EC311: Logic Design

Lecture notes for Logic Design

Giacomo Cappelletto

Published: September 12, 2025
Last updated: September 12, 2025

Contents

List of Figures

* * *

Chapter 1: Introduction to Logic Design

Digital logic design is the foundation of modern computing systems, from simple embedded controllers to complex processors. This course covers the systematic approach to designing digital circuits using Boolean algebra, logic gates, and systematic design methodologies.

1.1. Design Flow Overview

Digital System Design Flow — Definition 1.1.1

The modern digital design process follows a structured approach: Analog Input ⟶ ADC ⟶ Device ⟶ Digitized Data ⟶ Processing

This flow transforms real-world analog signals into digital representations that can be processed by digital logic circuits.

1.2. System Hierarchy

Digital systems are organized in a hierarchical structure for manageable design:

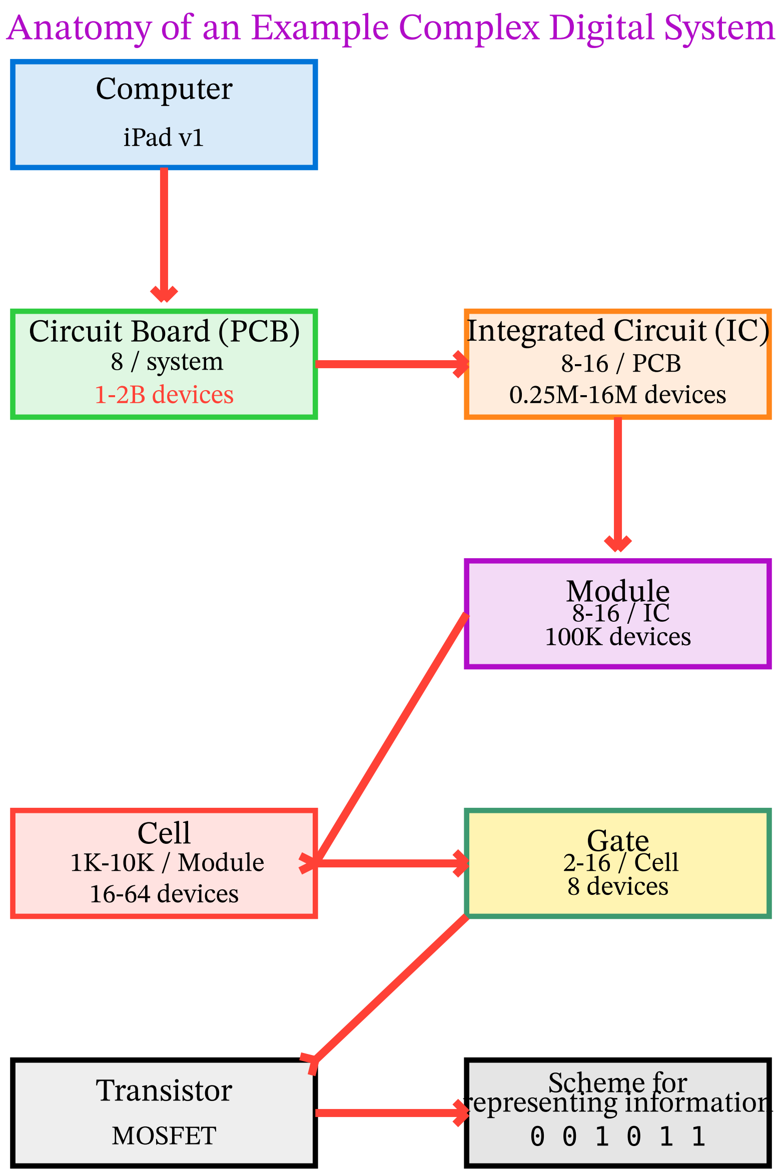


Figure 1: Complete anatomy of a complex digital system showing hierarchy from computer to transistor level

Claude Shannon — Note 1.2.1

Claude Shannon's work in the 1940s established the mathematical foundation for digital logic design, showing how Boolean algebra could be used to analyze and synthesize switching circuits.

Chapter 2: Digital Logic Fundamentals

2.1. Basic Logic Elements

Digital circuits are built from fundamental logic gates that perform Boolean operations on binary inputs.

Logic Gate Definition 2.1.1

A logic gate is a digital circuit that implements a Boolean function. It has one or more binary inputs and produces a single binary output based on the logical relationship defined by the gate type.

2.1.1. Truth Tables and Boolean Functions

For 2 input variables (X, Y), there are $2^{2^2} = 16$ possible Boolean functions:

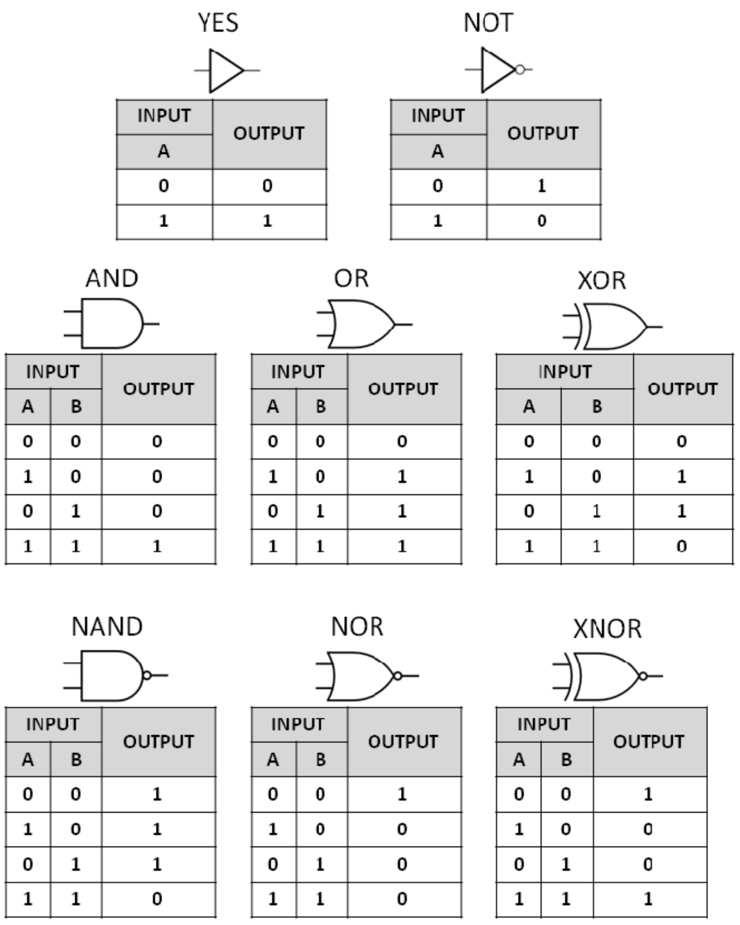
Table 1: Complete truth table showing all 16 possible Boolean functions (F_0-F_{15}) for inputs X and Y

X	Y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2: Complete identification of all 16 Boolean functions with their logical expressions

$F_0 =$	0	(Always FALSE)	$F_8 =$	$\overline{X+Y}$	(X NOR Y)
$F_1 =$	$X \cdot Y$	(X AND Y)	$F_9 =$	$\overline{X \oplus Y}$	(X XNOR Y)
$F_2 =$	$X \cdot \overline{Y}$	(X AND NOT Y)	$F_{10} =$	$\overline{Y}$	(NOT Y)
$F_3 =$	X	(Copy X)	$F_{11} =$	$X + \overline{Y}$	(X OR NOT Y)
$F_4 =$	$\overline{X} \cdot Y$	(NOT X AND Y)	$F_{12} =$	$\overline{X}$	(NOT X)
$F_5 =$	Y	(Copy Y)	$F_{13} =$	$\overline{X} + Y$	(NOT X OR Y)
$F_6 =$	$X \oplus Y$	(X XOR Y)	$F_{14} =$	$\overline{X \cdot Y}$	(X NAND Y)
$F_7 =$	$X + Y$	(X OR Y)	$F_{15} =$	1	(Always TRUE)

2.1.2. Standard Logic Gates



YES

INPUT	OUTPUT
A	
0	0
1	1

NOT

INPUT	OUTPUT
A	
0	1
1	0

AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

2.1.3. Boolean Expressions

XOR and XNOR Expressions — Example 2.1.3.1

The XOR and XNOR gates have expanded Boolean forms:

- XOR: $Z = X \oplus Y = X \cdot \overline{Y} + \overline{X} \cdot Y$
- XNOR: $Z = \overline{X \oplus Y} = X \cdot Y + \overline{X} \cdot \overline{Y}$

These expressions show that XOR outputs 1 when inputs differ, while XNOR outputs 1 when inputs are the same.

2.2. Modern Technology: MOS and CMOS

MOSFET Technology — Definition 2.2.1

Modern digital circuits primarily use MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) technology:

- NMOS: N-channel transistors that conduct when gate is HIGH
- PMOS: P-channel transistors that conduct when gate is LOW
- CMOS: Complementary MOS using both NMOS and PMOS for low power consumption

The CMOS inverter we studied earlier demonstrates how these transistors work together to create efficient digital switches with minimal power consumption except during transitions.

Chapter 3: Circuit Analysis and Abstraction

3.1. Abstraction Levels

Design Abstraction Note 3.1.1

Digital design uses multiple levels of abstraction:

1. Behavioral Description: Specification of what the circuit should do
2. Circuit Schematic: Gate-level implementation
3. Hardware Implementation: Physical realization in silicon

Each level abstracts away lower-level details while maintaining functionality.

3.2. CMOS NOT Gate Implementation

The CMOS (Complementary MOS) NOT gate demonstrates the fundamental principle of modern digital logic design using both NMOS and PMOS transistors.

3.2.1. Transistor Operation as Switches

MOS Transistor Switch Model Definition 3.2.1.1

MOSFET transistors can be modeled as voltage-controlled switches:

- NMOS: Acts like a switch between drain and source, controlled by gate voltage
  - Gate HIGH (VDD) ⟶ Switch CLOSED (conducts)
  - Gate LOW (GND) ⟶ Switch OPEN (does not conduct)
- PMOS: Acts like an inverted switch (note the bubble on gate symbol)
  - Gate LOW (GND) ⟶ Switch CLOSED (conducts)
  - Gate HIGH (VDD) ⟶ Switch OPEN (does not conduct)

3.2.2. Complete CMOS Inverter Circuit

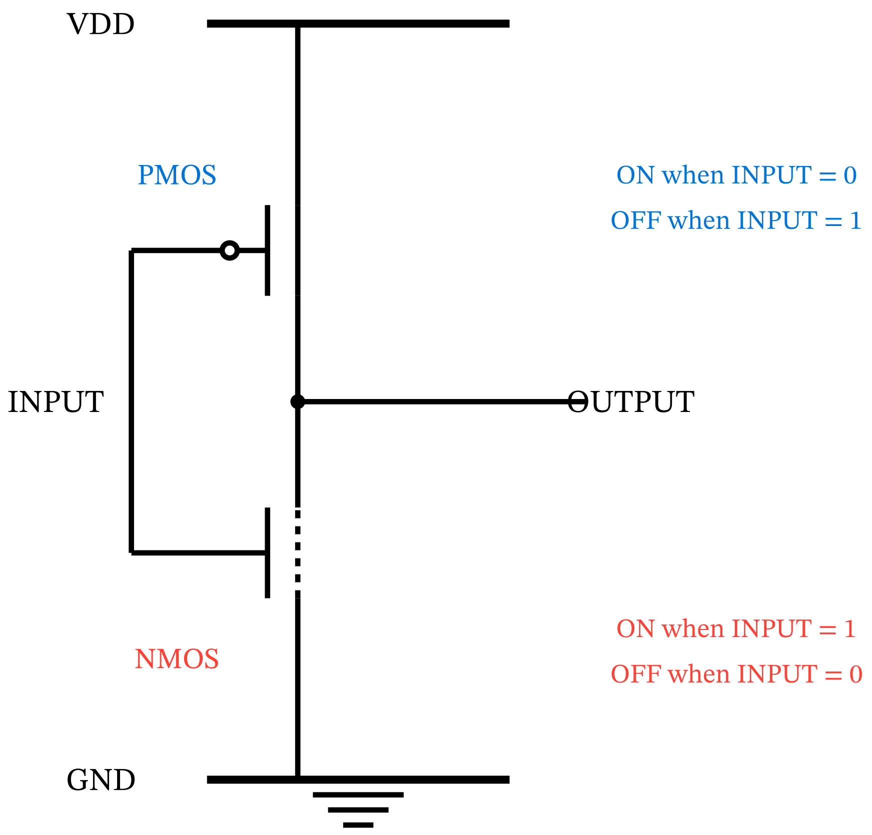


Figure 3: CMOS NOT gate schematic showing complementary operation

3.2.3. Switch Model Abstraction

To understand why we need both NMOS and PMOS, consider the resistor abstraction:

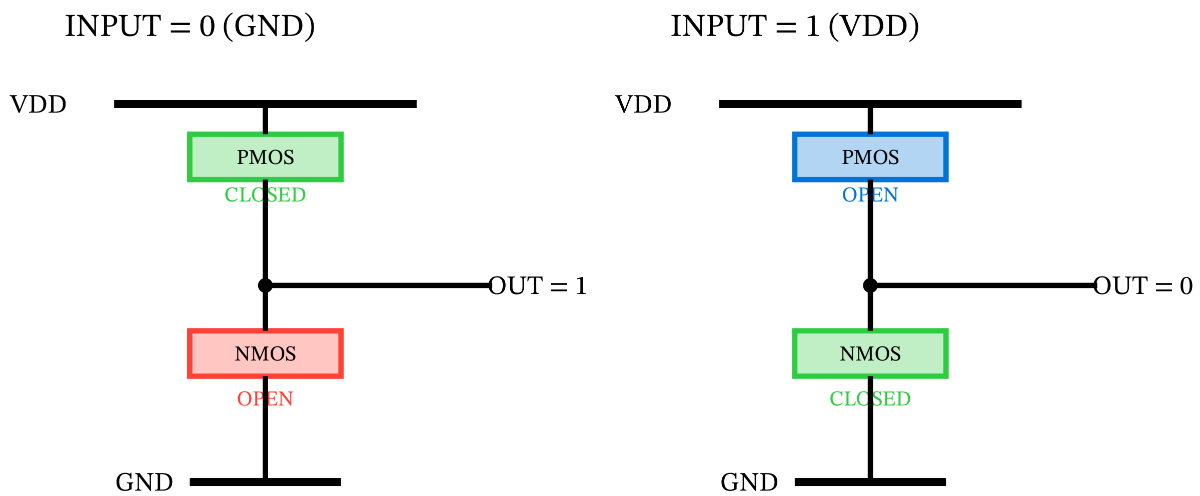


Figure 4: Switch model showing why both NMOS and PMOS are necessary

3.2.4. Why Both Transistor Types Are Essential

Necessity of Complementary Transistors Example 3.2.4.1

Each transistor type serves a specific role:

PMOS (Pull-up network):
- Connects output to VDD when input is LOW
- Good at "pulling up" to logic 1
- Poor at "pulling down" to logic 0

NMOS (Pull-down network):
- Connects output to GND when input is HIGH
- Good at "pulling down" to logic 0
- Poor at "pulling up" to logic 1

Together they provide:
- Strong drive in both directions (full rail-to-rail output)
- No static current path (one is always OFF)
- Fast switching with minimal power consumption

3.2.5. Operation Analysis

Table 3: CMOS inverter truth table and current paths

INPUT	PMOS	NMOS	OUTPUT	Current Path
0 (GND)	ON	OFF	1 (VDD)	VDD $\longrightarrow$ PMOS $\longrightarrow$ Output
1 (VDD)	OFF	ON	0 (GND)	Output $\longrightarrow$ NMOS $\longrightarrow$ GND

Power Consumption Advantage Note 3.2.5.1

The complementary nature ensures that in steady state, one transistor is always OFF, preventing any direct current path from VDD to GND. Power is only consumed during switching transitions, making CMOS extremely power-efficient compared to other logic families.

3.3. Alternative Single-Transistor Approaches (Why They Don't Work)

NMOS-only Inverter Problems Example 3.3.1

If we tried to build an inverter with only NMOS:
- Could pull output LOW when input is HIGH
- Cannot pull output HIGH when input is LOW (would need a resistor)
- Resistor would cause static power consumption
- Weak HIGH output level (degraded logic levels)

This is why early logic families like NMOS required large pull-up resistors and consumed significant power.

The CMOS approach solves all these problems by using the PMOS as an "active pull-up" device that strongly drives the output HIGH while consuming no static power.