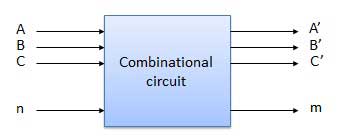
**Combinational Circuits :**

**Combinational circuit is circuit in which we combine the different gates in the circuit for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following.**

* **The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.**
* **The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.**
* **A combinational circuit can have a n number of inputs and m number of outputs.**

**Block diagram**

****

**The design procedure involves the following steps:**

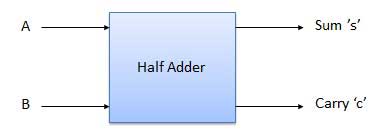
1. **The problem is stated.**
2. **The input and output variables are assigned letter symbols.**
3. **The truth table that defines the relationship between inputs and outputs is derived.**
4. **The simplified Boolean functions for each output are obtained.**
5. **The logic diagram is drawn.**

**We're going to elaborate few important combinational circuits as follows.**

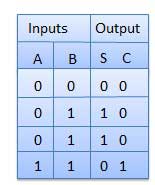
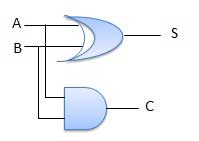
**Half Adder**

**Half adder is a combinational logic circuit with two input and two output. The half adder circuit is designed to add two single bit binary number A (augend) and B (addend). It is the basic building block for addition of two single bit numbers. This circuit has two outputs carry and sum.**

**Block diagram**

****

**Truth Table Circuit/LOGIC Diagram**

** **

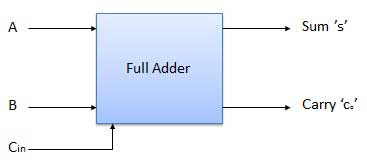
**Boolean functions for the output : S = A’B + AB’ = A B**

**C = AB**

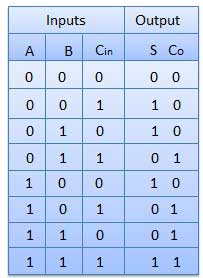
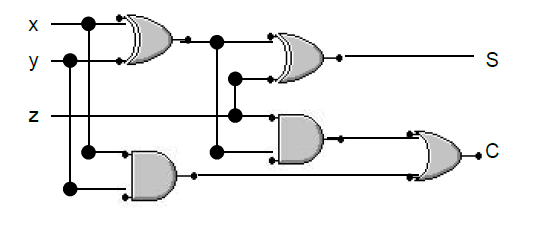
**Full Adder**

**Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.**

**Block diagram**

****

**Truth Table Circuit Diagram**

** **

**Boolean function S = x y z**

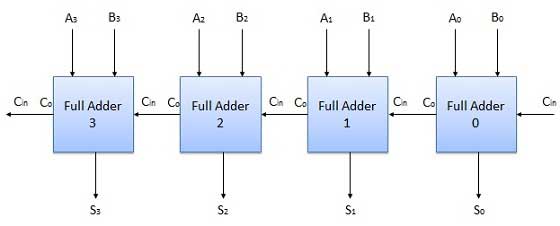
**C = xy + (x y) z  
N-Bit Parallel Adder**

**The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit. To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.**

**4 Bit Parallel Adder**

**In the block diagram, A0 and B0 represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its Cin has been permanently made 0. The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.**

**Block diagram**

****

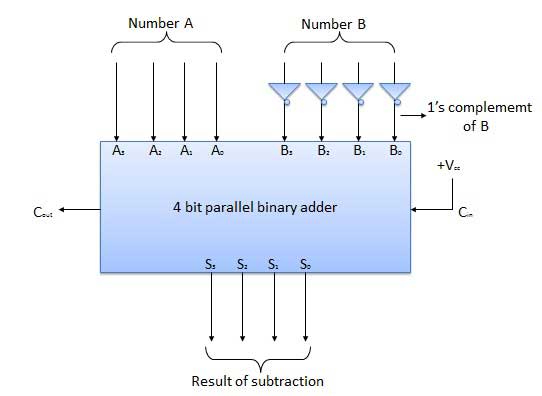
**N-Bit Parallel Subtractor**

**The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.**

**4 Bit Parallel Subtractor**

**The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction. S3 S2 S1 S0 represent the result of binary subtraction (A-B) and carry output Cout represents the polarity of the result. If A > B then Cout =0 and the result of binary form (A-B) then Cout = 1 and the result is in the 2's complement form.**

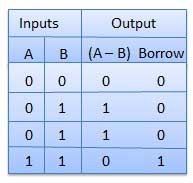
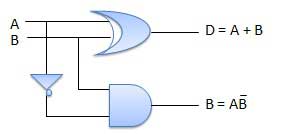
**Block diagram**

****

**Half Subtractors**

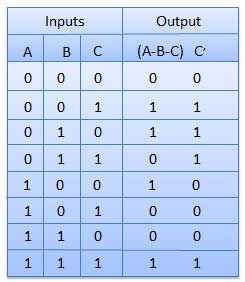
**Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces a output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.**

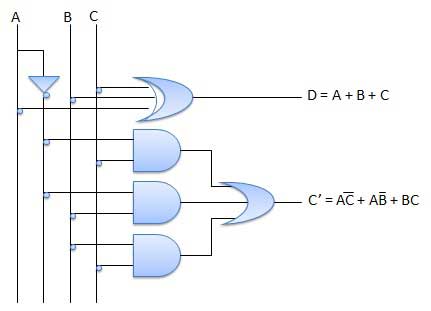
**Truth Table Circuit Diagram**

** **

**Full Subtractors**

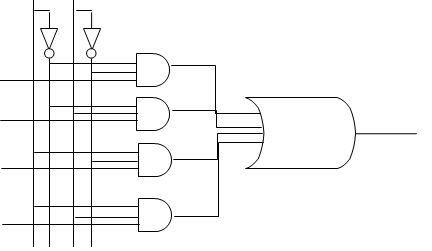
**The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,C and two output D and C'. A is the minuend, B is subtrahend, C is the borrow produced by the previous stage, D is the difference output and C' is the borrow output.**

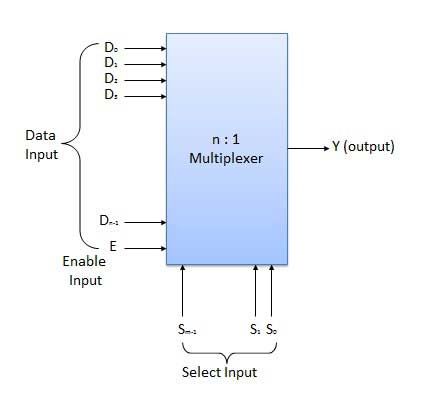
**Truth Table Circuit Diagram**

****

**Multiplexers**

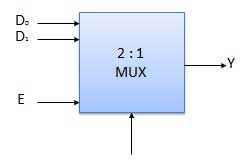
**Multiplexer is a special type of combinational circuit that receives binary information from one of 2n input data lines and directs it to a single output line. The selection of a particular input data line for the output is determined by a set of selection inputs. A 2n–to – 1 multiplexer ahs 2n input data lines and n input selection lines whose bit combinations determine which input data are selected for the output.**

**Block diagram**

**  
Multiplexers come in multiple variations**

* **2 : 1 multiplexer**
* **4 : 1 multiplexer**
* **16 : 1 multiplexer**
* **32 : 1 multiplexer**

**Block Diagram Truth Table**

****

|  |  |  |
| --- | --- | --- |
| **Select** | | **Output** |
| **S1** | **S0** | **Y** |
| **0** | **0** | **I0** |
| **0** | **1** | **I1** |
| **1** | **0** | **I2** |
| **1** | **1** | **I3** |

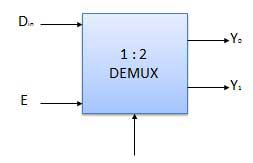
**Demultiplexers**

**A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line. A de-multiplexer is equivalent to a single pole multiple way switch as shown in fig.**

**Demultiplexers come in multiple variations**

* **1 : 2 demultiplexer**
* **1 : 4 demultiplexer**
* **1 : 16 demultiplexer**
* **1 : 32 demultiplexer**

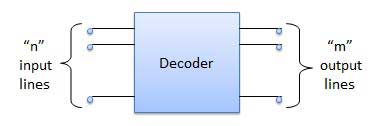
**Block diagram Truth Table**

****

**Decoder**

**A decoder is a combinational circuit that converts binary information from n coded input to a maximum of 2n unique outputs.. It has n input and to a maximum m = 2n outputs. Decoder is identical to a demultiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.**

**Block diagram**

****

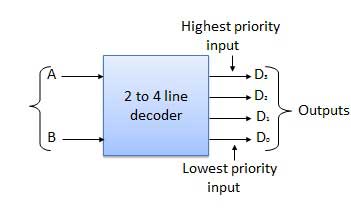
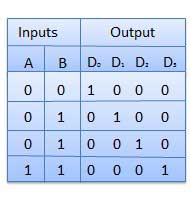
**Examples of Decoders are following.**

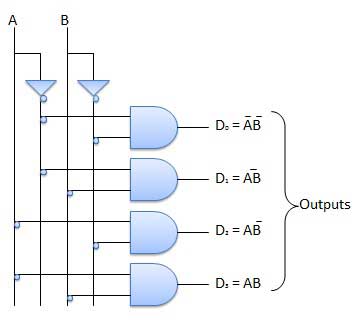
* **Code converters**
* **BCD to seven segment decoders**
* **Nixie tube decoders**
* **Relay actuator**

**2 to 4 Line Decoder**

**The block diagram of 2 to 4 line decoder is shown in the fig. A and B are the two inputs where D0 through D3 are the four outputs. Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs.**

**Block diagram Truth Table**

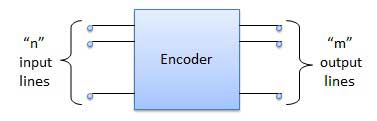
**  
Logic Circuit**

****

**Encoder**

**Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has 2n or less number of input lines and m number of output lines. An encoder produces an m bit binary code corresponding to the digital input number. The encoder accepts an n input digital word and converts it into an m bit another digital word.**

**Block diagram Truth table**

****

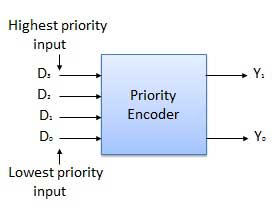
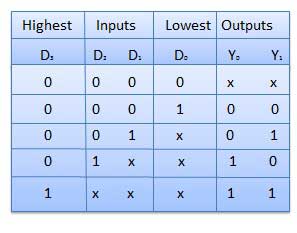
**Examples of Encoders are following.**

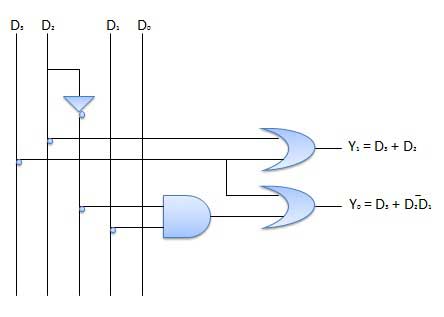
* **Priority encoders**
* **Decimal to BCD encoder**
* **Octal to binary encoder**
* **Hexadecimal to binary encoder**

**Priority Encoder**

**This is a special type of encoder. Priority is given to the input lines. If two or more input line are 1 at the same time, then the input line with highest priority will be considered. There are four input D0, D1, D2, D3 and two output Y0, Y1. Out of the four input D3 has the highest priority and D0 has the lowest priority. That means if D3 = 1 then Y1 Y0 = 11 irrespective of the other inputs. Similarly if D3 = 0 and D2 = 1 then Y1 Y0 = 10 irrespective of the other inputs.**

**Block diagram Truth Table**

**   
Logic Circuit**

****

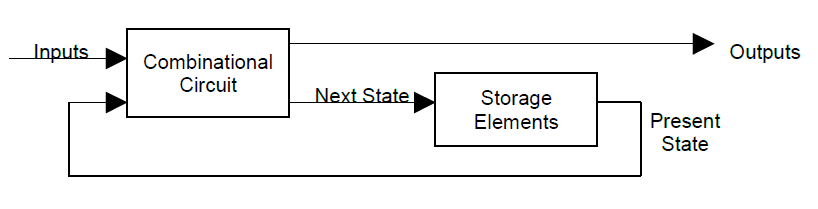
**SEQUENTIAL CIRCUITS :**

**Definition:**

**A sequential circuit consists of a combinational circuit and storage elements that together form a feedback system.**

**Storage Elements:**

**The storage elements are devices capable of storing binary information within them. The binary information stored at any given time defines the *state* of the sequential circuit.**

****

**Synchronous sequential circuit: is a system whose behaviour can be defined from the knowledge of its signals at discrete instants of time.**

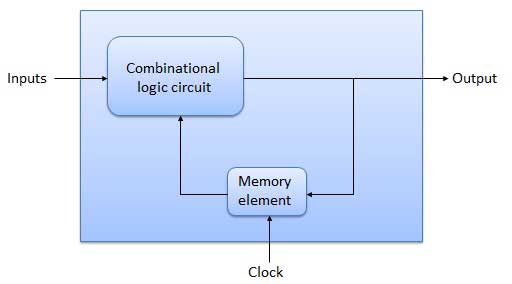
**Asynchronous sequential circuit: The behaviour of an asynchronous sequential circuit depends upon the order in which the inputs change, and the state of the circuit can be affected at any instant of time.**

**A synchronous sequential circuit employs signals that affect the storage elements only at discrete instants of time. Synchronization is achieved by a timing device called a clock generator that produces a periodic train of clock pulses.**

**The outputs of storage elements change only when clock pulses are present.**

**The combinational circuit does not use any memory. Hence the previous state of input does not have any effect on the present state of the circuit. But sequential circuit has memory so output can vary based on input. This type of circuits uses previous input, output, clock and a memory element.**

**Block diagram**

****

**Flip Flop**

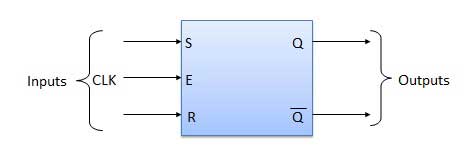
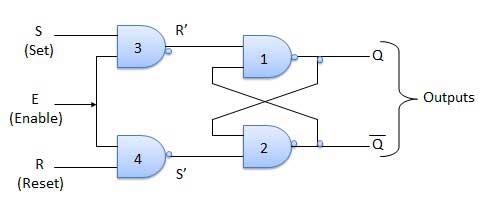
**Flip flop is a sequential circuit which generally samples its inputs and changes its outputs only at particular instants of time and not continuously. Flip flop is said to be edge sensitive or edge triggered rather than being level triggered like latches.**

**Each flip-flop is capable of holding one bit of information. Flip-flops are clocked storage elements. Flip-flops are not combinational circuits, because they have clock inputs, and the results change only at the time of clock pulses. The characteristic tables of flip-flops are used to determine the next state when the flip-flop input(s) are known.**

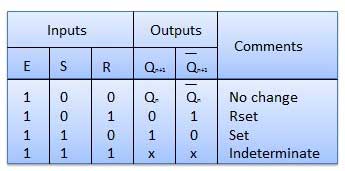
**S-R Flip Flop**

**It is basically S-R latch using NAND gates with an additional enable input. It is also called as level triggered SR-FF. For this circuit in output will take place if and only if the enable input (E) is made active. In short this circuit will operate as an S-R latch if E= 1 but there is no change in the output if E = 0.**

**Block Diagram Circuit Diagram**

** **

**Truth Table**

****

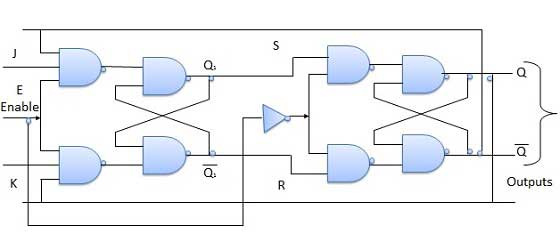
**Operation**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Condition** | **Operation** |
| **1** | **S = R = 0 : No change** | * **If S = R = 0 then output of NAND gates 3 and 4 are forced to become 1.** * **Hence R' and S' both will be equal to 1. Since S' and R' are the input of the basic S-R latch using NAND gates, there will be no change in the state of outputs.** |
| **2** | **S = 0, R = 1, E = 1** | * **Since S = 0, output of NAND-3 i.e. R' = 1 and E = 1 the output of NAND-4 i.e. S' = 0.** * **Hence Qn+1 = 0 and Qn+1 bar = 1. This is reset condition.** |
| **3** | **S = 1, R = 0, E = 1** | * **Output of NAND-3 i.e. R' = 0 and output of NAND-4 i.e. S' = 1.** * **Hence output of S-R NAND latch is Qn+1 = 1 and Qn+1 bar = 0. This is the reset condition.** |
| **4** | **S = 1, R = 1, E = 1** | * **As S = 1, R = 1 and E = 1, the output of NAND gates 3 and 4 both are 0 i.e. S' = R' = 0.** * **Hence the Race condition will occur in the basic NAND latch.** |

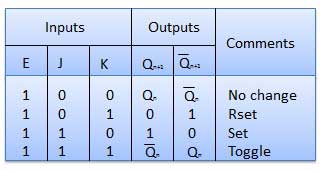
**Master Slave JK Flip Flop**

**Master slave JK FF is a cascade of two S-R FF with feedback from the output of second to input of first. Master is a positive level triggered. But due to the presence of the inverter in the clock line, the slave will respond to the negative level. Hence when the clock = 1 (positive level) the master is active and the slave is inactive. Whereas when clock = 0 (low level) the slave is active and master is inactive.**

**Circuit Diagram**

****

**Truth Table**

****

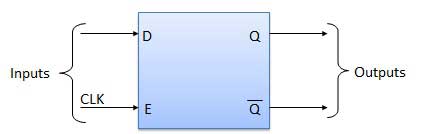
**Operation**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Condition** | **Operation** |
| **1** | **J = K = 0 (No change)** | * **When clock = 0, the slave becomes active and master is inactive. But since the S and R inputs have not changed, the slave outputs will also remain unchanged. Therefore outputs will not change if J = K =0.** |
| **2** | **J = 0 and K = 1 (Reset)** | * **Clock = 1: Master active, slave inactive. Therefore outputs of the master become Q1 = 0 and Q1 bar = 1. That means S = 0 and R =1.** * **Clock = 0: Slave active, master inactive Therefore outputs of the slave become Q = 0 and Q bar = 1.** * **Again clock = 1: Master active, slave inactive. Therefore even with the changed outputs Q = 0 and Q bar = 1 fed back to master, its outputs will Q1 = 0 and Q1 bar = 1. That means S = 0 and R = 1.** * **Hence with clock = 0 and slave becoming active the outputs of slave will remain Q = 0 and Q bar = 1. Thus we get a stable output from the Master slave.** |
| **3** | **J = 1 and K = 0 (Set)** | * **Clock = 1: Master active, slave inactive. Therefore outputs of the master become Q1 = 1 and Q1 bar = 0. That means S = 1 and R =0.** * **Clock = 0: Slave active, master inactive Therefore outputs of the slave become Q = 1 and Q bar = 0.** * **Again clock = 1: then it can be shown that the outputs of the slave are stabilized to Q = 1 and Q bar = 0.** |
| **4** | **J = K = 1 (Toggle)** | * **Clock = 1: Master active, slave inactive. Outputs of master will toggle. So S and R also will be inverted.** * **Clock = 0: Slave active, master inactive. Outputs of slave will toggle.** * **These changed output are returned back to the master inputs. But since clock = 0, the master is still inactive. So it does not respond to these changed outputs. This avoids the multiple toggling which leads to the race around condition. The master slave flip flop will avoid the race around condition.** |

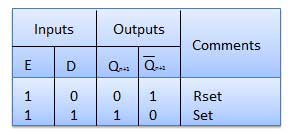
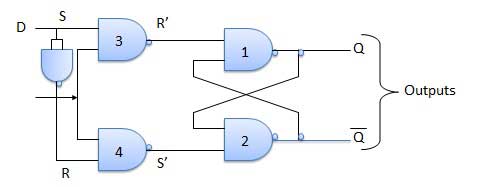
**Delay Flip Flop / D Flip Flop**

**Delay Flip Flop or D Flip Flop is the simple gated S-R latch with a NAND inverter connected between S and R inputs. It has only one input. The input data is appearing at the output after some time. Due to this data delay between i/p and o/p, it is called delay flip flop. S and R will be the complements of each other due to NAND inverter. Hence S = R = 0 or S = R = 1,these input condition will never appear. This problem is avoid by SR = 00 and SR = 1 conditions.**

**Block Diagram**

****

**Circuit Diagram Truth Table**

****

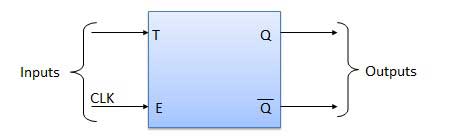
**Operation**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Condition** | **Operation** |
| **1** | **E = 0** | * **Latch is disabled. Hence is no change in output.** |
| **2** | **E = 1 and D = 0** | * **If E = 1 and D = 0 then S = 0 and R = 1. Hence irrespective of the present state, the next state is Qn+1 = 0 and Qn+1 bar = 1. This is the reset condition.** |
| **3** | **E = 1 and D = 1** | * **if E = 1 and D = 1, then S = 1 and R = 0. This will set the latch and Qn+1 = 1 and Qn+1 bar = 0 irrespective of the present state.** |

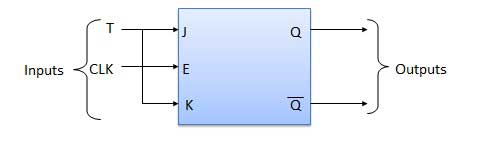
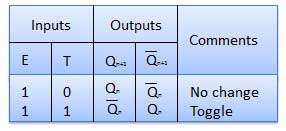
**Toggle Flip Flop / T Flip Flop**

**Toggle flip flop is basically a JK flip flop with J and K terminals permanently connected together. It has only input denoted by T is shown in the Symbol Diagram. The symbol for positive edge triggered T flip flop is shown in the Block Diagram.**

**Symbol Diagram**

****

**Block Diagram Truth Table**

** **

**Operation**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Condition** | **Operation** |
| **1** | **T = 0, J = K = 0** | * **The output Q and Q bar won't change** |
| **2** | **T = 1 ,J = K = 1** | * **output will toggle corresponding to every leading edge of clock signal.** |

**Classification of semiconductor memory and memory organization:**

**A computer memory must be able to temporarily store the patterns of bits with which the processor is working and provide it immediate access to any location it requests. The storage of information is accomplished by using collections of individual storage elements, each of which is capable of maintaining a single bit. For a device to be useful as a memory element it must have two stable states, a mechanism for setting the device to one state or the other, and a mechanism for reading the state. Memory systems have evolved through a variety of devices that match this characteristic, from relays, vacuum tubes, delay lines, ferrite cores to semiconductor materials. The cost and size of main memory, as well as their speed became a disadvantage as semiconductor memories were developed.**

**All microcomputers now use semiconductor memory which consists of RAM and ROM, made in the form of LSI circuits. The principal features of such circuits are low cost, high density and ease of use. Considerable differences exist in the types of semiconductor memory due to the wide range of manufacturing process available. These differences manifest themselves in the form of:**

* **power consumption**
* **packing density**
* **speed of operation**
* **internal organisation**
* **interface requirements**
* **methods of storage**
* **cost**

***Volatile memory***

**This describes a memory which loses its contents when the source of power is removed.**

***Read only memory***

**The contents of a read only memory may be read, but cannot be modified (written to).**

***Static memory***

**Semiconductor memory is either static or dynamic. Once data is written to a static memory cell it stays there until overwritten with new data, or the power is removed.**

***Dynamic memory***

**Semiconductor memory is either dynamic or static. Once data is written to a dynamic memory cell it must be refreshed (rewritten) periodically otherwise the electrical charge which represents the bit ‘leaks’ away. Dynamic memory, unlike static memory, requires a considerable amount of circuitry to control it. Despite this, it is still much cheaper than static memory.**

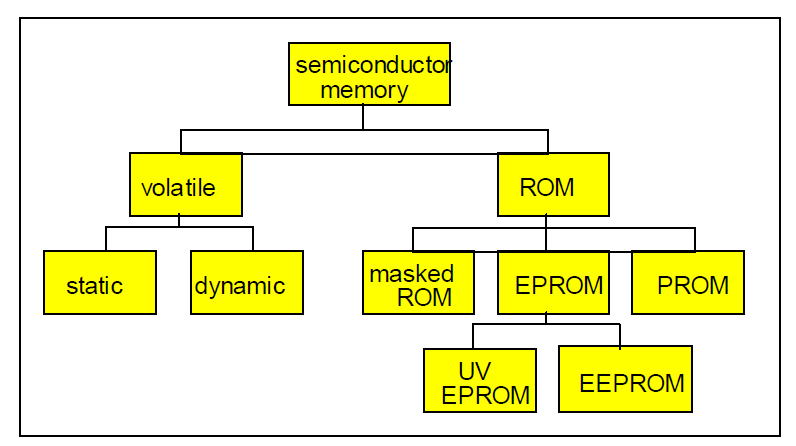
**Memory Types**

**Semiconductor memory is fabricated on silicon chips in the same fashion as microprocessors and other digital devices. The low cost of semiconductor memory (as compared to other memory devices) is the main reason for the ready availability and low cost of microcomputers nowadays. The main characteristics of semiconductor memory are low cost, high density (bits per chip), and ease of use. Apart from these characteristics, memory can be graded in terms of capacity and speed of access. A range of memory products exist, with differing characteristics. However, there are only two basic types:**

**• Those whose contents can be read and also written to (volatile). Examples of this type are DRAM and SRAM.**

**• Those whose contents can only be read (non-volatile). Some memory’s contents may be permanent, while other memory chips may be removed from the computer and reprogrammed. Examples of this type are ROM, PROM, EPROM and EEPROM.**

**The following Figure illustrates the classifications of the various types of semiconductor memory.**

****

**Volatile Memory**

**Computers use two types of RAM. These are termed Dynamic Ram (DRAM) and Static RAM (SRAM) and they have differing constructions and characteristics. These characteristics include, speed, complexity and cost. The speed of the chip is termed its *‘Access time’* and is measured in nano-seconds**

**Static Memory**

**Static memory is normally found only in smaller memories due to its high cost, and is easy to use from a designer’s point of view . A static RAM chip consists of a number of bistable elements called flip-flops. A flip-flop is a circuit consisting of 4 transistors, which can store 1 bit of data. The requirement of 4 transistors per bit of strorage is the reason why the packing density of static RAM is so low. This is the major drawback of SRAM: a high component count per bit of storage, making it very expensive with a high power consumption (as compared to dynamic RAM). Nevertheless, SRAM offers very fast access times typically about 10ns (nano - thousand millionths). It is unlikely to be used as a main computer memory, being used instead as a fast cache memory.**

**In summary, static memory**

**- is simplest from the designers point of view**

**- has a high cost**

**- consumes more power (than dynamic RAM)**

**- has a low storage density (bits per unit area)**

**- is used mainly for small fast (cache) memory**

**Dynamic Memory**

**The necessity for higher memory capacities has led to the development of dynamic memory. Compared to static ram, it has the advantages of high storage density, low cost and low power consumption. Standard dynamic RAM chips are available with capacities of about 256k x 1 bit, and typical access times of 70ns.**

**In summary, dynamic memory**

**- is complex to use from the designers point of view**

**- has a low cost**

**- consumes less power (than static RAM)**

**- has a high storage density (bits per unit area)**

**- is used in large memory systems**

**- is not as reliable as SRAM**

**Non-Volatile Memory - ROM**

**The main benefit of such chips is that their contents will not be lost if the power is removed. These chips are used in a wide variety of electronic control circuits, from industrial machine tools to domestic washing machines. They are also the ideal choice for computer control. A computer’s control programs require to be non-volatile. By placing part of the operating system software into a ROM chip, the system BIOS, the basic machine control programs are available to be run as soon as the computer is switched on. The programs in the ROM provide the machine’s basic input and output functions, to allow application programs to be loaded and run. Unfortunately, if the system is to be updated, the BIOS chip has to be replaced with a new chip mwhich contains the new program routines. This requires opening the computer case and is a job for experienced support staff or technicians. ROM chips are only capable of performing required pre-determined programs. Due to the cost of manufacturing ROMs, they are only used in large quantity runs. This, in turn, means that they are only used when the manufacturer is certain that the programs they contain are debugged.**

**MASKED ROM**

**The very first ROMs were hardwired devices that contained a preprogrammed set of data or instructions. The contents of the ROM had to be specified before chip production, so the actual data could be used to arrange the transistors inside the chip. Hardwired memories are still used, though they are now called masked ROMs to distinguish them from other types of ROM. The primary advantage of a masked ROM is its low production cost. Unfortunately, the cost is low only when large quantities of the same ROM are required.**

**PROM**

**The initials stand for *‘Programmable Read Only Memory’.* With ROM, the program was dedicated at the production stage; the program itself determined the physical construction of the ROM chip. A cheaper method for small and medium scale use is a ROM-type chip that can be programmed, after the construction stage. Such chips are mass produced by a chip manufacturer, who has no idea of the use to which they will be put. Once the chip is purchased by a computer manufacturer the company’s programs can be embedded in it. ‘This is achieved**

**by *‘blowing’* fusible links inside the chip, to form the binary codes representing the program’s machine code instructions. This is achieved using a special piece of equipment called a device programmer. Every intact link represents a binary 1, with a blown link representing a binary 0. Like the ROM, the PROM chip is also non-volatile. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as onetime programmable (OTP) devices.**

**EPROM**

**The initials stand for *‘Erasable Programmable Read Only Memory’* and it was introduced as a development tool. The problem with ROM and a programmed PROM was that, once produced, they were unalterable This is perfectly fine for computer manufacture - once the program contents are fully debugged. The EPROM is used to test an embedded program. Like PROM, its links are blown to the needs of the test program. The EPROM can then be used on the test computer. If the program is satisfactory, it can be used to create mass ROM or PROM versions.**

**If the program needs alteration, the EPROM is subjected to ultra-violet light for a few minutes. This ‘heals’ the ruptured links, allowing the chip to be blown to the next test program. The blowing and wiping clean process can be repeated many times over, before the chip starts to degenerate. An EPROM chip is easily identified, as it has a glass window on top of the chip to allow entry of the ultra-violet light. Due to its expensive construction, it is only a viable alternative to ROM and EPROM for small scale use.**

**EEPROM**

**A variation on the EPROM is the EEPROM - the *‘Electrically Erasable and Programmable Read Only Memory’*. Like the EPROM, it has the benefit of holding its contents when the power is removed. However its contents can be overwritten without resorting to prior cleaning with ultraviolet light It is currently significantly more expensive than other memory devices but is a likely candidate for future use in computers. Many palmtop computers use ROM to store application programs, to overcome the storage problems associated with small machines. Due to their size, there is no space for a hard disc to store application software, so the machine stores a word processor, spreadsheet, personal organiser, etc. in ROM. In most computers, however, the application software is loaded into, and run from, main memory.**

**Flash Memory**

**Flash memory is a form of non-volatile memory (EEPROM) that can be electrically erased and reprogrammed. It is erased and programmed in blocks consisting of multiple locations (usually 512 bytes in size). Flash memory costs far less than EEPROM and therefore has become the dominant technology wherever a significant amount of non-voltatile, solid-state storage is needed.**

**Here are a few examples of Flash memory:**

* **Your computer's BIOS chip**
* **USB flash drives**
* **CompactFlash (most often found in digital cameras)**
* **SmartMedia (most often found in digital cameras)**
* **Memory Stick (most often found in digital cameras)**