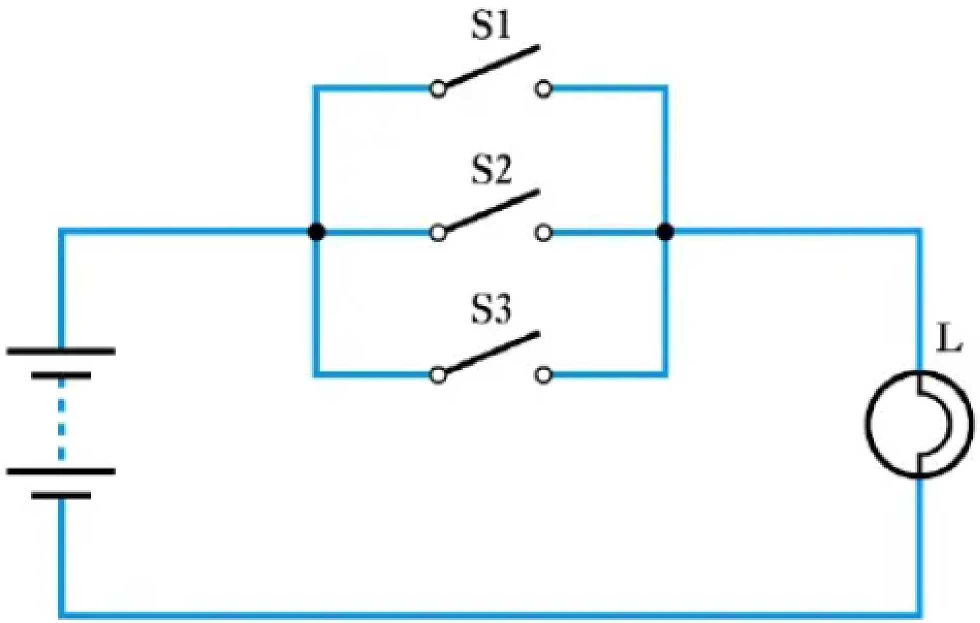


S1	S2	S3	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$L = S1$ AND $S2$ AND $S3$

Binary quantities and variables

- Three switches in parallel

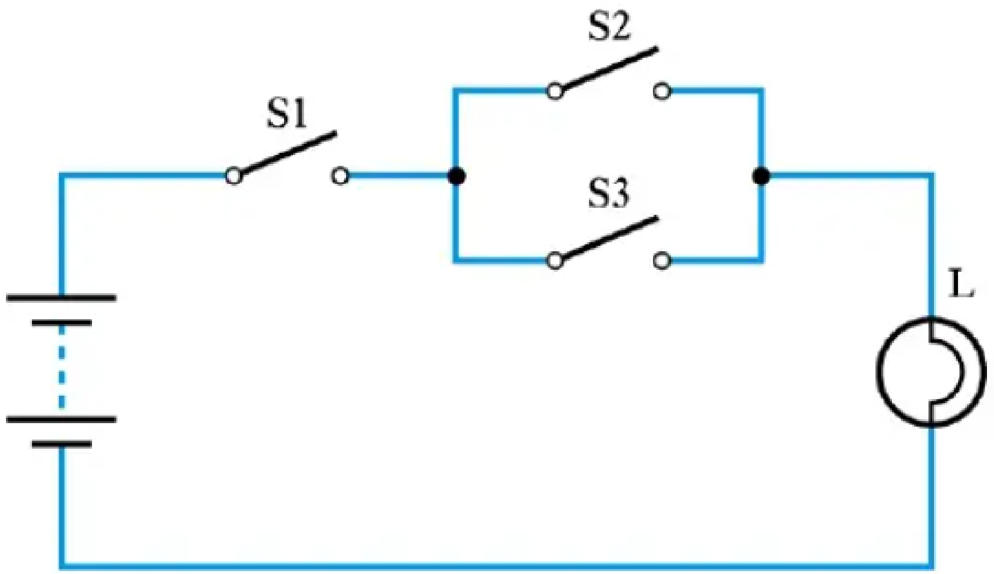


S1	S2	S3	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

L = *S1* OR *S2* OR *S3*

Binary quantities and variables

- A series/parallel arrangement

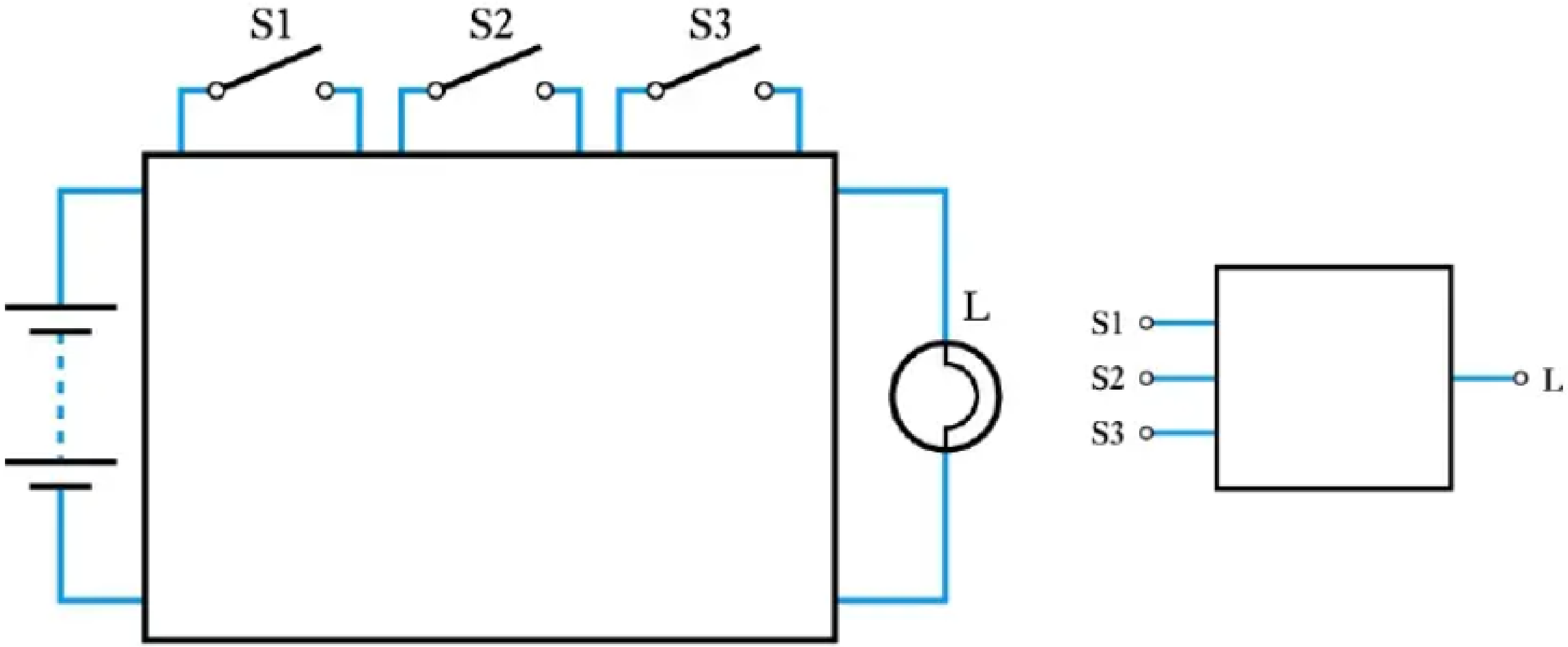


S1	S2	S3	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

L = *S1* AND (*S2* OR *S3*)

Binary quantities and variables

- Representing an unknown network



Logic gates

- The building blocks used to create digital circuits are **logic gates**
- There are three elementary logic gates and a range of other simple gates
- Each gate has its own **logic symbol** which allows complex functions to be represented by a logic diagram
- The function of each gate can be represented by a **truth table** or using **Boolean notation**

▪ The AND gate

(a) Circuit symbol

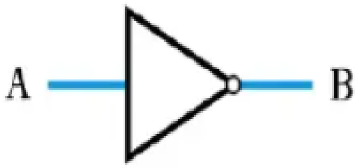
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth table

$C = A \cdot B$

(c) Boolean expression

▪ The NOT gate (or inverter)



(a) Circuit symbol

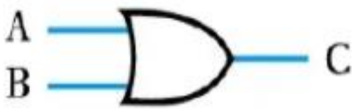
A	B
0	1
1	0

(b) Truth table

$B = \overline{A}$

(c) Boolean expression

▪ The OR gate



(a) Circuit symbol

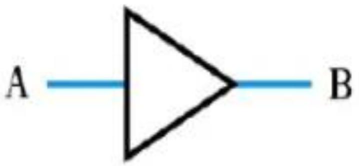
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

(b) Truth table

$C = A + B$

(c) Boolean expression

▪ A logic buffer gate



(a) Circuit symbol

A	B
0	0
1	1

(b) Truth table

$B = A$

(c) Boolean expression

▪ The NAND gate

(a) Circuit symbol

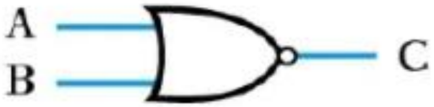
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

(b) Truth table

$$C = \overline{A \cdot B}$$

(c) Boolean expression

▪ The NOR gate



(a) Circuit symbol

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

(b) Truth table

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

(c) Boolean expression

Summary

- ✓ Introduction to digital systems
- ✓ Different binary circuits arrangement
- ✓ Different logic gates