Course Objectives:

This course provides the deep knowledge of designing techniques of digital circuits, which is the basic and essential for any digital circuit. The main objectives are:

- To learn basic techniques for the design of digital circuits and fundamental concepts used in the design of digital systems.
- To understand basics of number representation in digital electronic circuits and to be able to convert between different representations.
- To implement simple logical operations using combinational logic circuits
- To design combinational logic circuits, sequential logic circuits.
- To impart to student the concepts of sequential circuits, enabling them to analyze sequential systems in terms of state machines. To implement synchronous state machines using flip-flops.

Course Outcomes:

Upon completion of the course, students should possess the following skills:

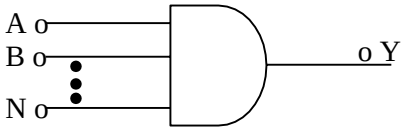
- Be able to manipulate numeric information in different forms, e.g. different bases, signed integers, various codes such as ASCII, Gray and BCD.
- Be able to manipulate simple Boolean expressions using the theorems and postulates of Boolean algebra and to minimize combinational functions.
- Be able to design and analyze small combinational circuits and to use standard combinational functions/building blocks to build larger more complex circuits.
- Be able to design and analyze small sequential circuits and devices and to use standard sequential functions/building blocks to build larger more complex circuits.

UNIT-1

BASIC DIGITAL CIRCUITS

The And Operation

Y = A AND B AND C … AND N



The standard symbol for an AND gate

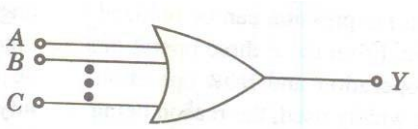
Truth Table

Truth table of a 2-input AND gate

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

The OR Operation

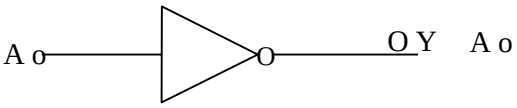
Y = A OR B OR C … OR N
= A + B + C + … + N



Truth table of a 2-input OR gate

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

The NOT Operation



The standard symbols for a NOT gate

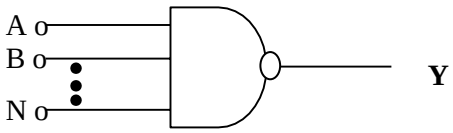
$$Y = \text{NOT } A$$

$$= \overline{A}$$

Input A	Output Y
0 1	1 0

The NAND Operation

$$Y = \overline{Y'} = \overline{(AB...N)}$$



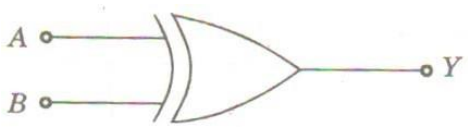
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

The NOR Operation

$$Y = \overline{A + B + ... + N}$$

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

EXCLUSIVE–OR OPERATION



Y = A EX – OR B = A ⊕ B

Inputs		Output
A	B	Y
0 0	0 1	0 1
1 1	0 1	1 0

NUMBER SYSTEM & CODES

DECIMAL NUMBER SYSTEM

We are all familiar with the decimal number system. It uses ten digits (0, 1, 2, 3, 4, 5, 6, 7, 9) and thus its base is 10.

Decimal 263 = $(2 \times 10^2) + (6 \times 10^1) + (3 \times 10^0) = 200 + 60 + 3 = 263$.

Since the base in decimal number system is 10, the number 263 can be written as 263_{10}. The suffix 10 emphasizes the fact that the base is 10.

BINARY NUMBER SYSTEM

The binary number system has only two digits 0 and 1. Thus a binary number is a string of zeros and ones. Since it has only two digits, the base is 2.

The binary number 1100 has 4 bits, 101011 has 6 bits and 11001010 has 8 bits.

A string of 8 bits is known as a **byte**.

Binary, decimal, hexadecimal and octal equivalence

Binary				**Decimal**	**Hexadecimal**	**Octal**
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	2	2	2
0	0	1	1	3	3	3
0	1	0	0	4	4	4
0	1	0	1	5	5	5
0	1	1	0	6	6	6
0	1	1	1	7	7	7
1	0	0	0	8	8	10
1	0	0	1	9	9	11
1	0	1	0	10	A	12
1	0	1	1	11	B	13
1	1	0	0	12	C	14
1	1	0	1	13	D	15
1	1	1	0	14	E	16
1	1	1	1	15	F	17

Binary to Decimal Conversion and vice versa

Convert 1101_2 into equivalent decimal number.

1	1	0	1	Binary number
8	4	2	1	Write weights
8	4	2	1	Cross out weights under zeros
8 +	4 +	0 +	1 = 13	Add weights

Therefore, $1101_2 = 13_{10}$

BINARY ARITHMETIC

The rules for addition of binary numbers are as under :

$0 + 0 = 0$

$0 + 1 = 1 + 0 = 1$

$1 + 1 = 10$ i.e., 1 + 1 equals 0 with a carry of 1 to next higher column

$1 + 1 + 1 = 11$ i.e., 1 + 1 + 1 equals 1 with a carry of 1 to next higher column.

Binary Subtraction

The rules for subtraction of binary numbers are as under :

$0 - 0 = 0$

$1 - 0 = 1$

$1 - 1 = 0$

$10 - 1 = 1$

1'S COMPLEMENT

The 1's complement of a binary number is obtained by complementing each bit (i.e., 0 for 1 and 1 for 0).

	1 1 0 0	1 0 0 1
1's complement is	0 0 1 1	0 1 1 0

2'S COMPLEMENT

2's complement is defined as the new word obtained by adding 1 to 1's complement* e.g.,

Let $A = 0\ 1\ 0\ 1$ i.e., 5

1's complement $\overline{A} = 1\ 0\ 1\ 0$

$+\ \ \ \ 1$

2's complement $A' = 1\ 0\ 1\ 1$ i.e., –5

2'S COMPLEMENT ADDITION, SUBTRACTION

$S = A + B' = A + (-B) = A - B$

Moreover, the final carry has no significance and is not used.

BINARY FRACTIONS

So far we have discussed only whole numbers. However, to represent fractions is also important. The decimal number 2568 is represented as

$$2568 = 2000 + 500 + 60 + 8 = 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$$

Similarly, 25.68 can be represented as

$$25.68 = 20 + 5 + 0.6 + 0.08 = 2 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

Conversion of Binary to Decimal

$$0.1011 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$
$$= 0.5 + 0 + 0.125 + 0.0625 = 0.6875 \text{ (decimal)}$$

.

OCTAL NUMEBR SYSTEM

$(6327.4051)_8$ is an octal number. Using the weights it can be written as

$$(6327.4051)_8 = 6 \times 8^3 + 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + .4 \times 8^{-1} + 0 \times 8^{-2} + 5 \times 8^{-3} + 1 \times 8^{-4}$$

$$= 3072 + 192 + 16 + 7 + \ldots\ldots\ldots\ldots$$

$$= (3287.5100098)_{10}$$

HEXADECIMAL NUMBER SYSTEM

The base for hexadecimal number system is 16 which requires 16 distinct symbols to represent the numbers. These are numerals 0 through 9 and alphabets A through F.

Binary and decimal equivalents of hexadecimal numbers

Gray Code

Decimal	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

INTRODUCTION OF DIGITAL LOGIC FAMILY

The basic building block in a digital system is logic gate logic circuits have evolved into families each of which has its own advantages and disadvantages. Generally a digital system is designed with circuits from one family only

The different logic functions are available in integrated circuit (IC) form. All digital systems are built with digital ICs. Typical chip sizes range from 40 × 40 mils** to about 300 × 300 mils and it contains both active and passive components. Many processes like wafer preparation, impurity diffusion, ion implantation, oxide growth, photolithography are used in their manufacture.

Levels of integration of digital ICs

Category	Numbers of equivalent basic gates on a single chip	Total number of components on single chip
SSI	Less than 12	Less than 100
MSI	12-99	100-999
LSI	100-999	1000-9999
VLSI	1000-9999	10,000 to 99,999
ULSI	10,000 and more	100,000 and more

Classification as per Technology

Digital ICS are manufactured by two technologies viz., Bipolar and MOS.The bipolar family uses transistor fabricated on a chip. This family includes DTL (Diode transistor logic using diodes and transistor), TTL (transistor-transistor logic which uses transistors only) and ECL (emitter coupled logic). TTL is the most poplar family in SSI and MSI category.

MOS family includes PMOS (p-channel MOSFET), NMOS (n0channel MOSFET) and CMOS (complementary MOSFET). PMOS is almost obsolete. NMOS is dominating the LSI field. CMOS is the most commonly used technology for digital wrist watches, pocket calculators etc.

The technologies being used now-a-days are TTL, ECL and CMOS.

CHARACTERISTICS OF DIGITAL ICs

The various characteristics of digital ICs used to compare their performances are:

1. Speed of operation,
2. Power dissipation,
3. Figure of merit,
4. Fan-out,
5. Current and voltage parameters,
6. Noise immunity,
7. Operating temperature range,
8. Power supply requirements, and
9. Flexibilities available.

UNIT 2

COMBINATIONAL LOGIC DESIGN

INTRODUCTION

In *combinational circuits*, the outputs at any instant of time depend upon the inputs present at that instant of time. This means there is no memory in these circuits.

The design requirements of combinational circuits may be specified in one of the following ways:

STANDARD FORMS OF LOGIC FUNCTIONS

A logic circuit to implement the above equation/truth table is to be synthesized. The logic expression can be either a sum of products or product of sums.[1]

Fundamental Products and Sum of Products

1, SOP (Sum of Product)
Having minterms (m)

2. POS (Product of sum)
Having maxterms (M)

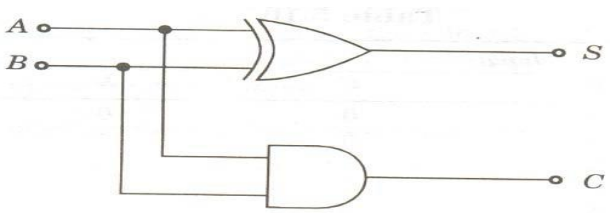
Arithmetic Circuits

Half-adder. A logic circuit for the addition of two one-bit numbers is referred to as an half-adder. A and B are the two inputs and S (SUM) and C(CARRY) are the two outputs.

$$S = A\,B + A\,B = A \oplus B$$

$$C = A$$

Inputs		Outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



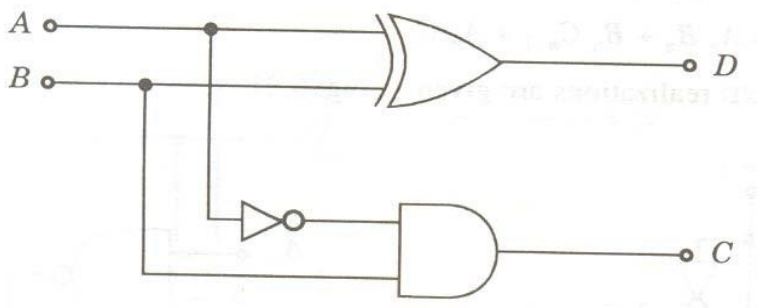
Half-sub-tractor. A logic circuit for the subtraction of B (subtrahend) from A (minuend) where A and B are 1-bit numbers is referred to as a *half-sub-tractor.*

A and B are the two inputs and D(difference) and C (borrow) are the two outputs.

Inputs		Outputs	
A	B	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \underline{A}\,B + A\,B = A \oplus B$

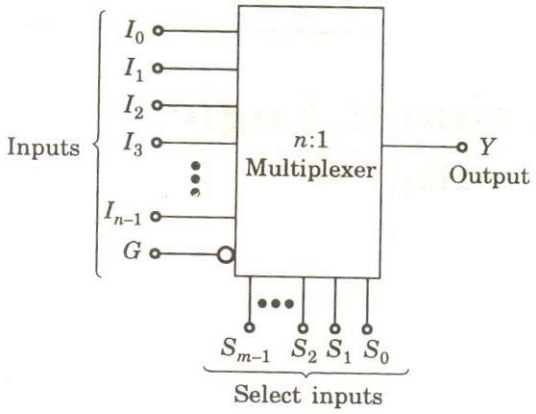
$C = \underline{A}\,B$



MULTIPLEXERS AND THEIR USE IN COMBINATIONAL LOGIC DESIGN

Multiplexer

m are select lines, n are inputs lines , 1 is Output lines. $\mathbf{n = 2^m}$

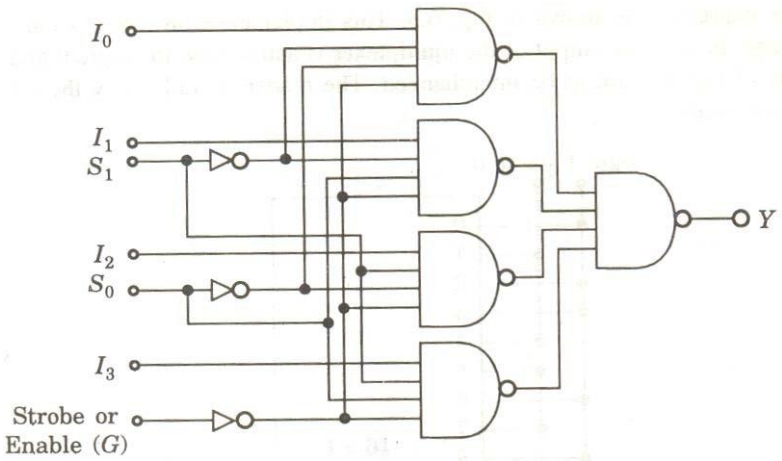


Select inputs		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = S_1 \overline{S_0}\ \overline{I_0} + S_1 \overline{S_0} I_1 + S_1 S_0 \overline{I_2} + S_1 S_0 I$$

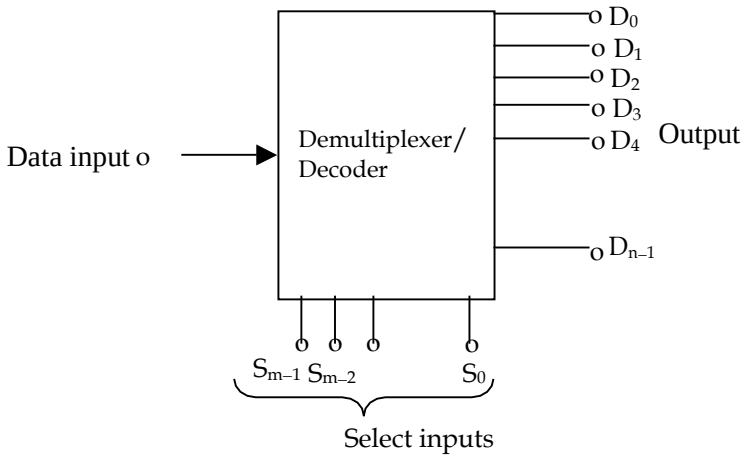
.

Available multiplexer IC



Demultiplexer

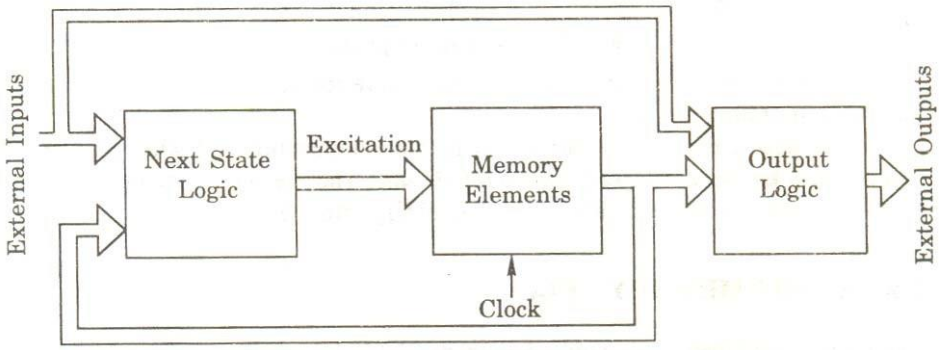
The demultiplexer performs the reverse operation of a multiplexer. It accepts a single input and distributes it over several outputs.



FLIP FLOPS

INTRODUCTION

A block diagram of a sequential circuit is shown in Fig. It consists of combinational circuits which accept digital signals from external inputs and from outputs of



Block diagram of sequential circuit.

memory elements and generates signals for external outputs and for inputs to memory elements referred to as excitation.

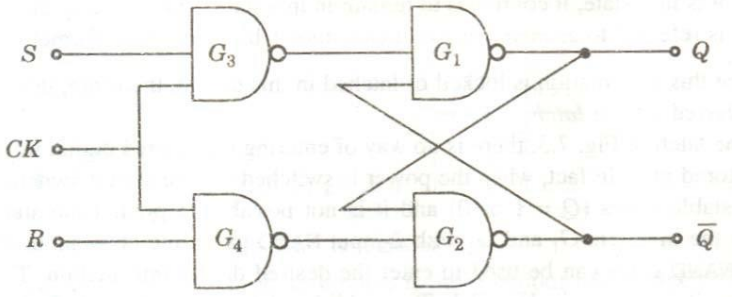
A memory element is some medium in which one bit of information (1 or 0) can be stored or retained until necessary, and thereafter its contents can be replaced by a new value. The contents of memory elements in Fig. 4.1 can be changed by the outputs of the combinational circuit which are connected to its input.

Sequential circuits are classified in two main categories, known as *Sychronous* and *Asynchronous* sequential circuits depending on timing of their signals.

CLOCKED S-R FLIP-FLOP

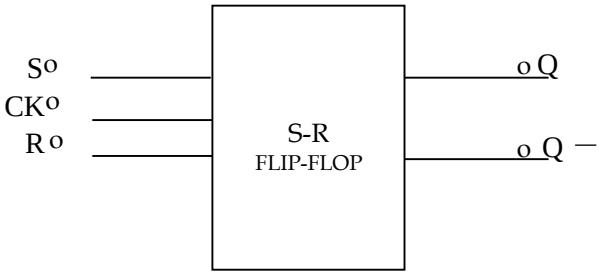
It is often required to set or reset the memory cell

A clocked S-R FOIP-FLOP.



Inputs		Output
S_n	R_n	Q_{n+1}

0	0	Q_n
1	0	1
0	1	0
1	1	?

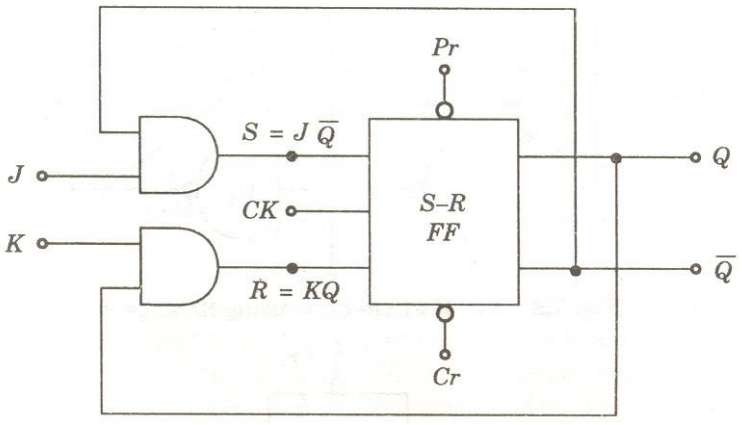


Logic symbol of clocked S–R FLIP-FLOP

J-K FLIP-FLOP

The uncertainty in the state of an S-R FLIP-FLOP when $S_n = R_n = 1$ (fourth row of the truth table) can be eliminated by converting it into a J-K FLIP-FLOP. The data inputs are J and K which are ANDed with $\overline{Q}$ and Q, respectively, to obtain S and R inputs, i.e.

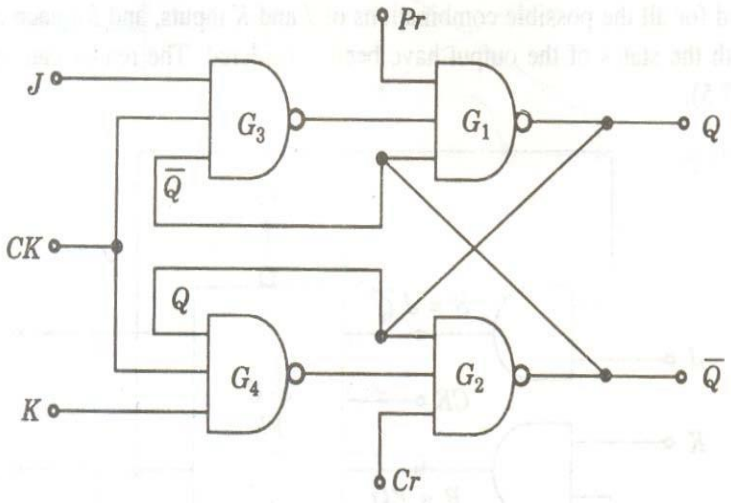
$$S = J.\ \overline{Q}$$
$$R = K.\ Q$$



An S-R FLIP-FLOP converted into J-K FLIP-FLOP.

Data inputs		Outputs		Inputs to S-R FF		Output
J_n	K_n	Q_n	$\overline{Q_n}$	S_n	R_n	Q_{n+1}
0	0	0	1	0	0	0] = Q_n

0	0	1	0	0	0	1
1 1	0 0	0 1	1 0	1 0	0 0	1 1] = 1
0 0	1 1	0 1	1 0	0 0	0 1	0 0] = 0
1 1	1 1	0 1	1 0	1 0	0 1	0 1] = $\overline{Q_n}$

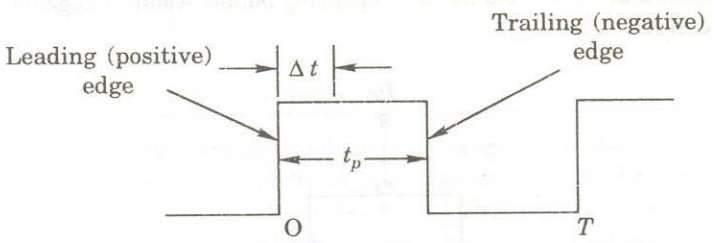


The Race-Around Condition

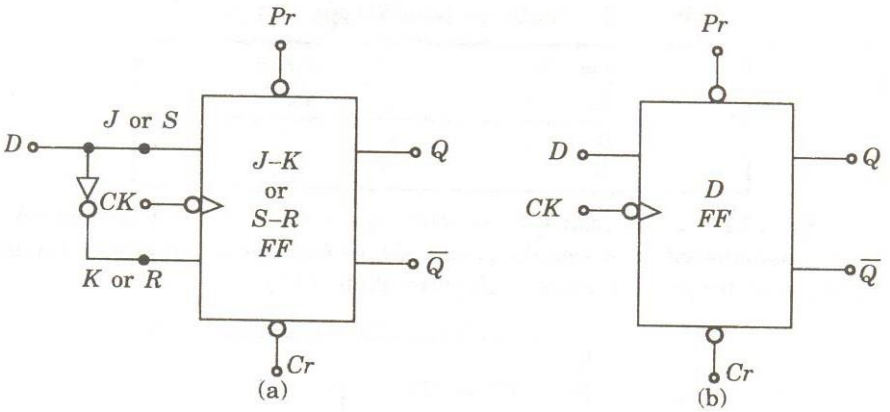
The difficulty of both inputs 1 (S = R = 1) being not allowed in an S-R FLIP-FLOP is eliminated in a J-K FLIP

J = K = 1 and Q =1 and after another interval of t the output will change backto Q = 0. Hence, we conclude that for the duration t_p of the clock p0ulse, the output will oscillate back and forth between 0 and 1. At the end of the clock pulse, the value of Q is uncertain. This situation is referred to as the race-round condition.

The race-around condition can be avoided if $t_p < t < T$. However, it may be difficult to satisfy this inequality because of very small propagation delays in ICs. A more practical method for overcoming this difficulty is the use of the master-slave (M-S) configuration discussed below.

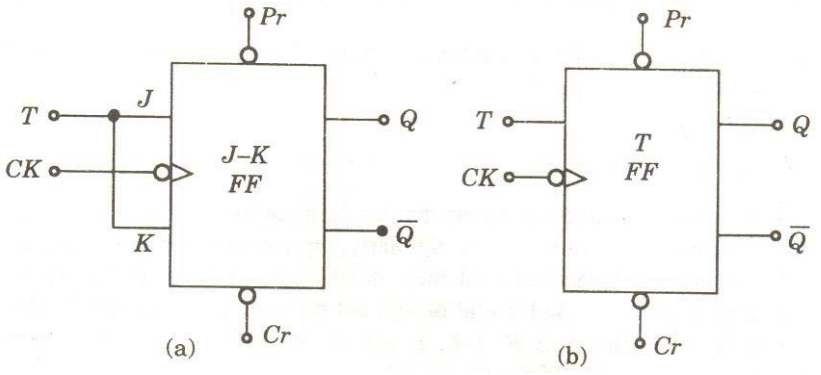


D-TYPE FLIP-FLOP



Input D_n	Output Q_{n+1}
0 1	0 1

T-TYPE FLIP-FLOP



Truth table of T-type FLIP-FLOP

Input T_n	Output Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

EXCITATION TABLE OF FLIP-FLOP

The truth table of a FLIP-FLOP is also referred to as the *characteristic table* and specifies the operational characteristic of the FLIP-FLOP.

Present State	Next State	S-R S_n	FF R_n	J-K J_n	FF K_n	T-FF T_n	D-FF D_n
0	0	O	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

UNIT 3

COUNTERS

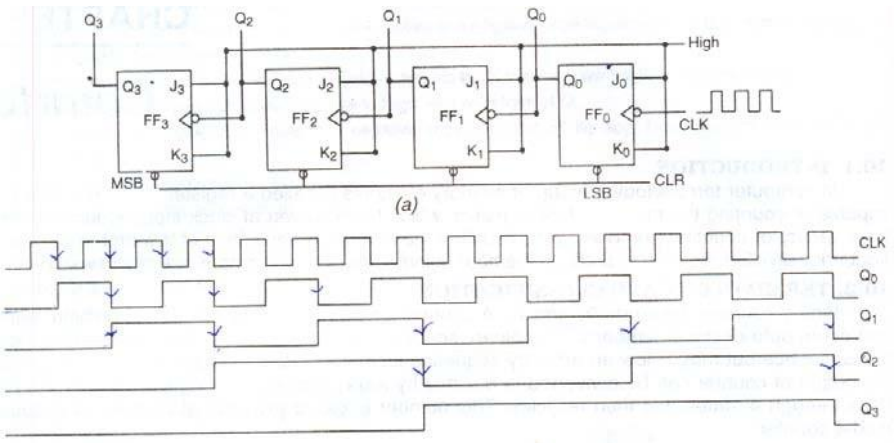
INTRODUCTION

In computer terminology a group of memory elements is called a register. A counter is a register capable of counting the number of clock pulses which have arrived at clock input. Counters are used for a variety of counting purposes, e.g., counting the number of revolution of a motor in a given time, frequency division required to produce the hour, minute and seconds output in a digital watch etc.

The word binary counter means a counter which counts and produces binary output 0000, 0001, 0010, 0011 etc. It goes through a binary sequence depending on its modulus.

CIRCUIT AND WORKING OF RIPPLE COUNTER:

.

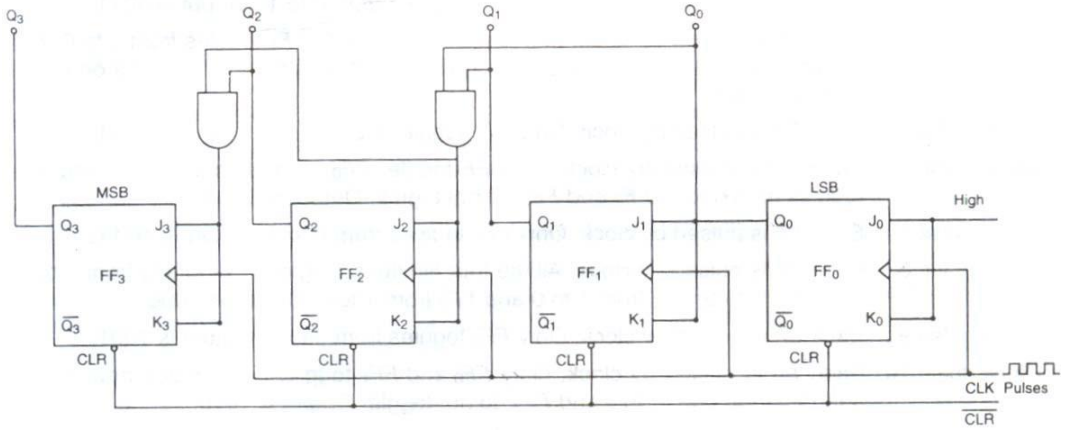


Ripple counter (a) circuit (b) timing diagram

Count	Q_3	Q_2	Q_1	Q_0	Count	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

SYNCHRONOUS COUNTER

In a synchronous counter, all the flip flops are clocked together.



Synchronous counter

Thus, each flip flop toggles on the next positive clock edge if all lower bits are 1.

Number of clock pulses	Counting sequence				Equivalent decimal
	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15

DIFFERENCE BETWEEN ASYNCHRONOUS AND SYNCHRONOUS COUNTER

The term asynchronous means that the counter is clocked in such a way that all the flip flops of the counter do not receive clock pulses at the same time. Ripple counter is an asynchronous counter. The clock pulses drive the clock input of the first flip flop. But the clock input of the second flip

flop is driven by Q output of the first flip flop. The clock input of third flip flop is driven by the Q output of the second flip flop and so on. This results in propagation delay in each flip flop. In a 4 bit ripple counter having flip flops with propagation delay of 10 ns each, there will be a total propagation delay of 40 ns. This may be undesirable in many cases.

The term synchronous means that all flip flops are clocked simultaneously. The clock pulses drive the clock input of all the flip flops together so that there is no propagation delay.

BASIC SHIFT OPERATIONS

A simple example of shift operation is that in a calculator. Suppose we have to enter 356 in the calculator. First we press and release 3. The digit 3 appears in the display. Next we press and release 5. The digit 3 appears in the display .

The digit 3 is shifted one place to the left and 5 appears on the extreme right. As we press and release 6, the digit 3 and 5 shift to the left and 6 appears in the extreme right position. This simple example illustrates two characteristics of a shift register.

(a) serial shift left, then out

(b) serial shift right, then out

(c) parallel shift in

(d) parallel shift out

(e) rotate left

(f) rotate right.

SHIFT REGISTER OPERATIONS

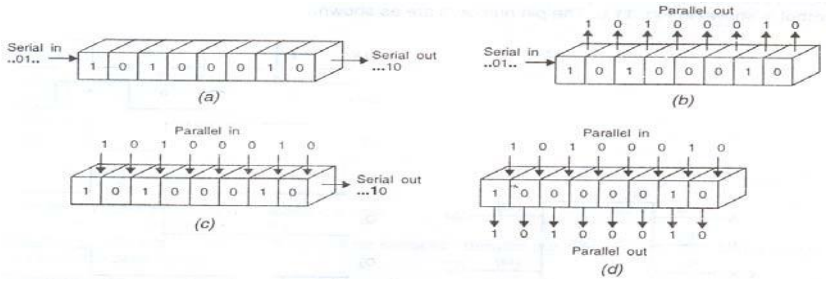
One method to describe the operation of shift register is the method of loading in and reading from the storage bits. There could be 4 such operations :

(a) **Serial in – Serial out :** The data is loaded into and read from the shift register serially.

(b) Serial in – **Parallel out** : The data is loaded into the register serially but read in parallel (i.e., data is available from all bits simultaneously.

(c) **Parallel in – Serial out :** The data is loaded in parallel, i.e., the bits are entered simultaneously in their respective stages and read serially.

(d) **Parallel in – Parallel out** : The data is loaded and read from the register in parallel, i.e., all bits are loaded simultaneously and read simultaneously.



Shift register operations (a) serial in-serial out (SISO) (b) serial in – parallel out (SIPO) (c) parallel in – serial out (PISO) (d) parallel in – parallel out (PIPO)

References

1. **http://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/**
2. **http://www.ni.com/example/14493/en/**
3. **http://195.134.76.37/applets/AppletGates/Appl_Gates2.html**
4. **https://www.allaboutcircuits.com/textbook/digital/**
5. **https://www.tutorialspoint.com/digital_electronics/index.asp**