IDE ALEXIUS AZIBANYE

COMPUTER ENGINEERING

17/ENG02/032

COE 312 ASSIGNMENT

SUMMARY OF THE TWO PDF DOCUMENTS: FLIP FLOP APPLICATIONS/REGISTERS & COUNTERS

FLIP FLOP APPLICATIONS

A flip-flop is a circuit that has two stable states and can be used to store state information – a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems.

TYPES OF FLIP FLOPS:

- **SR FLIP FLOP:** An SR flip-flop has two inputs, labeled SET and RESET. If the SET input is HIGH when the clock is triggered, the Q output goes HIGH. If the RESET input is HIGH when the clock is triggered, the Q output goes LOW.
- **JK FLIP FLOP:** A JK flip-flop has two inputs, labeled *J* and *K.* The J input corresponds to the SET input in an SR flip-flop, and the K input corresponds to the RESET input. When both the J and K inputs are HIGH, the Q output is *toggled*, which means that the output alternates between HIGH and LOW.
- **D FLIP FLOP:** Has just one input in addition to the CLOCK input. This input is called the DATA input. When the clock is triggered, the Q output is matched to the DATA input. Thus, if the DATA input is HIGH, the Q output goes HIGH, and if the DATA input is LOW, the Q output goes LOW.
- **T FLIP FLOP:** This is simply a JK flip-flop whose output alternates between HIGH and LOW with each clock pulse. Thus, whenever a clock pulse is received, the current state of the **Q** output is inverted (that's what the **Q-bar** output is) and fed back into the D input.

FLIP FLOP APPLICATIONS:

Application of the flip flop circuit mainly involves in bounce elimination switch, data storage, data transfer, latch, registers, counters, frequency division, memory, etc.

Flip-flops can be either simple (transparent or asynchronous) or clocked (synchronous). In the context of hardware description languages, the simple ones are commonly described as latches, while the clocked ones are described as flip-flops.

FLIP FLOP SYNCHRONIZATION:

Most digital systems are principally synchronous in their operation because most of the signals will change states in synchronism with the clock transitions. In many cases, however, there will be an external signal that is not synchronized to the clock; in other words, it is asynchronous. Asynchronous signals often occur as a result of human operators actuating an input switch at some random time relative to the clock signal. This randomness can produce unpredictable and undesirable results.

TIMING IMPLEMENTATIONS:

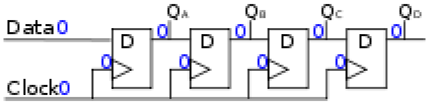
D FLIP FLOP:

The D flip-flop is widely used. It is also known as a "data" or "delay" flip-flop.

Most D-type flip-flops in ICs have the capability to be forced to the set or reset state (which ignores the D and clock inputs), much like an SR flip-flop. Usually, the illegal S = R = 1 condition is resolved in D-type flip-flops. Setting S = R = 0 makes the flip-flop behave as described above. Here is the truth table for the other possible S and R configurations:

Inputs				**Outputs**	
S	**R**	**D**	**>**	**Q**	**Q**
0	**1**	**X**	**X**	**0**	**1**
1	**0**	**X**	**X**	**1**	**0**
1	**1**	**X**	**X**	**1**	**1**

These flip-flops are very useful, as they form the basis for shift registers, which are an essential part of many electronic devices. The advantage of the D flip-flop over the D-type "transparent latch" is that the signal on the D input pin is captured the moment the flip-flop is clocked, and subsequent changes on the D input will be ignored until the next clock event. An exception is that some flip-flops have a "reset" signal input, which will reset Q (to zero), and may be either asynchronous or synchronous with the clock.

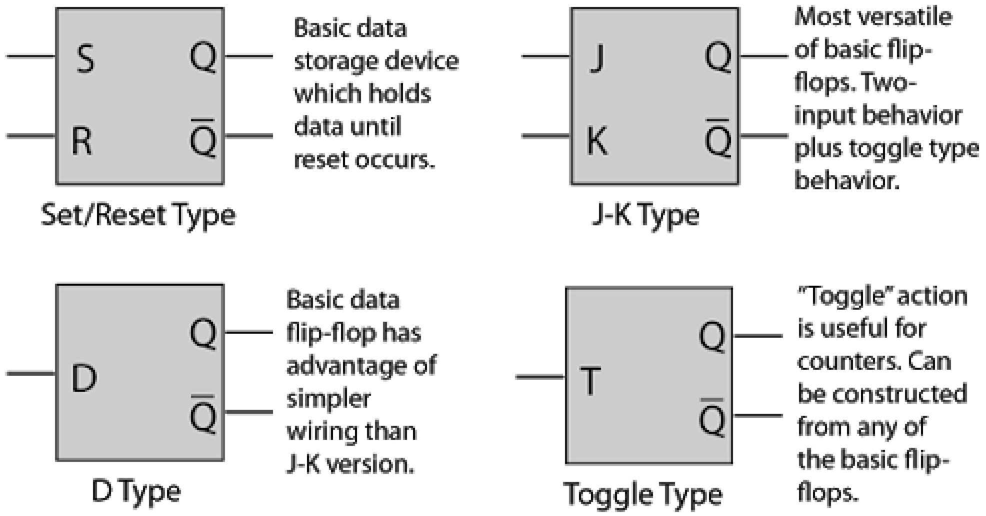


JK FLIP FLOP:

The JK flip-flop augments the behavior of the SR flip-flop (J: Set, K: Reset) by interpreting the J = K = 1 condition as a "flip" or toggle command. Specifically, the combination J = 1, K = 0 is a command to set the flip-flop; the combination J = 0, K = 1 is a command to reset the flip-flop; and the combination J = K = 1 is a command to toggle the flip-flop, i.e., change its output to the logical complement of its current value. Setting J = K = 0 maintains the current state. To synthesize a D flip-flop, simply set K equal to the complement of J (input J will act as input D). Similarly, to synthesize a T flip-flop, set K equal to J. The JK flip-flop is therefore a universal flip-flop, because it can be configured to work as an SR flip-flop, a D flip-flop, or a T flip-flop.

DATA STORAGE AND TRANSFER

Clock signal is applied simultaneously to all **flip flops**. With each clock pulse information is transferred to next **flip flop** as shown below. In the above four bit shift right the information from input could be transferred to output in four clock pulses. This is an example of **data transfer**. A **Flip-flop** is a clock-controlled memory device. ... It **stores** the input state and outputs the **stored** state only in response to the CLOCK signal. If a **Flip-flop** accepts its inputs at L to H (H to L) transition, it is Positive-Edge (Negative-Edge) Triggered. A **Flip-flop** is use to **store** one bit of information.

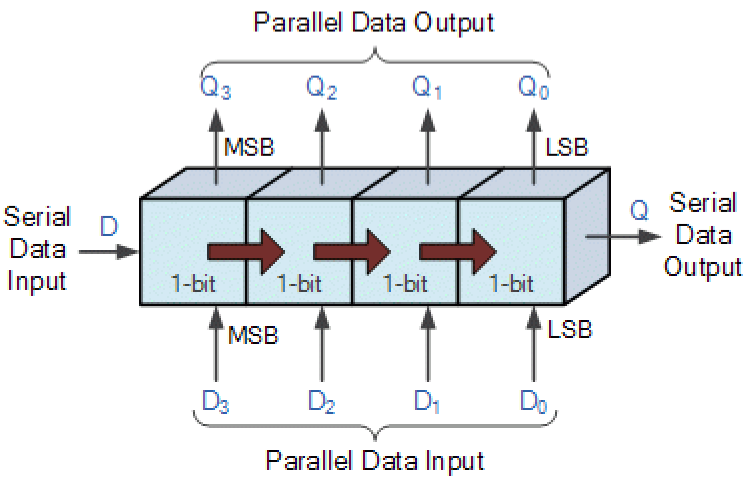

S
Q
R
$\overline{Q}$
Basic data storage device which holds data until reset occurs.
Set/Reset Type
J
Q
K
$\overline{Q}$
Most versatile of basic flip-flops. Two-input behavior plus toggle type behavior.
J-K Type
D
Q
$\overline{Q}$
Basic data flip-flop has advantage of simpler wiring than J-K version.
D Type
T
Q
$\overline{Q}$
"Toggle" action is useful for counters. Can be constructed from any of the basic flip-flops.
Toggle Type

SERIAL DATA TRANSFER: SHIFT REGISTER:

The Shift Register: The Shift Register is another type of sequential logic circuit that can be used for the storage or the transfer of binary data. This sequential device loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**.

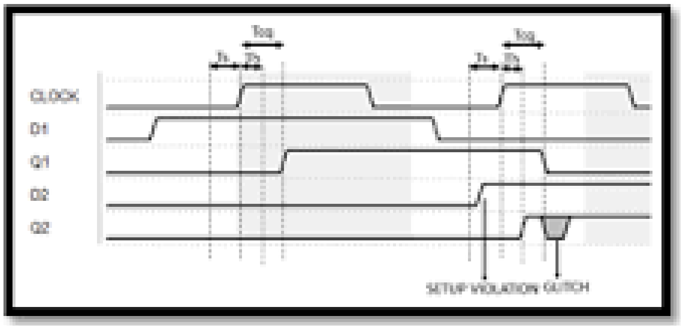
A *shift register* basically consists of several single bit “D-Type Data Latches”, one for each data bit, either a logic “0” or a “1”, connected together in a serial type daisy-chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Data bits may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction, or all together at the same time in a parallel configuration.



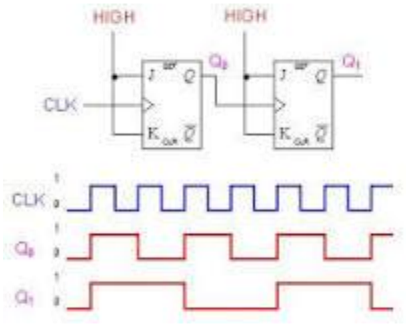
HOLD TIME REQUIREMENTS

Hold time is the minimum amount of **time** a synchronous data input should be held steady after the clock event so that the data input is reliably sampled by the clock event. If the input of the sequential logic changes in the setup window (setup **time** duration) then setup violation occurs.



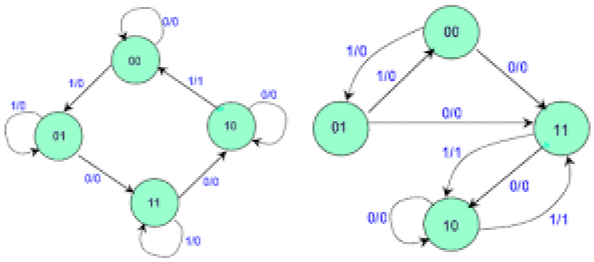
FREQUENCY DIVISION AND COUNTING

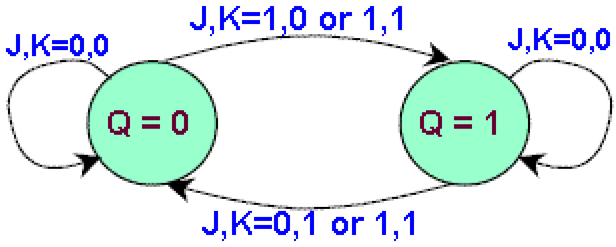
For **frequency division**, toggle mode **flip-flops** are used in a chain as a **divide** by two **counter**. The final output clock signal will have a **frequency** value equal to the input clock **frequency** divided by the MOD number of the **counter**. Such circuits are known as “**divide**-by-n” counters.



STATE DIAGRAM

State diagrams of the four types of **flip-flops**. You can see from the table that all four **flip-flops** have the same number of **states** and **transitions**. Each **flip-flop** is in the set **state** when Q=1 and in the reset **state** when Q=0. Also, each **flip-flop** can move from one **state** to another, or it can re-enter the same **state**.





REGISTERS AND COUNTERS

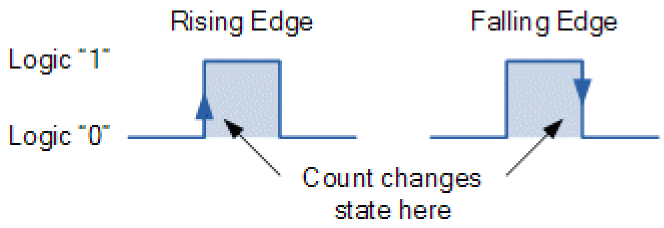
COUNTER:

A counter is essentially a register that goes through a predetermined sequence of states. a counter is a device which stores the number of times a particular event or process has occurred, often in relationship to a clock. The most common type is a sequential digital logic circuit with an input line called the clock and multiple output lines.

Types of Electronic Counters

- **Synchronous counter**.
- **Asynchronous** Counter or **Ripple Counter**.
- Up/Down Counter.
- Decade Counter.
- **Ring counter**.
- Cascaded counter.
- **Johnson counter**.
- Modulus counter.

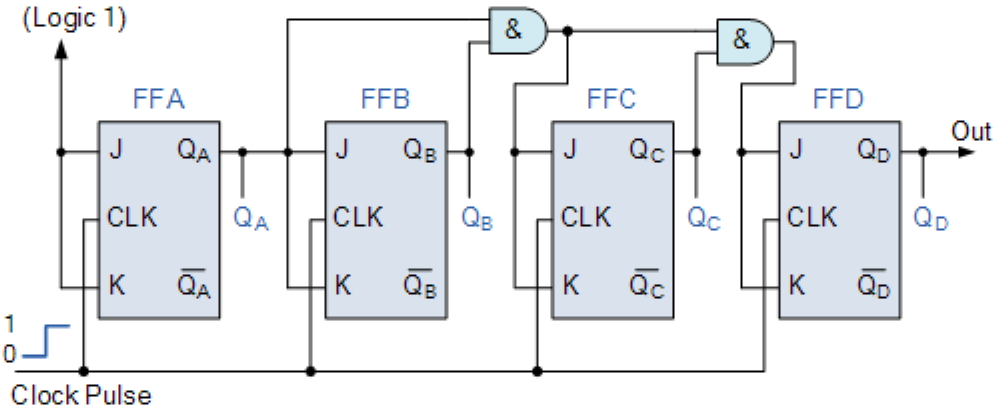
Synchronous Counter: Synchronous Counters are so called because the clock input of all the individual flip-flops within the counter are all clocked together at the same time by the same clock signal.



Synchronous Counter, the external clock signal is connected to the clock input of EVERY individual flip-flop within the counter so that all of the flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship. In other words, changes in the output occur in "synchronization" with the clock signal.

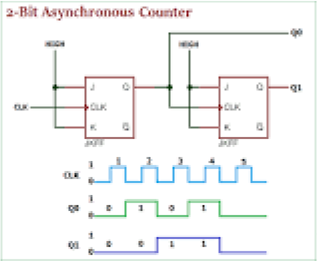
The result of this synchronization is that all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect and therefore, no propagation delay.

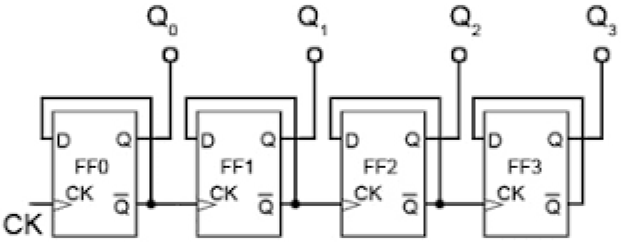
Binary 4-bit Synchronous Up Counter



It can be seen above, that the external clock pulses (pulses to be counted) are fed directly to each of the J-K flip-flops in the counter chain and that both the J and K inputs are all tied together in toggle mode, but only in the first flip-flop, flip-flop FFA (LSB) are they connected HIGH, logic "1" allowing the flip-flop to toggle on every clock pulse. Then the synchronous counter follows a predetermined sequence of states in response to the common clock signal, advancing one state for each pulse.

Asynchronous Counters: Asynchronous counters are those whose output is free from the clock signal. Because the flip flops in asynchronous counters are supplied with different clock signals, there may be delay in producing output. The required number of logic gates to design asynchronous counters is very less. So they are simple in design.

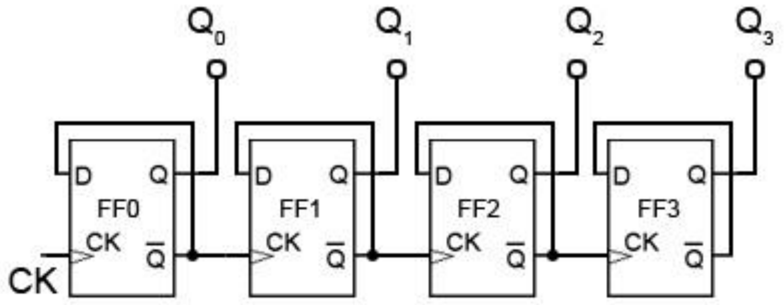




Different types of Asynchronous counters

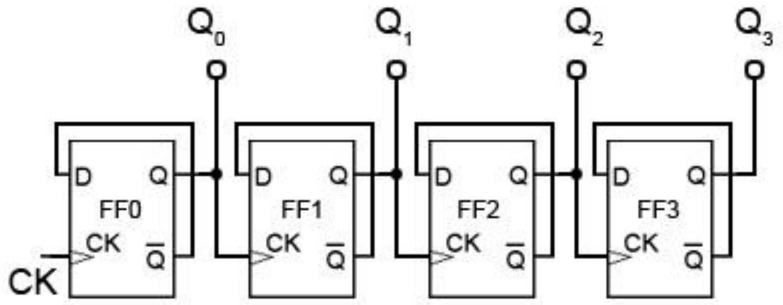
There are many types of Asynchronous counters available in digital electronics. They are

- 4 bit synchronous UP counter
- 4 bit synchronous DOWN counter
- 4 bit synchronous UP / DOWN counter



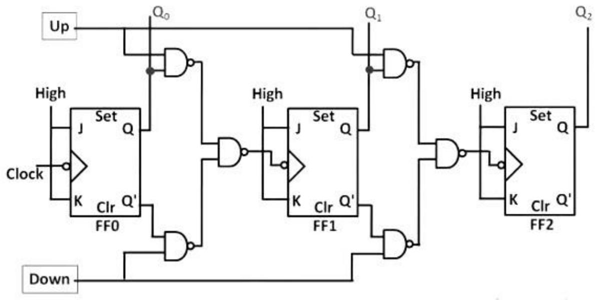
A 4 bit asynchronous UP counter with D flip flop is shown in above diagram. It is capable of counting numbers from 0 to 15.

4 bit synchronous DOWN counter:



A 4 bit asynchronous DOWN counter is shown in above diagram. It is simple modification of the UP counter. 4 bit DOWN counter will count numbers from 15 to 0, downwards.

Asynchronous 3-bit up/down counters: By adding up the ideas of UP counter and DOWN counters, we can design asynchronous up /down counter. The 3 bit asynchronous up/ down counter is shown below.



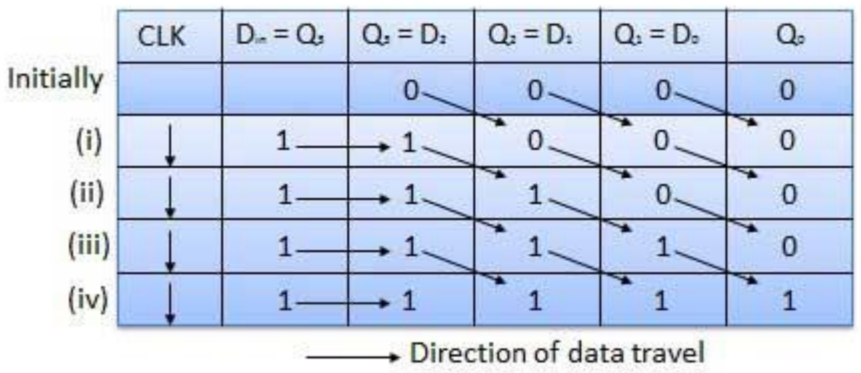
UP Counting: If the UP input and down inputs are 1 and 0 respectively, then the NAND gates between first flip flop to third flip flop will pass the non-inverted output of FF 0 to the clock input of FF 1.

DOWN Counting: If the DOWN input and up inputs are 1 and 0 respectively, then the NAND gates between first flip flop to third flip flop will pass the inverted output of FF 0 to the clock input of FF 1.

REGISTERS

Registers are data storage devices that are more sophisticated than latches. A **register** is a group of binary cells suitable for holding binary information. A group of cascaded flip-flops used to store related bits of information is known as a **register**.

TRUTH TABLE:



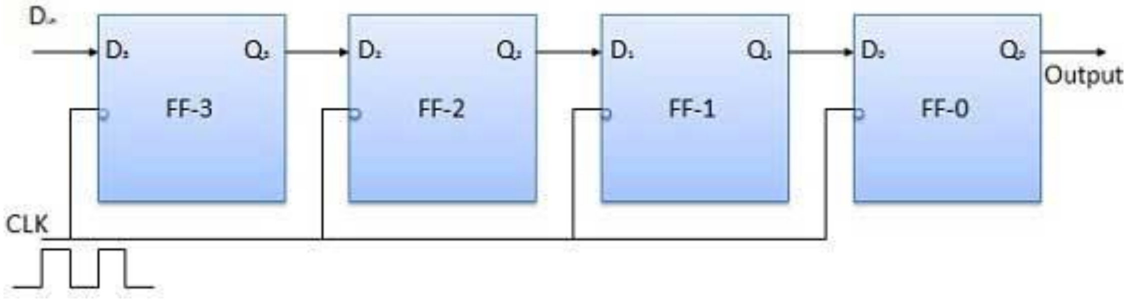
	CLK	$D_{in} = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

→ Direction of data travel

The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as shift registers. There are four mode of operations of a shift register.

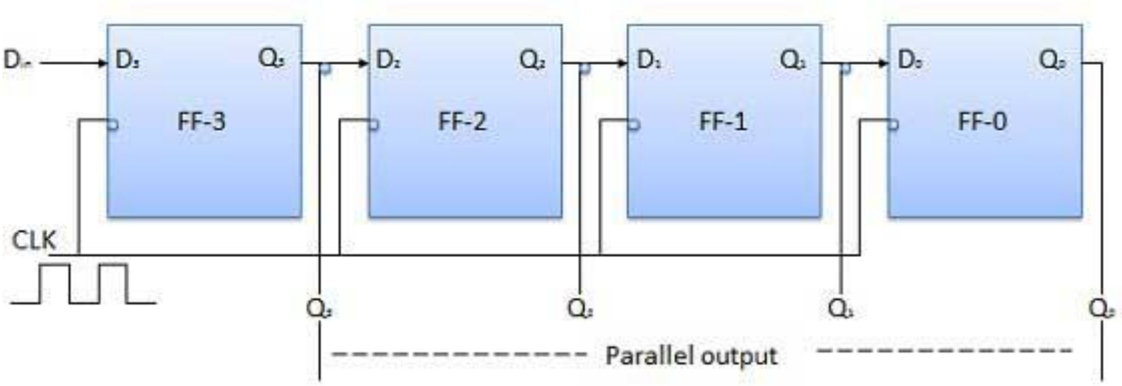
- Serial Input Serial Output (SISO)
- Serial Input Parallel Output (SIPO)
- Parallel Input Serial Output (PISO)
- Parallel Input Parallel Output (PIPO)

Serial Input Serial Output: Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four-bit binary number “1 1 1 1” is made into the register, this number should be applied to Din bit with the LSB bit applied first. The D input of FF-3 i.e. D3 is connected to serial data input Din. Output ofFF-3 i.e. Q3 is connected to the input of the next flip-flop i.e. D2 and so on.



Serial Input Parallel Output:

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four-bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

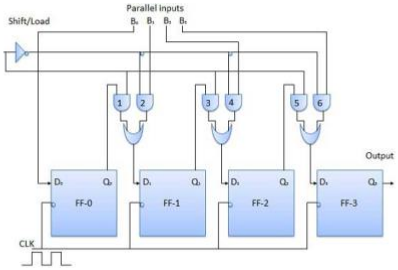


Parallel Input Serial Output PISO:

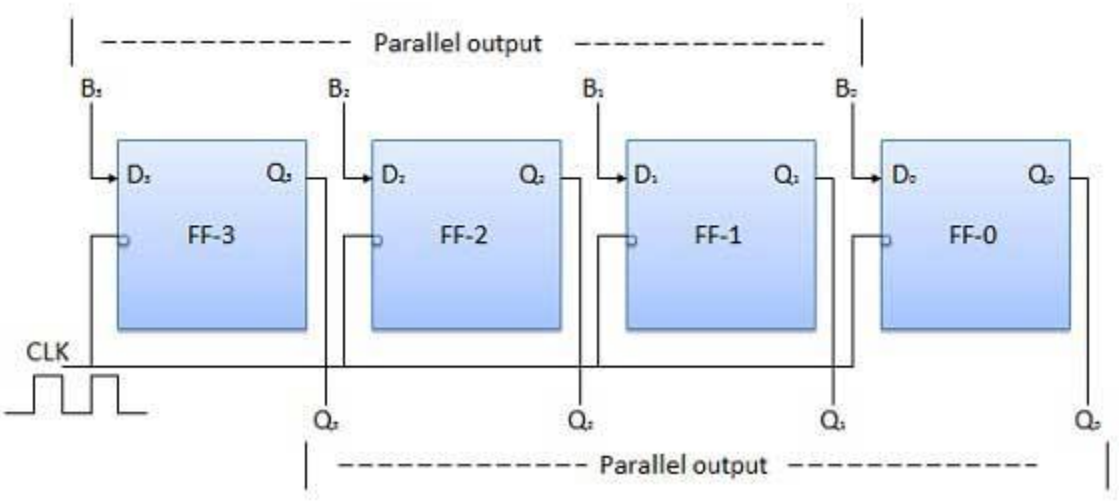
- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B0, B1, B2, B3 is applied though the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode: When the shift/load bar line is low 0, the AND gate 2, 4 and 6 become active they will pass B1, B2, B3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B0, B1, B2, B3 will get loaded into the corresponding flip-flops. Thus, parallel loading takes place.

Shift mode: When the shift/load bar line is low 1, the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore, the shifting of data from left to right bit by bit on application of clock pulses. Thus, the parallel in serial out operation takes place.



Parallel Input Parallel Output (PIPO): In this mode, the 4-bit binary input B0, B1, B2, B3 is applied to the data inputs D0, D1, D2, D3respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



Bidirectional Shift Register:

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly, if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four-bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input M.

Universal Shift Register: A shift register which can shift the data in only one direction is called a unidirectional shift register. A shift registers which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallelly, is known as a universal shift register. The shift register is capable of performing the following operation.

- Parallel loading
- Lift shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.