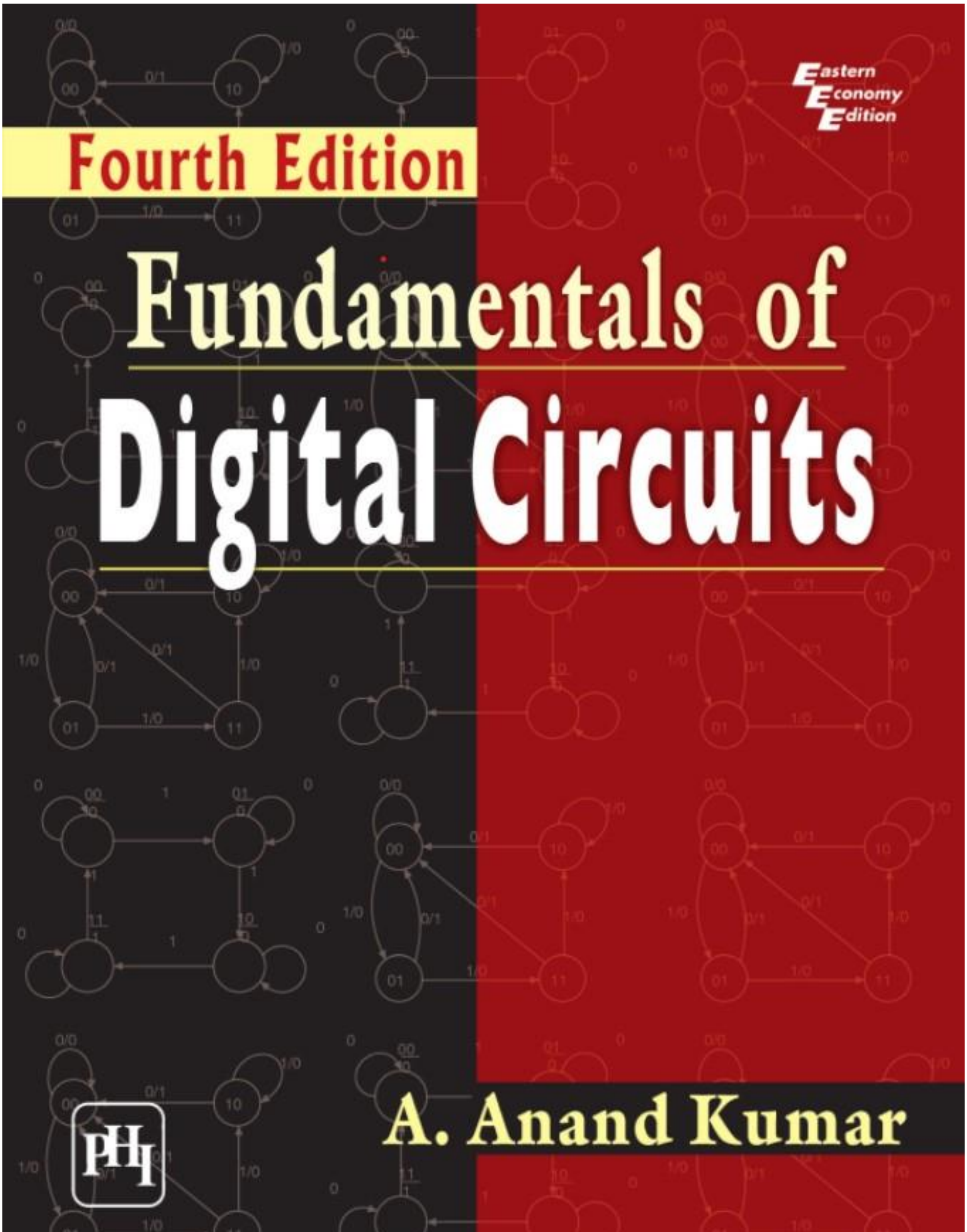

Eastern Economy Edition
Fourth Edition
Fundamentals of
Digital Circuits
A. Anand Kumar
PHI

1

INTRODUCTION

1.1 DIGITAL AND ANALOG SYSTEMS

Electronic circuits and systems are of two kinds—*analog* and *digital*. The distinction between them is not so much in the types of semiconductor devices used in these circuits as it is in voltage and current variations that occur when each type of circuit performs the function for which it is designed. Analog circuits are those in which voltages and currents vary continuously through the given range. They can take infinite values within the specified range. For example, the output voltage from an audio amplifier might be any one of the infinite values between –10 V and +10 V at any particular instant of time. Other examples of analog devices include signal generators, radio frequency transmitters and receivers, power supplies, electric motors and speed controllers, and many analog type instruments—those having *pointers* that move in a continuous arc across a calibrated scale. By contrast, a digital circuit is one in which the voltage levels assume a finite number of distinct values. In virtually all modern digital circuits, there are just two discrete voltage levels. However, each voltage level in a practical digital system can actually be a narrow *band* or *range* of voltages.

Digital circuits are often called switching circuits, because the voltage levels in a digital circuit are assumed to be switched from one value to another instantaneously, that is, the transition time is assumed to be zero.

Switching circuits may be combinational switching circuits or sequential switching circuits. In combinational switching circuits, the output depends only on the present inputs, whereas in sequential switching circuits, the output depends on the present inputs as well as the present state of the circuit, i.e. on the past inputs also. In other words we can say that sequential circuits have memory and combinational circuits have no memory. In fact sequential circuits are nothing but combinational circuits with memory.

Sequential switching circuits may be synchronous sequential circuits or asynchronous sequential circuits. In synchronous sequential switching circuits, state transitions can take place only when the inputs are applied along with a clock pulse, whereas in asynchronous sequential circuits state transitions can take place any time the inputs are applied.

Digital circuits are also called *logic* circuits, because each type of digital circuit obeys a certain set of logic rules. The manner in which a logic circuit responds to an input is referred to as the circuit's logic.

Digital systems are used extensively in computation and data processing, control systems, communications and measurement. Digital systems have a number of advantages over analog systems. Many tasks formally done by analog systems are now being performed digitally. The chief reasons for the shift to digital technology are summarized below:

Digital systems are easier to design: The switching circuits in which there are only two voltage levels, HIGH and LOW, are easier to design. The exact numerical values of voltages are not important because they have only logical significance; only the range in which they fall is important. In analog systems, signals have numerical significance; so, their design is more involved.

Information storage is easy: There are many types of semiconductor and magnetic memories of large capacity which can store digital data for periods as long as necessary.

Accuracy and precision are greater: Digital systems are much more accurate and precise than analog systems, because digital systems can be easily expanded to handle more digits by adding more switching circuits. Analog systems will be quite complex and costly for the same accuracy and precision.

Digital systems are more versatile: It is fairly easy to design digital systems whose operation is controlled by a set of stored instructions called the *program*. Any time the system operation is to be changed, it can easily be accomplished by modifying the program. Even though analog systems can also be programmed, the variety of the available operations is severely limited.

Digital circuits are less affected by noise: Unwanted electrical signals are called noise. Noise is unavoidable in any system. Since in analog systems the exact values of voltages are important and in digital systems only the range of values is important, the effect of noise is more severe in analog systems. In digital systems, noise is not critical as long as it is not large enough to prevent us from distinguishing a HIGH from a LOW.

More digital circuitry can be fabricated on IC chips: The fabrication of digital ICs is simpler and economical than that of analog ICs. Moreover, higher densities of integration can be achieved in digital ICs than in analog ICs, because digital design does not require high value capacitors, precision resistors, inductors and transformers (which cannot be integrated economically) like the analog design.

Reliability is more: Digital systems are more reliable than analog systems.

Limitations of digital techniques

Even though digital techniques have a number of advantages, they have only one major drawback.

THE REAL WORLD IS ANALOG

Most physical quantities are analog in nature, and it is these quantities that are often the inputs and outputs and continually monitored, operated and controlled by a system. When these quantities are processed and expressed digitally, we are really making a digital approximation to an inherently analog quantity. Instead of processing the analog information directly, it is first converted into digital form and then processed using digital techniques. The results of processing can be converted back to analog form for interpretation. Because of these conversions, the processing time increases and the system becomes more complex. In most cases, these disadvantages are outweighed by numerous advantages of digital techniques. However, there are situations where use of only analog techniques is simpler and more economical. Both the analog and digital techniques can be employed in the same system to advantage. Such systems are called *hybrid systems*. But the tendency today is towards employing digital systems because the economic benefits of integration are of overriding importance.

The design of digital systems may be roughly divided into three stages—SYSTEM DESIGN, LOGIC DESIGN, and CIRCUIT DESIGN. System design involves breaking the overall system into subsystems and specifying the characteristics of each subsystem. For example, the system design of a digital computer involves specifying the number and type of memory units, arithmetic units and input-output devices, as well as specifying the interconnection and control of these subsystems. Logic design involves determining how to interconnect basic logic building blocks to perform a specific function. An example of logic design is determining the interconnection of logic gates and flip-flops required to perform binary addition. Circuit design involves specifying the interconnection of specific components such as resistors, diodes and transistors to form a gate, flip-flop or any other logic building block. This book is largely devoted to a study of logic design and the theory necessary for understanding the logic design process.

1.2 LOGIC LEVELS AND PULSE WAVEFORMS

Digital systems use the binary number system. Therefore, two-state devices are used to represent the two binary digits 1 and 0 by two different voltage levels, called HIGH and LOW. If the HIGH voltage level is used to represent 1 and the LOW voltage level to represent 0, the system is called the *positive logic system*. On the other hand, if the HIGH voltage level represents 0 and the LOW voltage level represents 1, the system is called the *negative logic system*. Normally, the binary 0 and 1 are represented by the logic voltage levels 0 V and + 5 V. So, in a positive logic system, 1 is represented by + 5 V (HIGH) and 0 is represented by 0 V (LOW); and in a negative logic system, 0 is represented by + 5 V (HIGH) and 1 is represented by 0 V (LOW). Both positive and negative logics are used in digital systems, but the positive logic is more common. For this reason, we will use only the positive logic system in this book.

In reality, because of circuit variations, the 0 and 1 would be represented by voltage ranges instead of particular voltage levels. Usually, any voltage between 0 V and 0.8 V represents the logic 0 and any voltage between 2 V and 5 V represents the logic 1. Normally, all input and output signals fall within one of these ranges except during transition from one level to another. The range between 0.8 V and 2 V is called the *indeterminate range*. If the signal falls between 0.8 V and 2 V, the response is not predictable.

Digital circuits are designed to respond predictably to input voltages that are within the specified range. That means, the exact values of voltages are not important and the circuit gives the same response for all input voltages in the allowed range, i.e. a voltage of 0 V gives the same response as a voltage of 0.4 V or 0.6 V or 0.8 V. Similarly, a voltage of 2 V gives the same response as a voltage of 2.8 V or 3.6 V or 4.7 V or 5 V.

In digital circuits and systems, the voltage levels are normally changing back and forth between the HIGH and LOW states. So, pulses are very important in their operation. A pulse may be a positive pulse or a negative pulse. A single positive pulse is generated when a normally LOW voltage goes to its HIGH level and then returns to its normal LOW level as shown in Figure 1.1a. A single negative pulse is generated when a normally HIGH voltage goes to its LOW level and then returns to its normal HIGH level as shown in Figure 1.1b.

Figure 1.1 Ideal positive and negative pulses.

As indicated in Figure 1.1, a pulse has two edges: a leading edge and a trailing edge. For a positive pulse, the leading edge is a positive going transition (PGT or rising edge) and the trailing edge is a negative going transition (NGT or falling edge), whereas for a negative pulse, the leading edge is a negative going transition (NGT) and the trailing edge is a positive going transition (PGT). The pulses shown in Figure 1.1 are ideal, because the rising and falling edges change instantaneously, i.e. in zero time. Practical pulses do not change instantaneously from LOW to HIGH or from HIGH to LOW.

A non-ideal pulse is shown in Figure 1.2. It has finite rise and fall times. The time taken by the pulse to rise from LOW to HIGH is called the *rise time* and the time taken by the pulse to go from HIGH to LOW is called the *fall time*. Because of the nonlinearities that commonly occur at the bottom and top of the pulse, the rise time is defined as the time taken by the pulse to rise from 10% to 90% of the pulse amplitude, and the fall time is defined as the time taken by the pulse to fall from 90% to 10% of the pulse amplitude. The duration of the pulse is usually indicated by pulse width t_w which is defined as the time between the 50% points on the rising and falling edges.

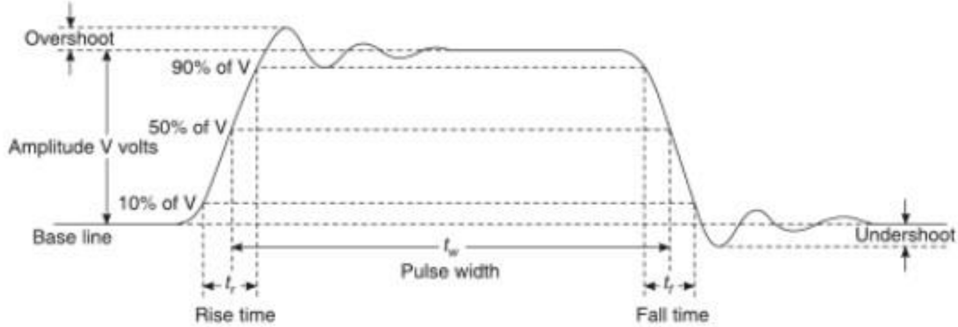


Figure 1.2 Non-ideal pulse characteristics.

Most waveforms encountered in digital systems are composed of a series of pulses and can be classified as periodic waveforms and non-periodic waveforms. A *periodic waveform* is one which repeats itself at regular intervals of time called the period, T. A *non-periodic* waveform, of course, does not repeat itself at regular intervals and may be composed of pulses of different widths and/or differing time intervals between the pulses. The reciprocal of the period is called the *frequency* of the periodic waveform. Another important characteristic of the periodic pulse waveform is its duty cycle which is defined as the ratio of the ON time (pulse width t_w) to the period of the pulse waveform. Thus,

$$f = \frac{1}{T} \text{ and duty cycle} = \frac{t_w}{T}$$

1.3 ELEMENTS OF DIGITAL LOGIC

In our daily life, we make many logic decisions. The term *logic* refers to something which can be reasoned out. In many situations, the problems and processes that we encounter can be expressed in the form of propositional or logic functions. Since these functions are true/false, yes/no statements, digital circuits with their two-state characteristics are extremely useful. Several logic statements when combined form logic functions. These logic functions can be formulated mathematically using Boolean algebra (which is a system of mathematical logic) and the minimal expression for the function can be obtained using minimization techniques. There are four basic logic elements using which any digital system can be built. They are the three basic gates—NOT, AND and OR gates, and a flip-flop. In fact, a flip-flop can be constructed using gates. So, we can say that any digital circuit can be constructed using only gates. In addition to the three basic gates, there are two universal gates called NAND and NOR gates. They are called universal gates, because any circuit of any complexity can be constructed using only NAND gates or only NOR gates. In addition, there are two more gates called X-OR and X-NOR gates. We will learn about the characteristics of these gates and flip-flops in the later chapters.

Using logic gates and flip-flops, more complex logic circuits like counters, shift registers, arithmetic circuits, comparators, encoders, decoders, code converters, multiplexers, demultiplexers, memories, etc. can be constructed. These more complex logic functions can then be combined to form complete digital systems to perform specified tasks.

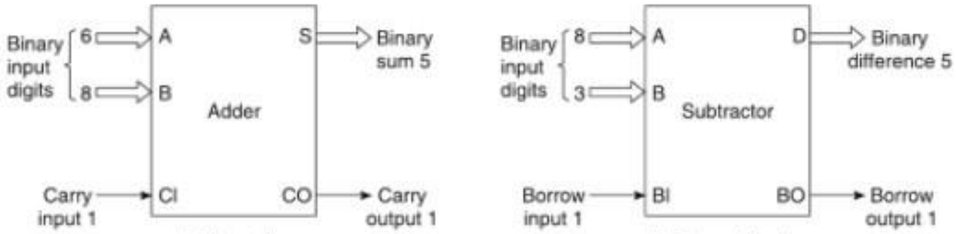
1.4 FUNCTIONS OF DIGITAL LOGIC

Many operations can be performed by combining logic gates and flip-flops. Some of the more common operations are arithmetic operations, comparison, code conversion, encoding, decoding, multiplexing, demultiplexing, shifting, counting and storing. These are all discussed thoroughly in the later chapters. The block diagram operation is given below.

1.4.1 Arithmetic Operations

The basic arithmetic operations are addition, subtraction, multiplication and division.

The arithmetic operation *addition* is performed by a digital logic circuit called the *adder*. Its function is to add two numbers *addend* (A) and *augend* (B) with a carry input (CI), and generate a sum term (S) and a carry output term (CO). Figure 1.3a is a block diagram of an adder. It illustrates the addition of the binary equivalents of 8 and 6 with a carry input of 1, which results in a binary sum term 5 and a carry output term 1.

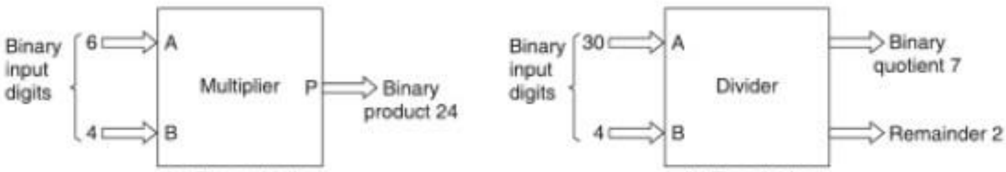


(a) The adder (b) The subtractor

Figure 1.3 The adder and the subtractor.

The arithmetic operation *subtraction* can be performed by a digital logic circuit called the *subtractor*. Its function is to subtract the subtrahend (A) from the minuend (B) considering the borrow input (BI) and to generate a difference term (D) and a borrow output term (BO). Since subtraction is equivalent to addition of a negative number, subtraction can be performed by using an adder. Figure 1.3b is a block diagram of a subtractor. It illustrates the subtraction of the binary equivalent of 8 from the binary equivalent of 3 with a borrow input of 1, which results in a binary difference term 5 and a borrow output term 1.

The arithmetic operation *multiplication* can be performed by a digital logic circuit called the *multiplier*. Its function is to multiply the *multiplicand* (A) by the *multiplier* (B) and generate the product term (P). Since multiplication is simply a series of additions with shifts in the positions of the partial products, it can be performed using an adder. Figure 1.4a is a block diagram of a multiplier. It illustrates the multiplication of 6 by 4, which results in the product term 24.



(a) The multiplier (b) The divider

Figure 1.4 The multiplier and the divider.

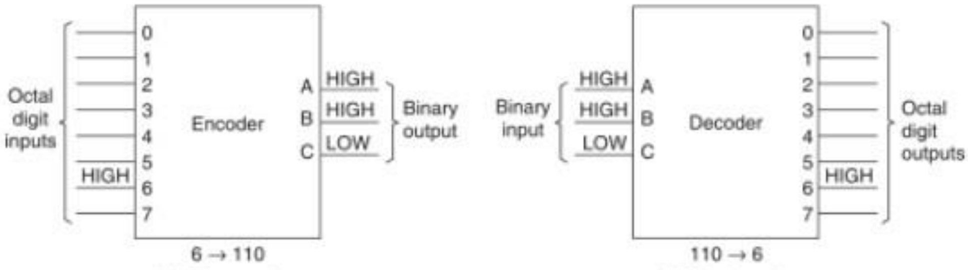
The arithmetic operation *division* can be performed by a digital logic circuit called the *divider*. Division can also be performed by an adder itself, since division involves a series of subtractions, comparisons and shifts. Its function is to divide the *dividend* (A) by the divisor (B) and generate a quotient term (Q) and a remainder term (R). Figure 1.4b is a block diagram of a divider. It illustrates the division of the binary equivalent of 30 by the binary equivalent of 4, which results in a binary quotient term 7 and a remainder term 2.

1.4.2 Encoding

Encoding is the process of converting a familiar number or symbol to some coded form. An *encoder* is a digital device that receives digits (decimal, octal, etc.), or alphabets, or special symbols and converts them to their respective binary codes. In the octal-to-binary encoder shown in Figure 1.5a, a HIGH level on a given input corresponding to a specific octal digit produces the appropriate 3-bit code (ABC) on the output levels. The figure illustrates encoding of the octal digit 6 to binary 110.

1.4.3 Decoding

Decoding is the inverse operation of encoding. A *decoder* converts binary-coded information (ABC) to unique outputs such as decimal, octal digits, etc. In the binary-to-octal decoder shown in Figure 1.5b, a combination of specific levels on the input lines produces a HIGH on the corresponding output line. The figure illustrates decoding of the binary 110 to octal digit 6.

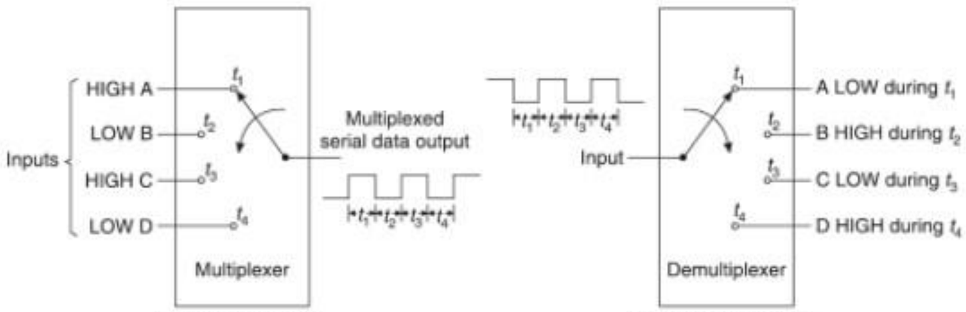


(a) The encoder (b) The decoder

Figure 1.5 The encoder and the decoder.

1.4.4 Multiplexing

Multiplexing means sharing. It is the process of switching information from several lines on to a single line in a specified sequence. A multiplexer or data selector is a logic circuit that accepts several data inputs and allows only one of them to get through to the output. It is an N-to-1 device. In the multiplexer shown in Figure 1.6a, if the switch is connected to input A for time t_1, to input B for time t_2, to input C for time t_3 and to input D for time t_4, the output of the multiplexer will be as shown in the figure. This figure illustrates a 4-to-1 multiplexer.



(a) The multiplexer (b) The demultiplexer

Figure 1.6 The multiplexer and the demultiplexer.

1.4.5 Demultiplexing

Demultiplexing operation is the inverse of multiplexing. Demultiplexing is the process of switching information from one input line on to several output lines. A demultiplexer is a digital circuit that takes a single input and distributes it over several outputs. It is a 1-to-N device. In the demultiplexer shown in Figure 1.6b, if the switch is connected to output A for time t_1, to output B for time t_2, to output C for time t_3 and to output D for time t_4, the output of the demultiplexer will be as shown in the figure. The figure illustrates a 1-to-4 demultiplexer.

1.4.6 Comparison

A logic circuit used to compare two quantities and give an output signal indicating whether the two input quantities are equal or not, and if not, which of them is greater, is called a *comparator*. Figure 1.7a shows the block diagram of a comparator. The binary representations of the quantities A and B to be compared are applied as inputs to the comparator. One of the outputs, A < B, A = B or A > B goes HIGH, depending on the magnitudes of the input quantities. The figure illustrates comparison of 8 and 4, and the result is HIGH (8 > 4).

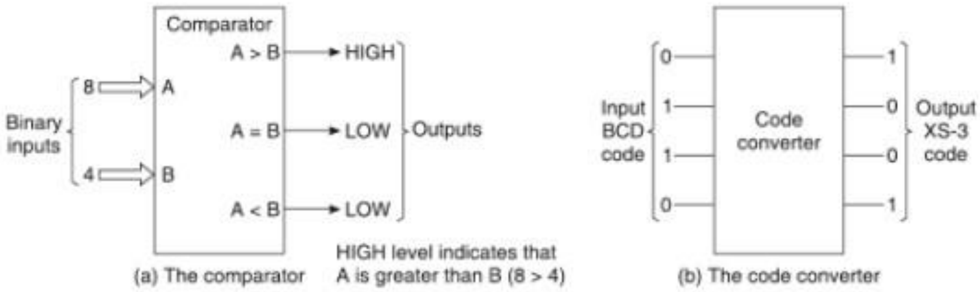


Figure 1.7 The comparator and the code converter.

1.4.7 Code Conversion

A logic circuit used to convert information coded in one form to another form is called a *code converter*. Figure 1.7b shows the block diagram of a BCD to XS-3 code converter. The figure illustrates conversion of decimal digit 6 coded as 0110 in 8421 BCD form to 1001 in XS-3 form.

1.4.8 Storage

Storage and shifting of information is very essential in digital systems. Digital circuits used for temporary storage and shifting of information (data), are called *registers*. Registers are made up of storage elements called flip-flops. Figure 1.8a shows the shifting or loading of data into a register made up of four flip-flops. After each clock pulse, the input bit is shifted into the first flip-flop and the content of each flip-flop is shifted to the flip-flop to its right. Figure 1.8b shows the shifting out of data from the register. The content of the last flip-flop is shifted out and lost.

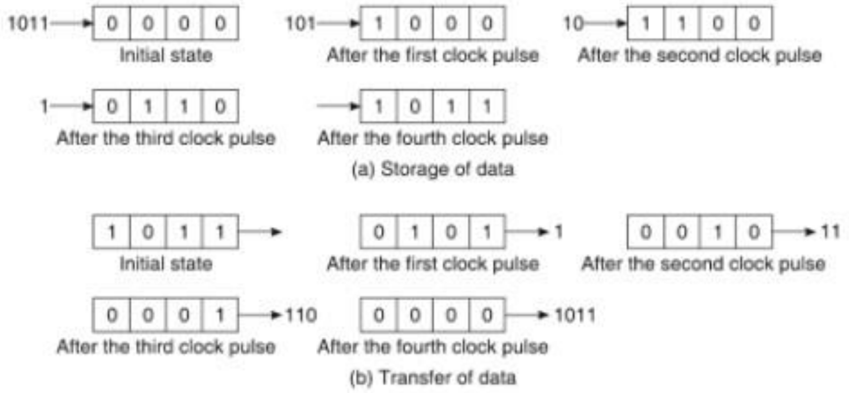


Figure 1.8 Storage and transfer of data.

1.4.9 Counting

The counting operation is very important in digital systems. A logic circuit used to count the number of pulses inputted to it, is called a *counter*. The pulses may represent some events. In order to count, the counter must remember the present number, so that it can go to the next proper number in the sequence when the next pulse comes. So, storage elements, i.e. flip-flops are used to build counters too. Figure 1.9a shows the block diagram of a counter.

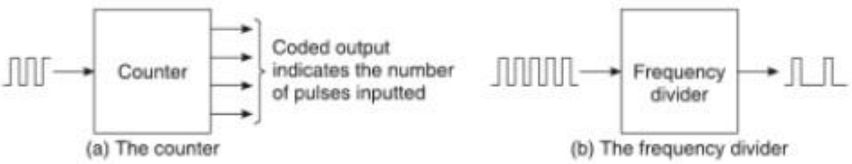


Figure 1.9 The counter and the frequency divider.

1.4.10 Frequency Division

A counter can also be used to perform *frequency division*. To divide a signal of frequency f by N, the signal is applied to a mod-N counter. The output of the counter will be of frequency f/N. Figure 1.9b shows the block diagram of a frequency divider.

1.4.11 Data Transmission

One of the most common operations that occurs in any digital system is the transmission of information (data) from one place to another. The distance over which information is transmitted may be very small or very large. The information transmitted is in binary form, representing voltages at the outputs of a sending circuit which are connected to the inputs of a receiving circuit. There are two basic methods for transmission of digital information: *parallel* and *serial*.

In parallel data transmission, all the bits are transmitted simultaneously. So, one connecting line is required for each bit. Though data transmission is faster, the number of lines used between the transmitter and the receiver is more. This system is therefore complex and costly. On the other