

Lecture 2 — Intro to Logic Gates

Table of contents

Today's outline	1
Digital logic	1
Quiz: 3-input NAND truth table	2
Quiz: 3-input XOR truth table	2
Boolean algebra conventions	2
Logic diagram conventions	2
Gates on ICs	2
Logic function implementation (sum of products)	3
Equivalent functions (simplification)	3
Universal (complete) gates	3
Reminders	4

See logic-gates¹ and boolean-algebra² for the reference definitions/tables this lecture introduces.

Today's outline

- Introduction to Logic Gates
- Logic Diagrams
- Boolean Algebra and Logic Expressions

Digital logic

- Digital circuits only have **two** logical levels present (binary): logic '0' ($\sim 0V$) and logic '1' ($\sim 0.8-5V$, depending on the logic family/transistor type).

¹courses/csse2010/logic-gates.pdf

²courses/csse2010/boolean-algebra.pdf

- **Logic gates** are the building blocks of computers: each has one or more inputs and exactly one output, and performs a logic operation.
- 7 basic gate types: **NOT, AND, OR, NAND, NOR, XOR, XNOR** — see logic-gates³ for symbols and truth tables.
- Every logic function can be described four equivalent ways: **logic symbol, truth table, Boolean expression, timing diagram**.

Quiz: 3-input NAND truth table

Given four candidate truth tables, which one is the 3-input NAND?

Reasoned from the definition ($\text{NAND} = \text{NOT}(\text{AND})$), and AND is 1 only when all inputs are 1): the correct table is the one where $X = 1$ for every combination **except** $A = B = C = 1$ (where $X = 0$).

Quiz: 3-input XOR truth table

XOR is the “odd function” — $X = 1$ exactly when an **odd** number of inputs are 1.

Reasoned from that definition, the correct table (in order $ABC = 000, 001, 010, 011, 100, 101, 110, 111$) is:

$$X = 0, 1, 1, 0, 1, 0, 0, 1$$

Boolean algebra conventions

See boolean-algebra⁴ for the full notation conventions, laws/identities table, and De Morgan’s law diagrams introduced in this lecture.

Logic diagram conventions

- A filled dot at a wire junction indicates an electrical connection; wires that cross **without** a dot are not connected.
- Inputs are conventionally drawn on the left, outputs on the right.

Gates on ICs

- Example: the **74HCT00** IC has four 2-input NAND gates in a single 14-pin DIP package.
- V_{cc} = power (e.g. 5V), GND = ground (0V). Pin spacing 0.1”×0.3”; chip ~15mm long.

³courses/csse2010/logic-gates.pdf

⁴courses/csse2010/boolean-algebra.pdf

Logic function implementation (sum of products)

Any logic function can be implemented as the **OR of AND combinations** of its inputs — called **sum of products**: for each row where the output is 1, write the AND of the inputs (complemented where 0) that produces that row, then OR all those terms together.

Worked example — truth table for M :

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

$$M = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Equivalent functions (simplification)

Sum-of-products doesn't necessarily give the minimum number of gates — Boolean algebra laws (see [boolean-algebra⁵](#)) can be used to manipulate an expression into an equivalent, simpler one.

Example: $Z = AB + AC = A(B + C)$ (distributive law) — the right-hand form needs fewer gates.

Universal (complete) gates

All circuits can be built from **NAND-only** or **NOR-only** gates (called *complete/universal* gates) — NOT, AND, and OR can each be constructed purely from NAND gates (or purely from NOR gates). This matters in practice because NAND and NOR gates are easier to build from transistors than AND/OR directly.

⁵[courses/csse2010/boolean-algebra.pdf](#)

Reminders

- Attend Learning Lab sessions for the second half of this week — only the session you're signed up to.
- From week 2, you'll have both P2 (Mon-Tue) and P1 (Thu-Fri) lab sessions.