Circuits

CISC 3310 — CUNY Brooklyn College
Lecture Notes

Circuits

In this chapter, we will:

1. Describe what ICs are, and why manufacturers prefer to create them instead of individual logic gates.
2. Learn what combinational circuits are, and see examples of combinational circuits.
3. Learn what sequential circuits are, and see examples of sequential circuits.

Circuits

In the last topic, we've found out how to create logic gates, which are tiny circuits made up of Transistors. This topic will introduce us to larger circuits, the purpose of each is to perform an important work inside a computer.

To increase portability and material utilization, manufacturers don't sell logic gates as individual devices. Instead, logic gates are sold in groups on top of chips called **Integrated Circuits** or **IC**s.

The logic gates present in ICs are usually NAND gates, which, as we learned in Topic 3, are universal gates and can be used to create any other type of gate you want.

Integrated Circuits range from Small-Scale Integration (SSI) models, which contain only a few logic gates on them, to Ultra-Large-Scale Integration (ULSI) models containing billions of logic gates on them.

Circuits

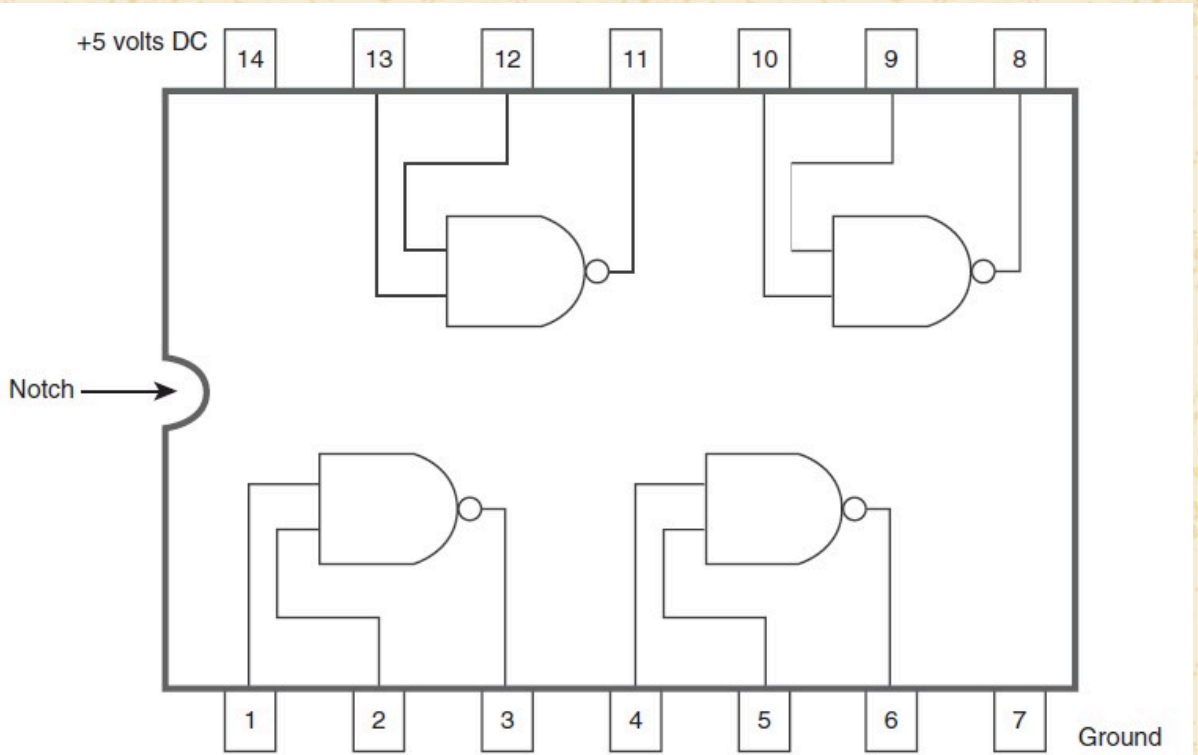
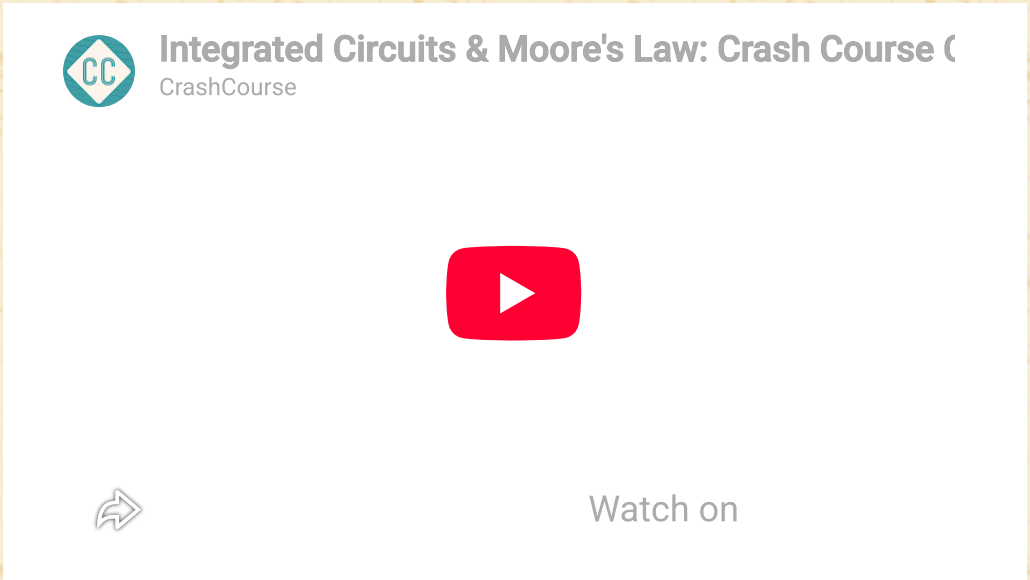


FIGURE 3.19 Simple SSI Integrated Circuit

A small integrated circuit. Figure 3.19 on page 165 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Circuits

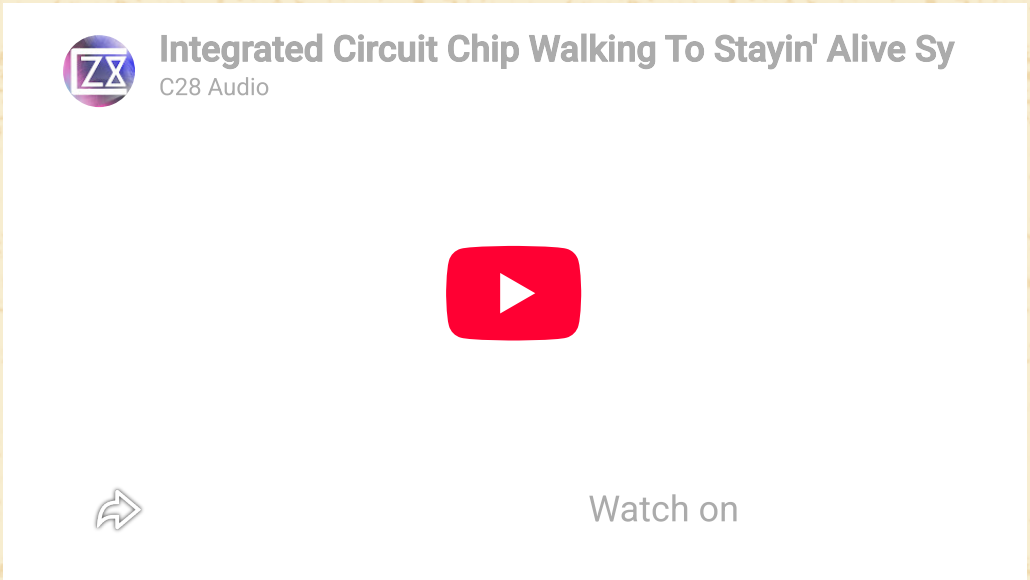
A video elaborating on the process of creating digital circuits.



https://www.youtube.com/watch?v=6-tKOHICqrI

Circuits

The Integrated Circuit in the video below is a NE555 timer.



https://www.youtube.com/watch?v=mDhNQPt8An0

Circuits

Any questions so far?

Combinational Circuits

We divide circuits into **two** kinds of categories: (1) Combinational Circuits, and (2) Sequential Circuits.

Combinational Circuits are simple circuits usually performing a computation. They don't store information, and their output(s) only depends on the input bits that they receive. Circuits that don't store memory or use history are called **stateless** circuits.

Sequential Circuits, on the other hand, are more complex than Combinational Circuits because they store information about previous uses (that is, the past inputs) of the circuit, and their output depends not only on the input bits but also on the memory they store. Such history-preserving circuits are called **stateful**.

Let's introduce examples of combinational circuits, and show how they work.

Combinational Circuits: Adders

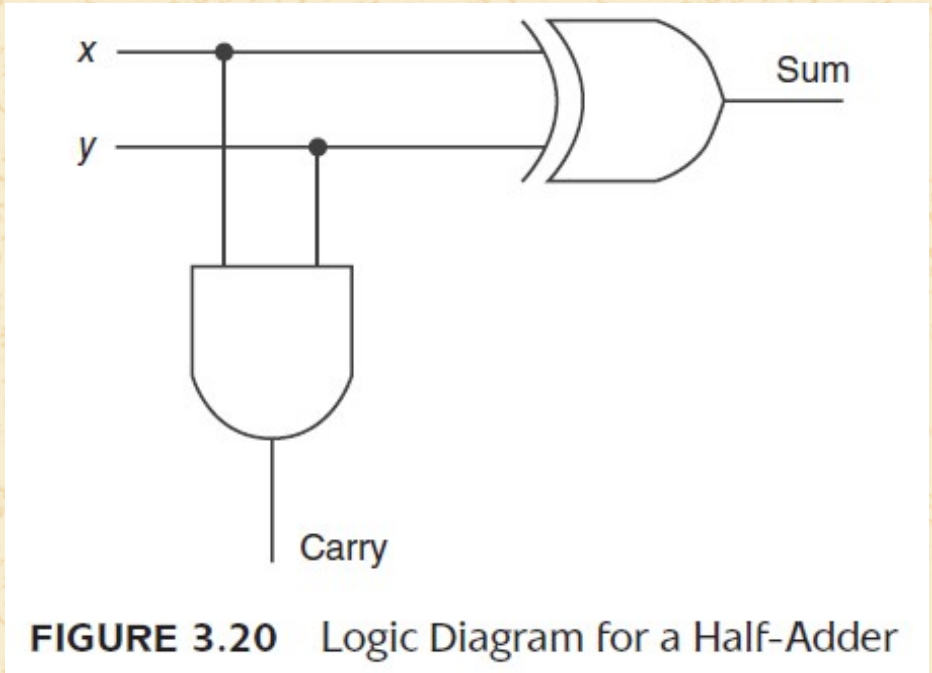
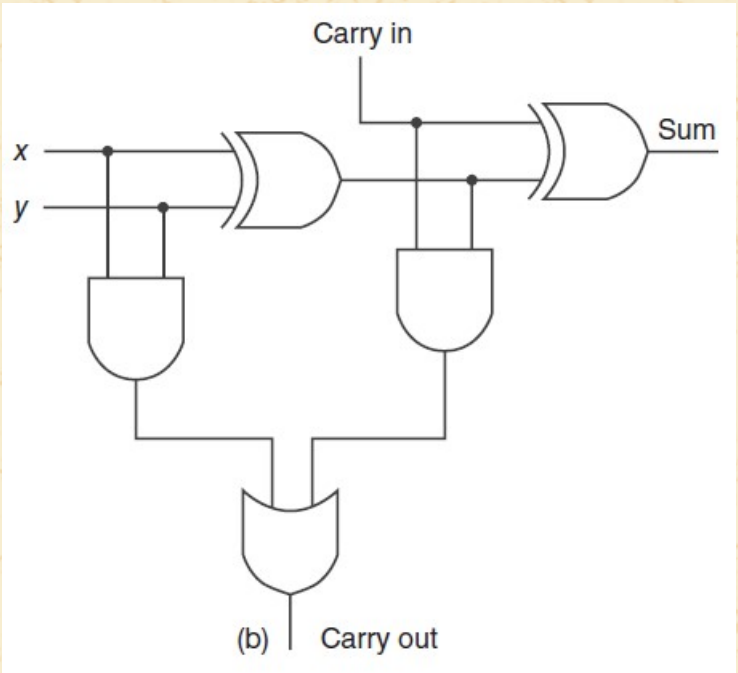
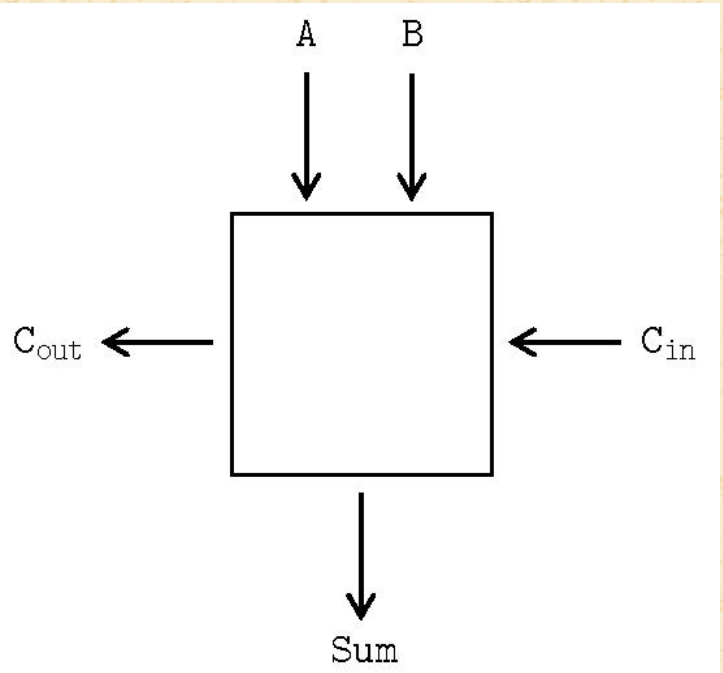


FIGURE 3.20 Logic Diagram for a Half-Adder

Half Adder. Figure 3.20 on page 171 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

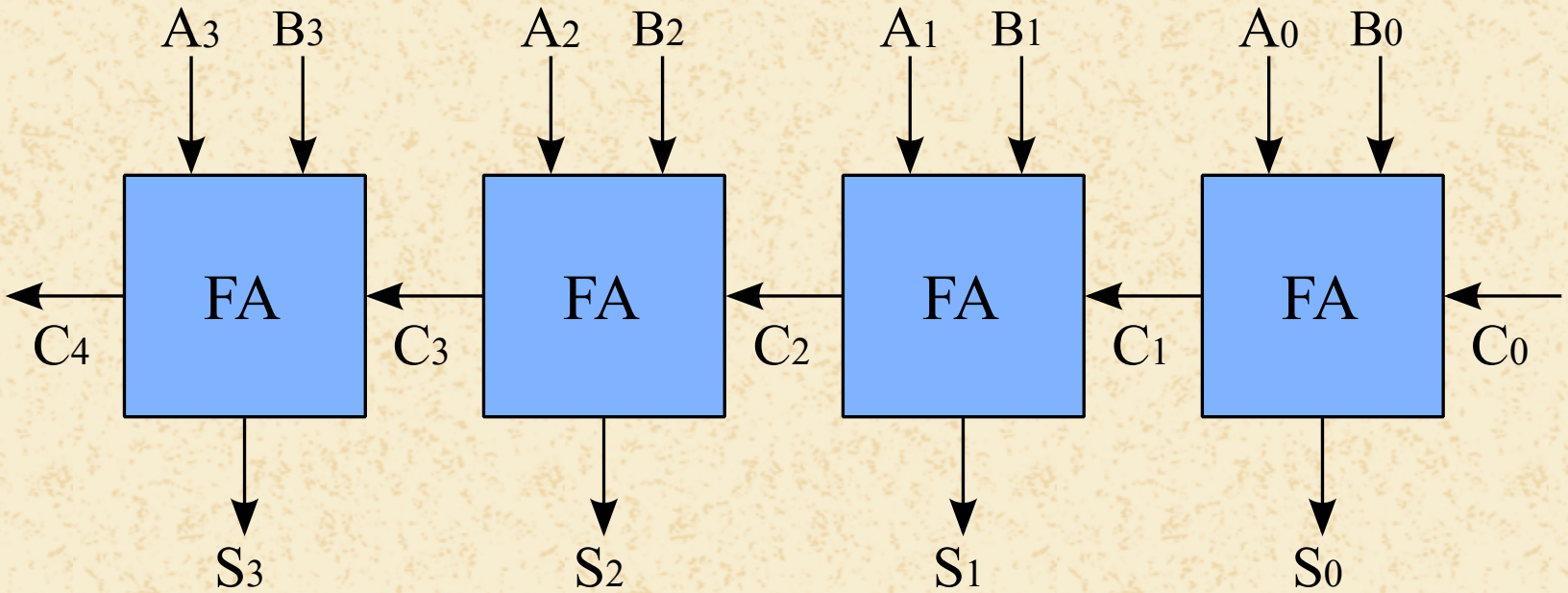


Full Adder. Figure 3.21.(b) on page 172 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



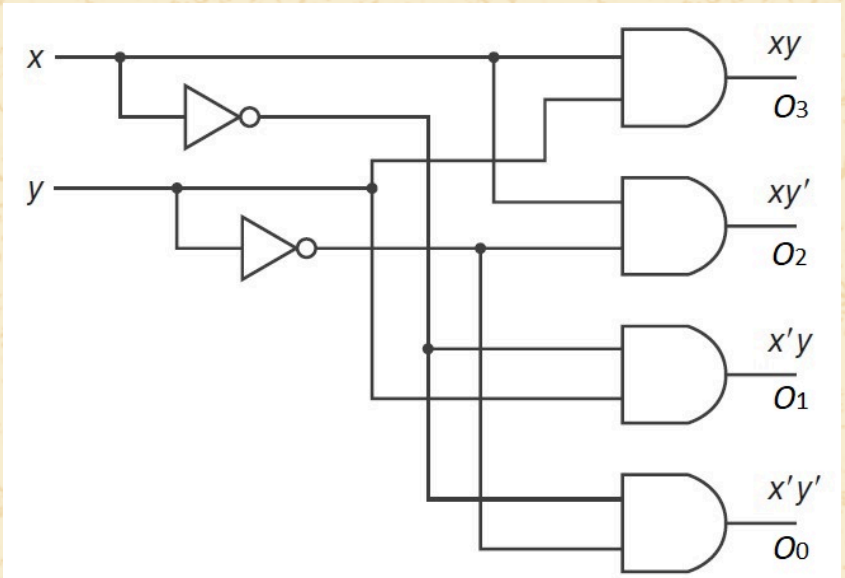
Full Adder: Blackbox. Miriam Briskman, CC BY-NC 4.0.

Combinational Circuits: Adders

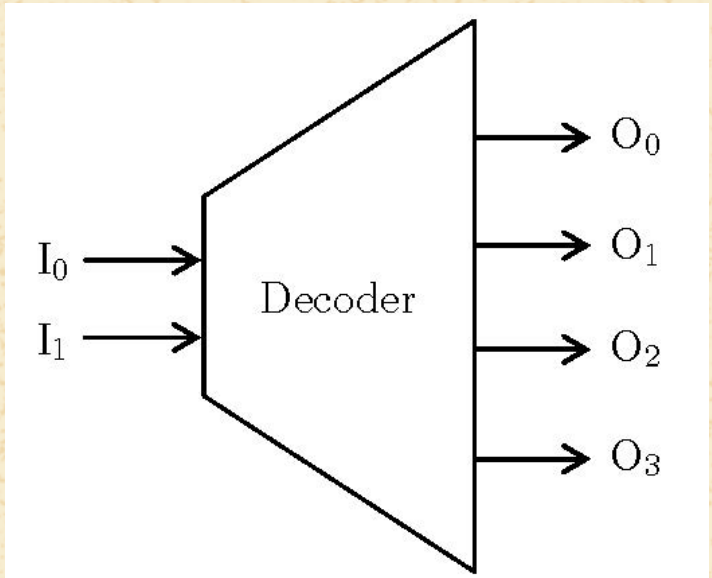


A ripple-carry adder of 4-bit numbers. derivative work: Moberg (talk)4-bit_ripple_carry_adder.svg: en:User:Cburnett, CC BY-SA 3.0, via Wikimedia Commons.

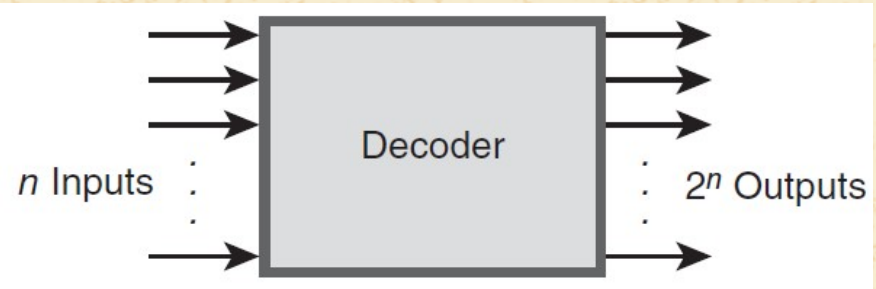
Combinational Circuits: Decoders



2 to 4 Decoder. Figure 3.23.(a) on page 175 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

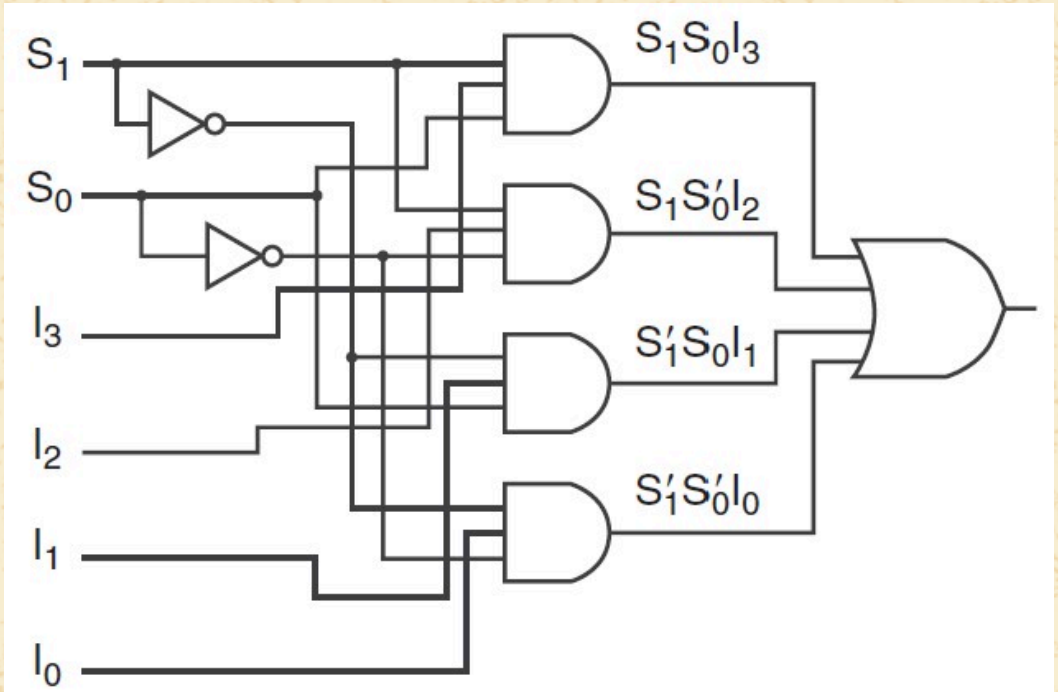


2 to 4 Decoder: Blackbox. Miriam Briskman, CC BY-NC 4.0.

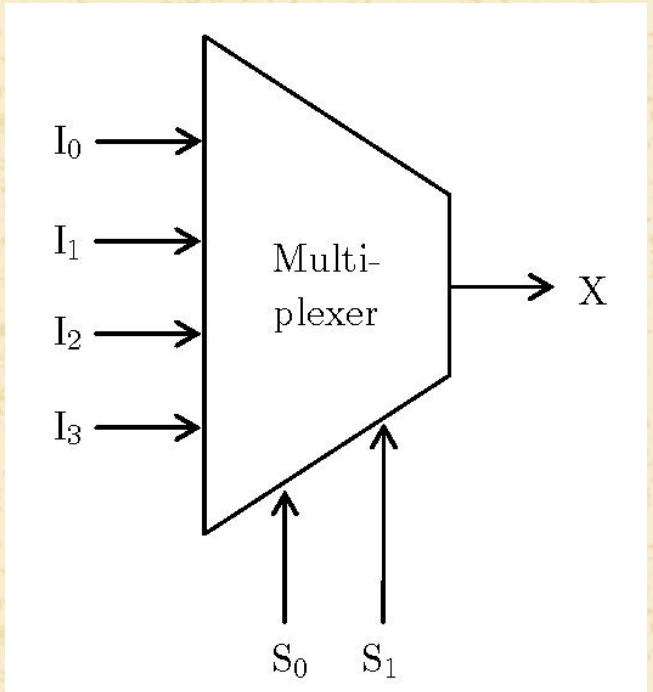


n to 2^n Decoder: Blackbox. Figure 3.23.(b) on page 175 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Combinational Circuits: Multiplexers (= Input Selectors)



4 Way Multiplexer. Figure 3.24.(a) on page 175 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



4 Way Multiplexer: Blackbox. Miriam Briskman, CC BY-NC 4.0.

Combinational Circuits

Any questions so far?

Combinational Circuits: Shifters

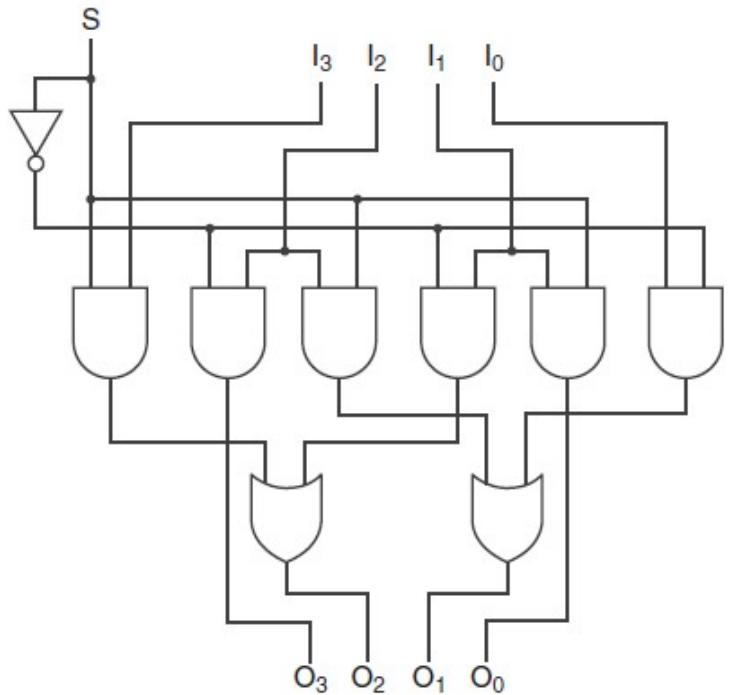


FIGURE 3.25 4-Bit Shifter

A shifter of 4 bits. Figure 3.25 on page 177 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Combinational Circuits: Multipliers

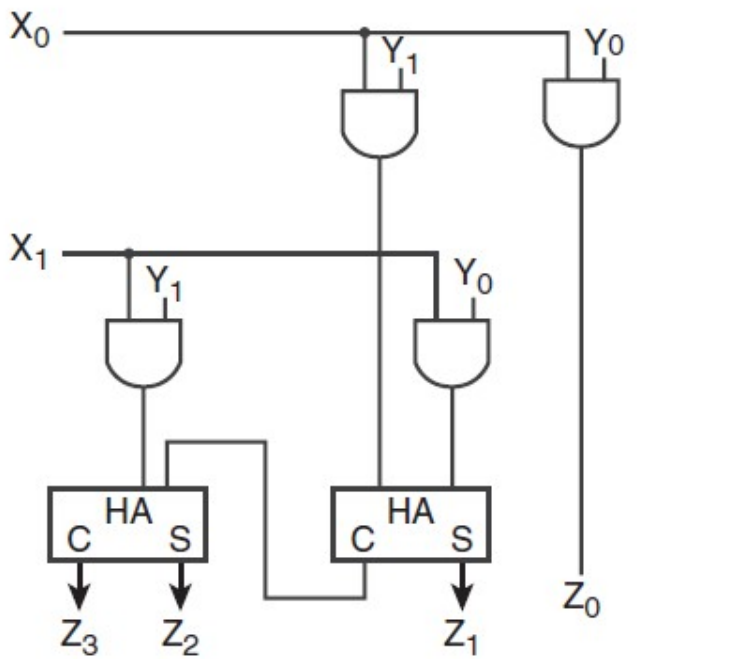


FIGURE 3.27 A Simple 2-Bit Multiplier

A multiplier of two 2-bit numbers. Figure 3.27 on page 178 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Combinational Circuits: Simple ALU

The **arithmetic logic unit** (**ALU**) is a mini calculator: it performs one of several types of calculations, as per your choice. The ALU below can do addition, NOT, OR, and AND on 2 bits based on the signal (f_0 and f_1) you pass to this circuit.

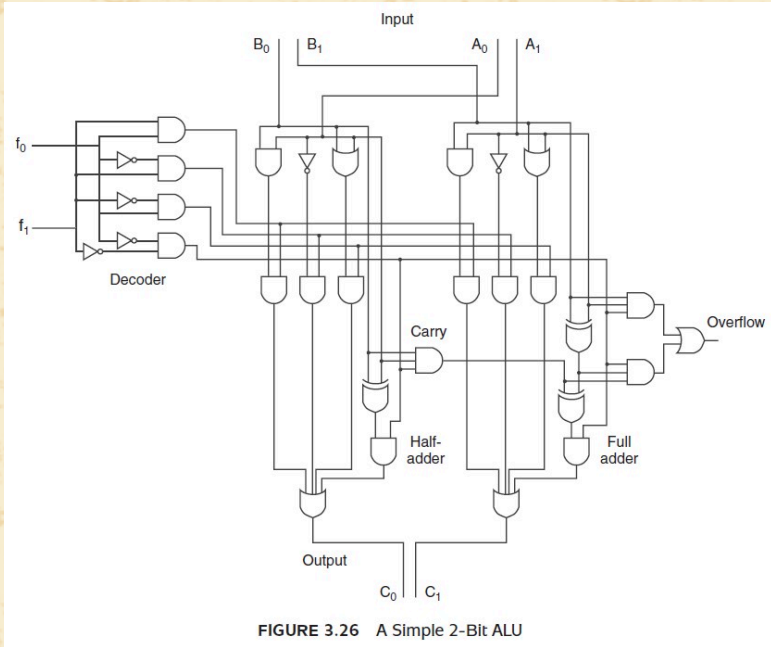


FIGURE 3.26 A Simple 2-Bit ALU

A simple ALU. Figure 3.26 on page 178 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Combinational Circuits

Any questions so far?

Sequential Circuits

Sequential circuits are digital circuits with an output that depends on both (1) current inputs and (2) past states (= previous inputs,) unlike combinational circuits which are purely input-dependent and don't store their past state.

- Think of combinational circuits as simple calculators - they give immediate answers but can’t save or remember them.

Sequential circuits retain past information, enabling them to "remember" states due to their ability to store binary data.

By storing these past states, sequential circuits play a key role in tasks that require memory, such as counters, registers (e.g., CPU registers,) and finite state machines. These circuits are critical in designing memory elements, control units, and other parts of a computer that rely on tracking past operations.

- Memory and state storage are essential in designing CPUs, memory units, and control units, which can’t be done with combinational circuits alone.

Sequential Circuits: Flip-Flops

The most basic circuit that keeps memory is called a flip-flop. **Flip-flops** are binary storage devices that store a single bit of data, acting as the building blocks of memory in sequential circuits.

- Flip-flops to sequential circuits are what logic gates are to combinational circuits are.

A flip-flop has two stable states (0 and 1), and it changes states based on control inputs or triggers. It holds its state until it receives a signal to switch.

Different flip-flops (SR, D, JK, T) are used based on requirements for control and state behavior. For instance, D flip-flops store input data directly, while JK flip-flops offer flexible control for toggling between states.

They use a concept called **feedback**, where the output is fed back into the input. This feedback allows flip-flops to hold onto information even after the input signal changes, providing the memory necessary for sequential circuits.

Sequential Circuits: Clocks

Before we describe how each flip-flop type works in details, we first need to introduce the concept of a clock.

The **clock** is a timing device that generates pulses to synchronize changes in a sequential circuit. It tells flip-flops when to update their states.

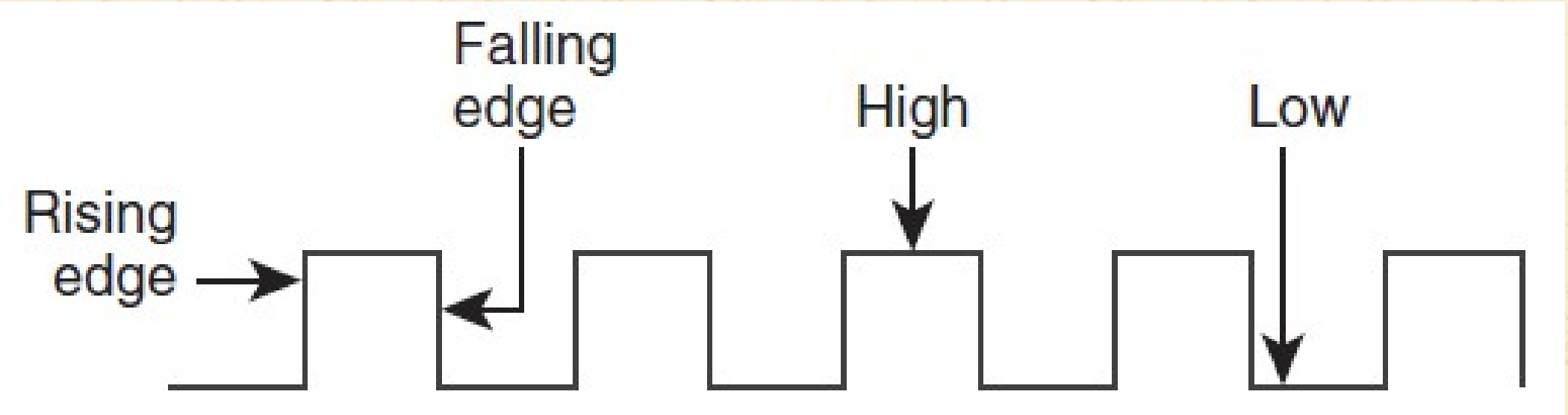
This signal is typically a square wave that alternates between high (1) and low (0) states at a specific frequency, coordinating all circuit operations to prevent data errors. The next slide contains a diagram showing these clock waves.

The clock is built using an oscillator sequential circuit, commonly based on quartz crystals or RC (resistor-capacitor) circuits to generate consistent timing pulses.

- A quartz oscillator or RC circuit generates a regular signal, which can then be shaped into a square wave using additional circuitry to ensure precise timing.

Sequential Circuits: Clocks

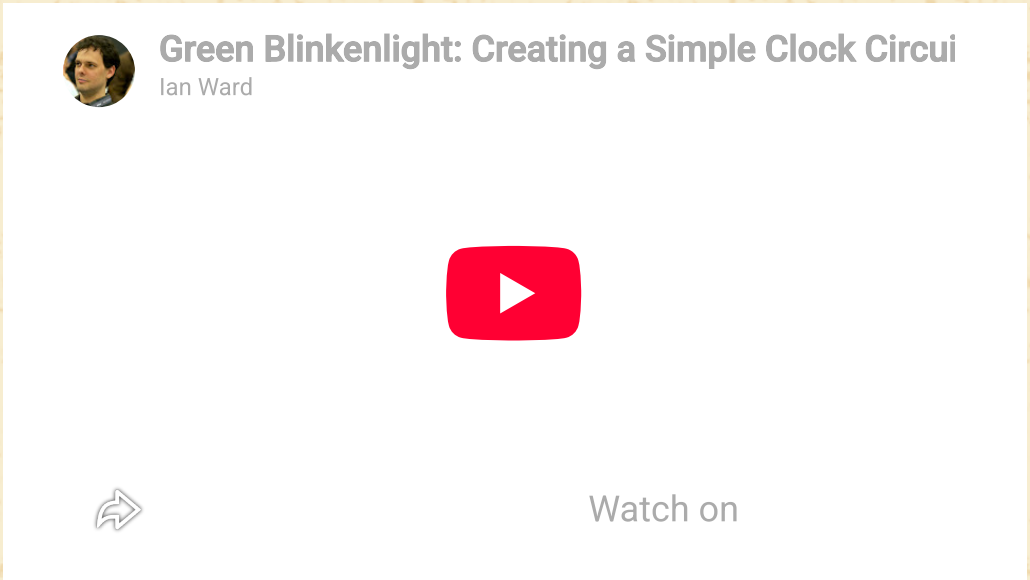
A diagram of the square waves of a clock:



Clock's square waves. Figure 3.28 on page 181 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: Clocks

One way to implement a clock using a timer integrated circuit:



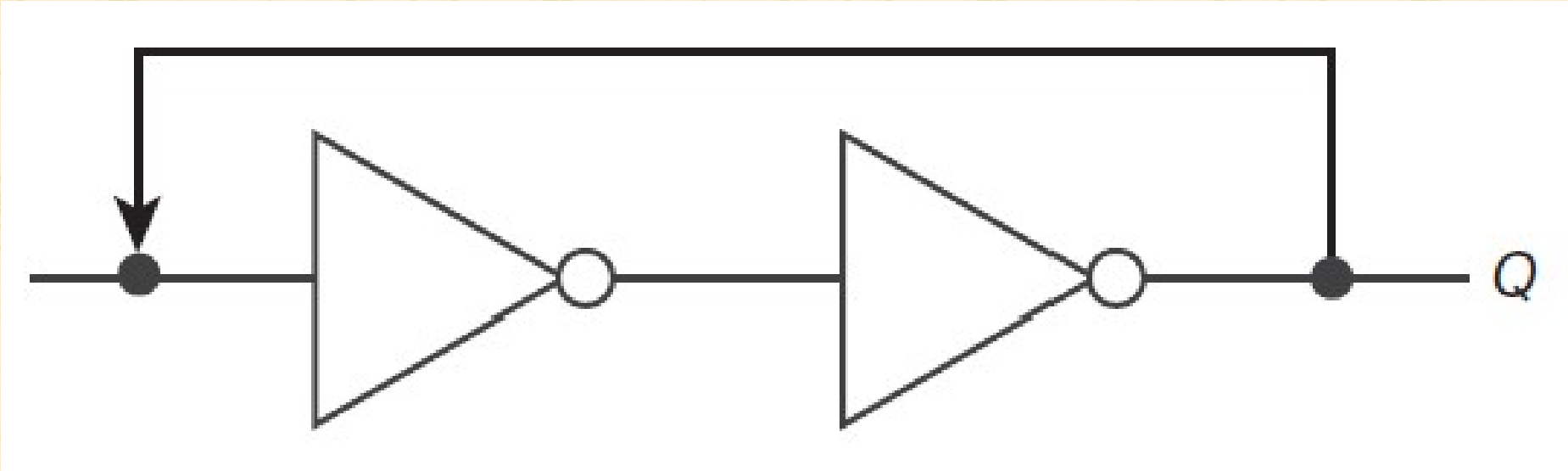
https://www.youtube.com/watch?v=QfnkuXDf6NE

Sequential Circuits: Clocks

Any questions so far?

Sequential Circuits: Feedback

This simple sequential circuit is called **feedback** and all it does is sending its output back into this circuit as an input; it literally feeds this output back to itself:

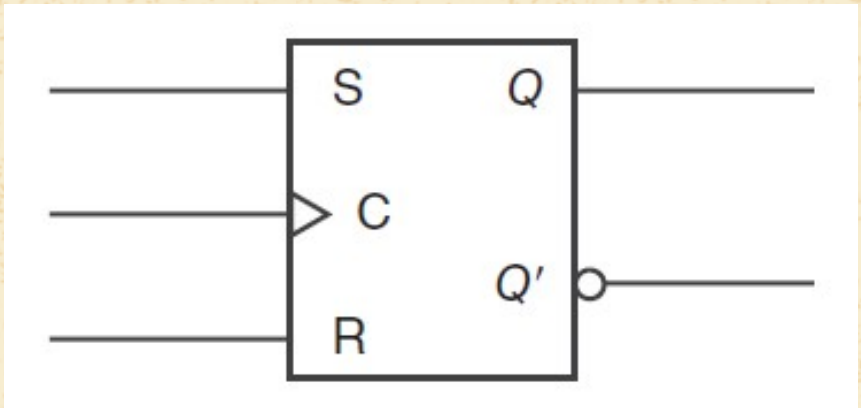


A feedback circuit. Figure 3.29 on page 181 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

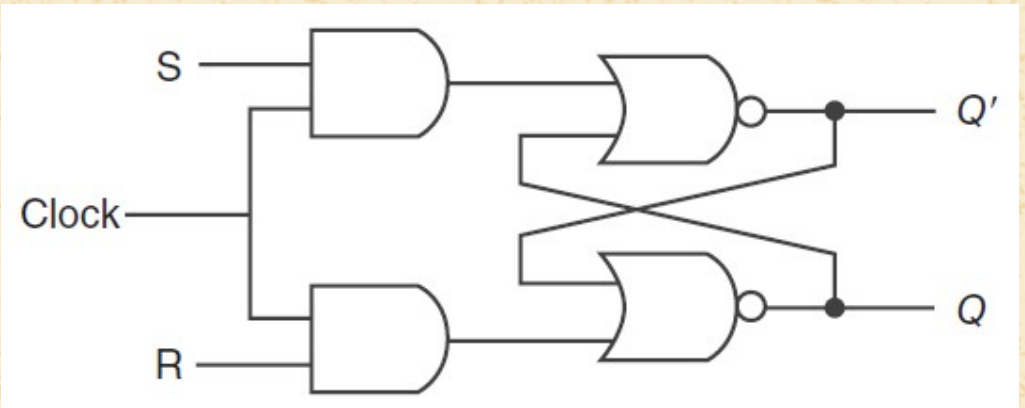
Sequential Circuits: SR Flip-Flop

The **SR** (**Set-Reset**) flip-flop has two inputs: **S** (Set) and **R** (Reset).

When S is activated (1) and R is deactivated (0), the flip-flop sets its output to 1. When R is activated and S is deactivated, the output resets to 0. If both inputs are 0, the output retains its previous state. However, if both inputs are 1, it creates an undefined or forbidden state.

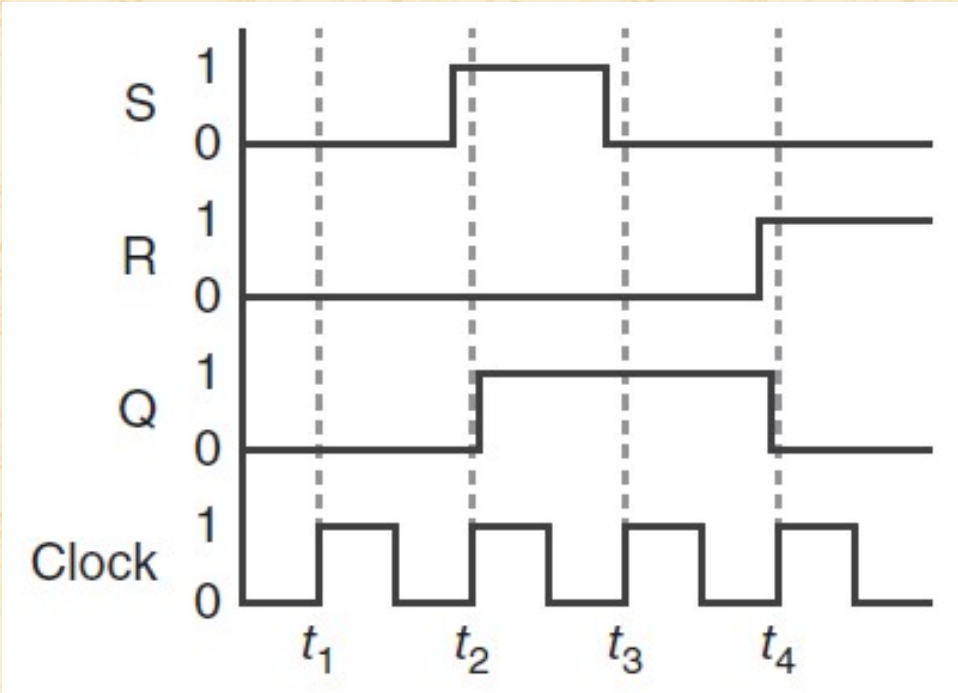


SR flip-flop: black-box representation. Figure 3.30 on page 182 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



SR flip-flop: logic gate diagram. Figure 3.31.(b) on page 182 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: SR Flip-Flop



SR flip-flop: clock-wave diagram. Figure 3.31.(d) on page 182 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

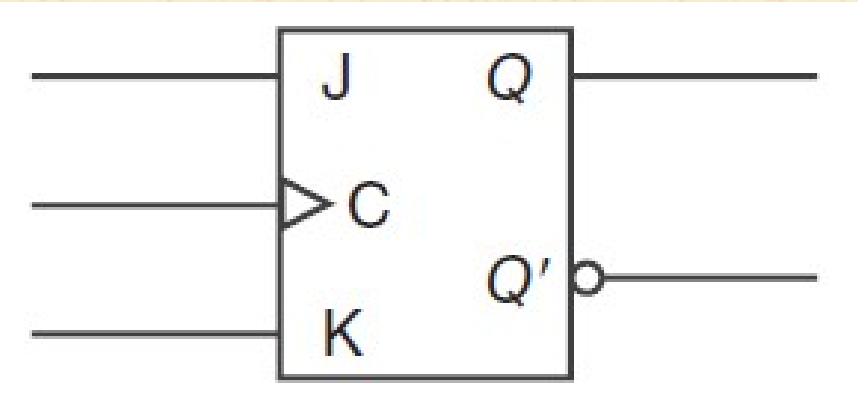
S	R	Q (Next State)
0	0	No Change
0	1	0
1	0	1
1	1	Undefined

Truth Table for SR

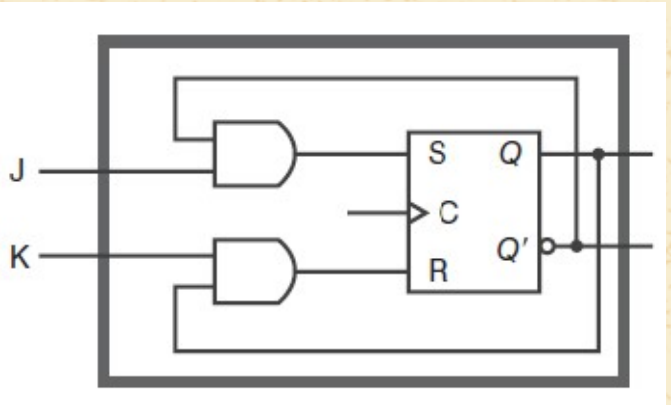
Sequential Circuits: JK Flip-Flop

The **JK** flip-flop is similar to the SR flip-flop but with the following upgrade: instead of entering an undefined state when both J and K inputs are 1, the JK flip-flop toggles the output.

With inputs J and K, the flip-flop can be set, reset, or toggle its output, depending on the input state and clock signal. It is highly versatile in digital systems requiring toggle functionality.

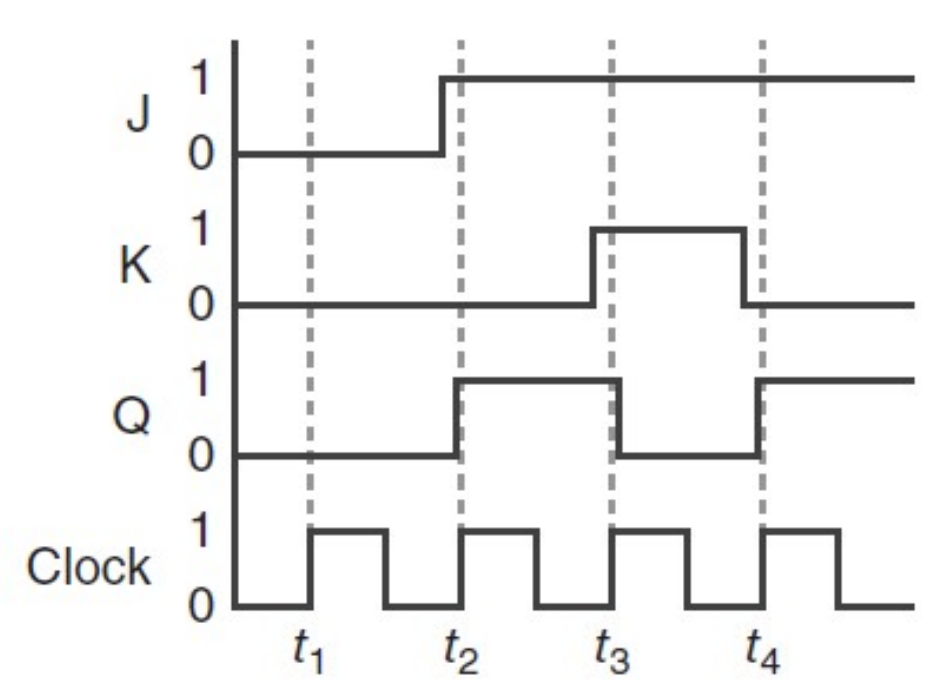


JK flip-flop: black-box representation. Figure 3.32.(a) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



JK flip-flop: logic gate diagram. Figure 3.32.(c) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: JK Flip-Flop

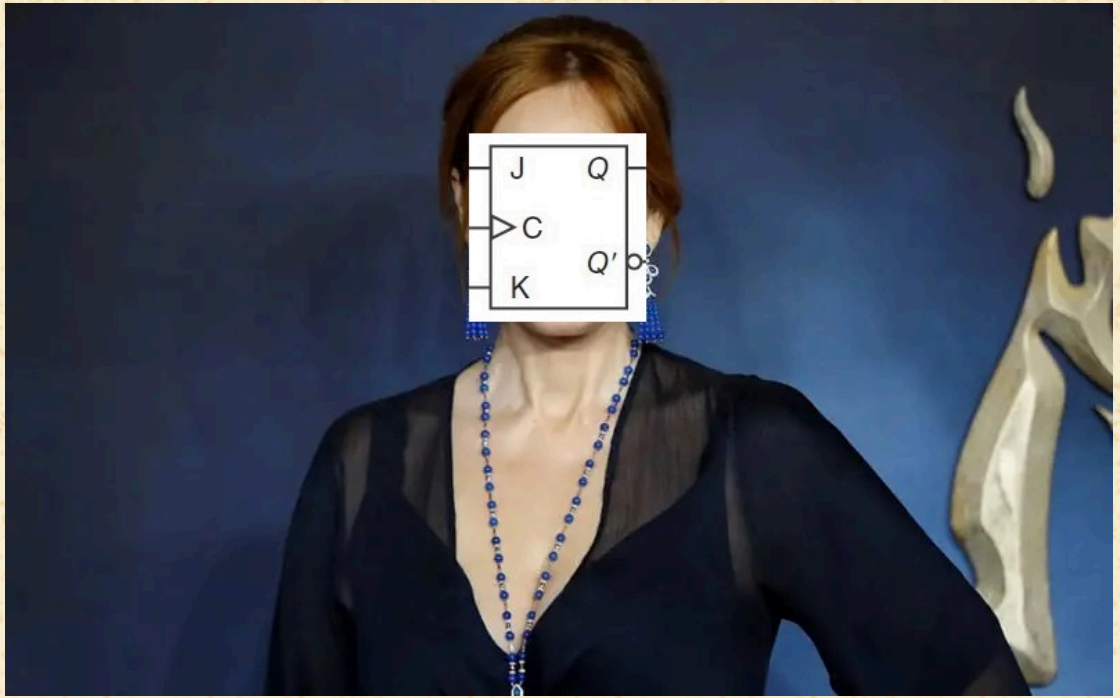


JK flip-flop: clock-wave diagram. Figure 3.32.(d) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

J	K	Q (Next State)
0	0	No Change
0	1	0
1	0	1
1	1	Toggle

Truth Table for JK

Sequential Circuits: JK Flip-Flop



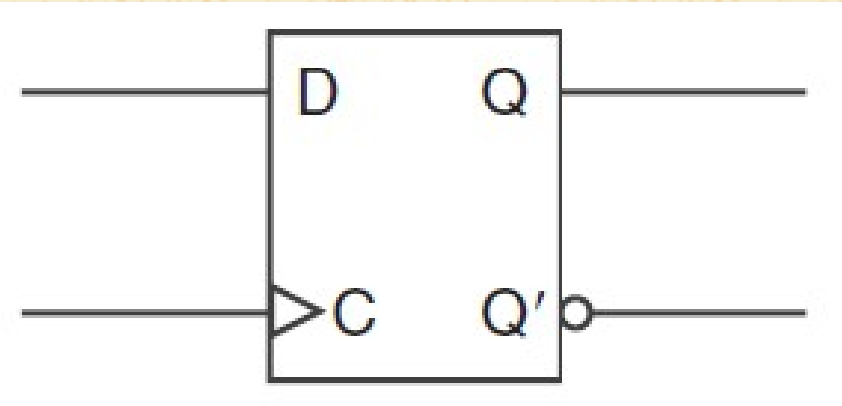
JK flipflop. Adapted from an image in this FMT article.

[Disclaimer: no offense intended. This is just a fun way to remember the JK flip-flop circuit.]

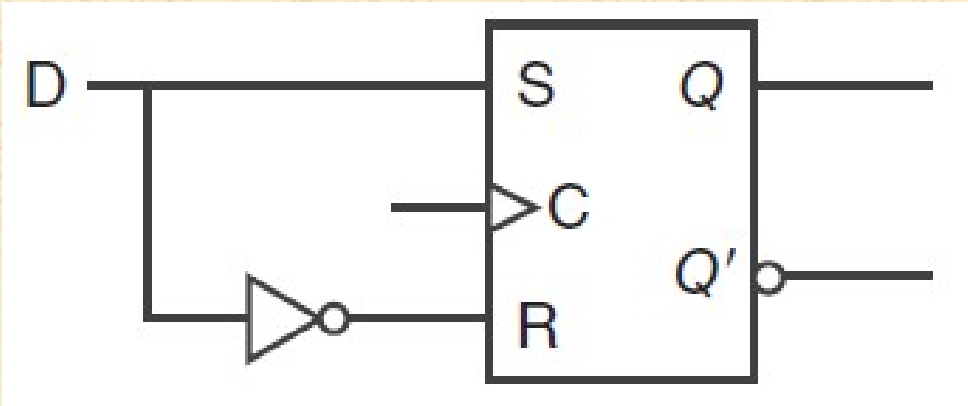
Sequential Circuits: D Flip-Flop

The **D** (Data or Delay) flip-flop has a single input, D, and is often used for data storage.

The D flip-flop stores the value of D at the moment of the clock pulse and retains this value until the next clock pulse. This design prevents the undefined state present in SR flip-flops, as there is only one input.

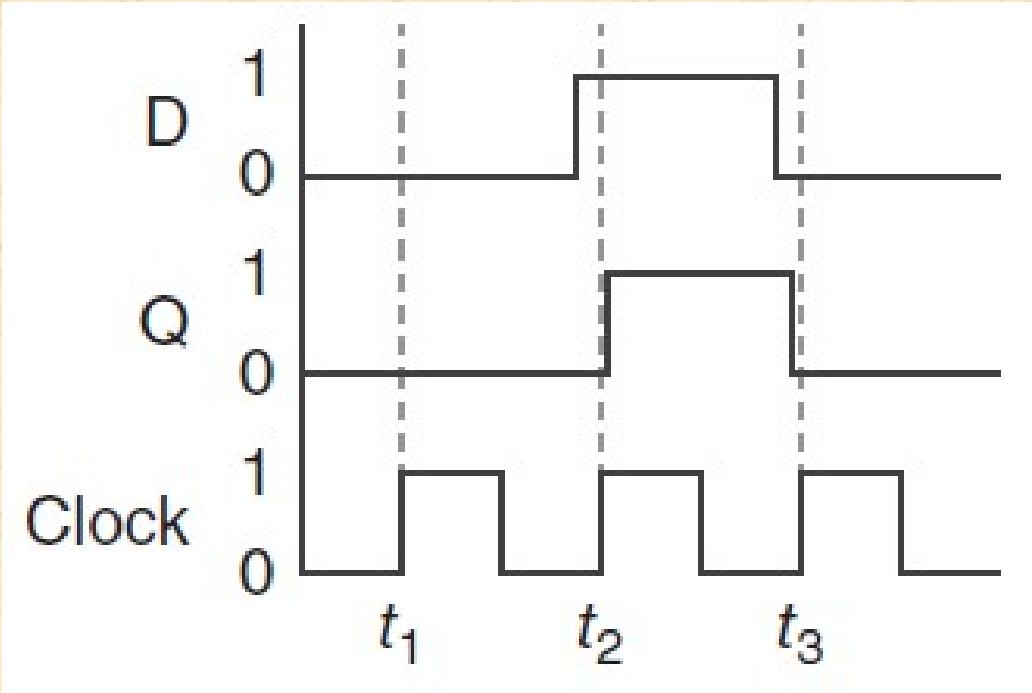


D flip-flop: black-box representation. Figure 3.32.(a) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



D flip-flop: logic gate diagram. Figure 3.32.(c) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: D Flip-Flop



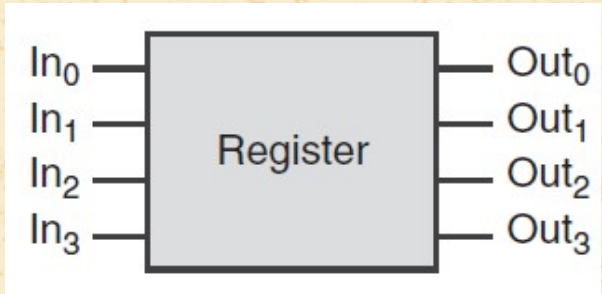
D flip-flop: clock-wave diagram. Figure 3.32.(d) on page 184 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

D	Q (Next State)
0	0
1	1

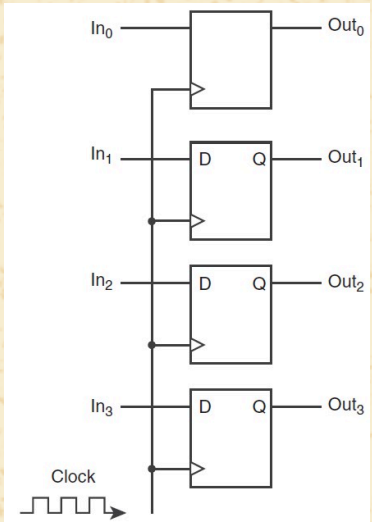
Truth Table for D

Sequential Circuits: Register

A **register** is a small, high-speed storage location within a CPU, designed to hold and quickly access specific data or instructions during processing. Registers are essential components in CPU architecture, enabling rapid data handling for various operations.

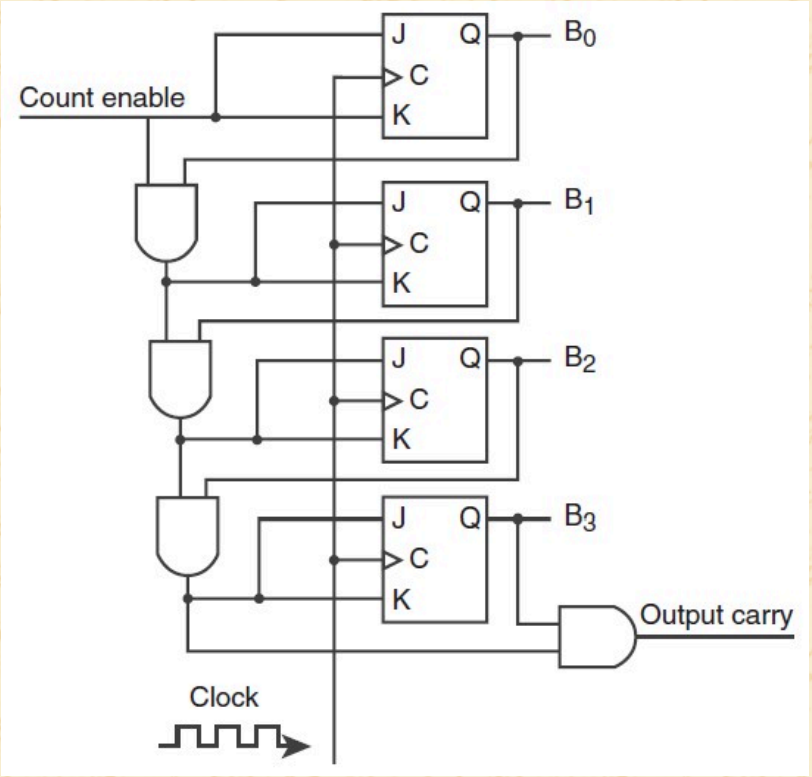


A Register storing 4 bits: black-box representation. Figure 3.40.(b) on page 194 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.



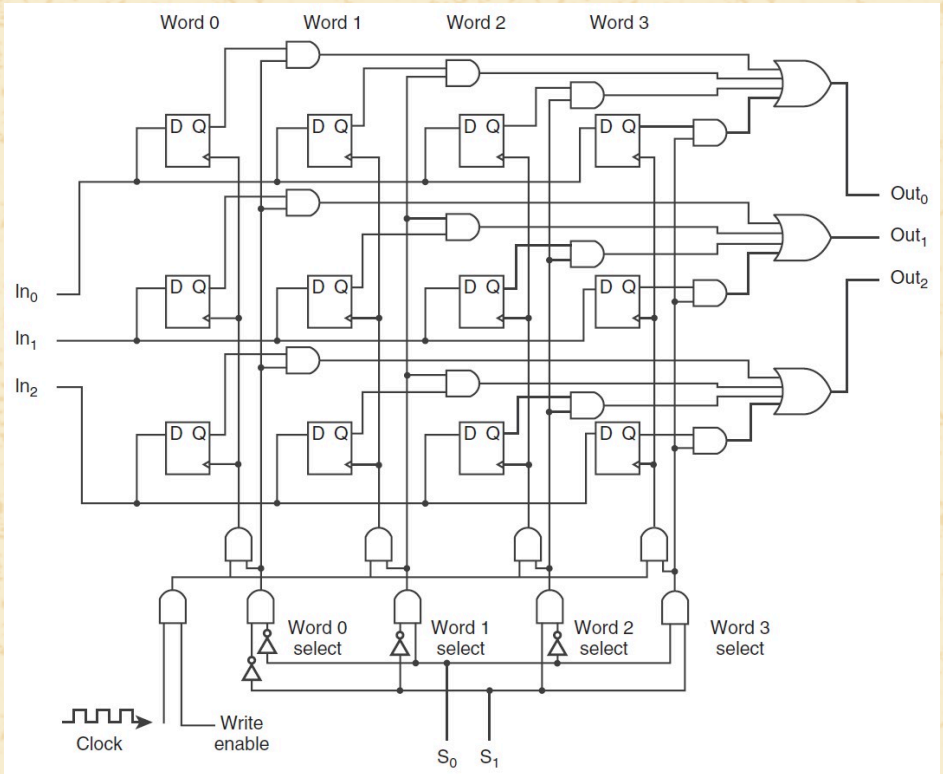
A Register storing 4 bits: logic gate diagram. Figure 3.40.(a) on page 194 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: Counter



A 4-Bit Counter built up of JK Flip-Flops. Figure 3.41 on page 195 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits: Memory



A 3x4 bit memory chip. Figure 3.42 on page 196 of *Essentials of Computer Organization and Architecture* by Linda Null, 2023.

Sequential Circuits

Any questions so far?

Circuits: Sources

Bergmann, Seth D. (2018). Section 6.3 and Section 6.4. *Computer Organization with MIPS*. License: CC BY-NC-ND: Attribution-NonCommercial-NoDerivs.

CS301: Computer Architecture. Unit 4. *Saylor Academy*, 2010. License: CC BY: Attribution.

Tarnoff, David. (2020). Episodes 6.08, 6.09, 6.10, 7.01, 7.02, 7.03, 7.04, 7.05, and 7.06. *Computer Organization and Design Fundamentals Series*. License: CC BY-NC-SA: Attribution-NonCommercial-ShareAlike.