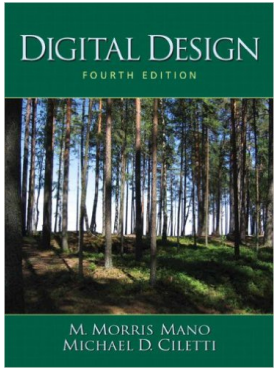
# CS221: Digital Design

Waleed A. Yousef, Ph.D.,

Human Computer Interaction Lab.,
Computer Science Department,
Faculty of Computers and Information,
Helwan University,
Egypt.

August 14, 2018

**Lectures follow (and some figures are adapted from):**




Mano, M. M., Ciletti, M. D., 2007. Digital design, 4th Edition. Prentice-Hall, Upper Saddle River, NJ

**We will proceed as follows:**

- Fast introduction (few sections from Chapter 1).
- Detailed study of Chapters 2-7; very few sections will be skipped. At the end of each chapter Verilog code for some circuits will be explained.
- If time permits, Chapter 8 will be covered in full or in parts

# Course Objectives

This course combines three approaches to teach students Digital Design, which is the fundamental prerequisite to understand computer design and architecture:

1. Theoretical aspects of the subject will be covered in lectures, along with exercises in sections. Students, by the end of the course, should be able to design, analyze, and implement combinational and synchronous digital circuits.

2. A second objective is to teach students the digital design using a Hardware Descriptive Language (HDL). Students by the end of the semester will be able to analyze logic circuits with Verilog (one of the available HDLs).

3. A third objective is to develop the practical sense of the students through lab. experiments. Students will be able to implement logic circuits using breadboards and ICs.

# Contents

# Chapter 1

# Introduction to Digital Systems

## 1.9 From Binary Logic (Mathematics) to Logic Gates (Circuits)

From what we have studied in Discrete Mathematics, we have the following basic three logic functions:

*Truth Tables of Logical Operations*

**AND**

| $x$ | $y$ | $x \cdot y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

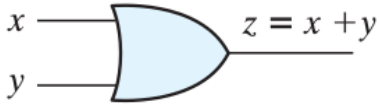
**OR**

| $x$ | $y$ | $x + y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

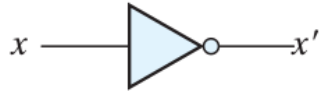
**NOT**

| $x$ | $x'$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

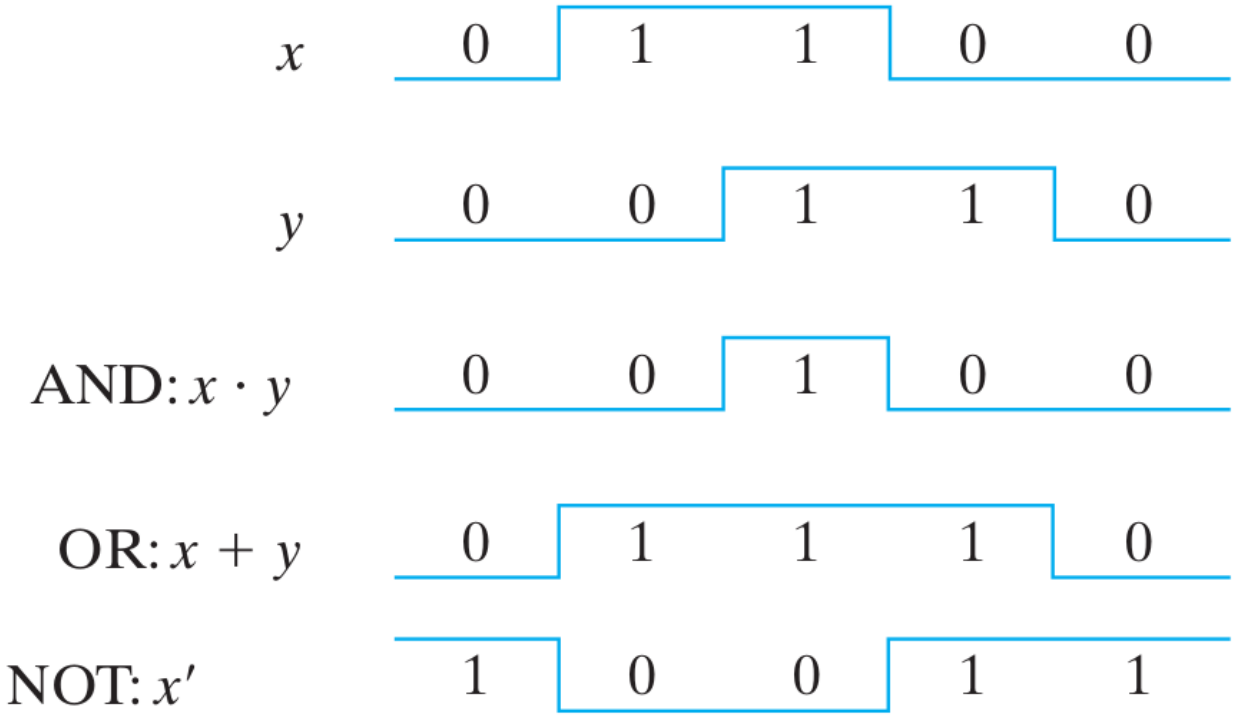
(a) Two-input AND gate

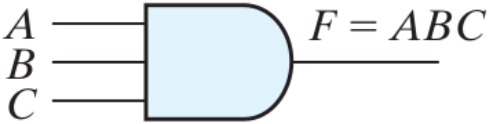


(b) Two-input OR gate

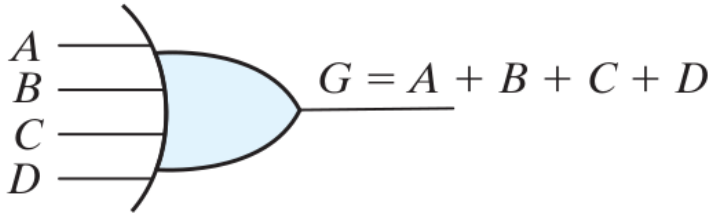


(c) NOT gate or inverter

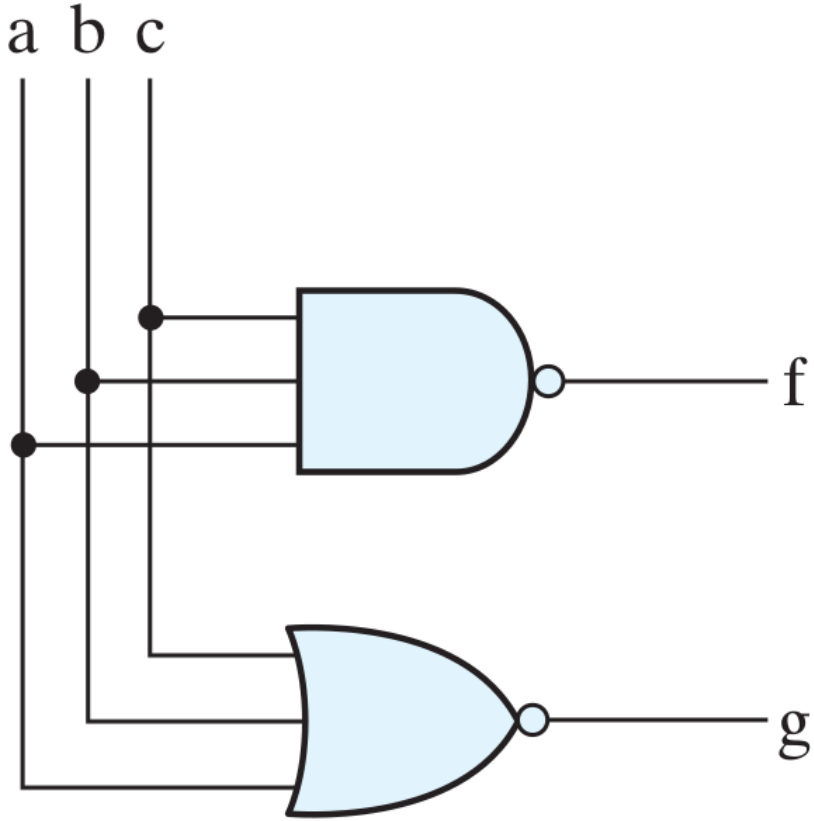

x
0 1 1 0 0
y
0 0 1 1 0
AND: x · y
0 0 1 0 0
OR: x + y
0 1 1 1 0
NOT: x′
1 0 0 1 1




(a) Three-input AND gate

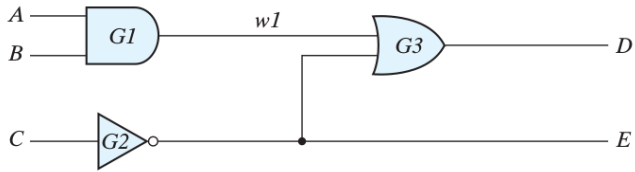



(b) Four-input OR gate


a
b
c
f
g

# Revisiting Course Objectives

## Theory (hands on paper)

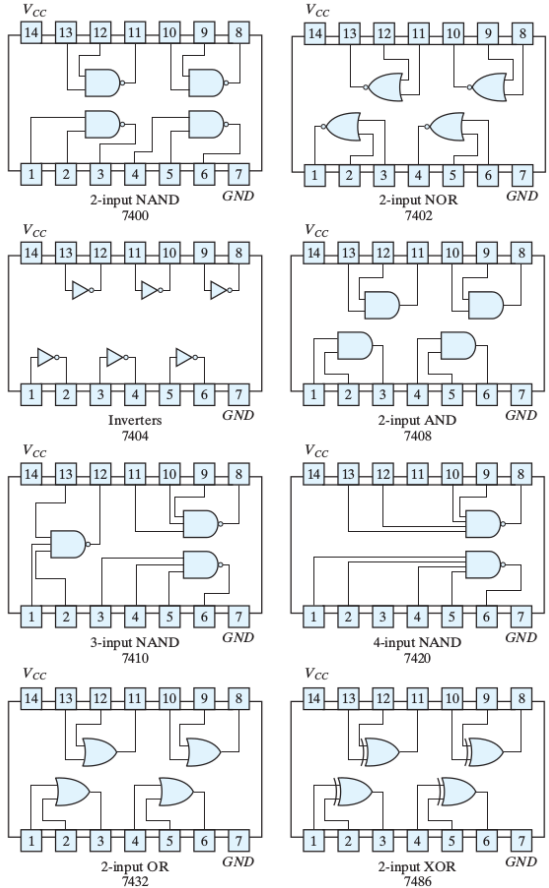



## Simulation using Verilog

```
// Verilog model: Simple_Circuit
module Simple_Circuit (A, B, C, D, E);
  output D, E;
  input  A, B, C;
  wire   w1;

  and    G1 (w1, A, B); // Optional gate instance
  not    G2 (E, C);
  or     G3 (D, w1, E);
endmodule
```

## Hardware Implementation


$V_{CC}$
14 13 12 11 10 9 8
1 2 3 4 5 6 7
GND
2-input NAND
7400
2-input NOR
7402
Inverters
7404
2-input AND
7408
3-input NAND
7410
4-input NAND
7420
2-input OR
7432
2-input XOR
7486

# Bibliography

Mano, M. M., Ciletti, M. D., 2007. Digital design, 4th Edition. Prentice-Hall, Upper Saddle River, NJ.