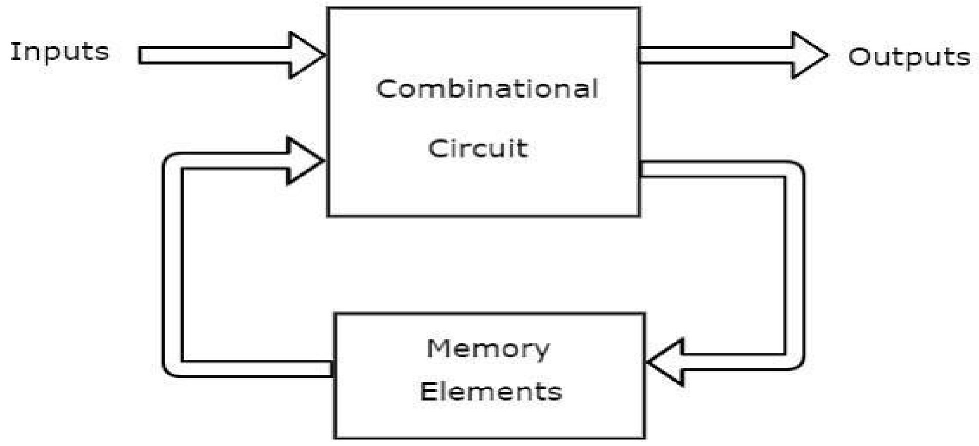
UNIT - V

SEQUENTIAL CIRCUITS

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.

Following table shows the **differences** between combinational circuits and sequential circuits.

Combinational Circuits	**Sequential Circuits**
Outputs depend only on present inputs.	Outputs depend on both present inputs and present state.
Feedback path is not present.	Feedback path is present.
Memory elements are not required.	Memory elements are required.

Clock signal is not required.	Clock signal is required.
Easy to design.	Difficult to design.

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**.Therefore, most of the outputs of asynchronous sequential circuits are **not in synchronous** with either only positive edges or only negative edges of clock signal.

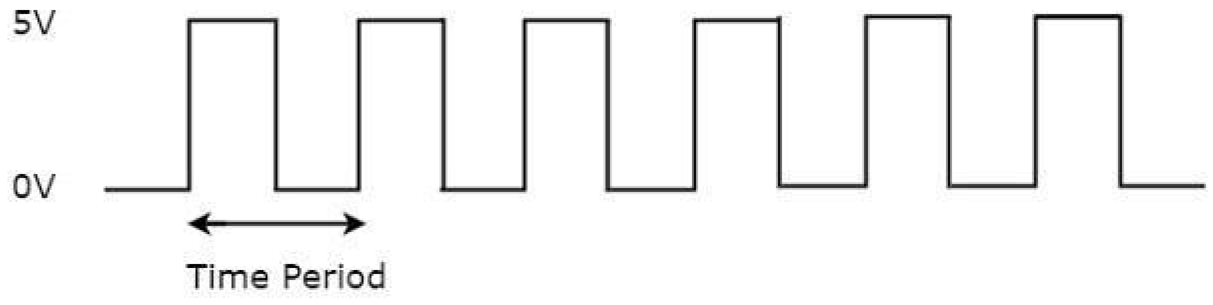
Synchronous sequential circuits

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**. That means, all the outputs of synchronous sequential circuits change (affect) at the same time. Therefore, the outputs of synchronous sequential circuits are in synchronous with either only positive edges or only negative edges of clock signal.

Clock Signal and Triggering

In this section, let us discuss about the clock signal and types of triggering one by one. **Clock signal**

Clock signal is a periodic signal and its ON time and OFF time need not be the same. We can represent the clock signal as a **square wave**, when both its ON time and OFF time are same. This clock signal is shown in the following figure.



This signal stays at logic High (5V) for some time and stays at logic Low (0V) for equal amount of time. This pattern repeats with some time period. In this case, the **time period** will be equal to either twice of ON time or twice of OFF time.

Types of Triggering

Following are the two possible types of triggering that are used in sequential circuits.

- Level triggering
- Edge triggering

Level triggering

There are two levels, namely logic High and logic Low in clock signal. Following are the two **types of level triggering**.

- Positive level triggering
- Negative level

triggering Edge triggering

There are two types of transitions that occur in clock signal. That means, the clock signal transitions either from Logic Low to Logic High or Logic High to Logic Low.

Following are the two **types of edge triggering** based on the transitions of clock signal.

- Positive edge triggering
- Negative edge triggering

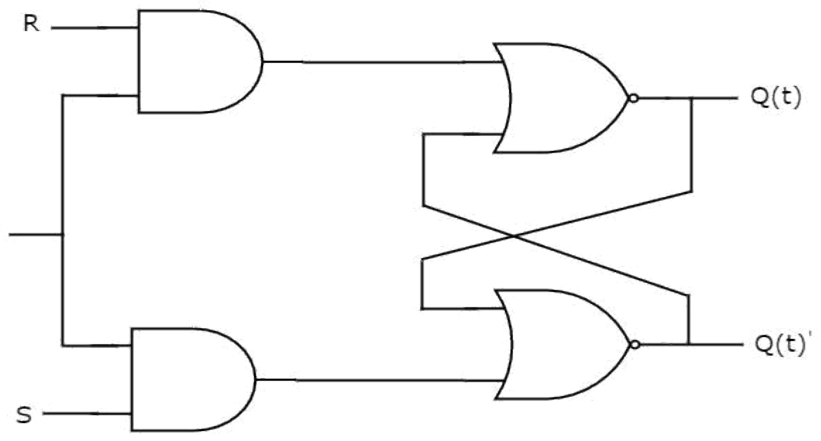
There are two types of memory elements based on the type of triggering that is suitable to operate it.

- Latches
- Flip-flops

Latches operate with enable signal, which is **level sensitive**. Whereas, flip-flops are edge sensitive. We will discuss about flip-flops in next chapter. Now, let us discuss about SR Latch & D Latch one by one.

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.



This circuit has two inputs S & R and two outputs Q(t) & Q(t)‟. The **upper NOR gate** has two inputs R & complement of present state, Q(t)‟ and produces next state, Q(t+1) when enable, E is „1‟.

Similarly, the **lower NOR gate** has two inputs S & present state, Q(t) and produces complement of next state, Q(t+1)‟ when enable, E is „1‟.

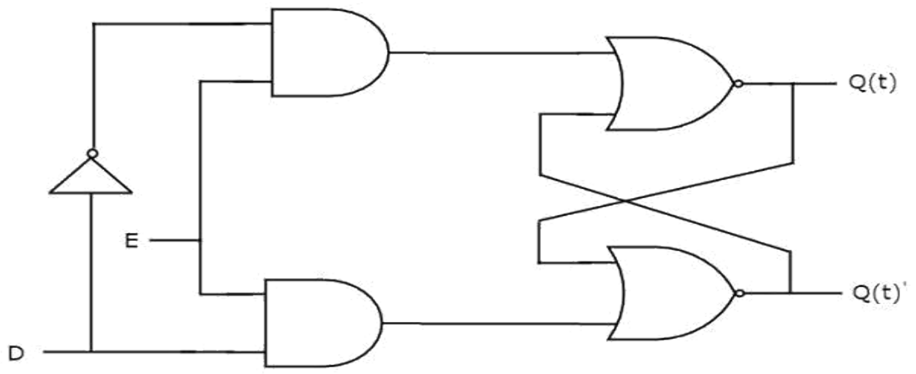
We know that a **2-input NOR gate** produces an output, which is the complement of another input when one of the input is „0‟. Similarly, it produces „0‟ output, when one of the input is „1‟.

The following table shows the **state table** of SR latch.

S	**R**	**Q(t+1)**
0	0	Q(t)
0	1	0
1	0	1
1	1	-

D Latch

There is one drawback of SR Latch. That is the next state value can‟t be predicted when both the inputs S & R are one. So, we can overcome this difficulty by D Latch. It is also called as Data Latch. The **circuit diagram** of D Latch is shown in the following figure.



The following table shows the **state table** of D latch.

D	**Q(t+1)**
0	0
1	1

In first method, **cascade two latches** in such a way that the first latch is enabled for every positive clock pulse and second latch is enabled for every negative clock pulse. So that the combination of these two latches become a flip-flop.

In second method, we can directly implement the flip-flop, which is edge sensitive. In this chapter, let us discuss the following **flip-flops** using second method.

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

SR Flip-Flop

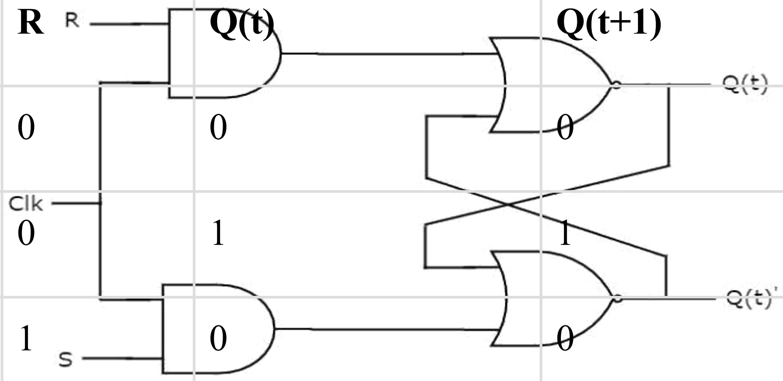
SR flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The **circuit diagram** of SR flip-flop is shown in the following figure.

The following table shows the **state table** of SR flip-flop.

S	R	Q(t+1)
0	0	Q(t+1)
0	1	0
1	0	1
1	1	-

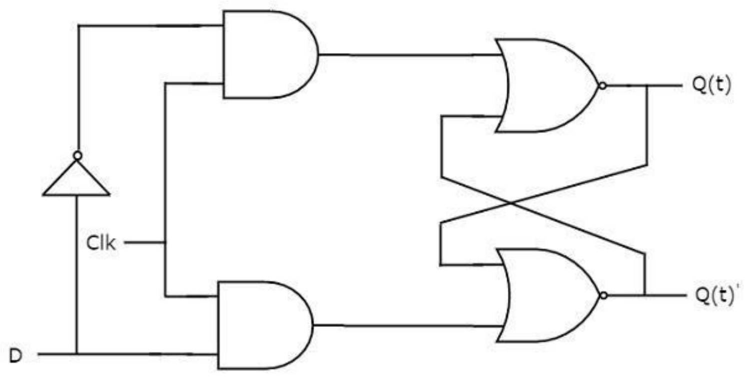
The following table shows the **characteristic table** of SR flip-flop.

Present Inputs		Present State	Next State
S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X



D Flip-Flop

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The **circuit diagram** of D flip-flop is shown in the following figure.



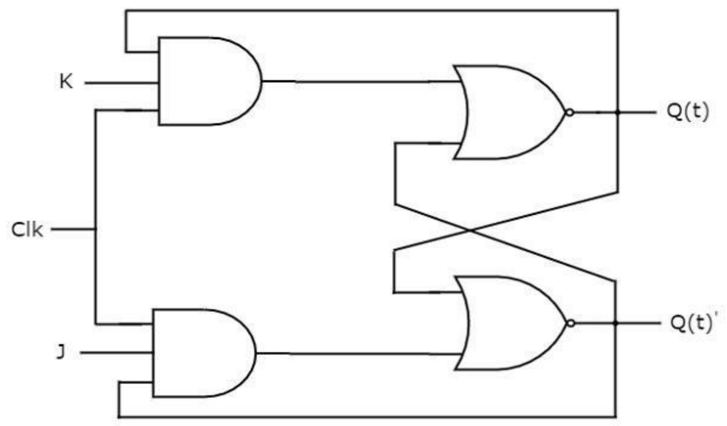
The following table shows the **state table** of D flip-flop.

D	**Q(t+1)**
0	0
0	1

From the above state table, we can directly write the next state equation as Q(t+1)=D

JK Flip-Flop

JK flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of JK flip-flop is shown in the following figure.



The following table shows the **state table** of JK flip-flop.

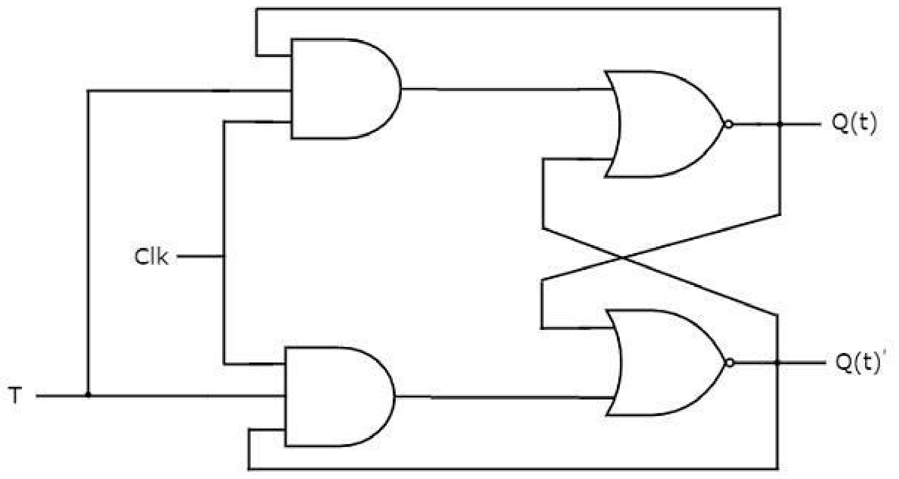
J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

The following table shows the **characteristic table** of JK flip-flop.

Present Inputs		**Present State**	**Next State**
J	**K**	**Q(t)**	**Q(t+1)**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

T Flip-Flop

T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input „T‟ to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions. The **circuit diagram** of T flip-flop is shown in the following figure.



The following table shows the **state table** of T flip-flop.

D	Q(t+1)
0	Q(t)
1	Q(t)‟

The following table shows the **characteristic table** of T flip-flop.

Inputs	**Present State**	**Next State**
T	**Q(t)**	**Q(t+1)**
0	0	0
0	1	1
1	0	1
1	1	0

From the above characteristic table, we can directly write the **next state equation** as

$Q(t+1)=T'Q(t)+TQ(t)'Q(t+1)=T'Q(t)+TQ(t)' \Rightarrow Q(t+1)=T\oplus Q(t)$

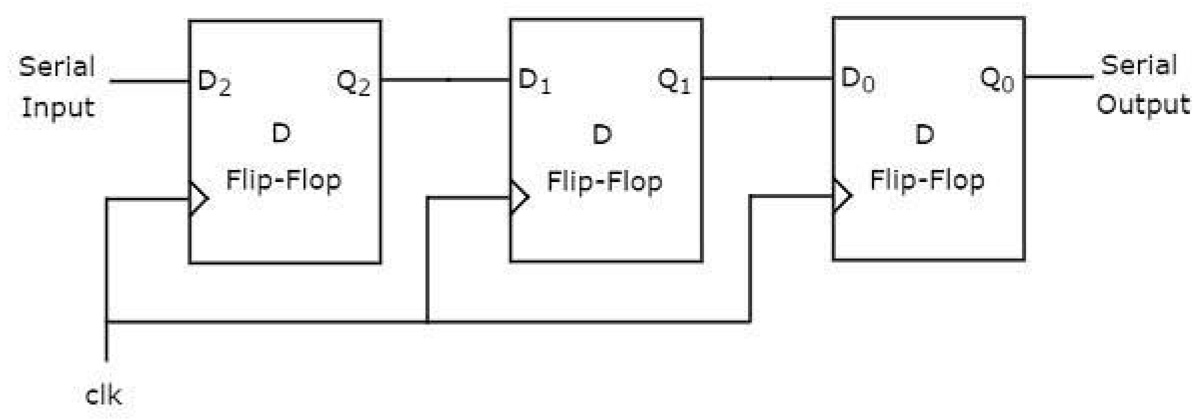
shift register:

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

Serial In - Serial Out (SISO) Shift Register

The shift register, which allows serial input and produces serial output is known as Serial In – Serial Out **(SISO)** shift register. The **block diagram** of 3-bit SISO shift register is shown in the following figure.

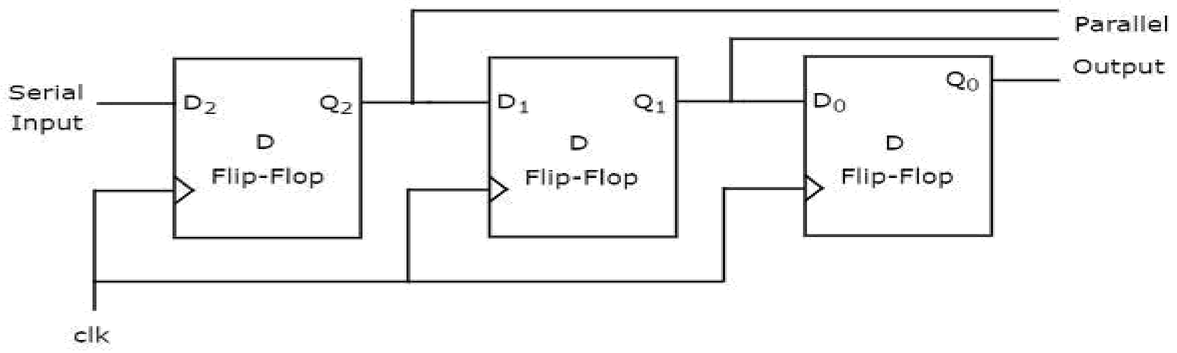


This block diagram consists of three D flip-flops, which are **cascaded**. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. So, we can receive the bits serially from the output of right most D flip-flop. Hence, this output is also called as **serial output**.

Serial In - Parallel Out (SIPO) Shift Register

The shift register, which allows serial input and produces parallel output is known as Serial In – Parallel Out **(SIPO)** shift register. The **block diagram** of 3-bit SIPO shift register is shown in the following figure.



This circuit consists of three D flip-flops, which are cascaded. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. In this case, we can access the outputs of each D flip-flop in parallel. So, we will get **parallel outputs** from this shift register.

Design of Asynchronous and Synchronous Circuits:

The 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

Now, let us discuss about these two state machines one by one.

Memory:

READ-ONLY MEMORY

Read-only memory (ROM) is a type of storage medium that permanently stores data on personal computers (PCs) and other electronic devices. It contains the programming needed to start a PC, which is essential for boot-up; it performs major input/output tasks and holds programs or software instructions.

Because ROM is read-only, it cannot be changed; it is permanent and non-volatile, meaning it also holds its memory even when power is removed. By contrast, random access memory (RAM) is volatile; it is lost when power is removed.

There are numerous ROM chips located on the motherboard and a few on expansion boards. The chips are essential for the basic input/output system (BIOS), boot up, reading and writing to peripheral devices, basic data management and the software for basic processes for certain utilities.

RANDOM ACCESS MEMORY

RAM (random access memory) is the place in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is much faster to read from and write to than other kinds of storage in a computer, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive. Data remains in RAM as long as the computer is running. When the computer is turned off, RAM loses its data. When the computer is turned on again, the OS and other files are once again loaded into RAM, usually from an HDD or SSD.

RAM TYPES

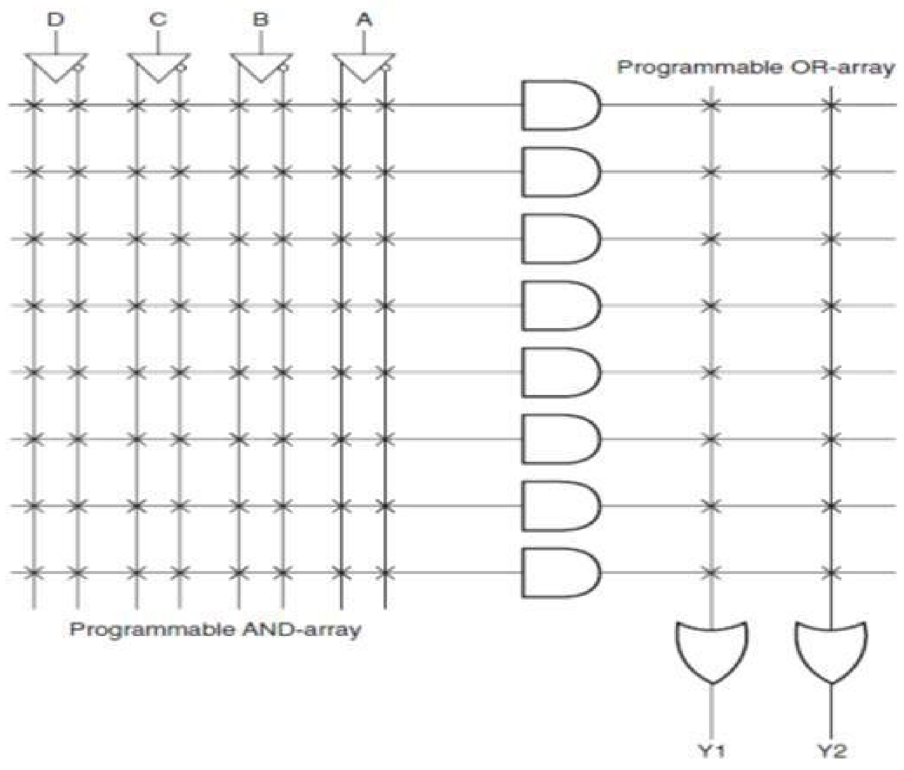
(1)Dynamic random access memory: DRAM is what makes up the typical computing device RAM and, as noted above, requires constant power to hold on to stored data.

(2)Static random access memory.:SRAM doesn't need constant power to hold on to data, but the way the memory chips are made means they are much larger and thousands of times more expensive than an equivalent amount of DRAM.

However, SRAM is significantly faster than DRAM. The price and speed differences mean SRAM is mainly used in small amounts as cache memory inside a device's processor.

PROGRAMMABLE LOGIC ARRAY

A programmable logic array (PLA) has a programmable AND array at the inputs and programmable OR array at the outputs. The PLA has a programmable AND array instead of hard-wired AND array. The number of AND gates in the programmable AND array are usually much less and the number of inputs of each of the OR gates equal to the number of AND gates. The OR gate generates an arbitrary Boolean function of minterms equal to the number of AND gates. Figure below shows the PLA architecture with four input lines, a programmable array of eight AND gates at the input and a programmable array of two OR gates at the output.



ADVANTAGES

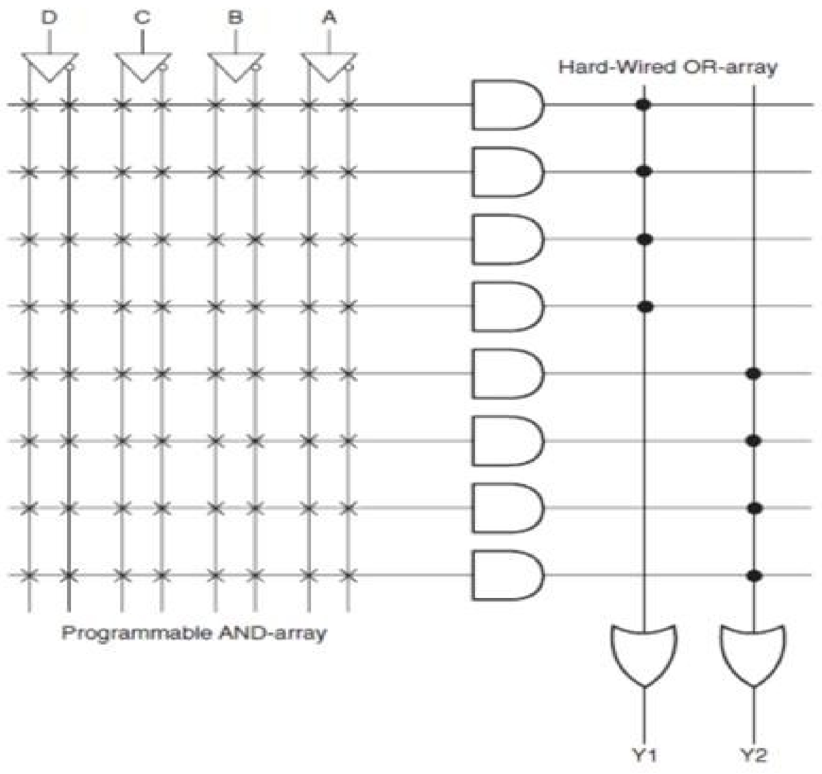
PLA architecture more efficient than a PROM.

DISADVANTAGE

PLA architecture has two sets of programmable fuses due to which PLA devices are difficult to manufacture, program and test.

PROGRAMMABLE ARRAY LOGIC

Programmable array logic (PAL) has a programmable AND array at the input and a fixed OR array at the output. The programmable AND array of a PAL architecture is same as that of the PLA architecture. The number of programmable AND gates in PAL architecture are smaller than the number of minterms. The OR array is fixed and the AND outputs are divided between OR gates.



Memory decoding:

Memory decoding :n The equivalent logic of a binary cell that stores one bit of information is shown below. Read/Write = 0, select = 1, input data to S-R latch Read/Write = 1, select = 1, output data from S-R latch.

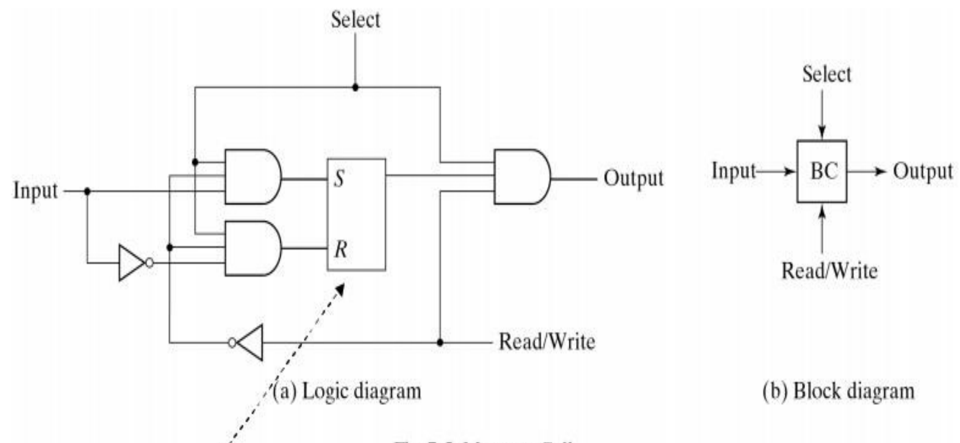


Fig. 7-5 Memory Cell

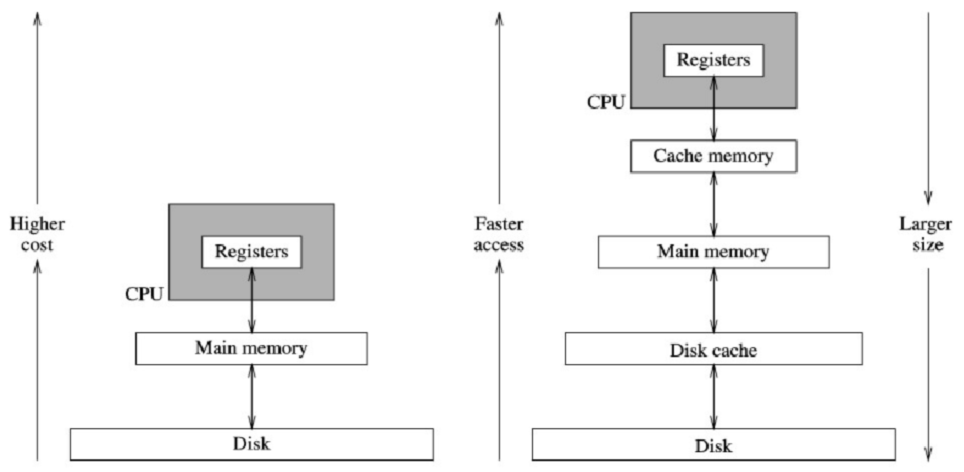
Cache memory:

Cache memory is a small amount of fast memory

∗Placed between two levels of memory hierarchy » To bridge the gap in access times

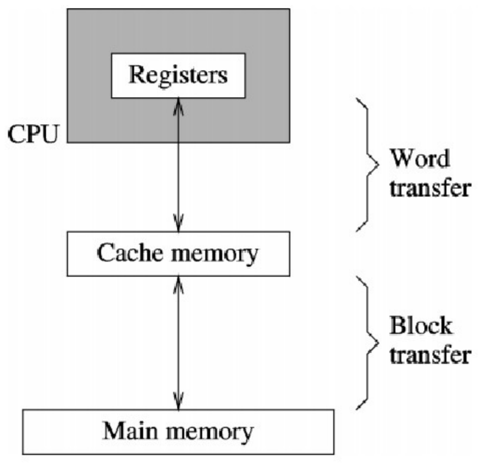
– Between processor and main memory (our focus)

– Between main memory and disk (disk cache)
∗Expected to behave like a large amount of fast memory



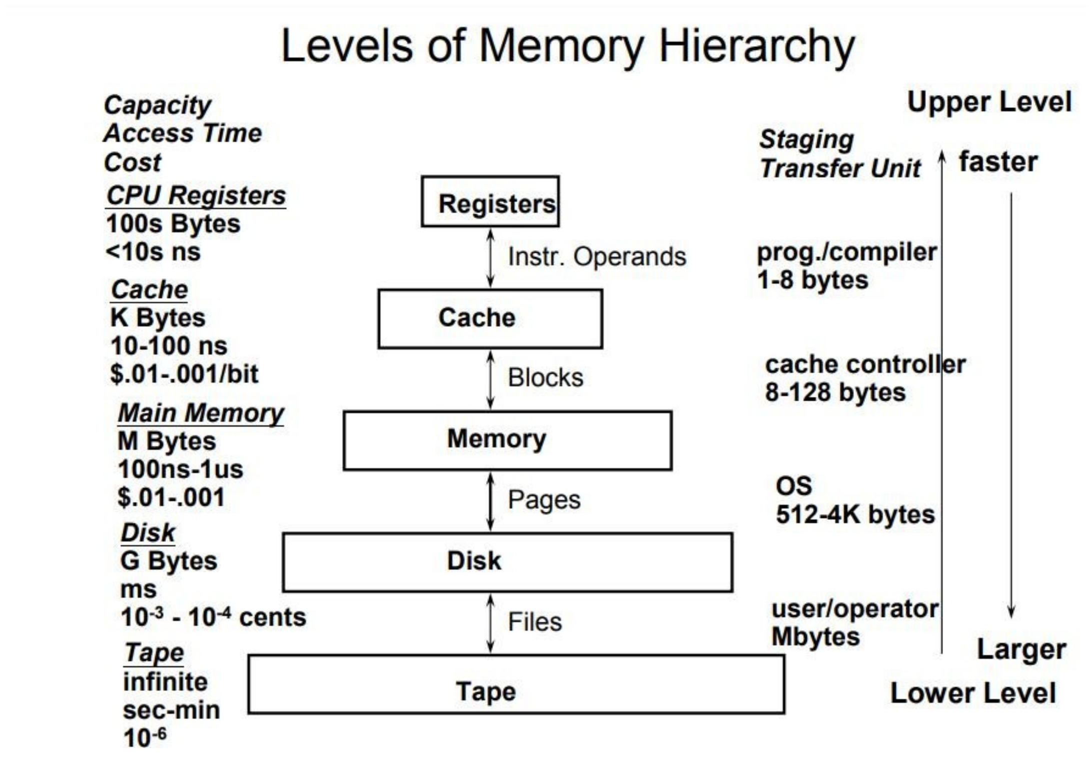
(a) (b)

- Transfer between main memory and cache
  - ∗ In units of blocks
  - ∗ Implements spatial locality
- Transfer between main memory and cache
  - ∗ In units of words
- Need policies for
  - ∗ Block placement
  - ∗ Mapping function
  - ∗ Block replacement
  - ∗ Write policies



- Determines how memory blocks are mapped to cache lines
- Three types
  - ∗ Direct mapping
    - » Specifies a single cache line for each memory block
  - ∗ Set-associative mapping
    - » Specifies a set of cache lines for each memory block
  - ∗ Associative mapping
    - » No restrictions
      - – Any cache line can be used for any memory block

Levels of memory Hierarchy:


Levels of Memory Hierarchy
Capacity
Access Time
Cost
CPU Registers
100s Bytes
<10s ns
Cache
K Bytes
10-100 ns
$.01-.001/bit
Main Memory
M Bytes
100ns-1us
$.01-.001
Disk
G Bytes
ms
10-3 - 10-4 cents
Tape
infinite
sec-min
10-6
Registers
Instr. Operands
Cache
Blocks
Memory
Pages
Disk
Files
Tape
Upper Level
Staging
Transfer Unit
faster
prog./compiler
1-8 bytes
cache controller
8-128 bytes
OS
512-4K bytes
user/operator
Mbytes
Larger
Lower Level