OVERVIEW

SWITCHING THEORY AND LOGIC DESIGN

Types of Number Systems

Some of the important types of number system are

1. Decimal Number System
2. Binary Number System
3. Octal Number System
4. Hexadecimal Number System

1. Decimal Number Systems

The number system is having digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; this number system is known as a decimal number system because total ten digits are involved. The base of the decimal number system is 10.

2. Binary Number Systems

The modern computers do not process decimal number; they work with another number system known as a binary number system which uses only two digits 0 and1.The base of binary number system is 2 because it has only two digit 0 and 1.

3. Octal Numbers

The base of a number system is equal to the number of digits used, i.e., for decimal number system the base is ten while for the binary system the base is two. The octal system has the base of eight as it uses eight digits 0, 1, 2, 3, 4, 5, 6, 7.

4. Hexadecimal Numbers

The hexadecimal number system has a base of 16, and hence it consists of the following sixteen numbers of digits.0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

Binary system complements

As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement.

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement

2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Classification of binary codes

The codes are broadly categorized into following four categories.

- Weighted Codes
- Non-Weighted Codes
- Binary Coded Decimal Code
- Alphanumeric Codes
- Error Detecting Codes
- Error Correcting Codes

Boolean algebra is used to analyze and simplify the digital (logic) circuits. It uses only the binary numbers i.e. 0 and 1. It is also called as **Binary Algebra** or **logical Algebra**

Boolean Laws

There are six types of Boolean Laws.

Commutative law

Any binary operation which satisfies the following expression is referred to as commutative operation.

(i) $A.B = B.A$ (ii) $A + B = B + A$

Commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit.

Associative law

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

(i) $(A.B).C = A.(B.C)$ (ii) $(A + B) + C = A + (B + C)$

Distributive law

Distributive law states the following condition.

$A.(B + C) = A.B + A.C$

AND law

These laws use the AND operation. Therefore they are called as **AND** laws.

(i) $A.0 = 0$ (ii) $A.1 = A$

(iii) $A.A = A$ (iv) $A.\overline{A} = 0$

OR law

These laws use the OR operation. Therefore they are called as **OR** laws.

(i) $A + 0 = A$ (ii) $A + 1 = 1$

(iii) $A + A = A$ (iv) $A + \overline{A} = 1$

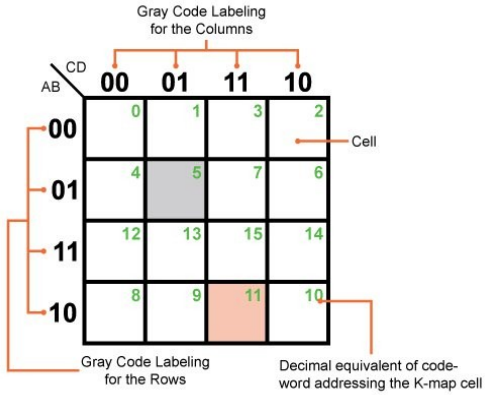
INVERSION law

This law uses the NOT operation. The inversion law states that double inversion of variable results in the original variable itself.

$\overline{\overline{A}} = A$

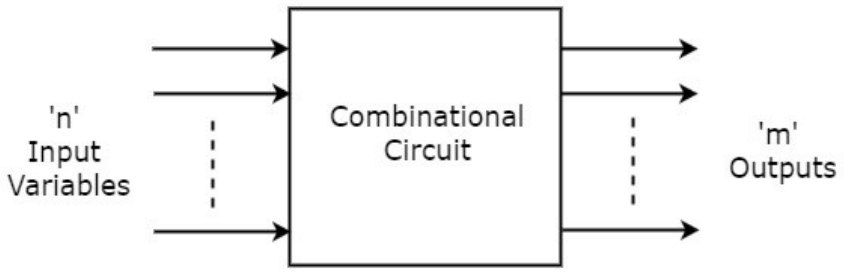
K-Map Simplification Technique

The K-map method of solving the logical expressions is referred to as the graphical technique of simplifying Boolean expressions



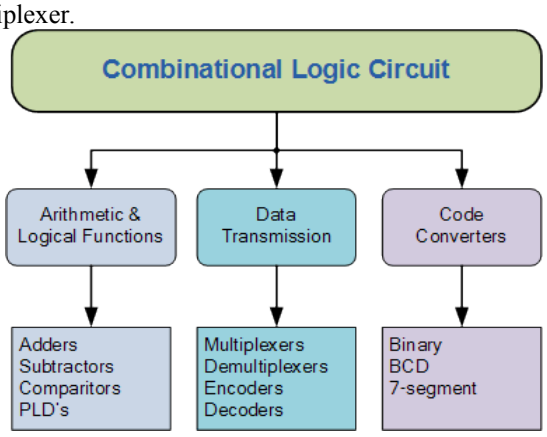
Combinational Circuits

Combinational circuits consist of Logic gates. These circuits operate with binary values. The output(s) of combinational circuit depends on the combination of present inputs. The following figure shows the **block diagram** of combinational circuit.



This combinational circuit has ‘n’ input variables and ‘m’ outputs. Each combination of input variables will affect the output(s).

Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer.



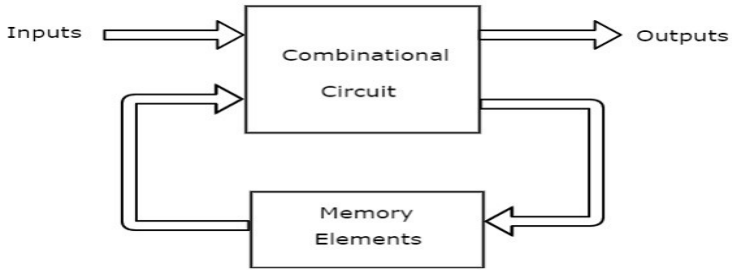
Programmable Logic Devices

Programmable Logic Devices **(PLDs)** are the integrated circuits. They contain an array of AND gates & another array of OR gates. There are three kinds of PLDs based on the type of array(s), which has programmable feature.

- Programmable Read Only Memory
- Programmable Array Logic
- Programmable Logic Array

Sequential Circuits

We discussed various combinational circuits in earlier chapters. All these circuits have a set of output(s), which depends only on the combination of present inputs. The following figure shows the **block diagram** of sequential circuit.



This sequential circuit contains a set of inputs and output(s). The output(s) of sequential circuit depends not only on the combination of present inputs but also on the previous output(s). Previous output is nothing but the **present state**. Therefore, sequential circuits contain combinational circuits along with memory (storage) elements. Some sequential circuits may not contain combinational circuits, but only memory elements.

Types of Sequential Circuits

Following are the two types of sequential circuits −

- Asynchronous sequential circuits
- Synchronous sequential circuits

Asynchronous sequential circuits

If some or all the outputs of a sequential circuit do not change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Asynchronous sequential circuit**.

Synchronous sequential circuits

If all the outputs of a sequential circuit change (affect) with respect to active transition of clock signal, then that sequential circuit is called as **Synchronous sequential circuit**.

Latches

There are two types of memory elements based on the type of triggering that is suitable to operate it.

Latches

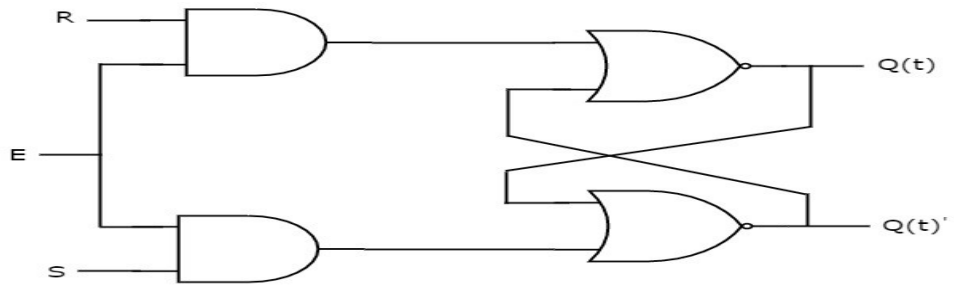
- SR Latch
- T Latch
- D Latch
- JK Latch

Flip-flops

- SR Flip Flop
- D Flip Flop
- JK Flip Flop
- T Flip Flop

SR Latch

SR Latch is also called as **Set Reset Latch**. This latch affects the outputs as long as the enable, E is maintained at '1'. The **circuit diagram** of SR Latch is shown in the following figure.



Conversion of Flip-Flops

In previous chapter, we discussed the four flip-flops, namely SR flip-flop, D flip-flop, JK flip-flop & T flip-flop. We can convert one flip-flop into the remaining three flip-flops by including some additional logic. So, there will be total of twelve **flip-flop conversions**.

Shift Registers

We know that one flip-flop can store one-bit of information. In order to store multiple bits of information, we require multiple flip-flops. The group of flip-flops, which are used to hold (store) the binary data is known as **register**.

If the register is capable of shifting bits either towards right hand side or towards left hand side is known as **shift register**. An 'N' bit shift register contains 'N' flip-flops. Following are the four types of shift registers based on applying inputs and accessing of outputs.

- Serial In − Serial Out shift register
- Serial In − Parallel Out shift register
- Parallel In − Serial Out shift register
- Parallel In − Parallel Out shift register

In this chapter, let us discuss about these two counters one by one.

Ring Counter

- In previous chapter, we discussed the operation of Serial In - Parallel Out **(SIPO)** shift register. It accepts the data from outside in serial form and it requires 'N' clock pulses in order to shift 'N' bit data.
- Similarly, **'N' bit Ring counter** performs the similar operation. But, the only difference is that the output of rightmost D flip-flop is given as input of leftmost D flip-flop instead of applying data from outside. Therefore, Ring counter produces a sequence of states (pattern of zeros and ones) and it repeats for every **'N' clock cycles**.

Johnson Ring Counter

The operation of **Johnson Ring counter** is similar to that of Ring counter. But, the only difference is that the complemented output of rightmost D flip-flop is given as input of leftmost D flip-flop instead of normal output. Therefore, 'N' bit Johnson Ring counter produces a sequence of states (pattern of zeros and ones) and it repeats for every **'2N' clock cycles**.

Counters

In previous two chapters, we discussed various shift registers & **counters using D flip flops**. Now, let us discuss various counters using T flip-flops. We know that T flip-flop toggles the output either for every positive edge of clock signal or for negative edge of clock signal.

An ‘N’ bit binary counter consists of ‘N’ T flip-flops. If the counter counts from 0 to $2^N - 1$, then it is called as binary **up counter**. Similarly, if the counter counts down from $2^N - 1$ to 0, then it is called as binary **down counter**.

There are two **types of counters** based on the flip-flops that are connected in synchronous or not.

- Asynchronous counters
- Synchronous counters

Asynchronous Counters

If the flip-flops do not receive the same clock signal, then that counter is called as **Asynchronous counter**. The output of system clock is applied as clock signal only to first flip-flop. The remaining flip-flops receive the clock signal from output of its previous stage flip-flop. Hence, the outputs of all flip-flops do not change (affect) at the same time.

Now, let us discuss the following two counters one by one.

- Asynchronous Binary up counter
- Asynchronous Binary down counter

Synchronous Counters

If all the flip-flops receive the same clock signal, then that counter is called as **Synchronous counter**.

Hence, the outputs of all flip-flops change (affect) at the same time.

Now, let us discuss the following two counters one by one.

- Synchronous Binary up counter
- Synchronous Binary down counter

Finite State Machines

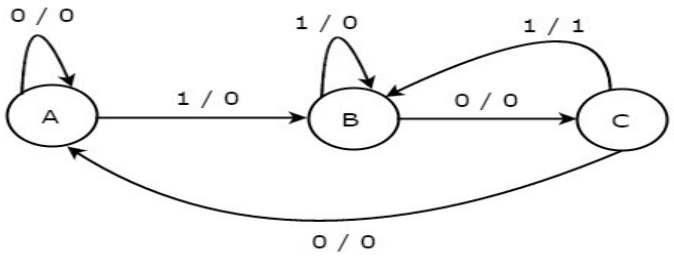
We know that synchronous sequential circuits change (affect) their states for every positive (or negative) transition of the clock signal based on the input. So, this behavior of synchronous sequential circuits can be represented in the graphical form and it is known as **state diagram**.

A synchronous sequential circuit is also called as **Finite State Machine** (FSM), if it has finite number of states. There are two types of FSMs.

- Mealy State Machine
- Moore State Machine

Mealy State Machine

The **state diagram** of Mealy state machine is shown in the following figure.



Moore State Machine

The **state diagram** of Moore state machine is shown in the following figure.