Data Storage, Transfer (Resisters) and Counters

Data Storage and Data Transfer

- Asynchronous transfers are controlled by PRE and CLR inputs.
- Transferring the bits of a register simultaneously is a parallel transfer.
- Transferring the bits of a register a bit at a time is a serial transfer.
- A group of flip flop (FFs) used for temporary data storage is known as register.

° Multiple flip flops can be combined to form a data register

° Shift registers allow data to be transported one bit at a time

° Registers also allow for parallel transfer

- **Many bits transferred at the same time**
- The operation most often performed on data that are stored in a FF or resister is the data transfer operation. This involves the transfer of data from one FF or resister to another.

Register Data Transfer

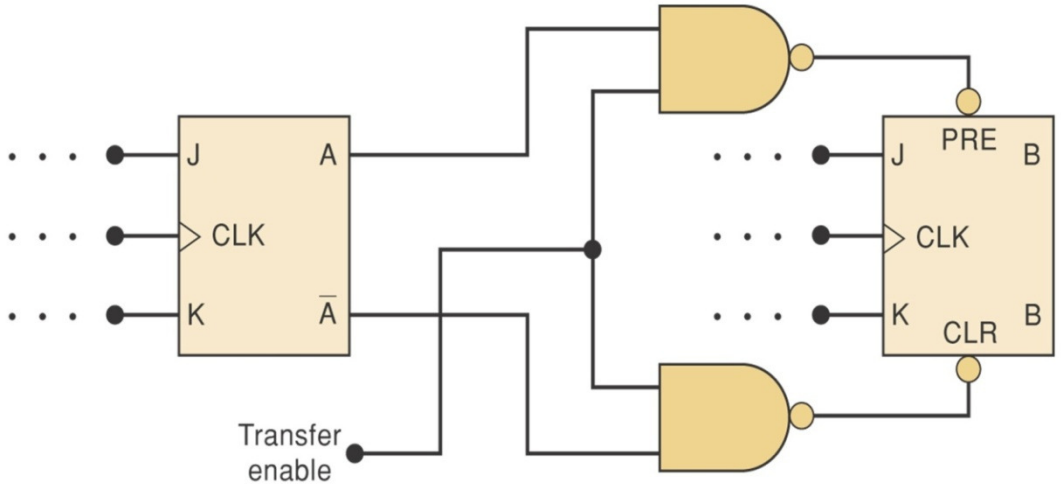
- The various types of registers can be classified according to…
  - The manner in which data can be entered into the register for storage.
  - The manner in which data are outputted from the register.
- Serial data flow through a register is generally called *shifting*—either to the left or to the right.
  - Serial output data fed back to the input of the same register is called a *data rotate*.

- Parallel inputting of data is often described as a *register load*.

- Shift registers can be used with adders to build arithmetic units

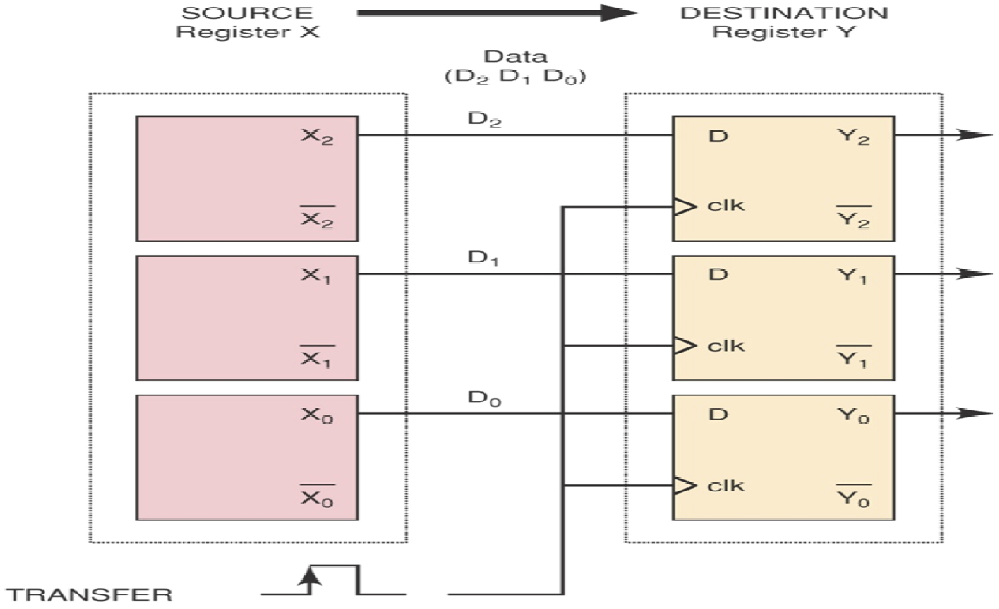
° Remember: most digital hardware can be built from combinational logic (and, or, invert) and flip flops : Basic components of most computers

Asynchronous Data Transfer Operation



- Uses PRE and CLR inputs to load data into FF
- PRE and CLR won't be both low at the same time
  - A = 1, EN =1, PRE = 0, sets B = 1
  - A =0, EN =1, CLR = 0, sets B = 0

Parallel transfer of contents of register *X* into register *Y*



Shift Resister and Resister's Classifications

Serial Data Transfer: Shift Registers

- **Parallel transfers – register contents are transferred simultaneously with a single clock cycle.**
- **Serial transfers – register contents are transferred one bit at a time, with a clock pulse for each bit.**
- **Serial transfers are slower, but the circuitry is simpler. Parallel transfers are faster, but circuitry is more complex.**
- **Serial and parallel are often combined to exploit the benefits of each.**

Resister Classifications :-

a) Serial in Serial Out (SISO)

b) Serial in Parallel Out (SIPO)

c) Parallel in Serial Out (PISO)

d) Parallel in Parallel Out (PIPO)

Bit Shift Register with Asynchronous Parallel Loading

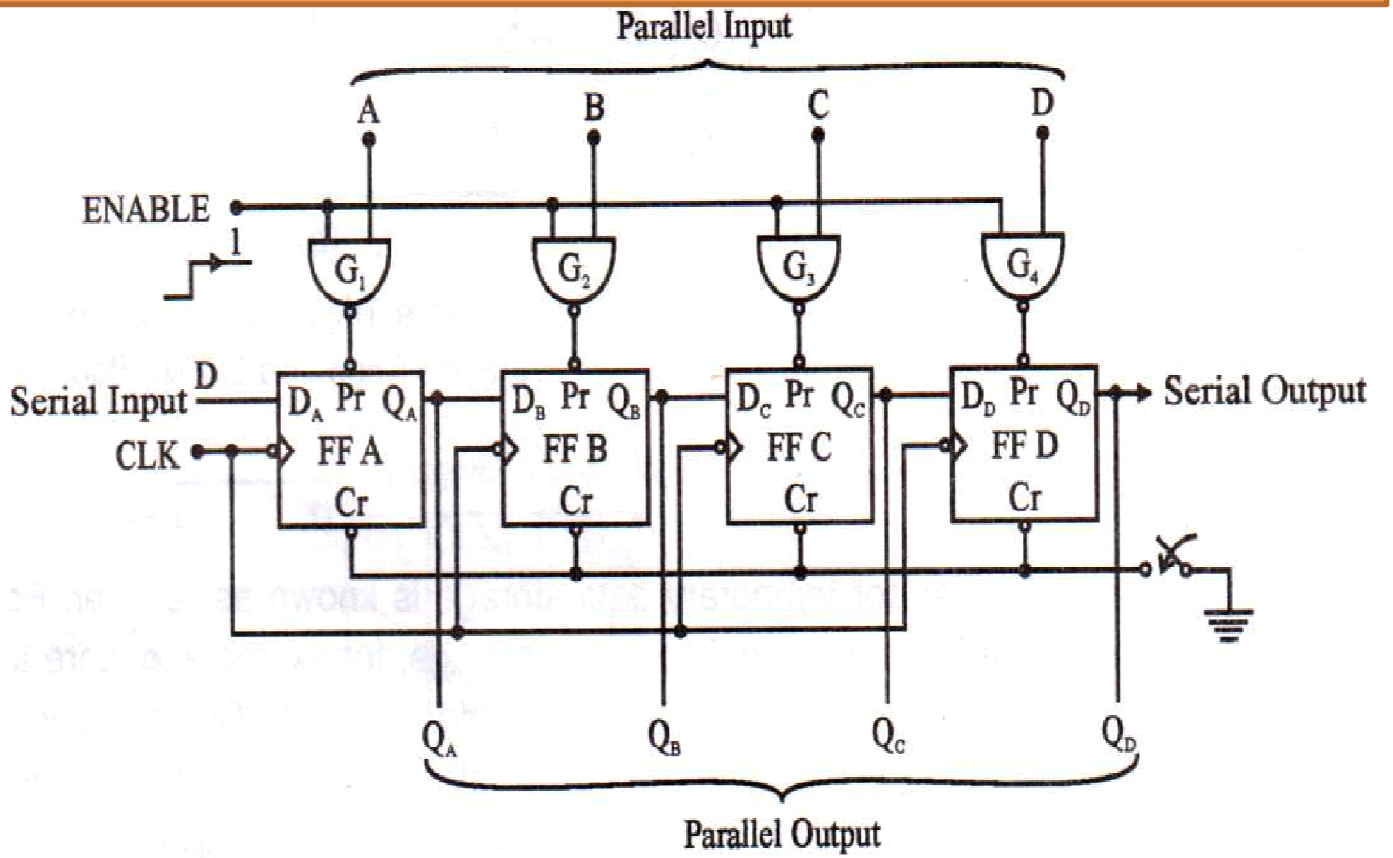


Fig. 4 Bit Shift Register

1. Four D filp flops used to construct resister.
2. Before loading the data D input, first cleared all contents by giving low(=0) pulse at CLEAR
3. Then keep it at high logic Open switch for normal operation.

Serial in Serial Out (SISO)

(for binary data $(1101)_2$

Step 1

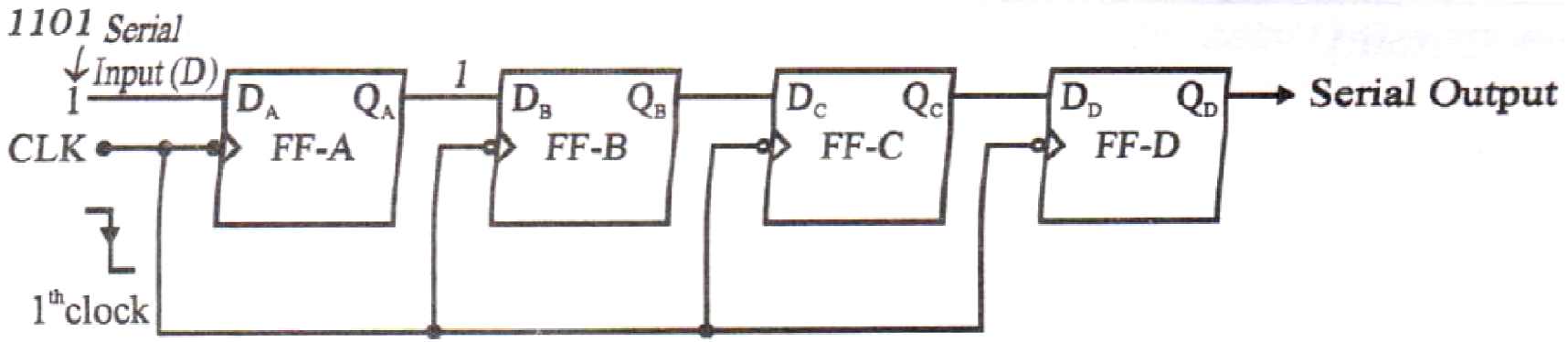


Fig. Data Bit Position After 1st Clock

Step 2

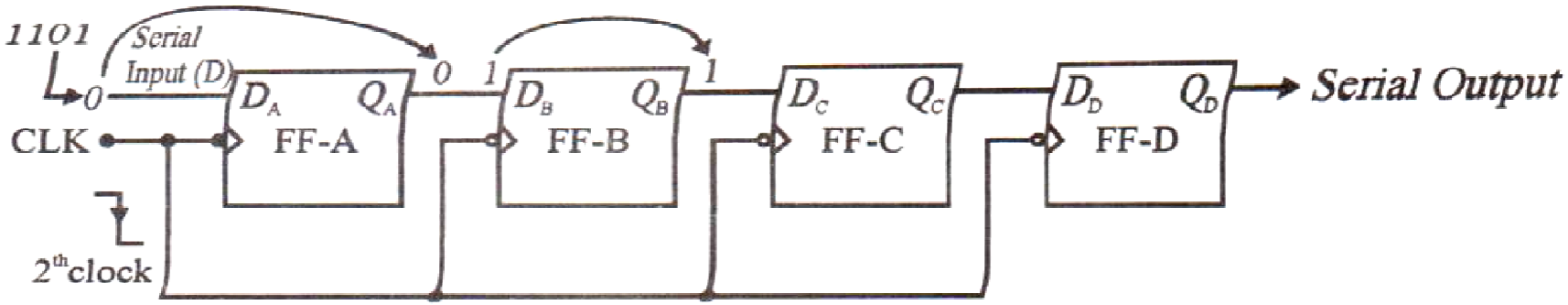


Fig. Data Bit Position After 2nd Clock

Step 3

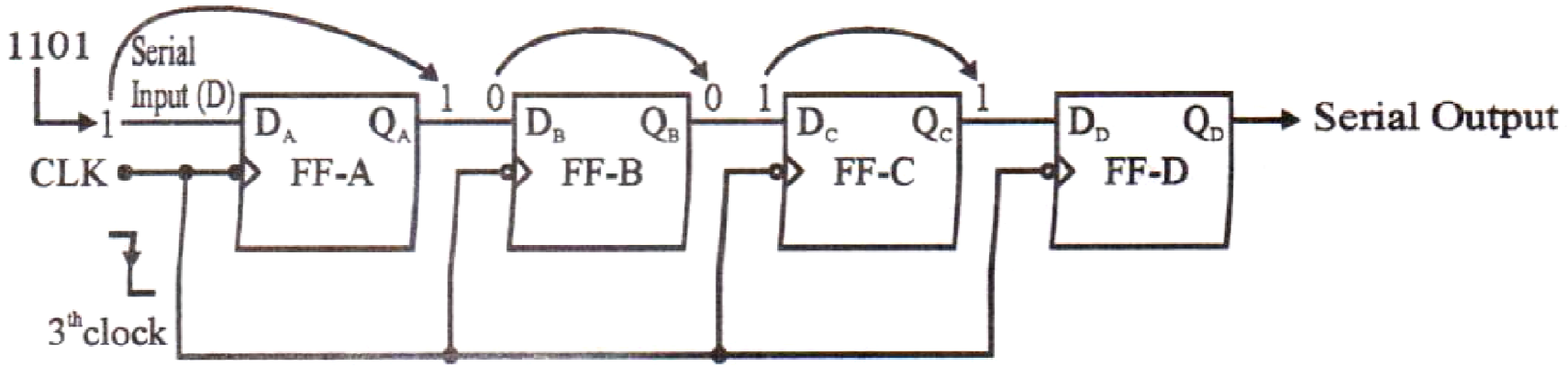


Fig. Data Bit Position After 3rd Clock

Step 4

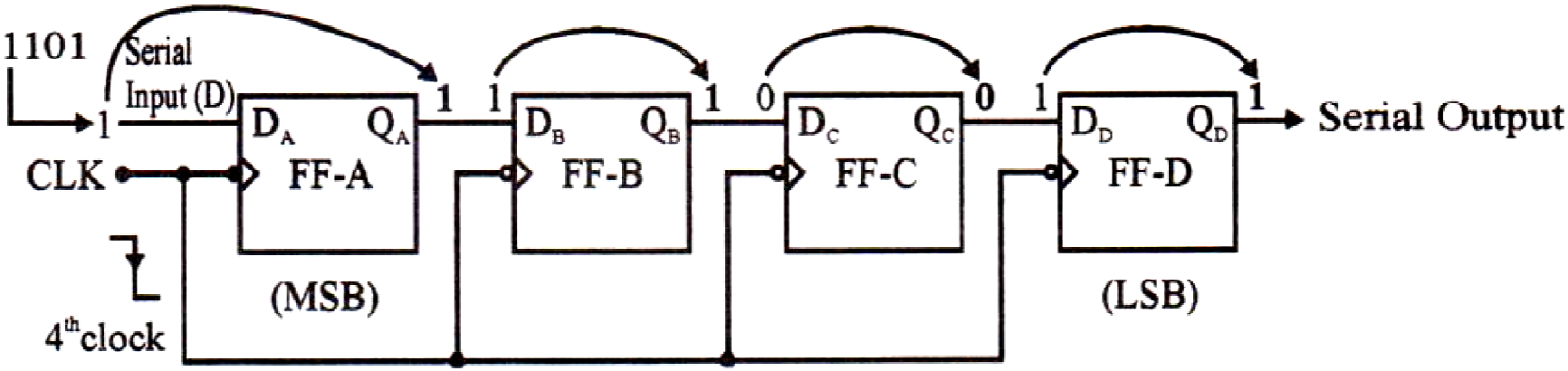


Fig. Data Bit Position After 4th Clock

Serial in/serial out (SISO)

Serial Output :

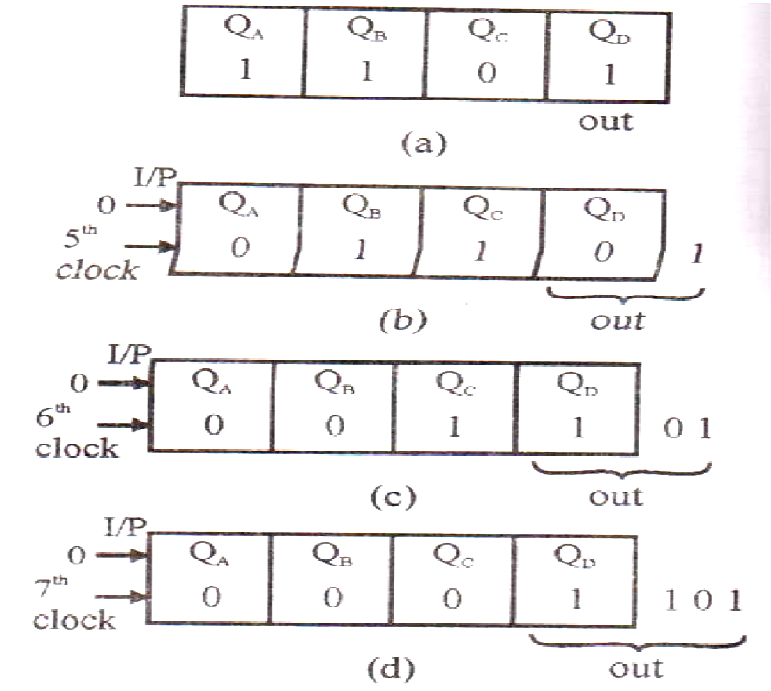
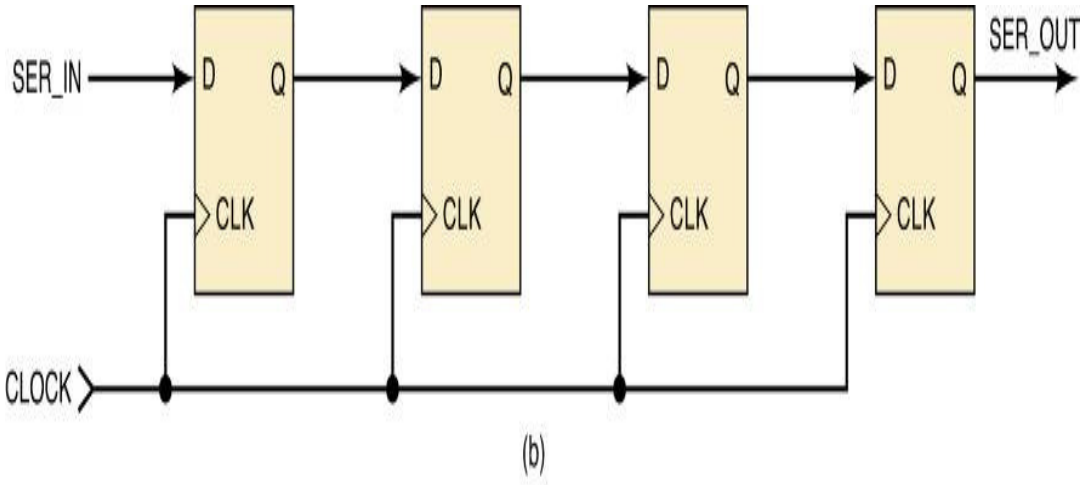


Fig. (a) Data Position after 4th clock pulses (b),(c)&(d) data Position after 5th ,6th & 7th Clock

74ALS166/74HC166 Serial in/serial out (SISO)



Data loaded one bit at a time moves one bit at a time, with each clock pulse through the flip-flops toward the other end of the register, and exit one bit at a time in the same order as originally loaded.

Serial in Parallel Out (SIPO)

(for binary data $(1101)_2$

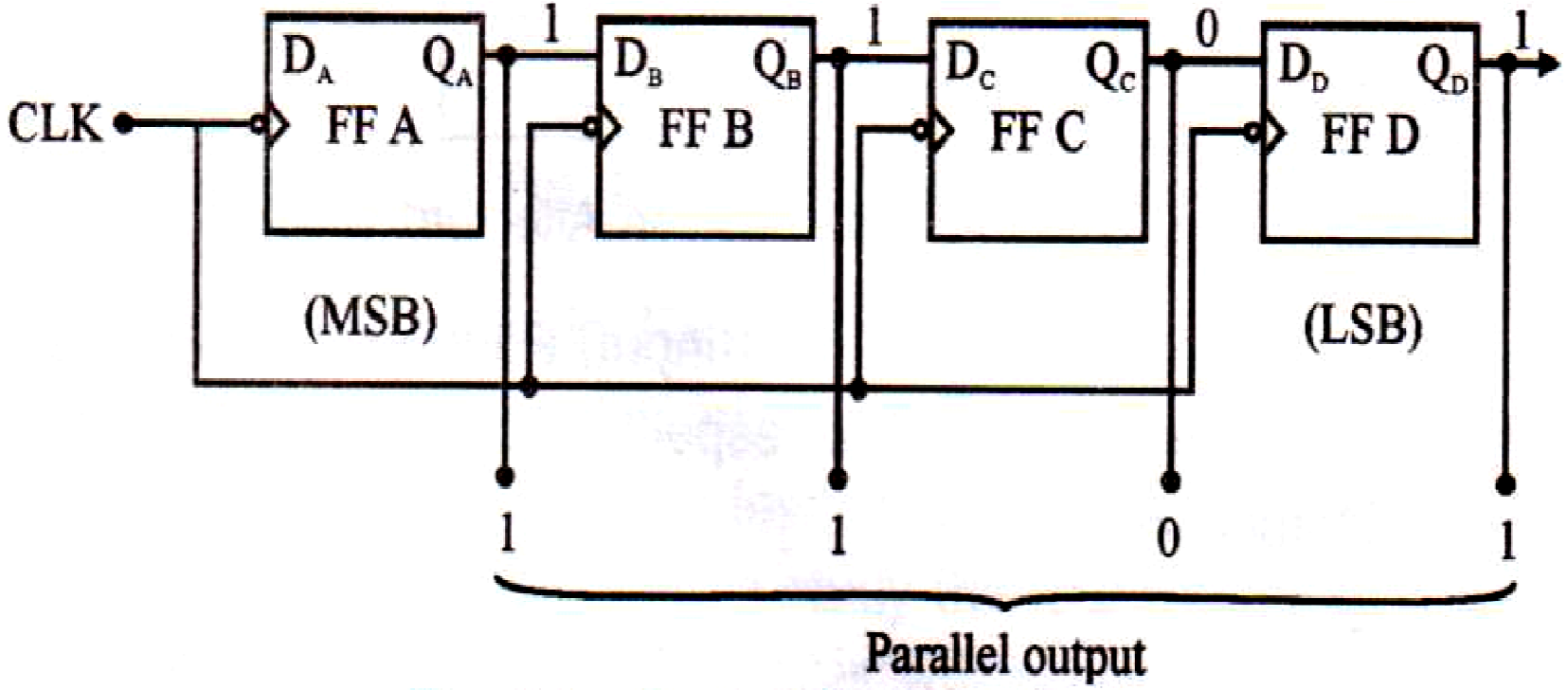


Fig. Parallel Out Configuration

Parallel in Serial Out (PISO)

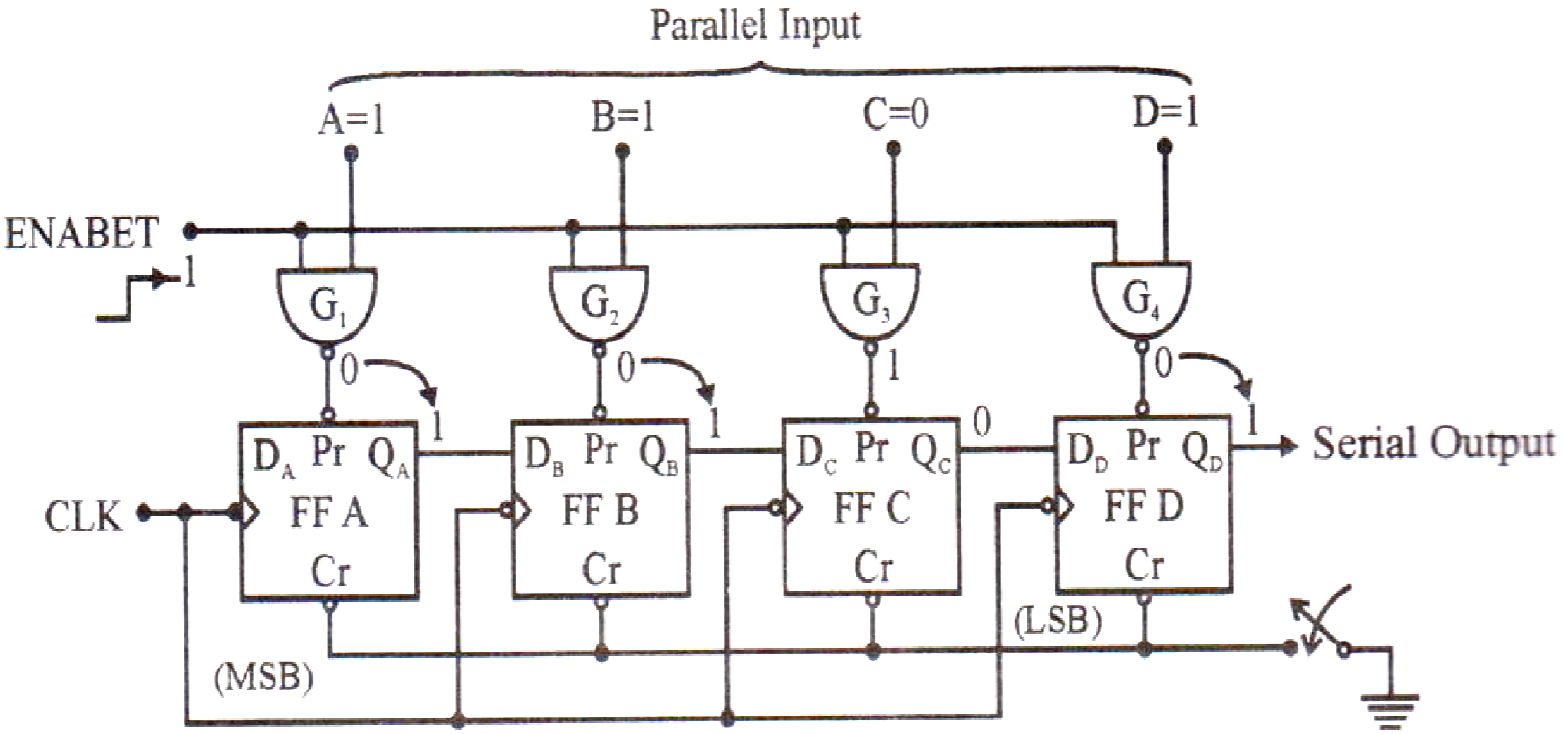


Fig. Parallel In Serial Out Configuration

1. Apply the **"Cr"** pulse in input to clear the contents and output becomes 0000.
2. Input the data in parallel form, when clock pulse is **"ENABLE".**
3. The **"Pr"** input of respected flip-flop becomes 0010 etc.
4. As preset inputs are active low inputs a binary low (0), sets (1) the respective flip flop and high at preset input make no change at output of the FF.
5. This type of data loading is called asynchronous loading as the loading is independent of the system clock.

Parallel in Parallel Out (PIPO)

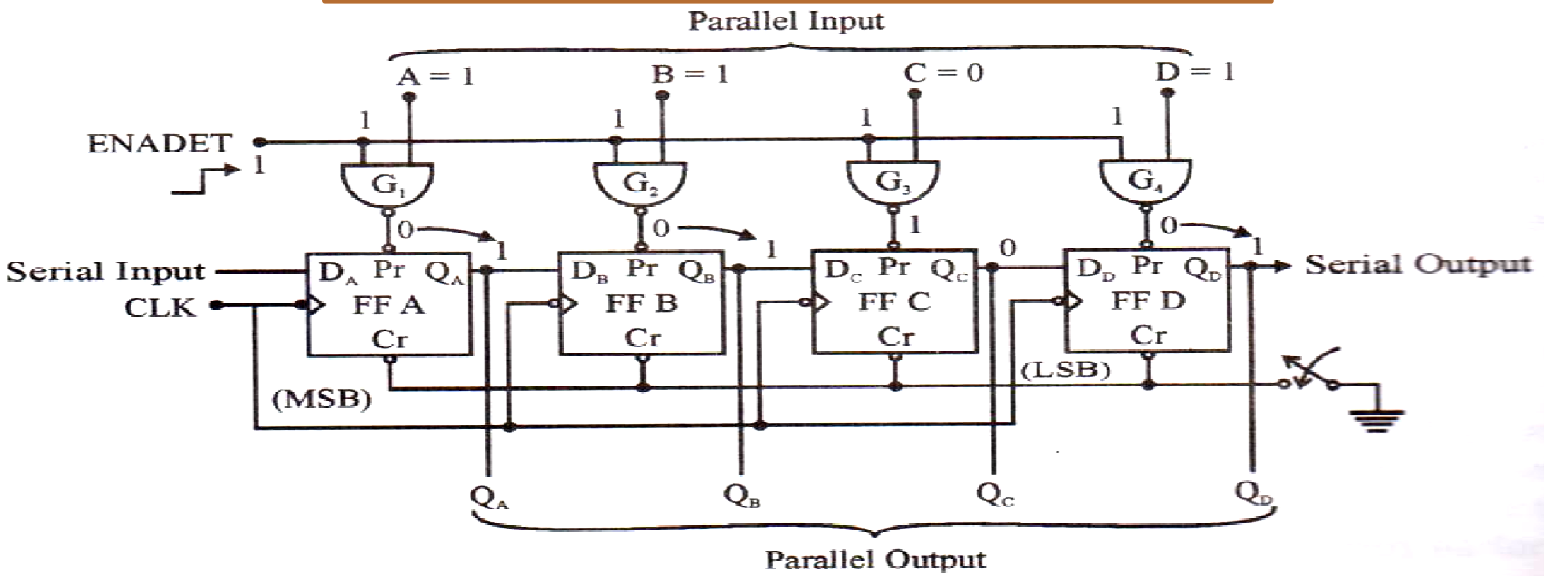
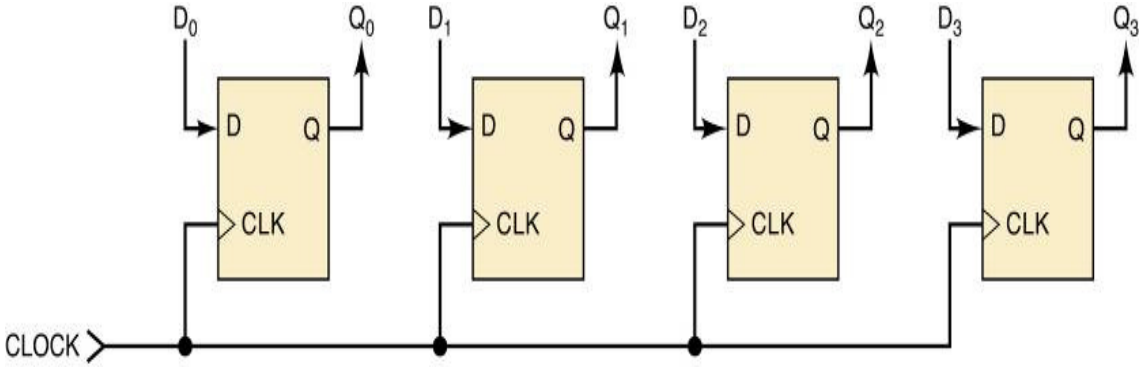


Fig. Parallel In Parallel Out Configuration

74ALS174/74HC174 Parallel in/parallel out (PIPO)



A group of flip-flops that can store multiple bits simultaneously and in which all bits of the stored binary value are directly available.

Shift Register Counters

° **Counters are important components in computers**

- Counter: A register that goes through a prescribed series of states
- The increment or decrement by one in response to input
- Shift-register counters use *feedback*—the output of the last FF in the register is connected back to the first FF in some way.

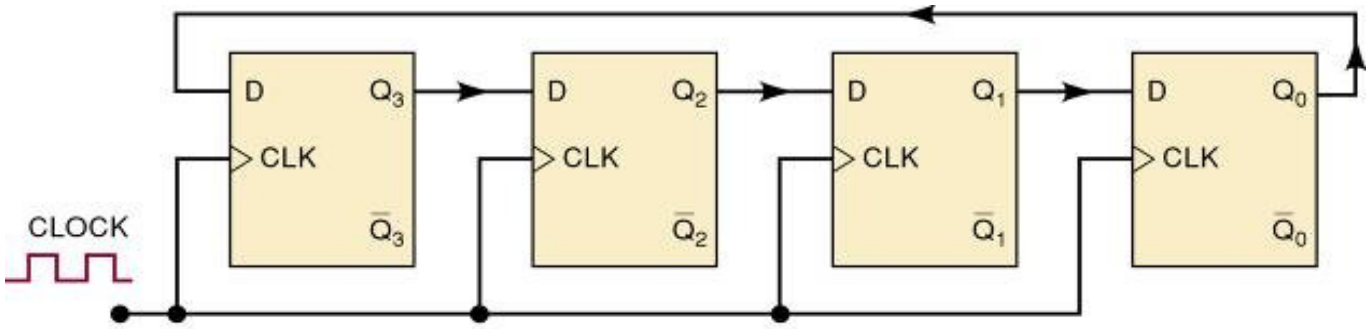
° A **ring counter** is a **circulating shift register** connected so the last FF shifts its value to the first.

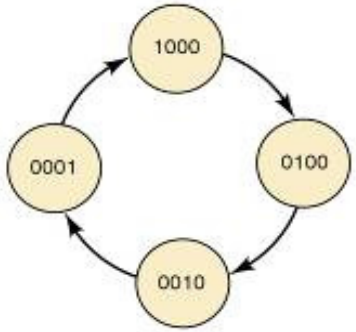
Two main types of counters

- **Ripple (asynchronous) counters :** *Flip flop output serves as a source for triggering other flip flops*
- **Synchronous counters :** *All flip flops triggered by a clock signal*

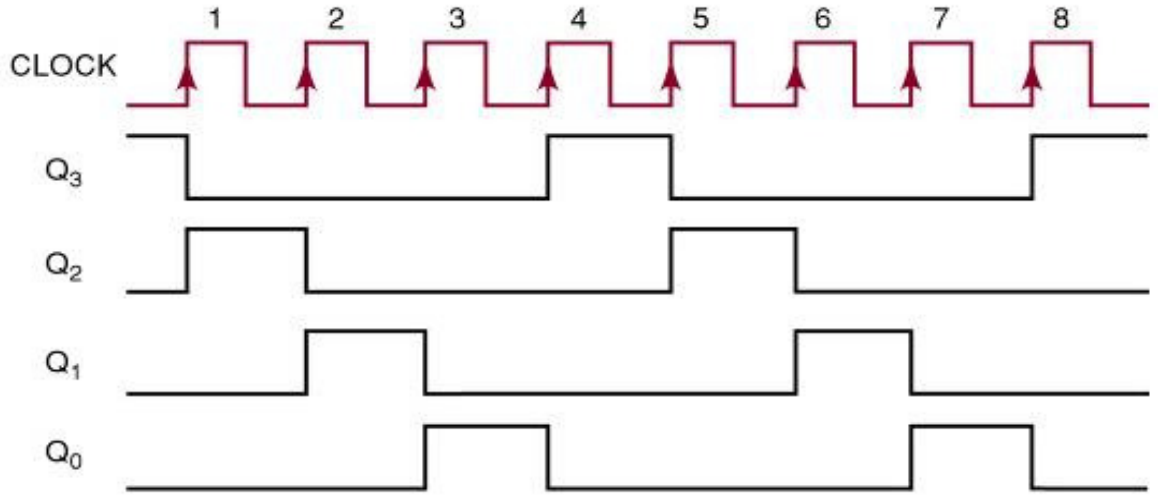
Applications: ***Watches, Clocks, Alarms, Web browser refresh***

Shift Register Ring Counter





Q_3	Q_2	Q_1	Q_0	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.
.	.	.	.	.



- To operate properly, a ring counter must start off with only one FF in the 1 state and all the others in the 0 state.

- As power-up starting states will be unpredictable, the counter is preset *before* clock pulses are applied.