Vietnam National University HCMC

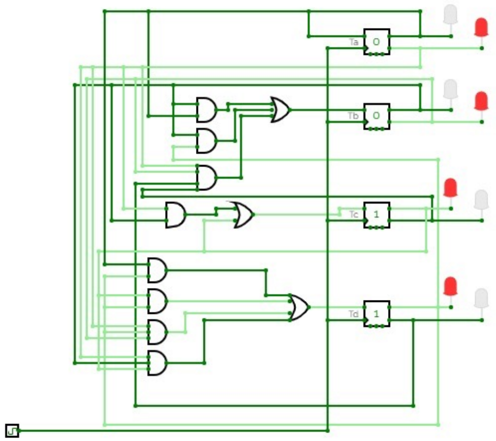
International University

School of Electrical Engineering

EE053IU

Digital Logic Design

Lecture 1: Introductory Concepts



INSTRUCTOR: Dr. Vuong Quoc Bao

Instructor's Information

- **Instructor:** Dr. Vuong Quoc Bao
- **Email address:** vqbao@hcmiu.edu.vn
- **Office location:** LA2. 109 (RF & Microwave Lab)
- **Office hours:** Tuesday (8:00 AM – 11:00 AM)

General Information

- **Course Name:** Digital Logic Design
- **Course Code:** EE053IU
- **Program:** Undergraduate
- **No. of Credits: 3**
- **Prerequisite/ Parallel Courses:** Digital Logic Design Lab (EE054IU)
- **Textbook:** ***2015 - Digital Fundamentals 11th Edition Global Ed - Thomas L Floyd***
- **Assessment Methods:**
  - Attendance/Assignments/ Quizzes: 30%
  - Midterm: 30%
  - Final exam: 40%

- **Course Description:**

This course introduces the basic tools for design with combinational and sequential digital logic and state machines. To learn simple digital circuits in preparation for computer engineering. Main content: Binary arithmetic, Boolean algebra, K-maps, Combinational circuit synthesis, Combinational MSI circuits, Sequential logic, Synchronous state machine design, Sequential MSI circuits.

- **Learning Objectives:**

- Understand the logical thinking and general concepts related to digital systems
- Analyze typical designs of digital system: combinational logic circuit and sequential logic circuit
- Using methods in designing to develop some typical applications of digital logic circuits
- Have ability to engage life-long learning
- Learn Simple digital circuits in preparation for computer engineering and science (optional)

Lecture Objectives

- Explain the basic differences between digital and analog quantities
- Show how voltage levels are used to represent digital quantities
- Describe various parameters of a pulse waveform such as rise time, fall time, pulse width, frequency, period, and duty cycle
- Explain the basic logic functions of NOT, AND, and OR
- Describe several types of logic operations and explain their application in an example system
- Describe programmable logic, discuss the various types, and describe how PLDs are programmed

I. Digital and Analog Quantities

Electronic circuits can be divided into two broad categories, digital and analog.

Digital electronics involves quantities with discrete values, and analog electronics involves quantities with continuous

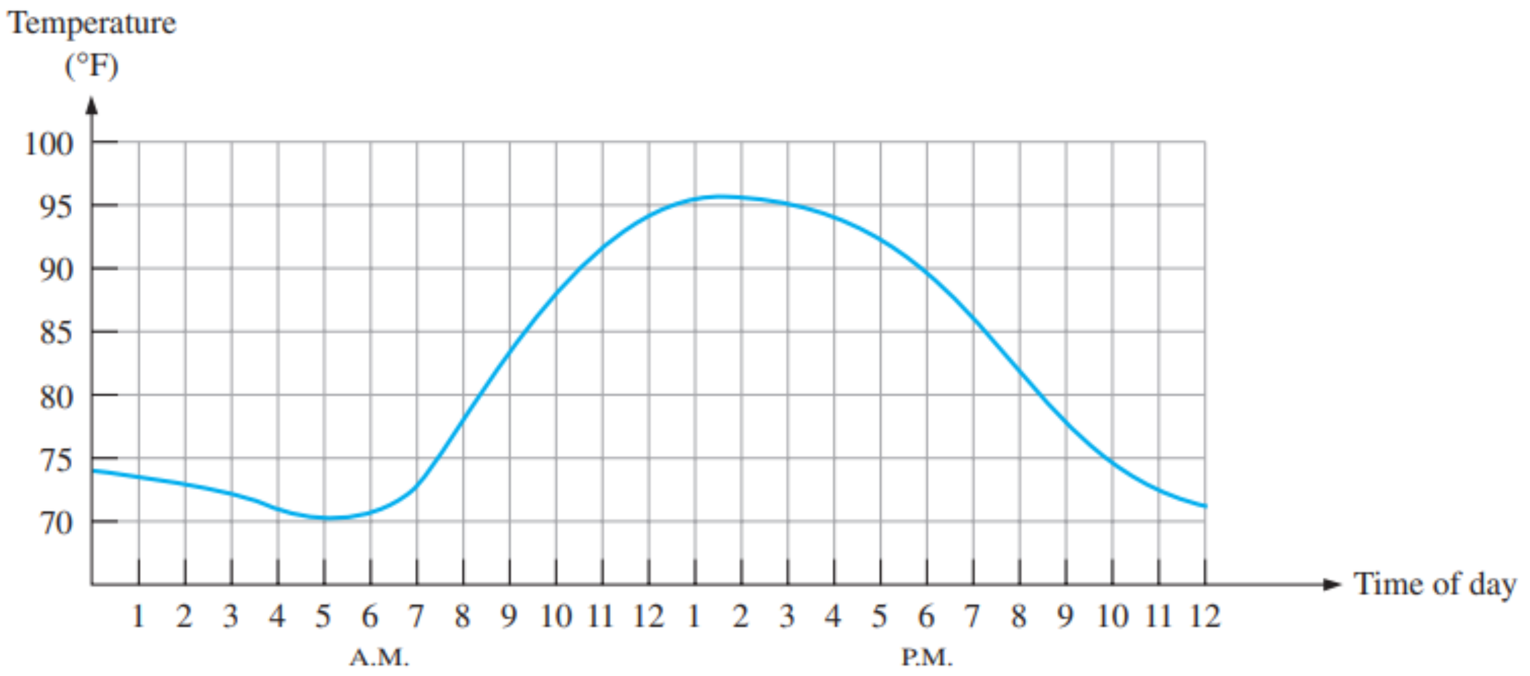


FIGURE 1–1 Graph of an analog quantity (temperature versus time).

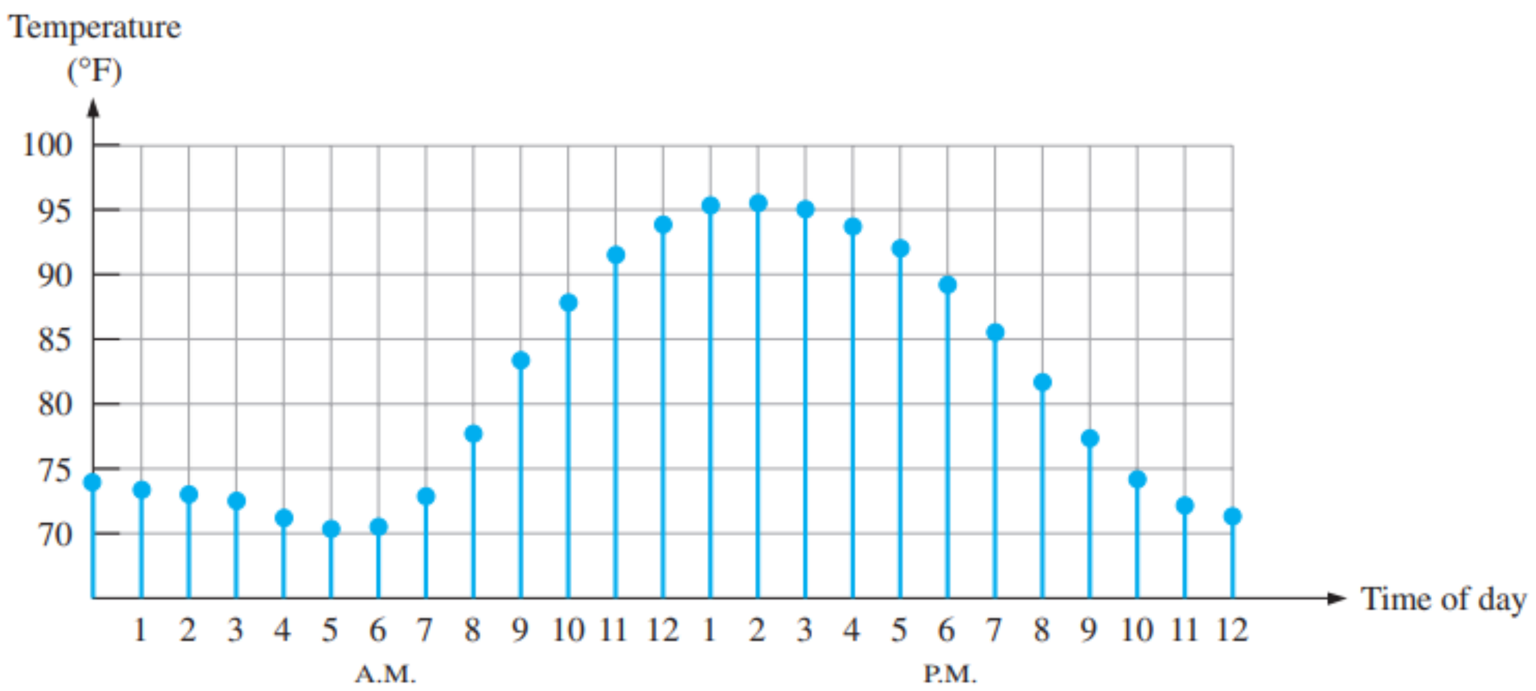


FIGURE 1–2 Sampled-value representation (quantization) of the analog quantity in Figure 1–1. Each value represented by a dot can be digitized by representing it as a digital code that consists of a series of 1s and 0s.

=> You have effectively converted an analog quantity to a form that can now be digitized by representing each sampled value by a digital code.

The Digital Advantage

An Analog System

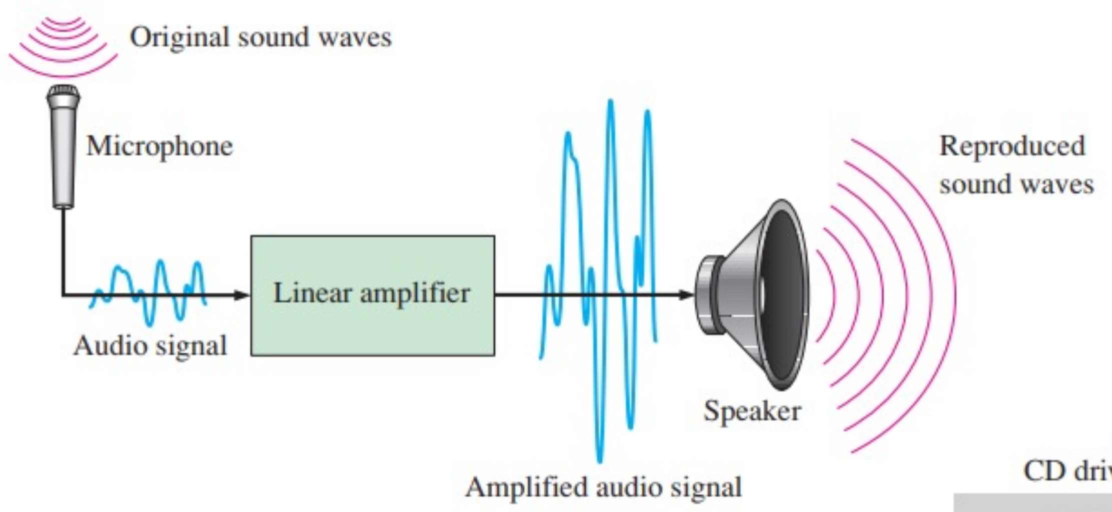


FIGURE 1–3 A basic audio public address system.

A System Using Digital and Analog Methods

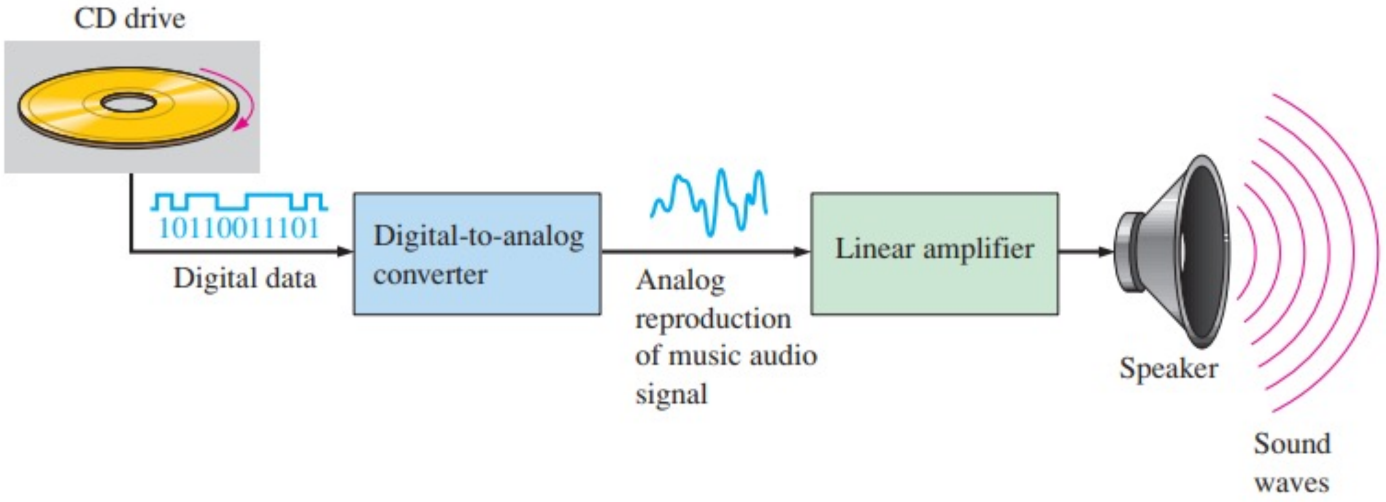


FIGURE 1–4 Basic block diagram of a CD player. Only one channel is shown.

The Digital Advantage

Mechatronics

Both digital and analog electronics are used in the control of various mechanical systems. The interdisciplinary field that comprises both mechanical and electronic components is known as ***mechatronics.***

FIGURE 1–5 Example of a mechatronic system and application. Part (b) Beawolf/Fotolia; Part (c) Small Town Studio/Fotolia.

2. Binary Digits, Logic Levels, and Digital Waveforms

Binary Digits

Each of the two digits in the binary system, 1 and 0, is called a bit, which is a contraction of the words binary digit.

In digital circuits, two different voltage levels are used to represent the two bits. Generally, 1 is represented by the higher voltage, which we will refer to as a HIGH, and a 0 is represented by the lower voltage level, which we will refer to as a LOW.

This is called **positive logic** and will be used throughout the book.

HIGH = 1 and LOW = 0

Logic Levels

- The voltages used to represent a 1 and a 0 are called logic levels. Ideally, one voltage level represents a HIGH and another voltage level represents a LOW.
- In a practical digital circuit, however, a HIGH can be any voltage between a specified minimum value and a specified maximum value.
- Likewise, a LOW can be any voltage between a specified minimum and a specified maximum.
- There can be no overlap between the accepted range of HIGH levels and the accepted range of LOW levels

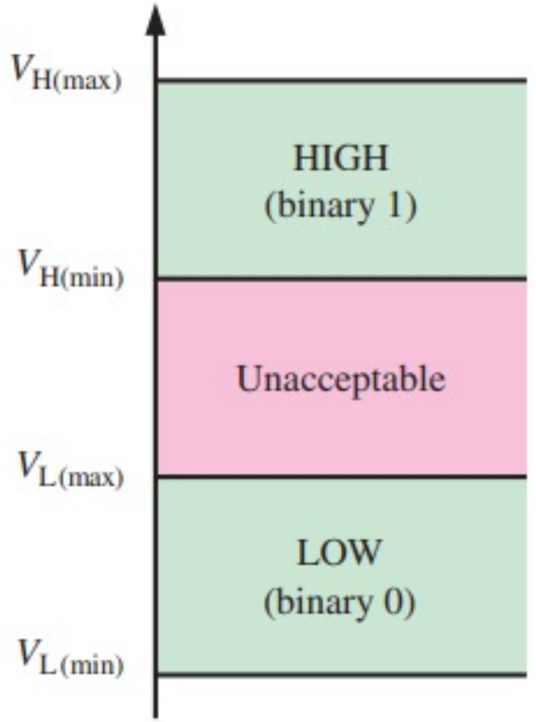


FIGURE 1–6 Logic level ranges of voltage for a digital circuit.

Digital Waveforms

- Digital waveforms consist of voltage levels that are changing back and forth between the HIGH and LOW levels or states.
- A digital waveform is made up of a series of pulses.

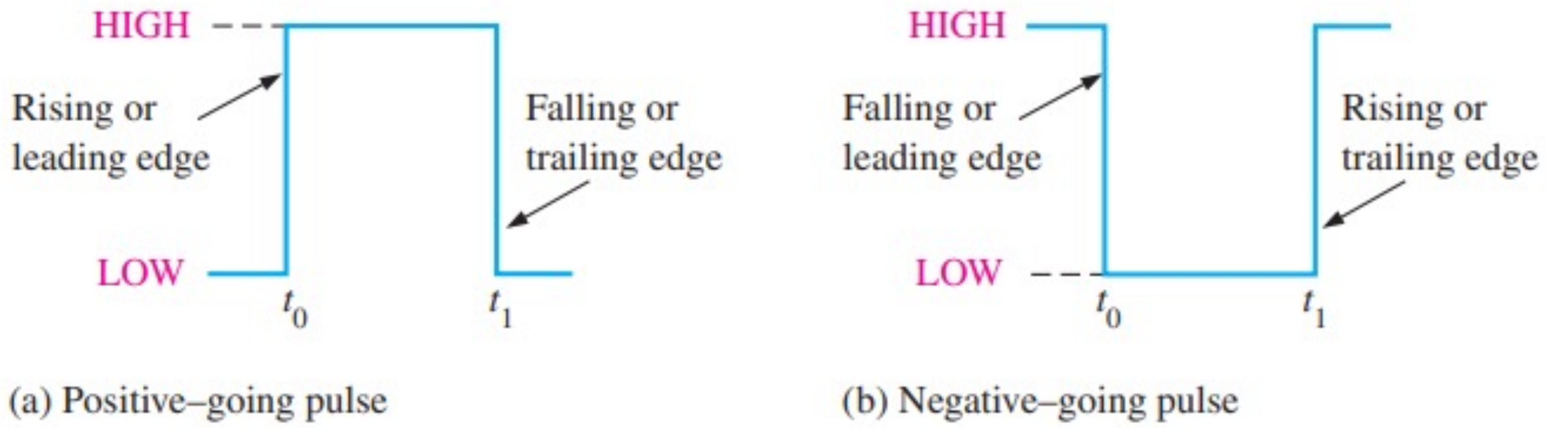


FIGURE 1–7 Ideal pulses.

The Pulse

A pulse has two edges: a leading edge that occurs first at time t_0 and a trailing edge that occurs last at time t_1. For a positive-going pulse, the leading edge is a **rising edge**, and the trailing edge is a **falling edge**.

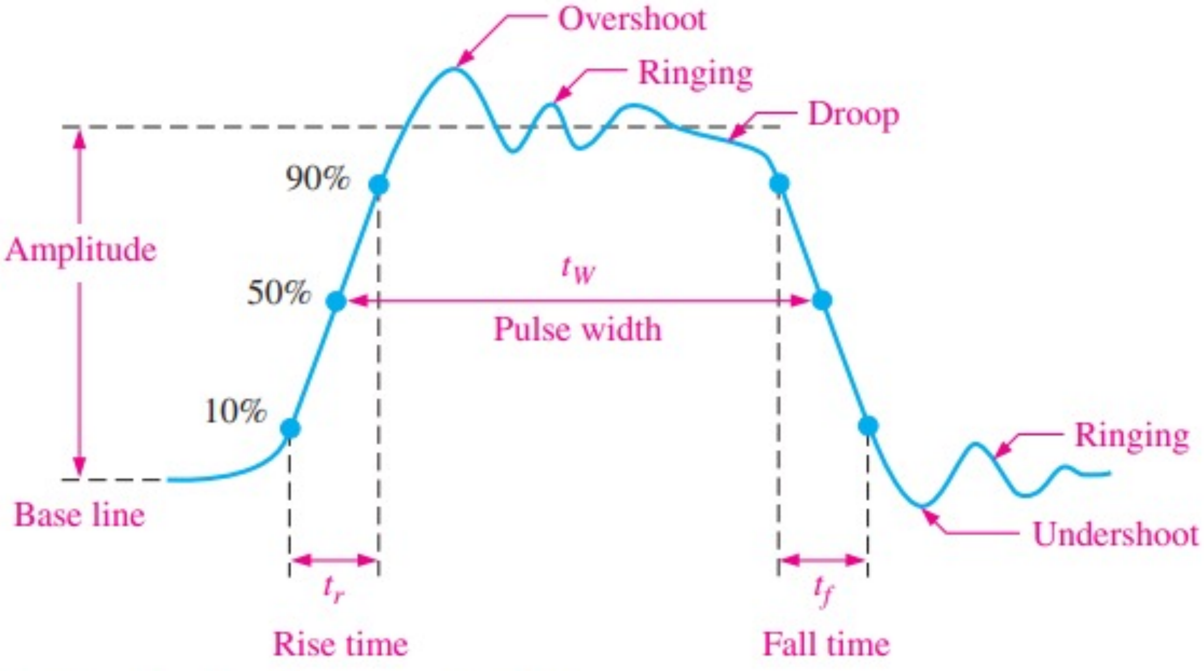


FIGURE 1–8 Nonideal pulse characteristics.

The time required for a pulse to go from its LOW level to its HIGH level is called the **rise time** (***t_r***), and the time required for the transition from the HIGH level to the LOW level is called the **fall time** (***t_f***). The **pulse width** (***t_W***) is a measure of the duration of the pulse and is often defined as the time interval between the 50% points on the rising and falling edges

Waveform Characteristics

- A periodic pulse waveform is one that repeats itself at a fixed interval, called a period (T). The frequency (f) is the rate at which it repeats itself and is measured in hertz (Hz).
- A non-periodic pulse waveform, of course, does not repeat itself at fixed intervals and may be composed of pulses of randomly differing pulse widths and/or randomly differing time intervals between the pulses.

(a) Periodic (square wave) (b) Nonperiodic

FIGURE 1–9 Examples of digital waveforms.

- An important characteristic of a periodic digital waveform is its duty cycle, which is the ratio of the pulse width (t_W) to the period (T). It can be expressed as a percentage.

$$\textbf{Duty cycle} = \left(\frac{t_W}{T}\right)100\%$$

EXAMPLE 1–1

A portion of a periodic digital waveform is shown in Figure 1–10. The measurements are in milliseconds. Determine the following:

(a) period **(b)** frequency **(c)** duty cycle

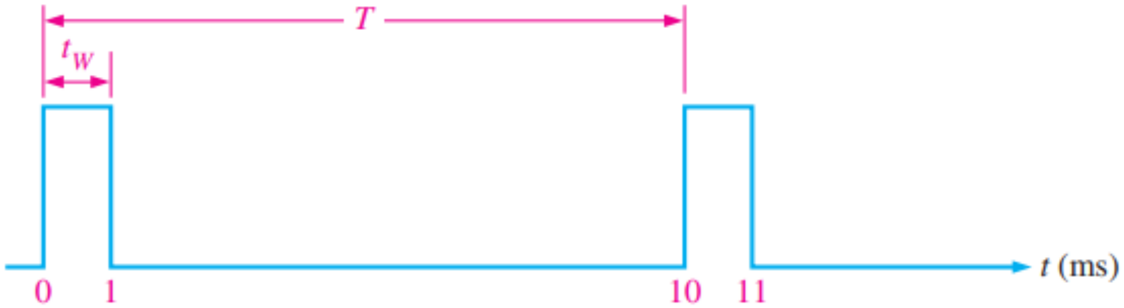


FIGURE 1–10

Solution

(a) The period (T) is measured from the edge of one pulse to the corresponding edge of the next pulse. In this case T is measured from leading edge to leading edge, as indicated. T equals **10 ms**.

(b) $$f = \frac{1}{T} = \frac{1}{10\text{ ms}} = \mathbf{100\ Hz}$$

(c) $$\text{Duty cycle} = \left(\frac{t_W}{T}\right)100\% = \left(\frac{1\text{ ms}}{10\text{ ms}}\right)100\% = \mathbf{10\%}$$

Related Problem*

A periodic digital waveform has a pulse width of 25 μs and a period of 150 μs. Determine the frequency and the duty cycle.

A Digital Waveform Carries Binary Information

Binary information that is handled by digital systems appears as waveforms that represent sequences of bits. When the waveform is **HIGH**, a binary 1 is present; when the waveform is **LOW**, a binary 0 is present. Each bit in a sequence occupies a defined time interval called a bit time.

The Clock

- In digital systems, all waveforms are synchronized with a basic timing waveform called **the clock**.
- The clock is a periodic waveform in which each interval between pulses (the period) equals the time for one bit.

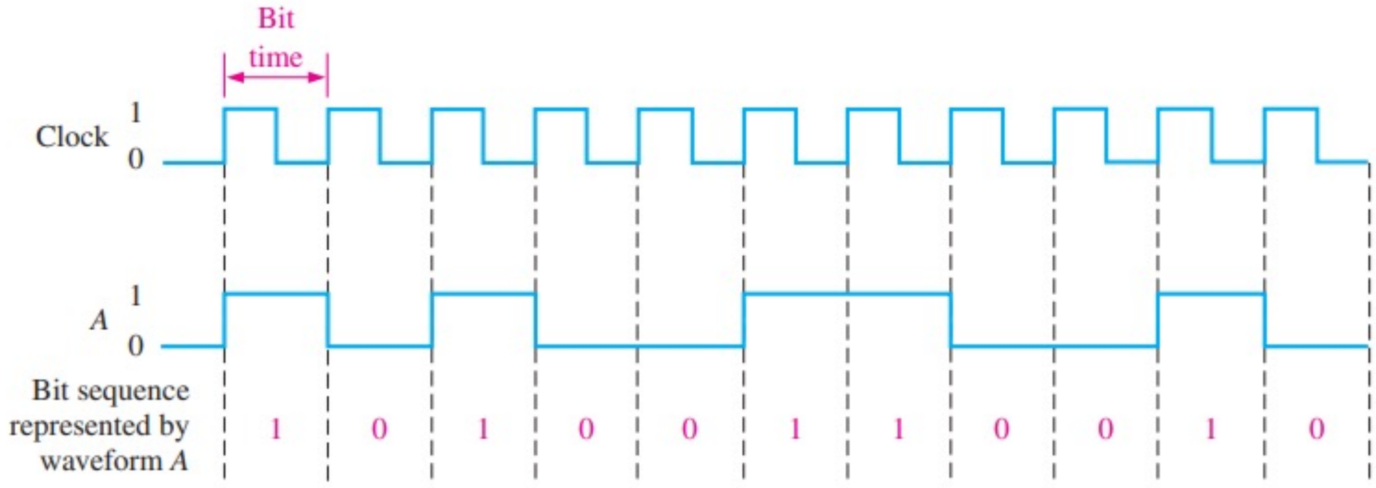


FIGURE 1–11 Example of a clock waveform synchronized with a waveform representation of a sequence of bits.

Timing Diagrams

- A **timing diagram** is a graph of digital waveforms showing the actual time relationship of two or more waveforms and how each waveform changes in relation to the others.
- By looking at a timing diagram, you can determine the states (HIGH or LOW) of all the waveforms at any specified point in time and the exact time that a waveform changes state relative to the other waveforms.

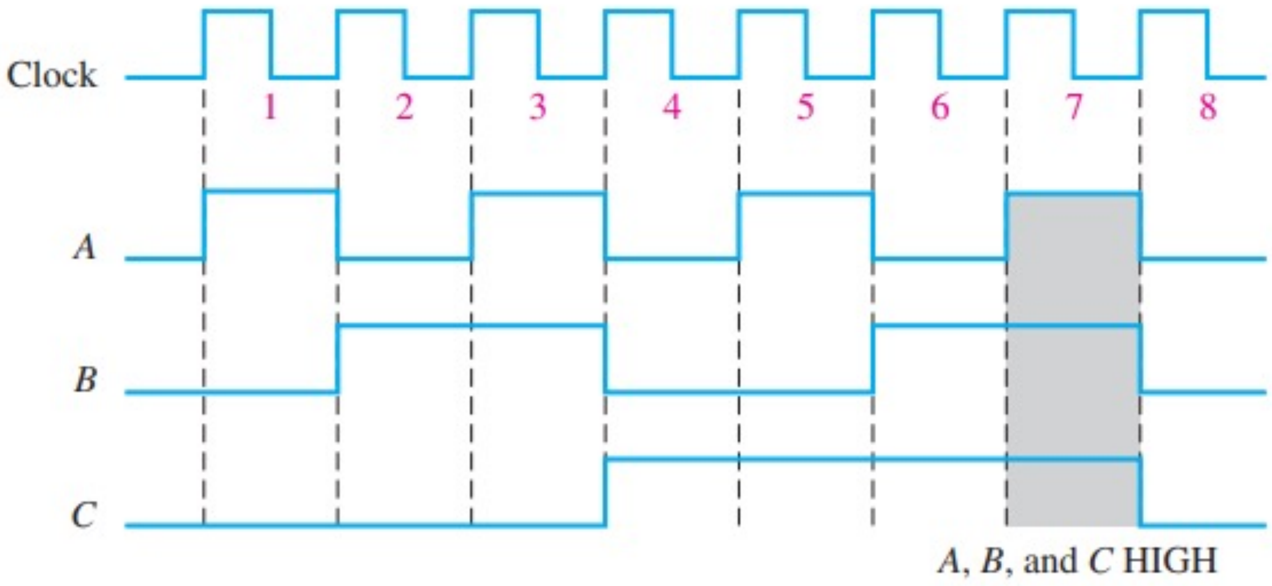
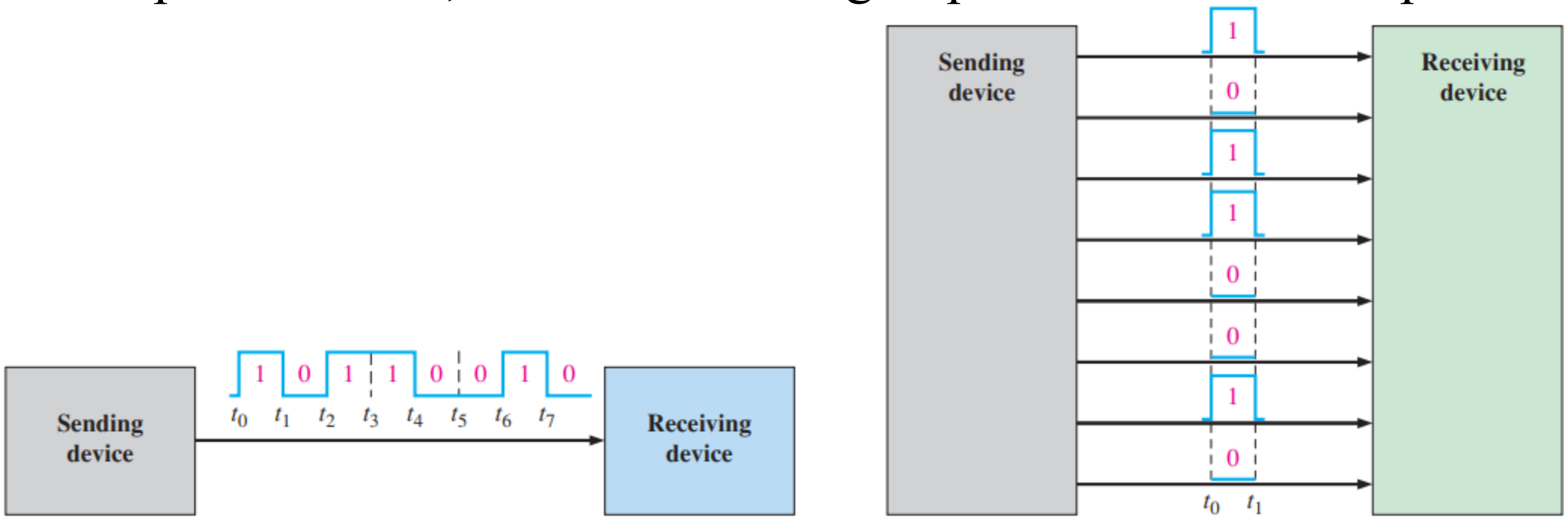


FIGURE 1–12 Example of a timing diagram.

Data Transfer

- Data refers to groups of bits that convey some type of information. Binary data, which are represented by digital waveforms, must be transferred from one device to another within a digital system or from one system to another in order to accomplish a given purpose.
- When bits are transferred in serial form from one point to another, they are sent one bit at a time along a single line
- When bits are transferred in parallel form, all the bits in a group are sent out on separate lines at the same time.



(a) Serial transfer of 8 bits of binary data. Interval t_0 to t_1 is first.

(b) Parallel transfer of 8 bits of binary data. The beginning time is t_0.

FIGURE 1–13 Illustration of serial and parallel transfer of binary data. Only the data lines are shown.

EXAMPLE 1–2

(a) Determine the total time required to serially transfer the eight bits contained in waveform A of Figure 1–14, and indicate the sequence of bits. The left-most bit is the first to be transferred. The 1 MHz clock is used as reference.

(b) What is the total time to transfer the same eight bits in parallel?

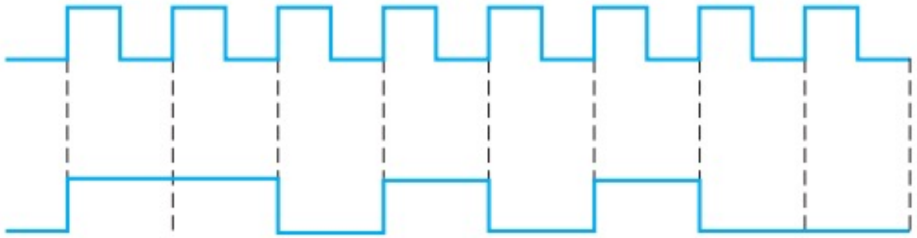


FIGURE 1–14

Solution

(a) Since the frequency of the clock is 1 MHz, the period is

$$T = \frac{1}{f} = \frac{1}{1\text{ MHz}} = 1\ \mu\text{s}$$

It takes 1 μs to transfer each bit in the waveform. The total transfer time for 8 bits is

$$8 \times 1\ \mu\text{s} = \mathbf{8\ \mu s}$$

To determine the sequence of bits, examine the waveform in Figure 1–14 during each bit time. If waveform A is HIGH during the bit time, a 1 is transferred. If waveform A is LOW during the bit time, a 0 is transferred. The bit sequence is illustrated in Figure 1–15. The left-most bit is the first to be transferred.

FIGURE 1–15

(b) A parallel transfer would take **1 μs** for all eight bits.

Related Problem

If binary data are transferred on a USB at the rate of 480 million bits per second (480 Mbps), how long will it take to serially transfer 16 bits?

3. Basic Logic Functions

NOT

- The **NOT** function changes one logic level to the opposite logic level, as indicated in Figure 1–17.
- When the input is High (1), the output is Low (0). When the input is Low, the output is High.
- In either case, the output is *not* the same as the input. The Not function is implemented by a logic circuit known as an **inverter**.

FIGURE 1–17 The NOT function.

AND

- AND function produces a HIGH output only when all the inputs are HIGH, as indicated in Figure 1–18 for the case of two inputs.
- When one input is HIGH and the other input is HIGH, the output is HIGH. When any or all inputs are LOW, the output is LOW.
- The AND function is implemented by a logic circuit known as an AND gate.

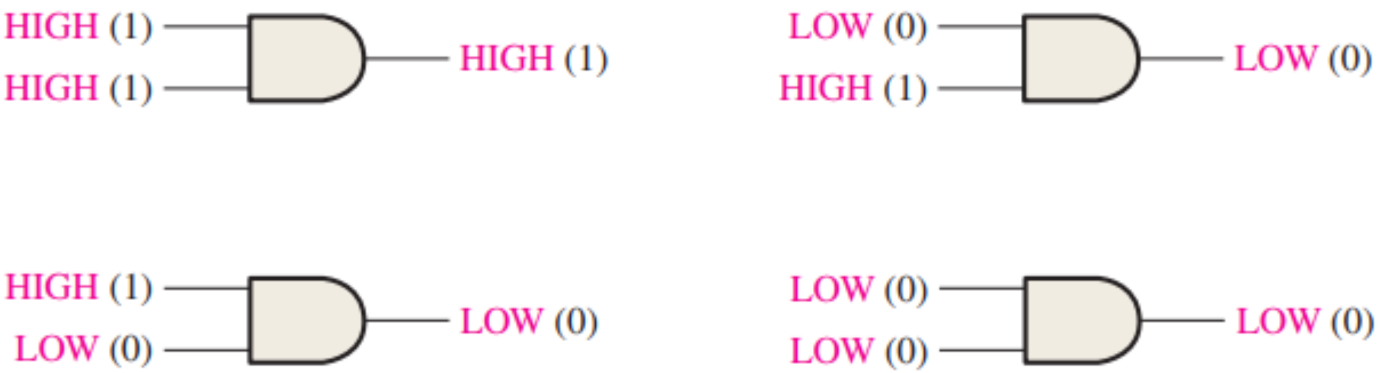


FIGURE 1–18 The AND function.

OR

- The OR function produces a HIGH output when one or more inputs are HIGH, as indicated in Figure 1–19 for the case of two inputs.
- When one input is HIGH or the other input is HIGH or both inputs are HIGH, the output is HIGH.
- When both inputs are LOW, the output is LOW.
- The OR function is implemented by a logic circuit known as an OR gate.

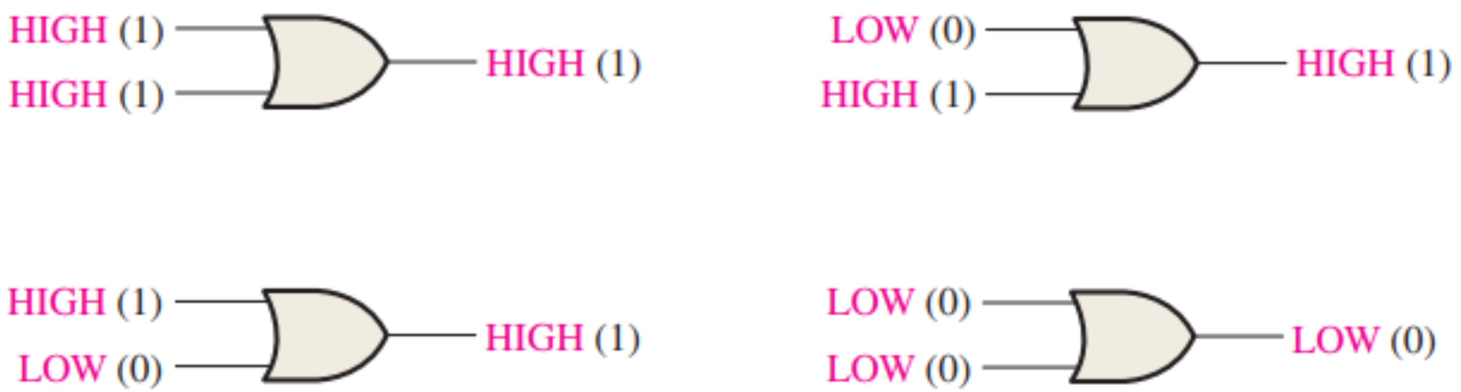
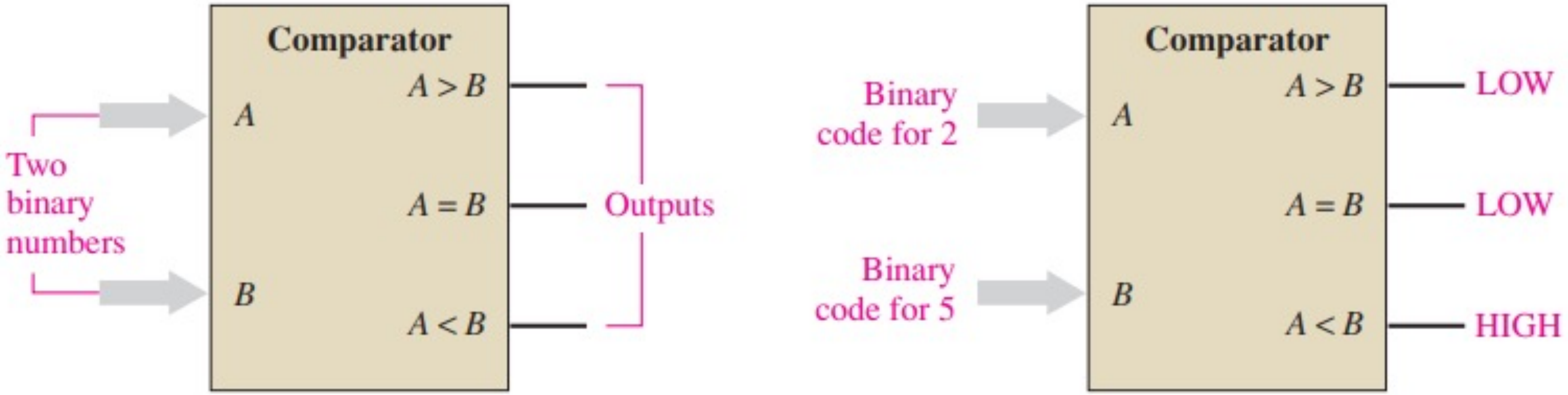


FIGURE 1–19 The OR function.

4. Combinational and Sequential Logic Functions

The Comparison Function

- Magnitude comparison is performed by a logic circuit called a comparator.
- A comparator compares two quantities and indicates whether or not they are equal.



(a) Basic magnitude comparator

(b) Example: A is less than B ($2 < 5$) as indicated by the HIGH output ($A < B$)

FIGURE 1–20 The comparison function.

The Arithmetic Functions

Addition

- Addition is performed by a logic circuit called an **adder**.
- An adder adds two binary numbers (on inputs A and B with a carry input C_{in}) and generates a sum ($\sum$) and a carry output (C_{out})

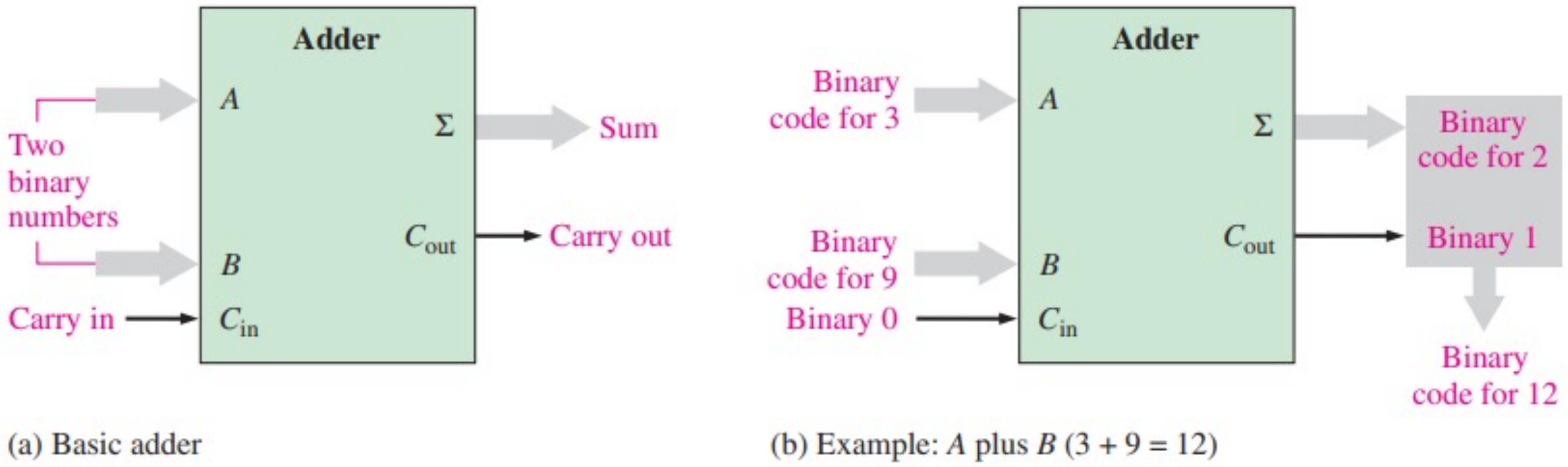


FIGURE 1–21 The addition function.

Subtraction

Subtraction is also performed by a logic circuit. A subtracter requires three inputs: the two numbers that are to be subtracted and a borrow input. The two outputs are the difference and the borrow output.

Multiplication

Multiplication is performed by a logic circuit called a multiplier. Numbers are always multiplied two at a time, so two inputs are required. The output of the multiplier is the product.

Division

Division can be performed with a series of subtractions, comparisons, and shifts, and thus it can also be done using an adder in conjunction with other circuits. Two inputs to the divider are required, and the outputs generated are the quotient and the remainder.

The Code Conversion Function

A code is a set of bits arranged in a unique pattern and used to represent specified information. A code converter changes one form of coded information into another coded form. Examples are conversion between binary and other codes such as the binary coded decimal (BCD) and the Gray code.

The Encoding Function

The encoding function is performed by a logic circuit called an encoder. The encoder converts information, such as a decimal number or an alphabetic character, into some coded form.

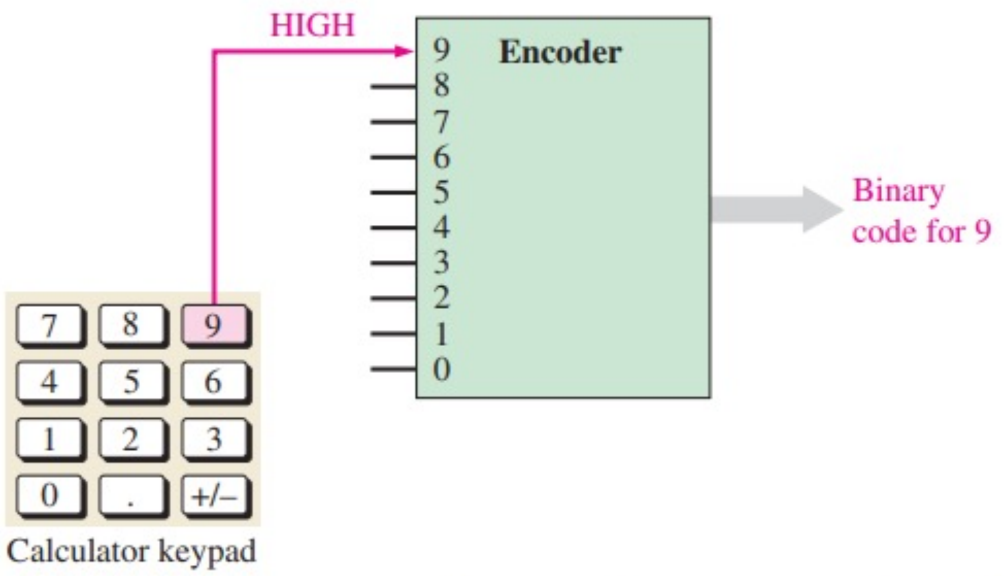


FIGURE 1–22 An encoder used to encode a calculator keystroke into a binary code for storage or for calculation.

The Decoding Function

- The decoding function is performed by a logic circuit called a decoder.
- The decoder converts coded information, such as a binary number, into a noncoded form, such as a decimal form.
- For example, one particular type of decoder converts a 4-bit binary code into the appropriate decimal digit.

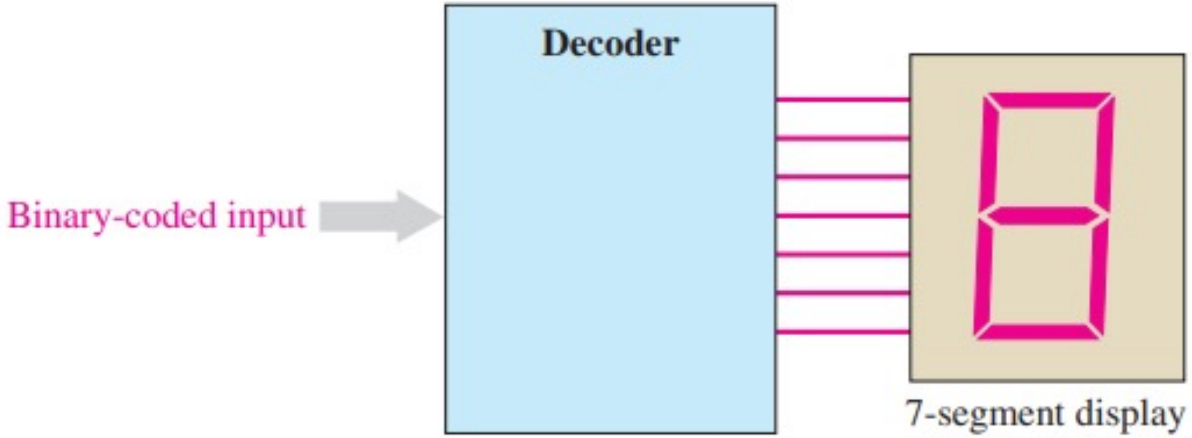


FIGURE 1–23 A decoder used to convert a special binary code into a 7-segment decimal readout.

The Data Selection Function

- Two types of circuits that select data are the multiplexer and the demultiplexer.
- The multiplexer, or mux for short, is a logic circuit that switches digital data from several input lines onto a single output line in a specified time sequence.
- The demultiplexer (demux) is a logic circuit that switches digital data from one input line to several output lines in a specified time sequence.

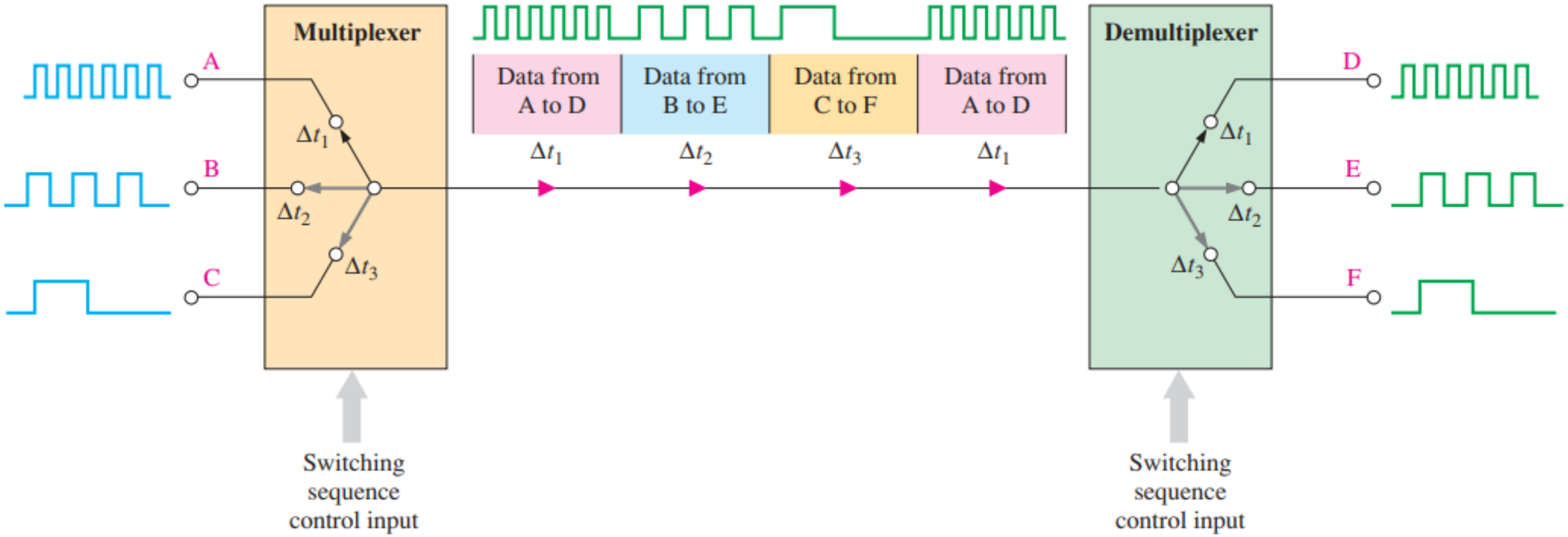


FIGURE 1–24 Illustration of a basic multiplexing/demultiplexing application.

The Storage Function

- Storage is a function that is required in most digital systems, and its purpose is to retain binary data for a period of time.
- Some storage devices are used for short-term storage and some are used for long-term storage.
- A storage device can “memorize” a bit or a group of bits and retain the information as long as necessary.
- Common types of storage devices are flip-flops, registers, semiconductor memories, magnetic disks, magnetic tape, and optical disks (CDs).

Flip-flops

- A flip-flop is a bistable (two stable states) logic circuit that can store only one bit at a time, either a 1 or a 0.
- The output of a flip-flop indicates which bit it is storing.

Registers

- A register is formed by combining several flip-flops so that groups of bits can be stored.
- The two basic types of shift registers are serial and parallel.

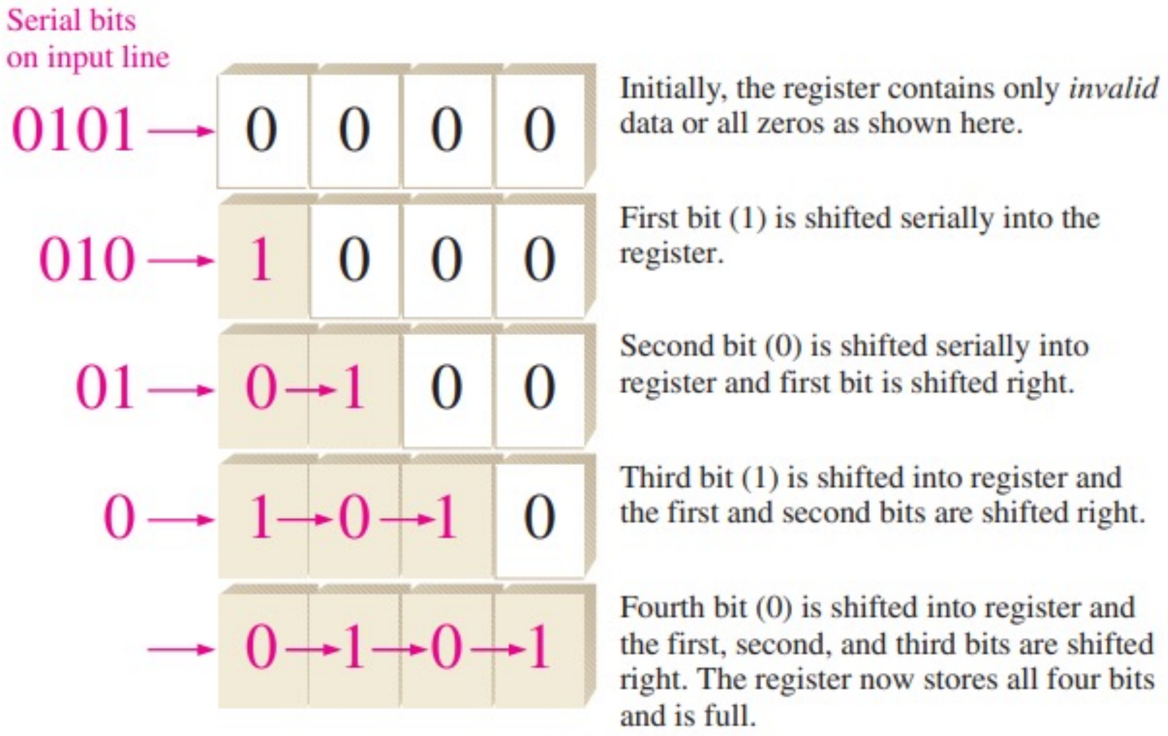


FIGURE 1–25 Example of the operation of a 4-bit serial shift register. Each block represents one storage "cell" or flip-flop.

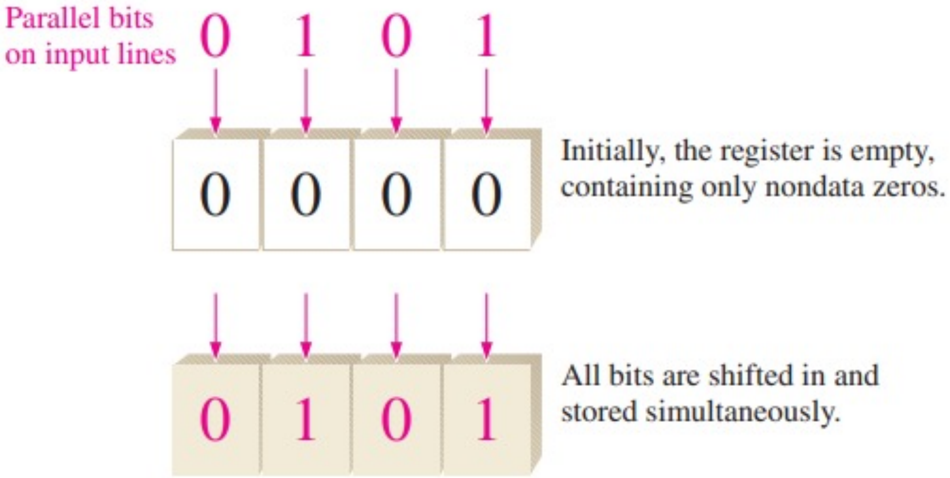


FIGURE 1–26 Example of the operation of a 4-bit parallel shift register.

The Counting Function

- The counting function is important in digital systems.
- There are many types of digital counters, but their basic purpose is to count events represented by changing levels or pulses.
- To count, the counter must "remember" the present number so that it can go to the next proper number in sequence.
- Therefore, storage capability is an important characteristic of all counters, and flip-flops are generally used to implement them.

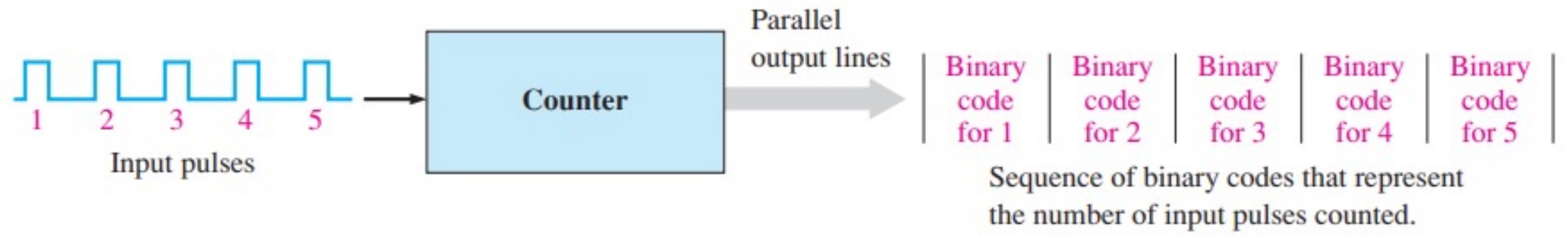
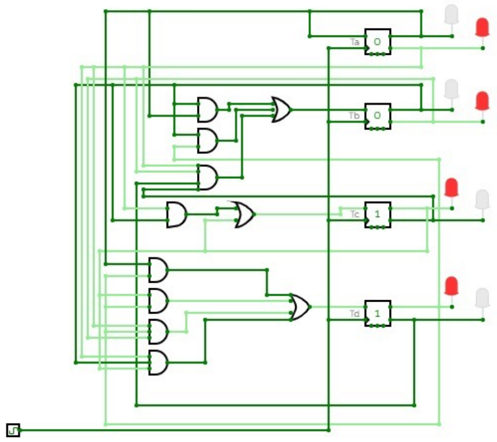


FIGURE 1–27 Illustration of basic counter operation.

THE END

Lecture 1: Introductory Concepts



INSTRUCTOR: Dr. Vuong Quoc Bao