Welcome to the Electroverse

A Journey Through Digital Circuits

The digital world around you—from your phone to supercomputers—runs on billions of tiny electronic decisions happening every second.
Today, you'll discover how these digital systems actually work.

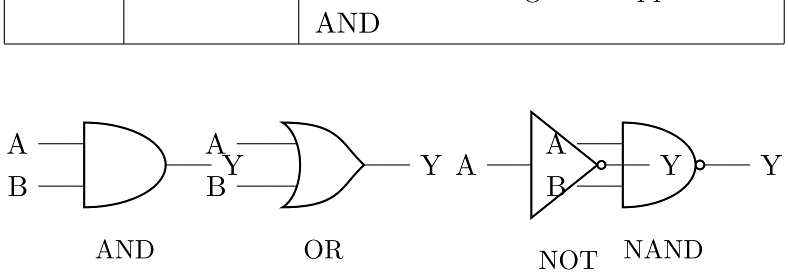
> **Your Mission:** Learn the fundamental building blocks of digital circuits, then use your newfound knowledge to solve real engineering challenges. Everything you need is explained below—read carefully!

Part 1: Your Digital Toolkit

Logic Gates: The Foundation

Imagine you're building with LEGO blocks, but instead of plastic pieces, you have gates that make decisions. Every gate takes binary inputs (0 or 1) and produces an output based on a simple rule.

Gate	Symbol	What It Does
AND	$Y = A \cdot B$	Output is 1 only when BOTH inputs are 1
OR	$Y = A + B$	Output is 1 when AT LEAST ONE input is 1
NOT	$Y = \overline{A}$	Flips the input: 0 becomes 1, 1 becomes 0
XOR	$Y = A \oplus B$	Output is 1 when inputs are DIFFERENT
NAND	$Y = \overline{A \cdot B}$	The "universal gate"—opposite of AND



Quick Example: Here's what an AND gate does with all possible inputs:

A	B	Y = A·B
0	0	0
0	1	0
1	0	0
1	1	1

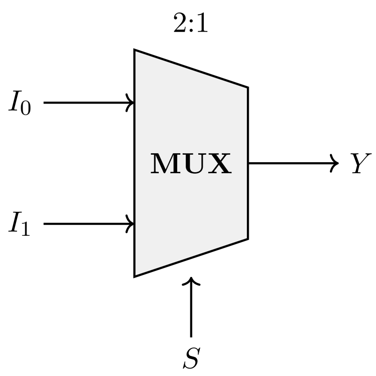
Table 1: *
Only when both A AND B are 1, output is 1

Multiplexer (MUX): The Digital Selector

Think of a MUX as a railroad switch—it selects which track (input) to send to the destination (output).

2-to-1 Multiplexer: Chooses between two inputs based on a select signal.

+

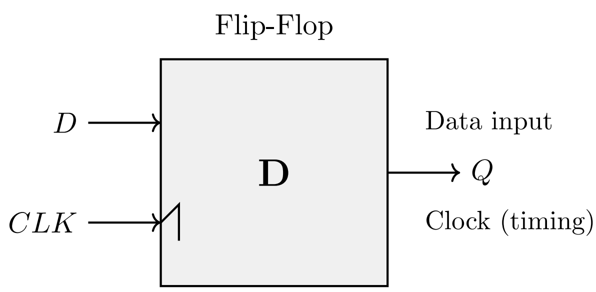


If $S = 0$: Output $= I_0$
If $S = 1$: Output $= I_1$

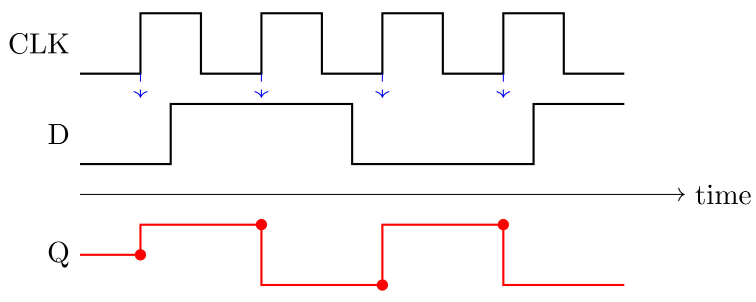
The Boolean expression: $Y = \overline{S} \cdot I_0 + S \cdot I_1$

D Flip-Flop: The Memory Element

Here's where things get interesting! Unlike gates that just process signals, a flip-flop can *remember* information.



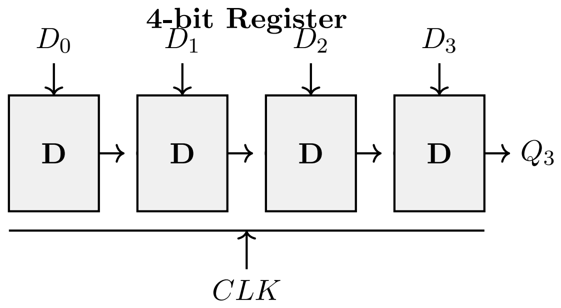
How it works: At the moment the clock signal goes from 0→1 (rising edge), the flip-flop captures whatever value is at input D and stores it in Q. Between clock pulses, Q stays constant even if D changes.



Q captures D at each rising clock edge

Register: Multi-Bit Memory

A register is just multiple flip-flops working together. If you want to store a 4-bit number like 1011, you need four flip-flops.



All flip-flops share the same clock, so they all capture their inputs simultaneously.

Part 2: Challenge Missions

Now that you understand the building blocks, let's put them to work! Start with the easier challenges and work your way up.

Mission 1: Building with Universal Gates

NAND gates are special—they're called "universal" because you can build ANY other gate using only NANDs. This is why they're so popular in real chip design.

Your Task: Show how to create these gates using ONLY NAND gates:

(a) A NOT gate (Hint: What happens if both inputs of a NAND are the same?)

(b) An AND gate

(c) An OR gate

Draw the circuits clearly with inputs and outputs labeled.

Mission 2: Binary Addition

Computers need to add numbers. Let's build the most basic adder—one that adds two single bits.

When you add in binary: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, but $1 + 1 = 10$ (that's "two" in binary: Sum = 0, Carry = 1)

Your Task: Design a **Half Adder** circuit that takes two bits A and B and produces:

- Sum bit (S)
- Carry bit (C)

(a) Create the complete truth table

(b) Write Boolean expressions for S and C

(c) Draw the circuit using basic logic gates

Hint: Look at when the outputs are 1. Does one of them remind you of a gate you've learned?

Mission 3: Building a Smart Selector

You have four data sources (I_0, I_1, I_2, I_3) and need to select one based on a 2-bit control signal.

Your Task: Design a 4-to-1 Multiplexer using three 2-to-1 MUX blocks (as shown in Part 1).

Draw the complete circuit showing:

- How all four inputs connect
- How select signals S_1 and S_0 control which input reaches the output
- Label everything clearly

Think hierarchically: First, use two MUXes to choose between pairs, then...

Mission 4: Pattern Detector

This is where it gets really interesting! You need to build a circuit that monitors a data stream and detects a specific pattern.

Your Task: Design a sequential circuit that watches input X and outputs $P = 1$ whenever it detects three consecutive 1s (pattern "111").

For example:

- Input stream: 0, 1, 1, 1, 0, 1, 1, 1, 0...
- Output P: 0, 0, 0, 1, 0, 0, 0, 1, 0...

(a) How many D Flip-Flops do you need? Explain your reasoning.

(b) Draw the complete circuit including:

- All flip-flops with their connections
- The combinational logic that generates output P
- Clock signal

Hint: You need to "remember" what you've seen. If you just saw two 1s and get another 1, that's when P should be 1.

Mission 5: The Counter Mystery

A counter is a circuit that counts clock pulses. A 2-bit counter counts: 00 → 01 → 10 → 11 → 00 (and repeats).

Your Task: Design a 2-bit up-counter using two D flip-flops.

(a) Create the state transition table showing current state (Q_1Q_0) and next state after one clock pulse.

(b) Derive Boolean expressions for the inputs D_1 and D_0 in terms of current outputs Q_1 and Q_0.

(c) Draw the complete circuit.

Hint: Look at each bit. Q_0 toggles every cycle. When does Q_1 toggle?

These challenges mirror real problems that digital designers solve every day.
Take your time, think logically, and sketch out your ideas.
Remember: every complex chip started with these same building blocks!