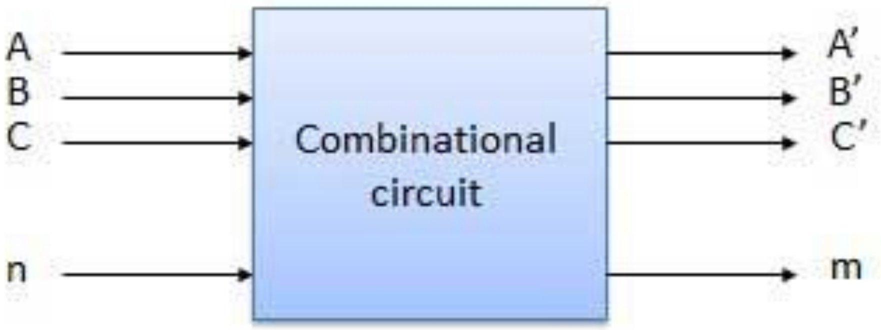
Combinational and Sequential Circuit

TOPICS

- **Combinational Circuit and Sequential Circuit Criterions**
- **Some Examples of Combinational Circuit: Adder, subtractor, Encoder, Decoder, Multiplexers, De-multiplexers.**
- **Some Examples of Sequential Circuits: Flip-flop, Register, Counters.**

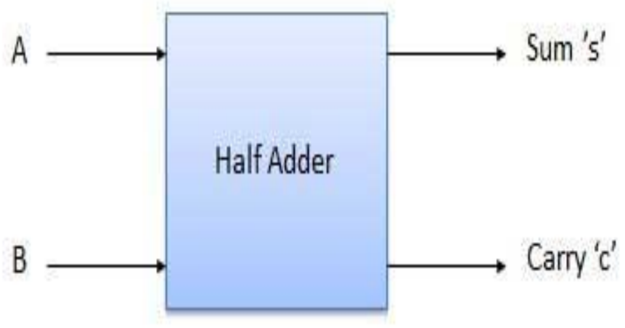
Combinational Circuit

- Combinational circuit is a circuit in which we combine the different gates in the circuit.
- 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 an n number of inputs and m number of outputs.

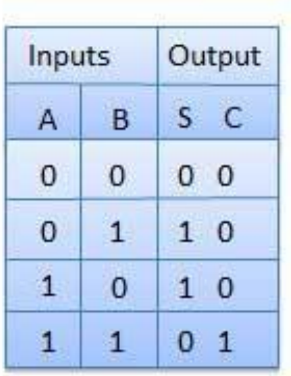


Combinational Circuit

- Half Adder
  - Half adder is a combinational logic circuit with two inputs and two outputs.
  - The half adder circuit is designed to add two single bit binary number A and B.
  - It is the basic building block for addition of two single bit numbers.
  - This circuit has two outputs carry and sum.

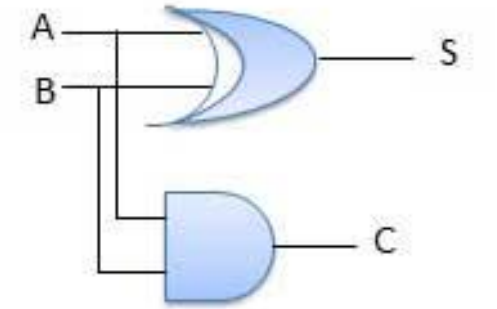


Block Diagram of Half Adder



Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

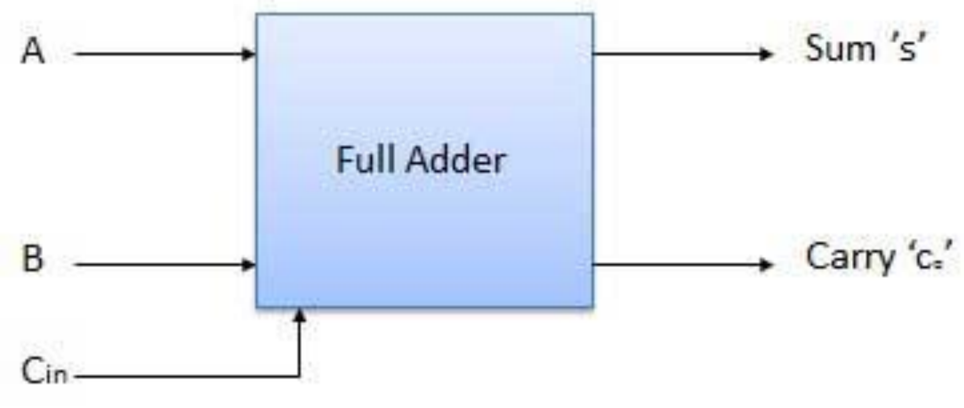
Half Adder Truth Table



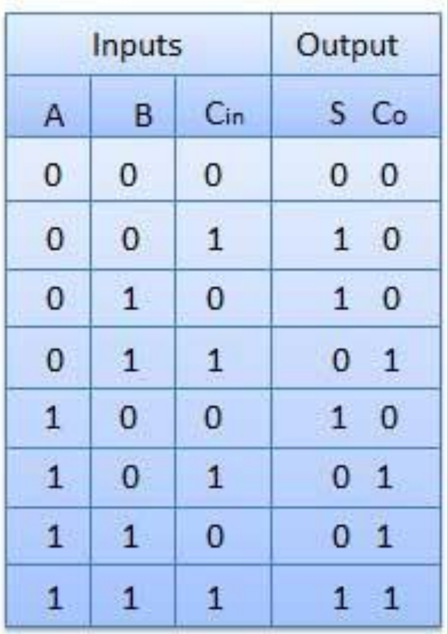
Half Adder Circuit Diagram

COMBINATIONAL CIRCUIT

- 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 of Full Adder

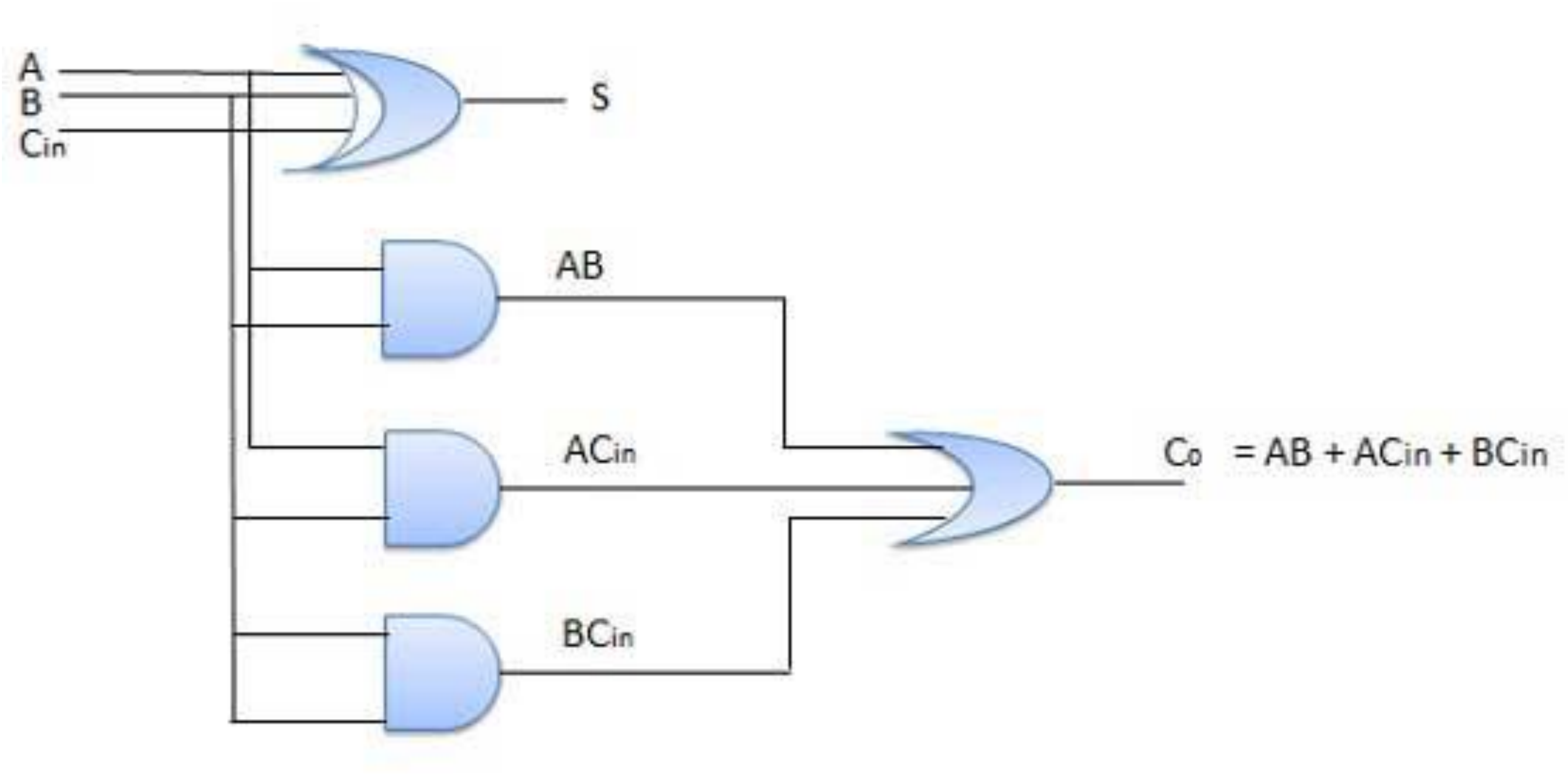


Inputs			Output	
A	B	Cin	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Adder
Truth Table

COMBINATIONAL CIRCUIT

- Full Adder



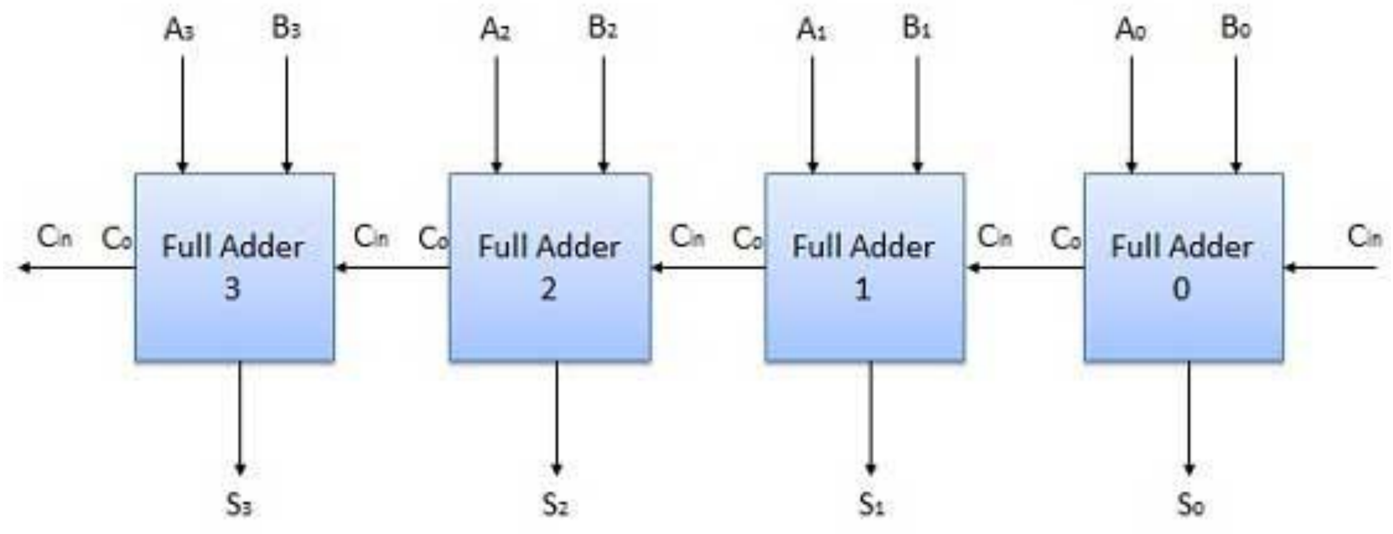
Full Adder Circuit Diagram

Combinational Circuit

- 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.

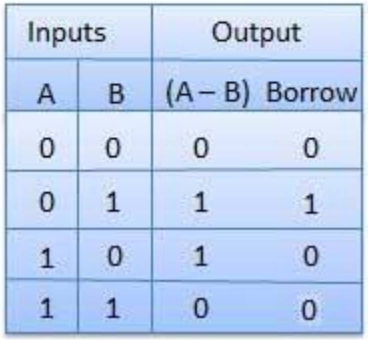
COMBINATIONAL CIRCUIT

- 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 block diagram.



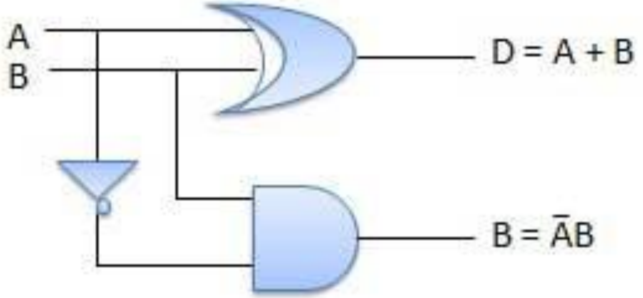
Combinational Circuit

- Half Subtractor
  - 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 an 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.



Inputs		Output	
A	B	(A – B)	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Substractor Truth Table

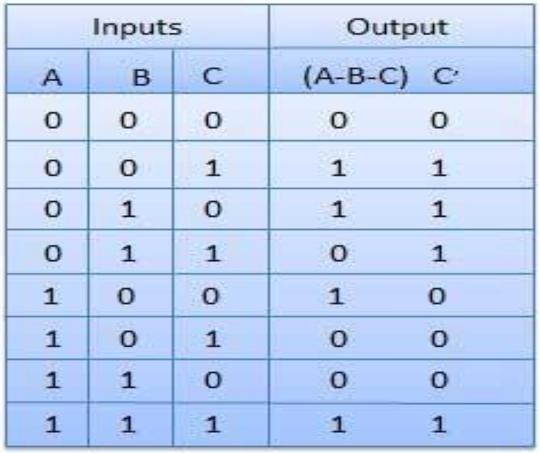


Half Substractor Circuit Diagram

Combinational Circuit

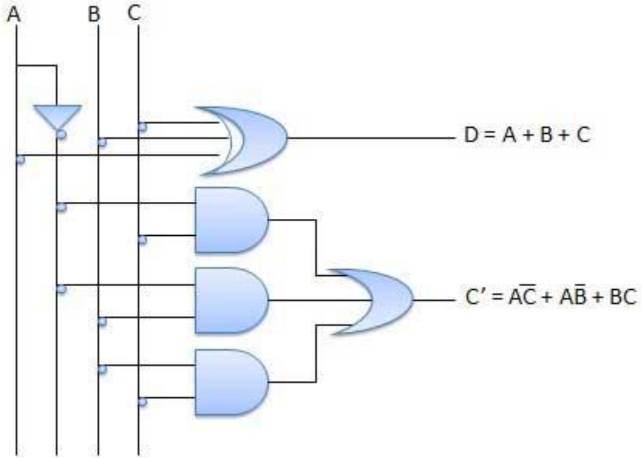
Full Subtractor

- 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.



Inputs			Output	
A	B	C	(A-B-C)	C'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Full Substractor Truth Table



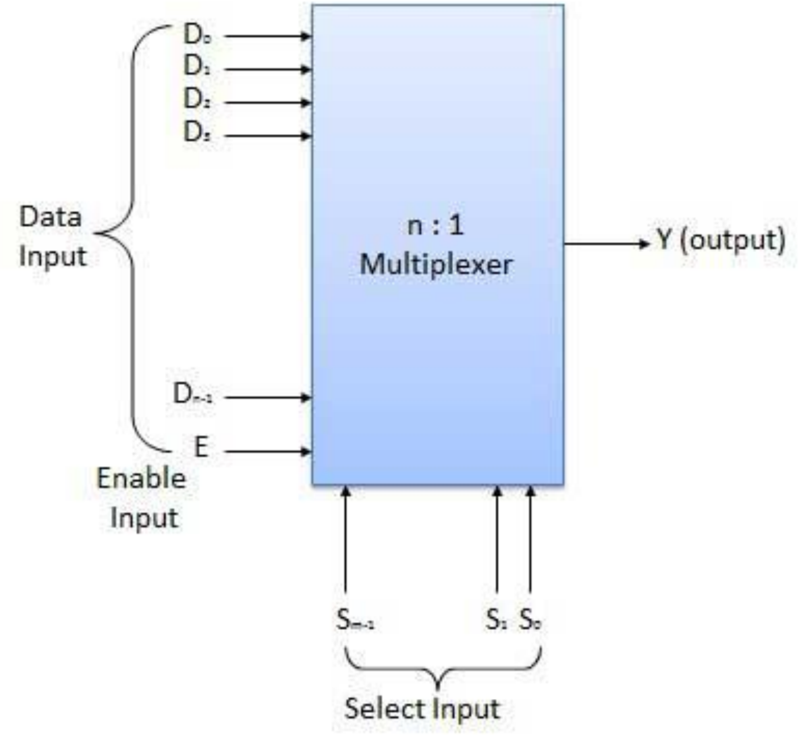
Full Substractor Circuit Diagram

Combinational Circuit

- Multiplexers
  - Multiplexer is a special type of combinational circuit. There are n-data inputs, one output and m select inputs with 2m = n.
  - It is a digital circuit which selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the selected inputs.
  - Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y.

Combinational Circuit

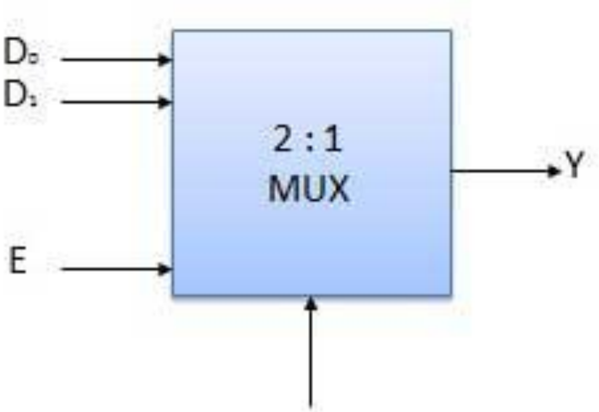
- Multiplexers
  - E is called the strobe or enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low.



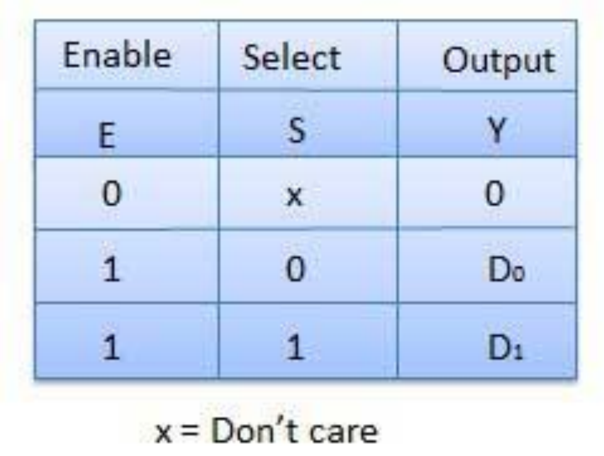
Block Diagram of n:1 Multiplexer

Combinational Circuit

- Multiplexers come in multiple variations
  - 2 : 1 multiplexer
  - 4 : 1 multiplexer
  - 16 : 1 multiplexer
  - 32 : 1 multiplexer



2:1 Multiplexer Block Diagram



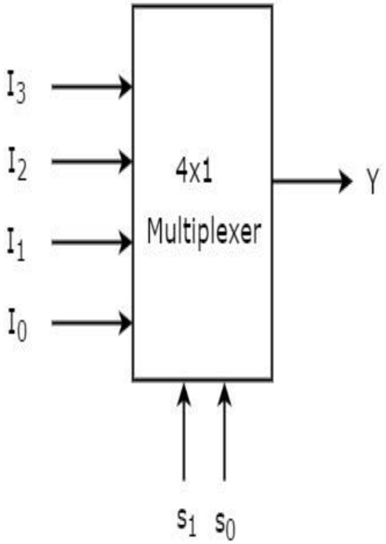
Enable	Select	Output
E	S	Y
0	x	0
1	0	D_0
1	1	D_1

x = Don't care

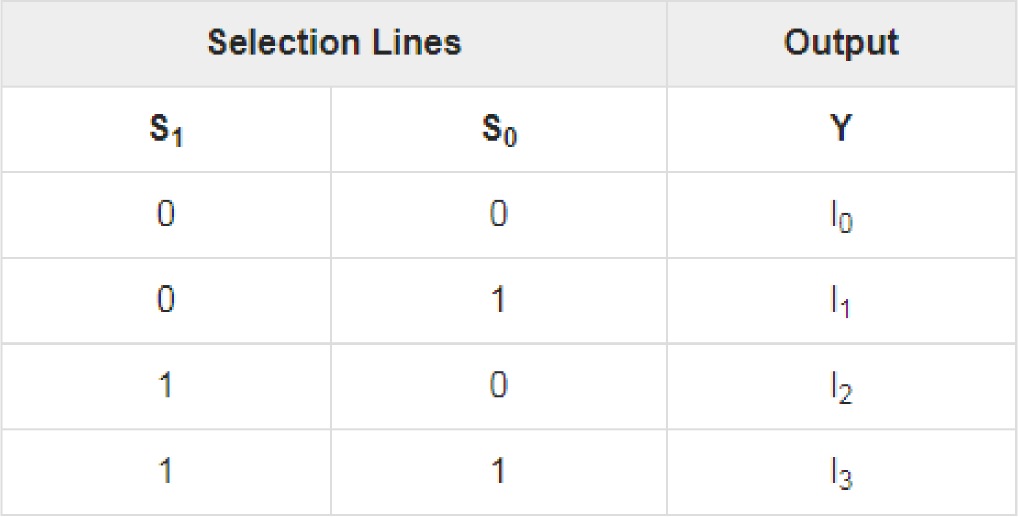
2:1 Multiplexer Truth Table

Combinational Circuit

- 4 : 1 multiplexer:
  - 4:1 Multiplexer has four data inputs I_0, I_1, I_2 & I_3, two selection lines s_0 & s_1 and one output Y. The **block diagram** of 4:1 Multiplexer is shown in the following figure.
  - One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.



4:1 Multiplexer Block Diagram

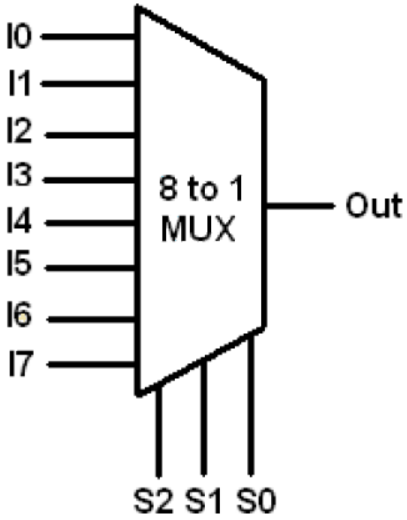


Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

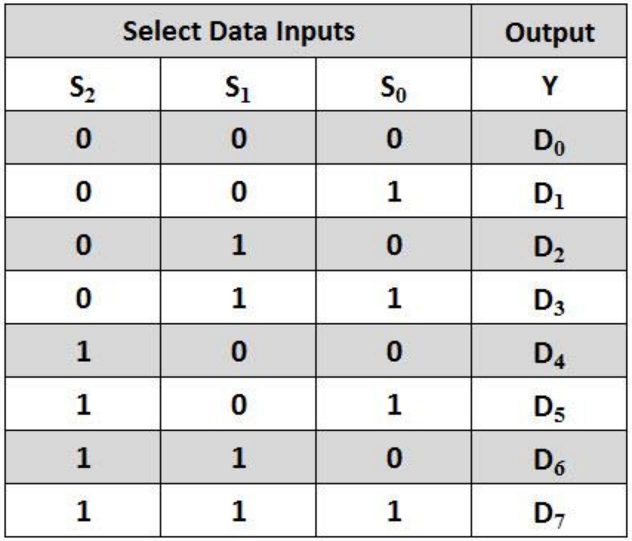
4:1 Multiplexer Truth Table

COMBINATIONAL CIRCUIT

- 8: 1 multiplexer:
  - 8:1 Multiplexer has eight data inputs $I_0 I_1$, I_2 , I_3 ... I_7 and three selection lines s_0 , s_1 & s_2 and one output Y.
    The **block diagram and truth table** of 8:1 Multiplexer is shown in the following figure.
  - One of these 8 inputs will be connected to the output based on the combination of inputs present at these three selection lines.



8:1 Multiplexer Block Diagram



Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

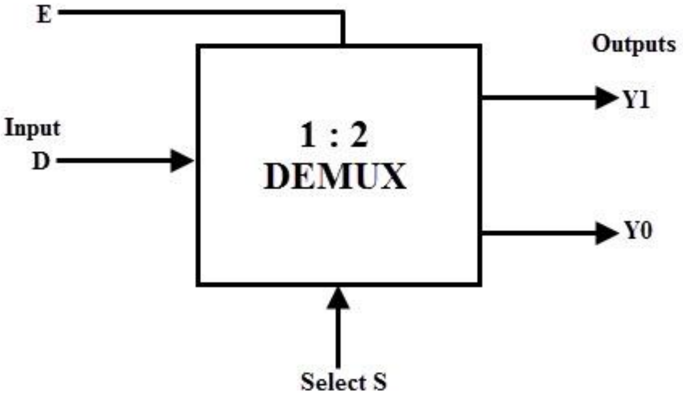
8:1 Multiplexer Truth Table

Combinational Circuit

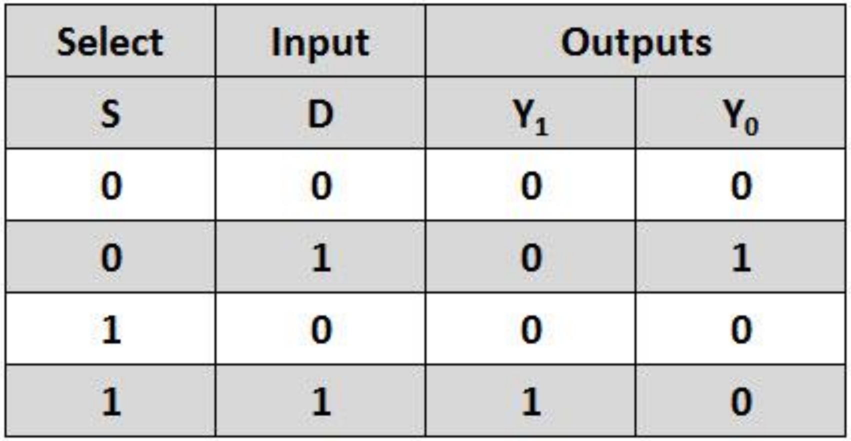
- De-multiplexer
  - A de-multiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs.
  - It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to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

Combinational Circuit

- De-multiplexers comes in multiple variations.
  - 1 : 2 de-multiplexer
  - 1 : 4 de-multiplexer
  - 1 : 16 de-multiplexer
  - 1 : 32 de-multiplexer



Block Diagram of 1:2 De-multiplexer

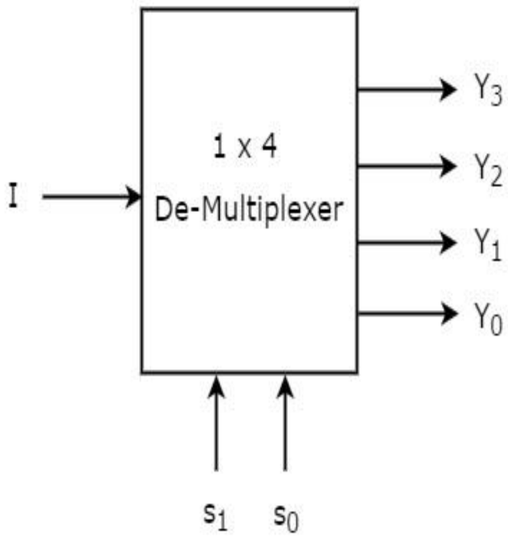


Select	Input	Outputs	
S	D	Y_1	Y_0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

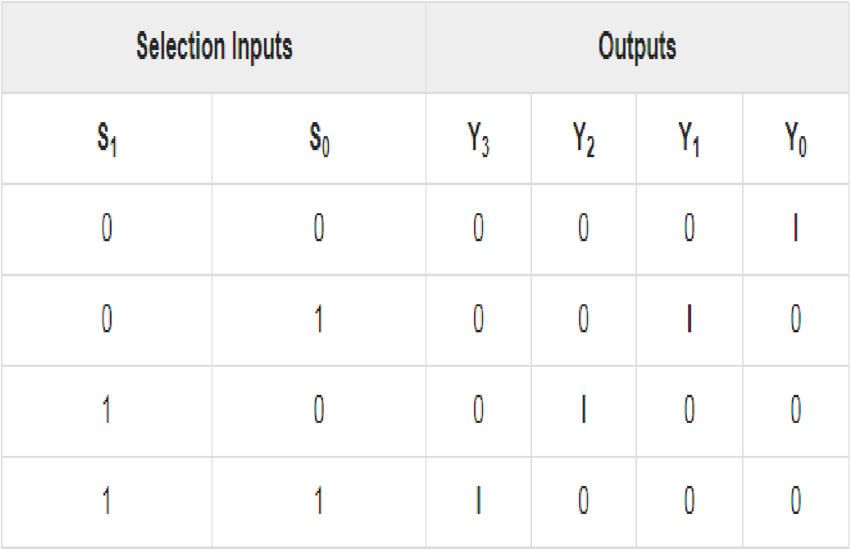
1:2 De-multiplexer Truth Table

Combinational Circuit

- 1 : 4 de-multiplexer:
  - 1 : 4 De-Multiplexer has one input I, two selection lines, s_0 & s_1 and four outputs Y_0, Y_1, Y_2 &Y_3. The **block diagram** & **truth table** of 1:4 De-Multiplexer is shown in the following figure.
  - The single input 'I' will be connected to one of the four outputs, Y_0 to Y_3 based on the values of selection lines S_0 & S_1 .



Block Diagram of 1:4 De-multiplexer

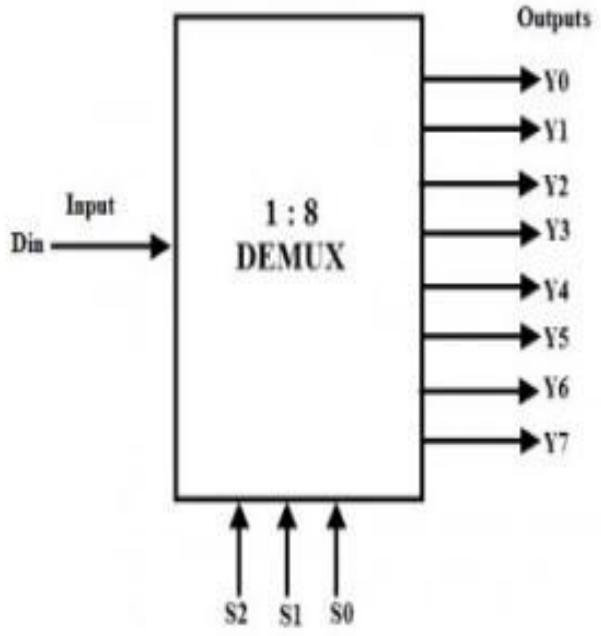


Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

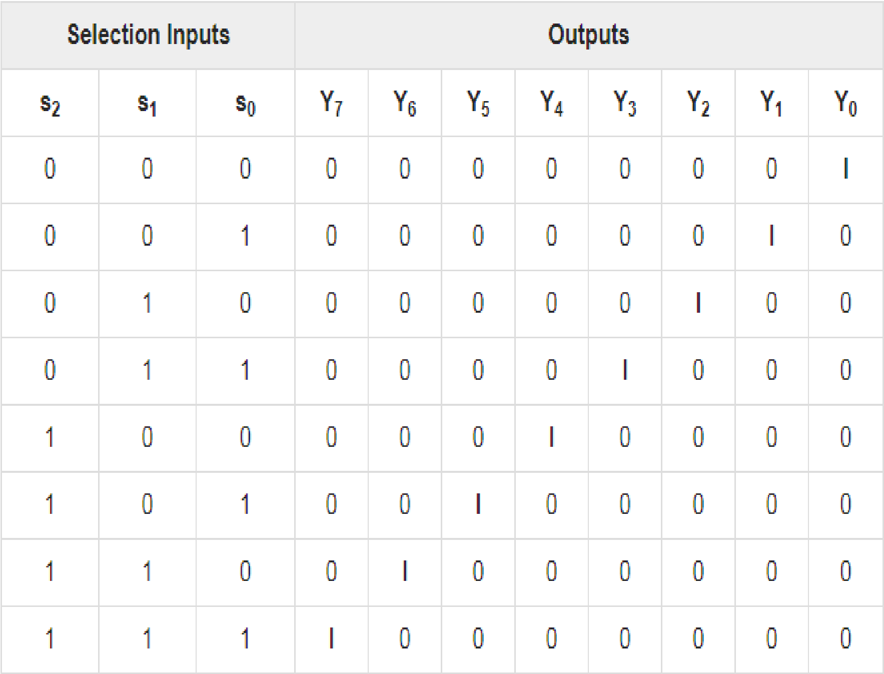
1:4 De-multiplexer Truth Table

COMBINATIONAL CIRCUIT

- 1 : 8 de-multiplexer:
  - 1 : 8 De-Multiplexer has one input I, three selection lines, S_0 , S_1 & S_2 and four outputs Y_0, Y_1, Y_2 , Y_3 … Y_7. The **block diagram** & **truth table** of 1:8 De-Multiplexer is shown in the following figure.
  - The single input ‘I’ will be connected to one of the eight outputs, Y_0 to Y_7 based on the values of selection lines S_0, S_1 & S_2.



Block Diagram of 1:8 De-multiplexer

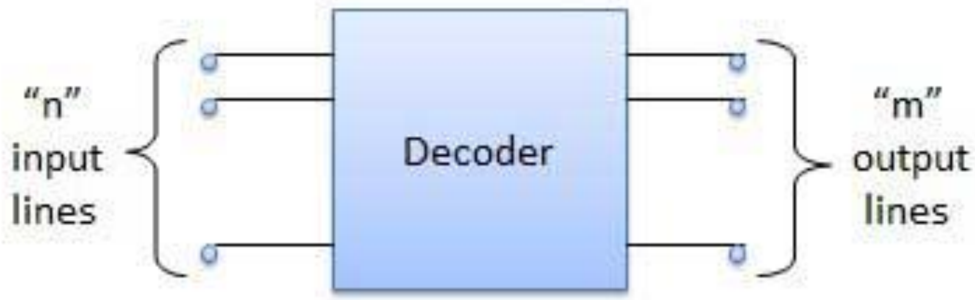


Selection Inputs			Outputs							
s_2	s_1	s_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

1:8 De-multiplexer Truth Table

Combinational Circuit

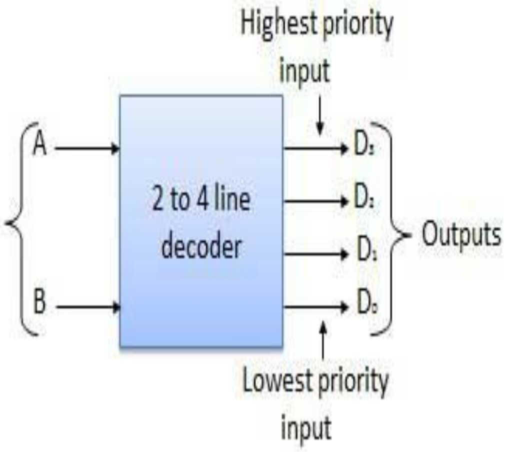
- Decoder
  - A decoder is a combinational circuit. It has n input and to a maximum m = 2n outputs.
  - Decoder is identical to a de-multiplexer without any data input.
  - It performs operations which are exactly opposite to those of an encoder.



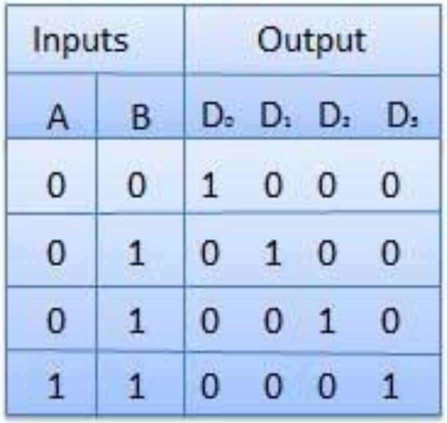
Block Diagram of Decoder

COMBINATIONAL CIRCUIT

- 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 and 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.

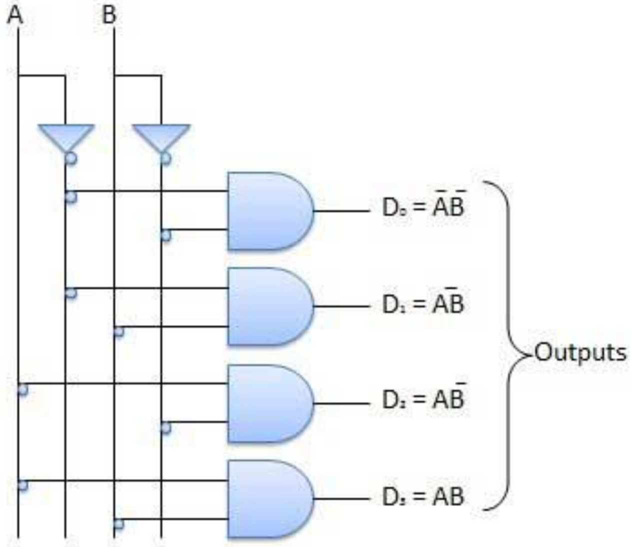


Block Diagram of 2 to 4 Decoder



Inputs		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

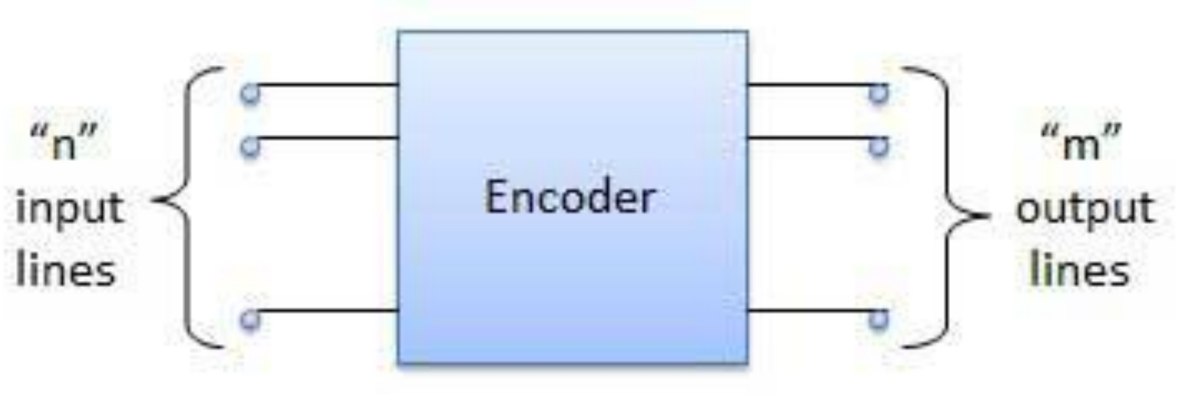
Truth Table of 2 to 4 Decoder



Logic Circuit

Combinational Circuit

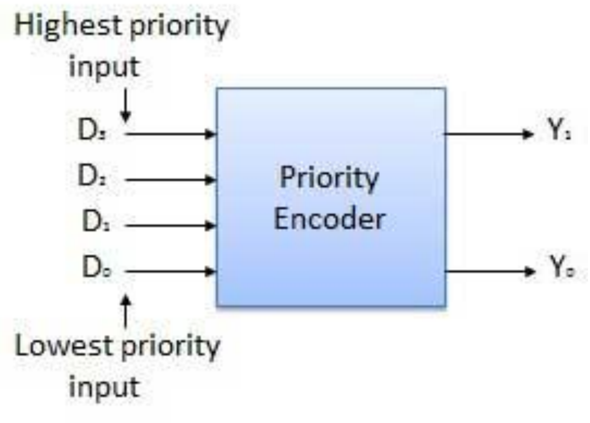
- Encoder
  - Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder.
  - An encoder has n number of input lines and m number of output lines.
  - The encoder accepts an n input digital word and converts it into an m bit another digital word.



Block Diagram of encoder

COMBINATIONAL CIRCUIT

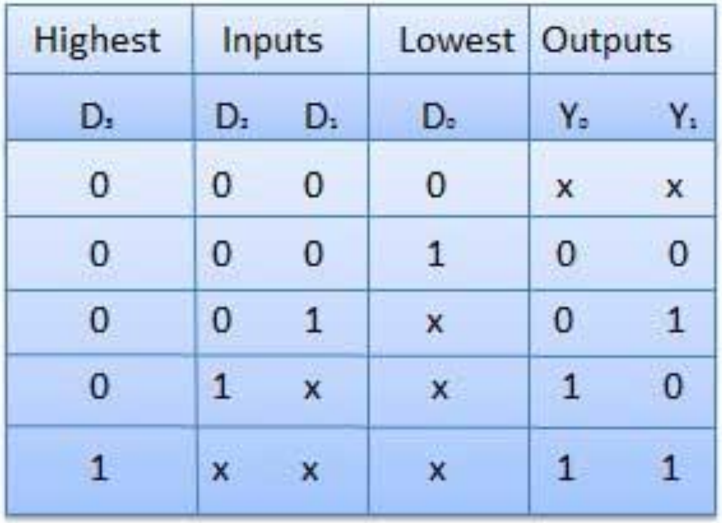
- Priority Encoder
  - This is a special type of encoder. Priority is given to the input lines.
  - If two or more input line are 1 at the same time, then the input line with highest priority will be considered.
  - There are four input D0, D1, D2, D3 and two output Y0, Y1.
  - Out of the four input D3 has the highest priority and D0 has the lowest priority.



Block Diagram of encoder

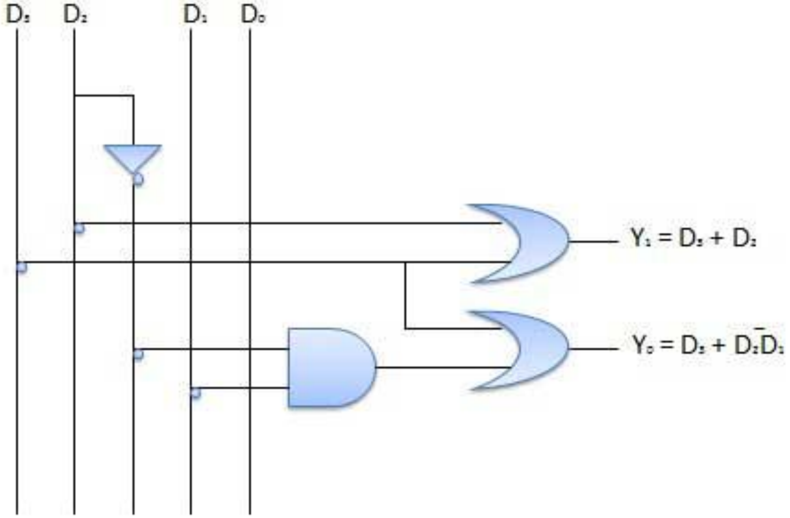
Combinational Circuit

- Priority Encoder
  - That means if D3 = 1 then Y1 Y1 = 11 irrespective of the other inputs. Similarly if D3 = 0 and D2 = 1 then Y1 Y0 = 10 irrespective of the other inputs.



Highest	Inputs		Lowest	Outputs	
D_3	D_2	D_1	D_0	Y_0	Y_1
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

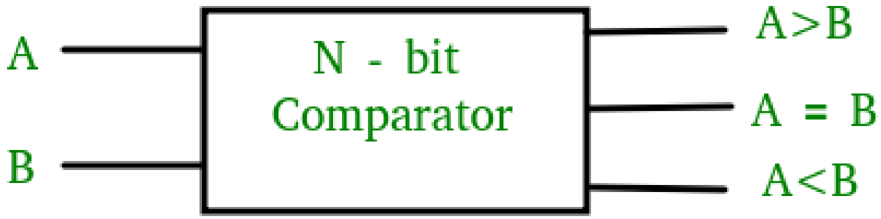
Truth table of priority encoder



Logic Circuit

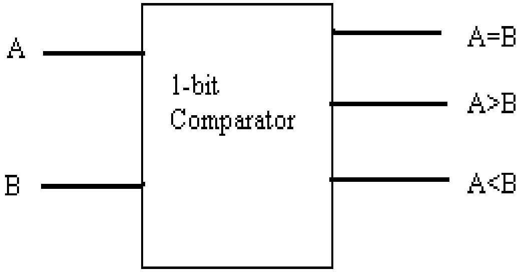
Combinational Circuit

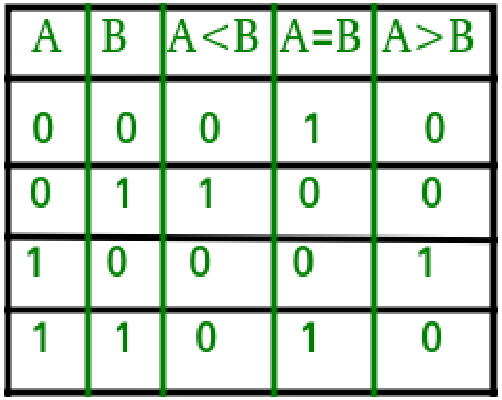
- Comparators:
  - A magnitude digital Comparator is a combinational circuit that **compares two digital or binary numbers** in order to find out whether one binary number is equal, less than or greater than the other binary number.
  - We logically design a circuit for which we will have two inputs one for A and other for B and have three output terminals, one for A > B condition, one for A = B condition and one for A < B condition.



Combinational Circuit

- 1-Bit Magnitude Comparator :
  - A comparator used to compare two bits is called a single bit comparator. It consists of two inputs each for two single bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

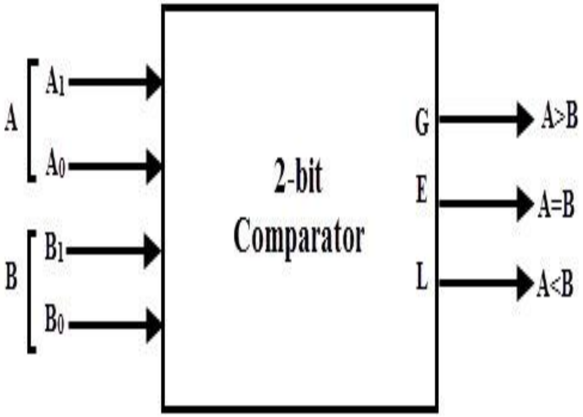


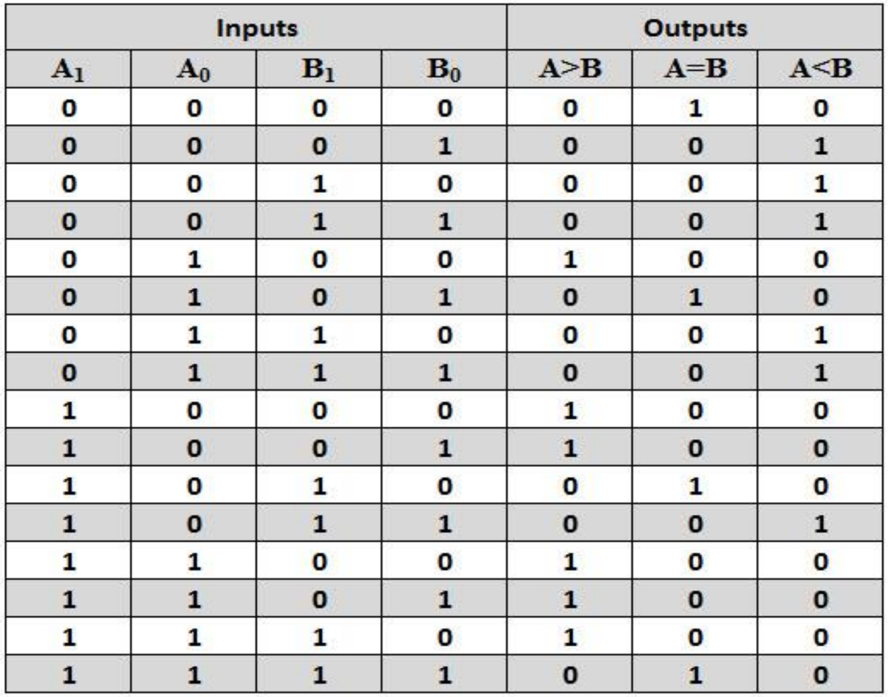


A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

COMBINATIONAL CIRCUIT

- 2-Bit Comparator:
  - A 2-bit comparator compares two binary numbers, each of two bits and produces their relation such as one number is equal or greater than or less than the other.
  - The first number A is designated as A = A1A0 and the second number is designated as B = B1B0. This comparator produces three outputs as G (G = 1 if A>B), E (E = 1, if A = B) and L (L = 1 if A<B).

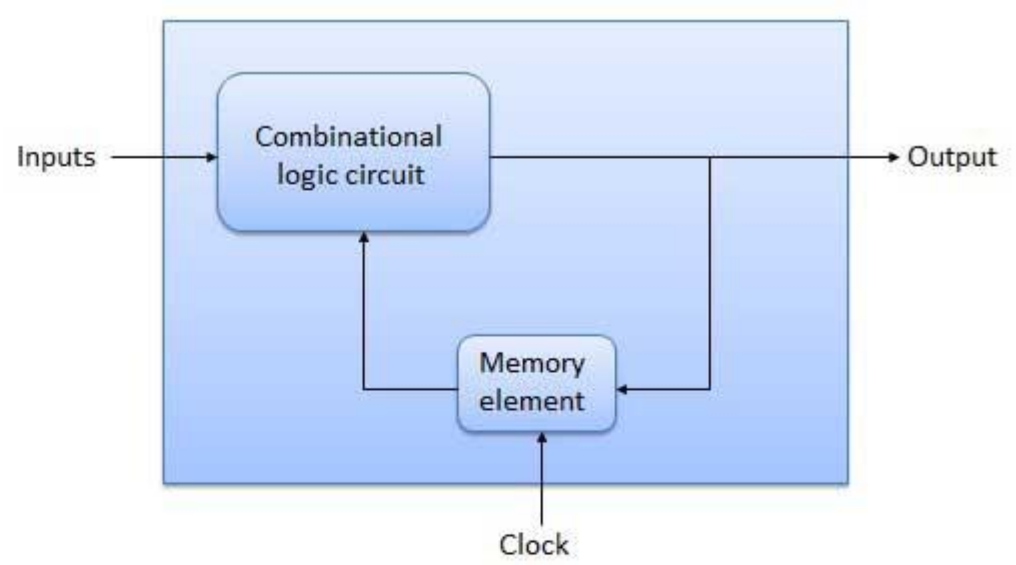




Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

SEQUENTIAL CIRCUIT

- 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.

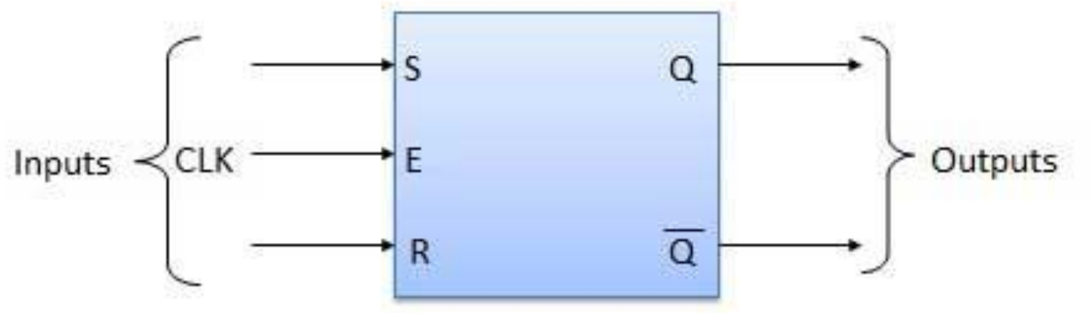


Sequential Circuit

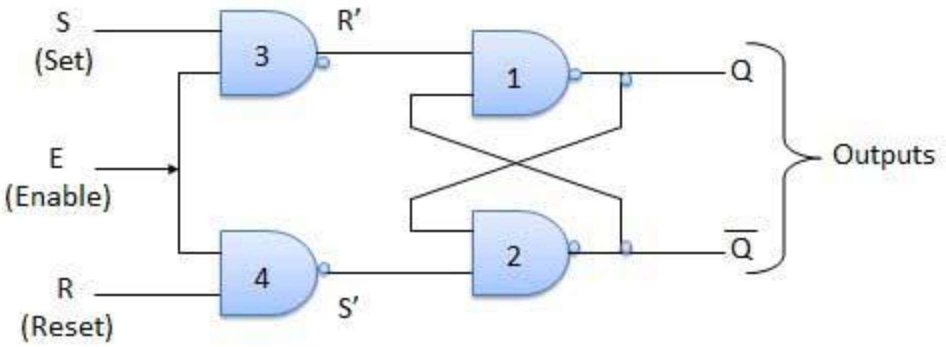
- Flip Flop:
  - Flip flop is a sequential circuit which generally samples its inputs and changes its outputs only at particular instants of time and not continuously.
  - Flip flop is said to be edge sensitive or edge triggered rather than being level triggered like latches.

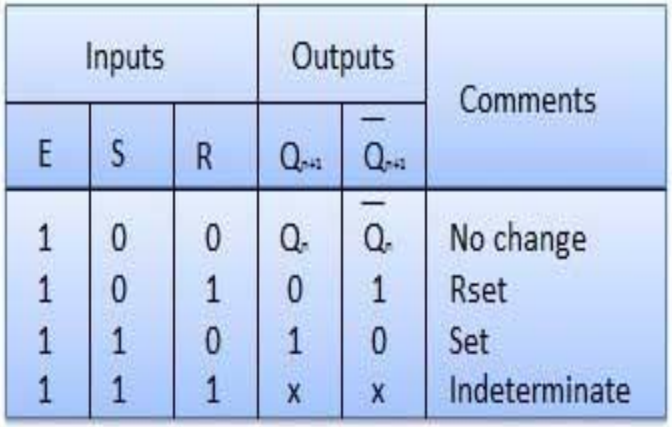
Sequential Circuit

- 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.

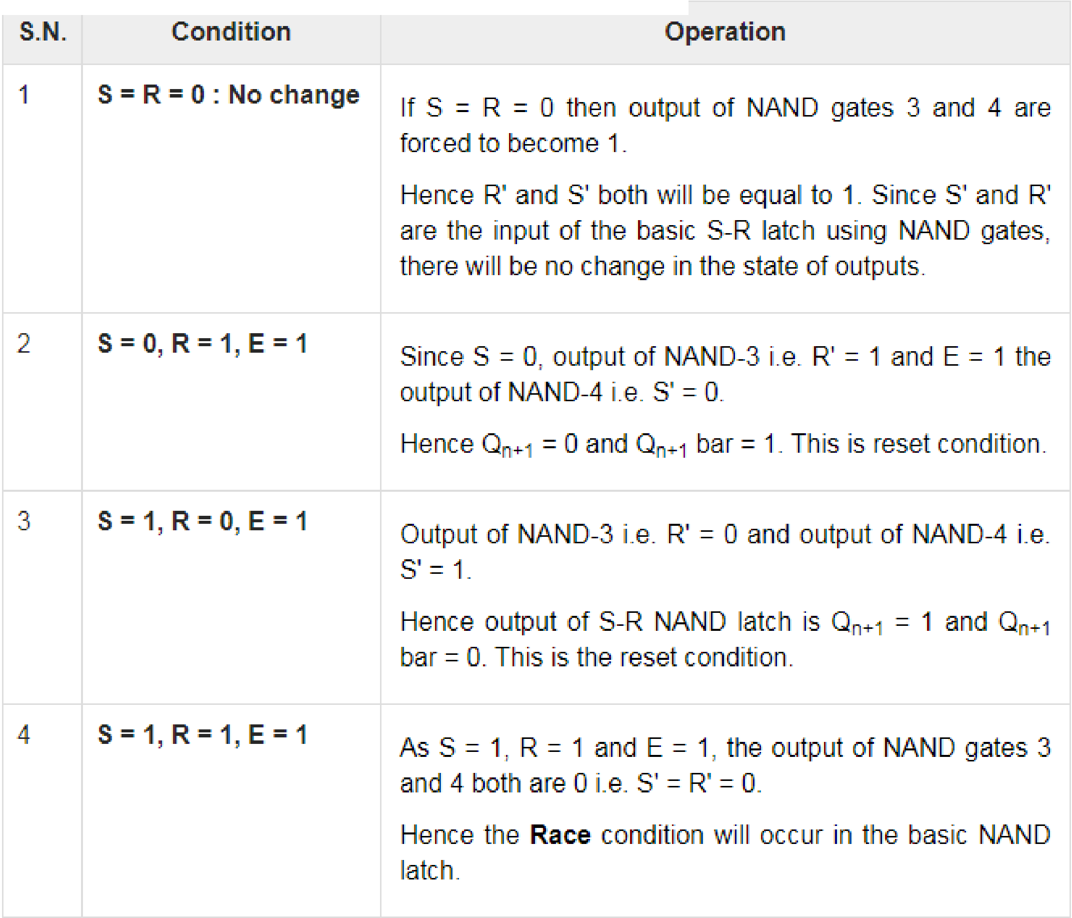


S-R Flip Flop operation:





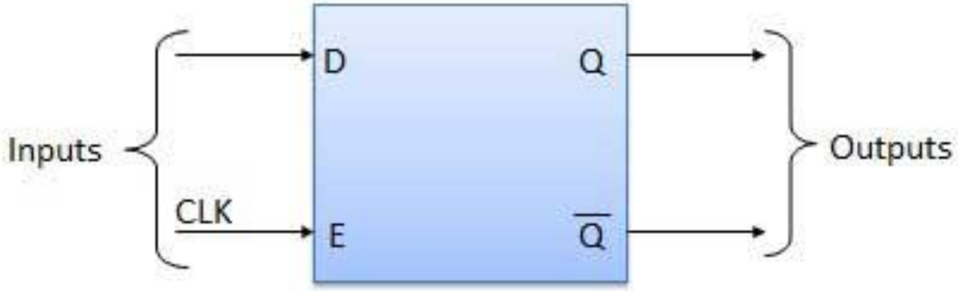
Inputs			Outputs		Comments
E	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	Q_n	$\overline{Q}_n$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	x	x	Indeterminate



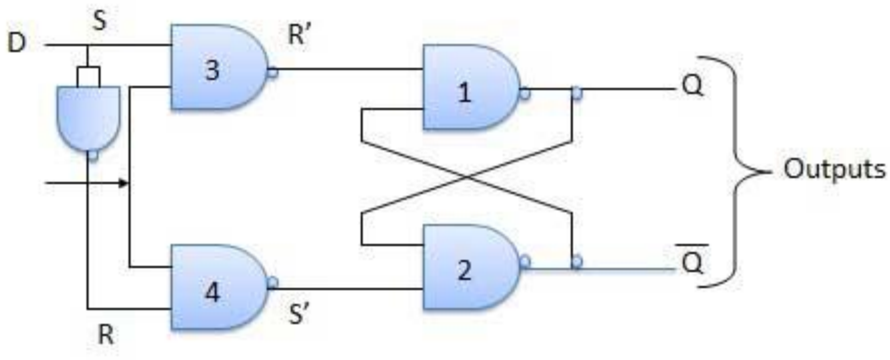
S.N.	Condition	Operation
1	**S = R = 0 : No change**	If S = R = 0 then output of NAND gates 3 and 4 are forced to become 1. Hence R' and S' both will be equal to 1. Since S' and R' are the input of the basic S-R latch using NAND gates, there will be no change in the state of outputs.
2	**S = 0, R = 1, 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 Q_{n+1} = 0 and Q_{n+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 Q_{n+1} = 1 and Q_{n+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.

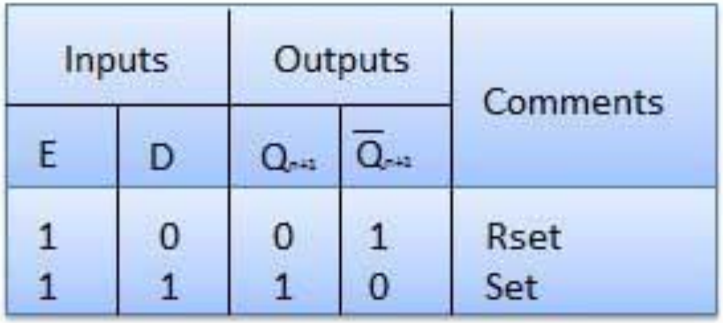
Sequential Circuit

- 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.

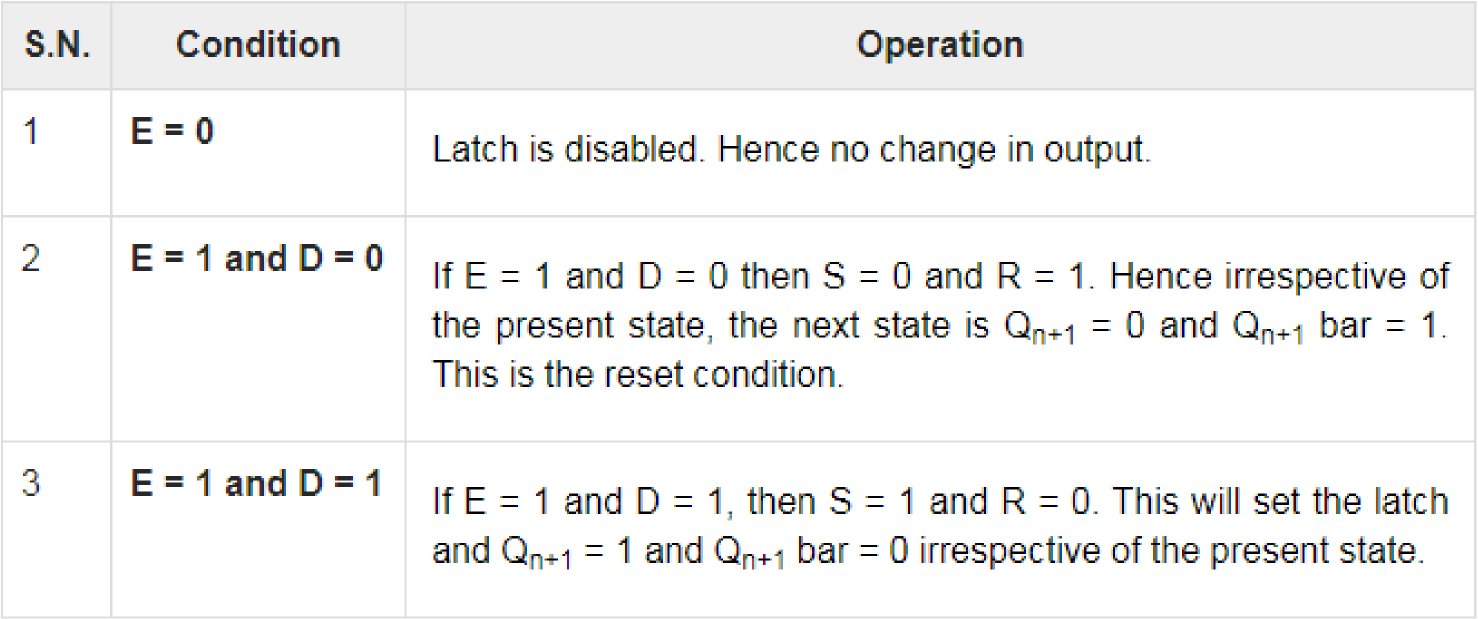


D Flip Flop operation:





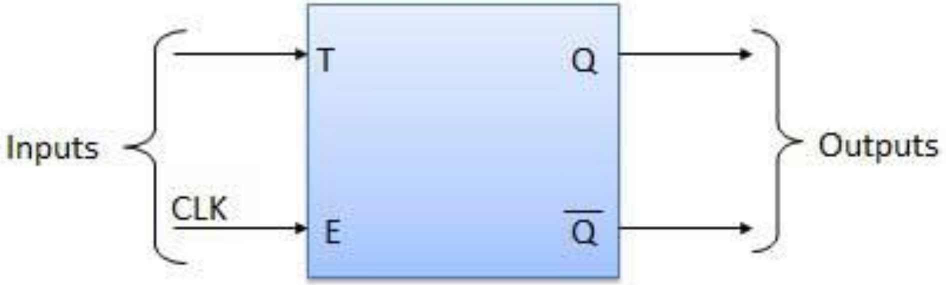
Inputs		Outputs		Comments
E	D	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	1	Rset
1	1	1	0	Set



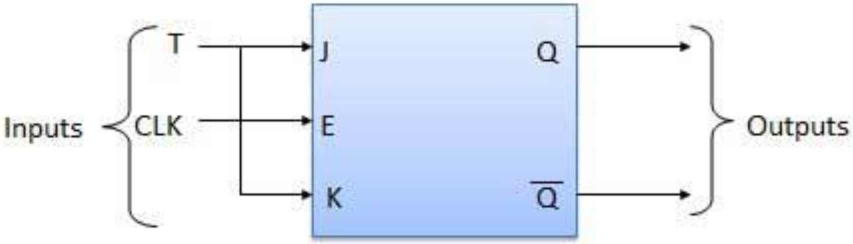
S.N.	Condition	Operation
1	**E = 0**	Latch is disabled. Hence no change in output.
2	**E = 1 and D = 0**	If E = 1 and D = 0 then S = 0 and R = 1. Hence irrespective of the present state, the next state is Q_{n+1} = 0 and Q_{n+1} bar = 1. This is the reset condition.
3	**E = 1 and D = 1**	If E = 1 and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

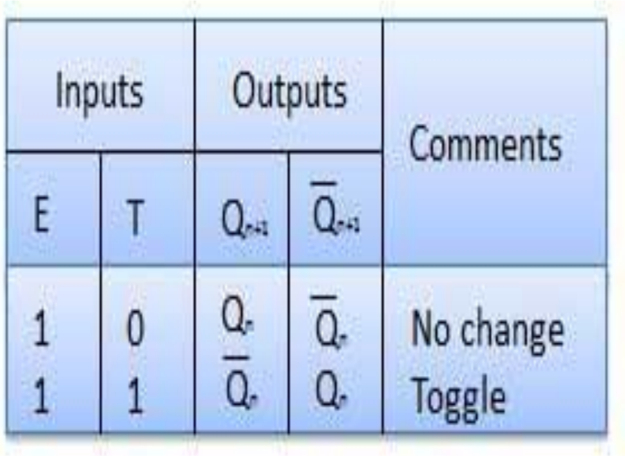
SEQUENTIAL CIRCUIT

- 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** as shown in the Symbol Diagram.
  - The symbol for positive edge triggered T flip flop is shown in the Block Diagram.



- T Flip Flop operation:





Inputs		Outputs		Comments
E	T	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	Q_n	$\overline{Q}_n$	No change
1	1	$\overline{Q}_n$	Q_n	Toggle

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.

SEQUENTIAL CIRCUIT

- Shift registers:
  - 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.
  - Shift Registers are used for data storage or for the movement of data.

Sequential Circuit

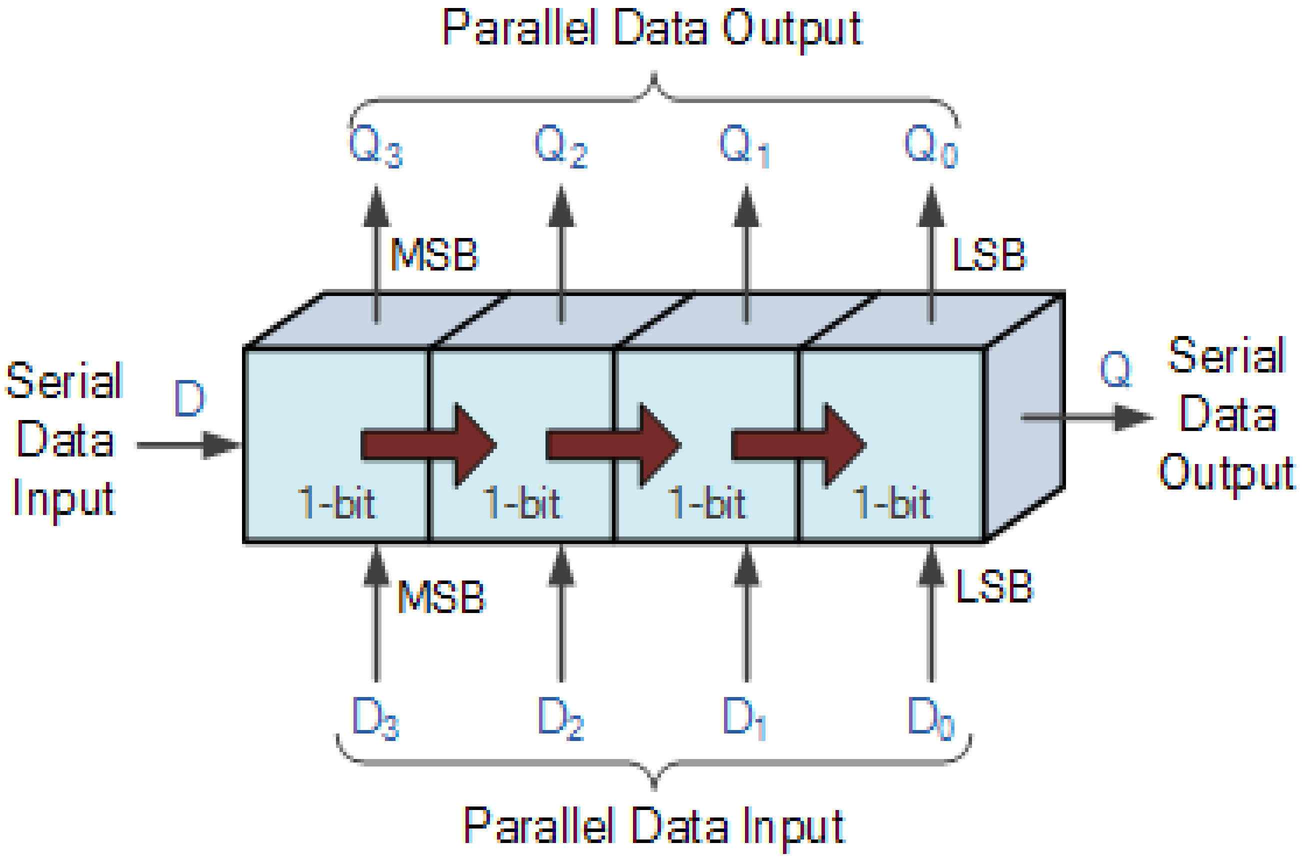
- Therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format.
- The individual data latches that make up a single shift register are all driven by a common clock (Clk) signal making them synchronous devices.
- Shift register IC's are generally provided with a clear or reset connection so that they can be "SET" or "RESET" as required.

SEQUENTIAL CIRCUIT

- Generally, shift registers operate in one of four different modes with the basic movement of data through a shift register being:
  - Serial-in to Parallel-out (SIPO): The register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
  - Serial-in to Serial-out (SISO): The data is shifted serially “IN” and “OUT” of the register, one bit at a time in either a left or right direction under clock control.
  - Parallel-in to Serial-out (PISO): The parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
  - Parallel-in to Parallel-out (PIPO): The parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

Sequential Circuit

- The effect of data movement from left to right through a shift register can be presented graphically as:



Sequential Circuit

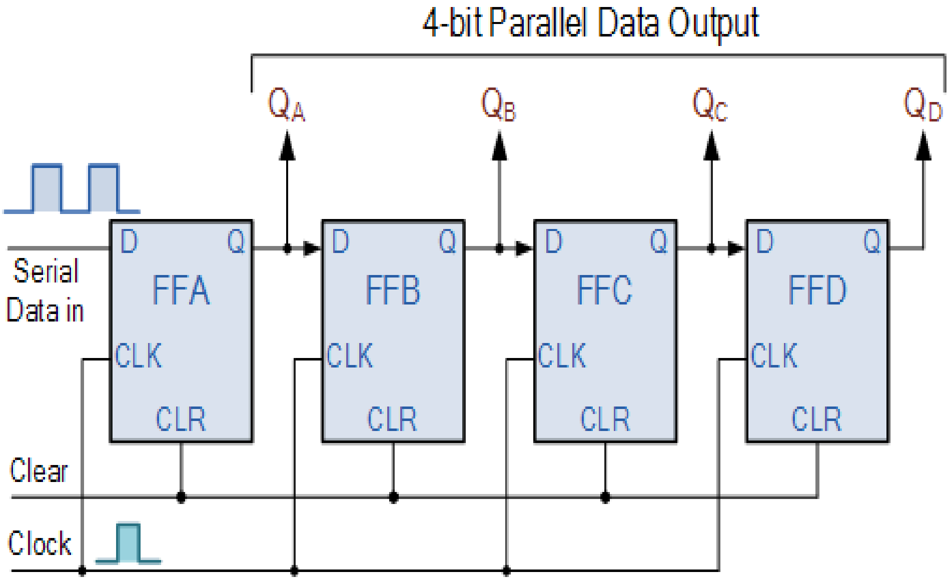
- 4-bit Serial-in to Parallel-out (SIPO) Shift Register:
  - The operation is as follows. Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs QA to QD are at logic level "0" ie, no parallel data output.
  - If a logic "1" is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting QA will be set HIGH to logic "1" with all the other outputs still remaining LOW at logic "0".
  - Assume now that the DATA input pin of FFA has returned LOW again to logic "0" giving us one data pulse or 0-1-0.
  - The second clock pulse will change the output of FFA to logic "0" and the output of FFB and QB HIGH to logic "1" as its input D has the logic "1" level on it from QA. The logic "1" has now moved or been "shifted" one place along the register to the right as it is now at QA.

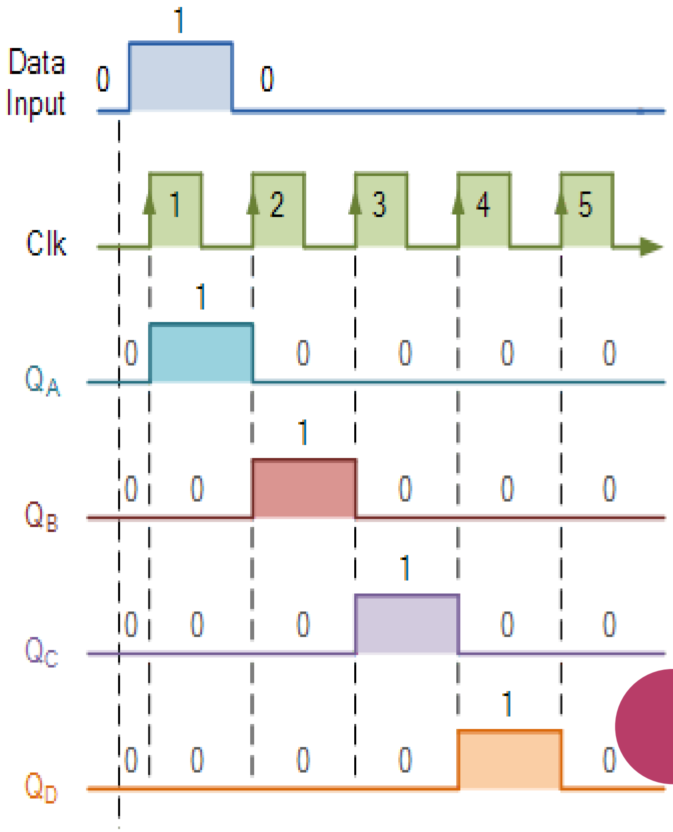
Sequential Circuit

- 4-bit Serial-in to Parallel-out (SIPO) Shift Register:
  - When the third clock pulse arrives this logic “1” value moves to the output of FFC (QC) and so on until the arrival of the fifth clock pulse which sets all the outputs QA to QD back again to logic level “0” because the input to FFA has remained constant at logic level “0”.
  - The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1 is stored in the register.
  - This data value can now be read directly from the outputs of QA to QD.
  - Then the data has been converted from a serial data input signal to a parallel data output.

Sequential Circuit

- 4-bit Serial-in to Parallel-out (SIPO) Shift Register:



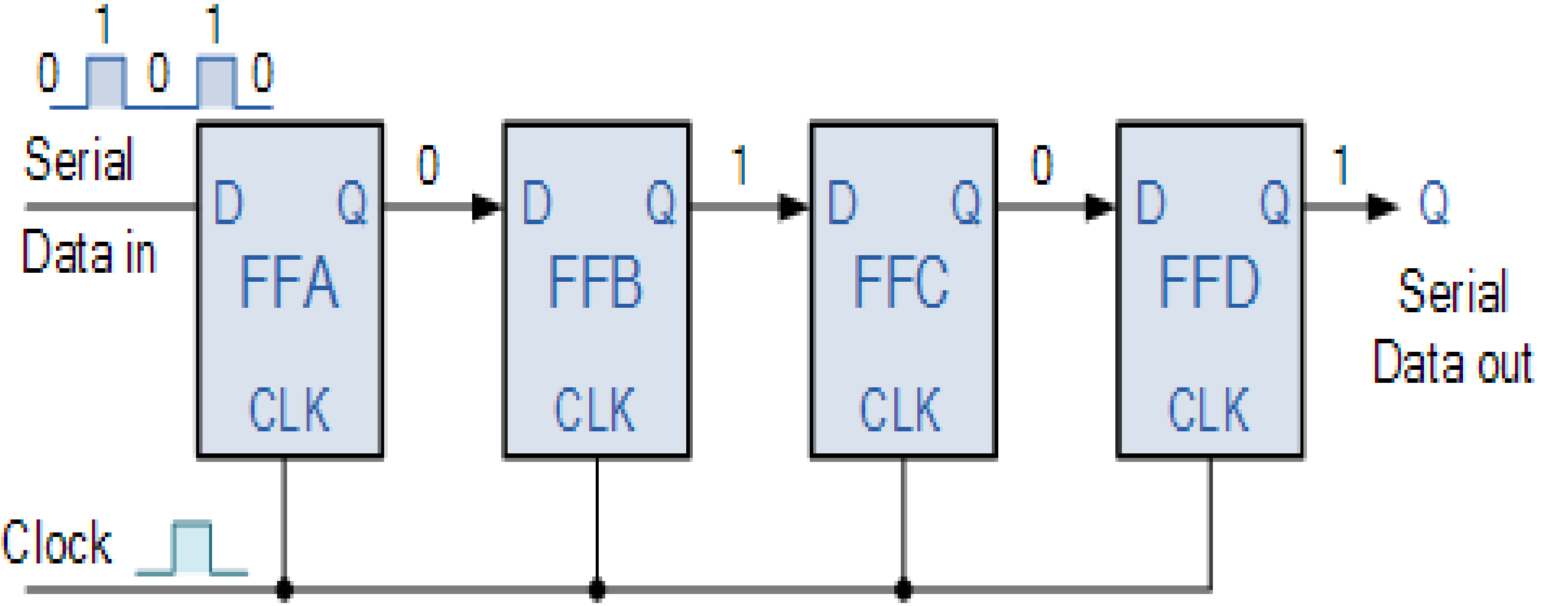


Sequential Circuit

- 4-bit Serial-in to Serial-out (SISO) Shift Register:
  - This shift register is very similar to the SIPO above, except were before the data was read directly in a parallel form from the outputs QA to QD, this time the data is allowed to flow straight through the register and out of the other end.
  - Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name Serial-in to Serial-Out Shift Register or SISO.
  - The SISO shift register is one of the simplest of the four configurations as it has only three connections.
  - The serial input (SI) determines what enters the left hand flip-flop and the serial output (SO) is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk).
  - The logic circuit diagram below shows a generalized serial-in serial-out shift register.

Sequential Circuit

- 4-bit Serial-in to Serial-out (SISO) Shift Register:

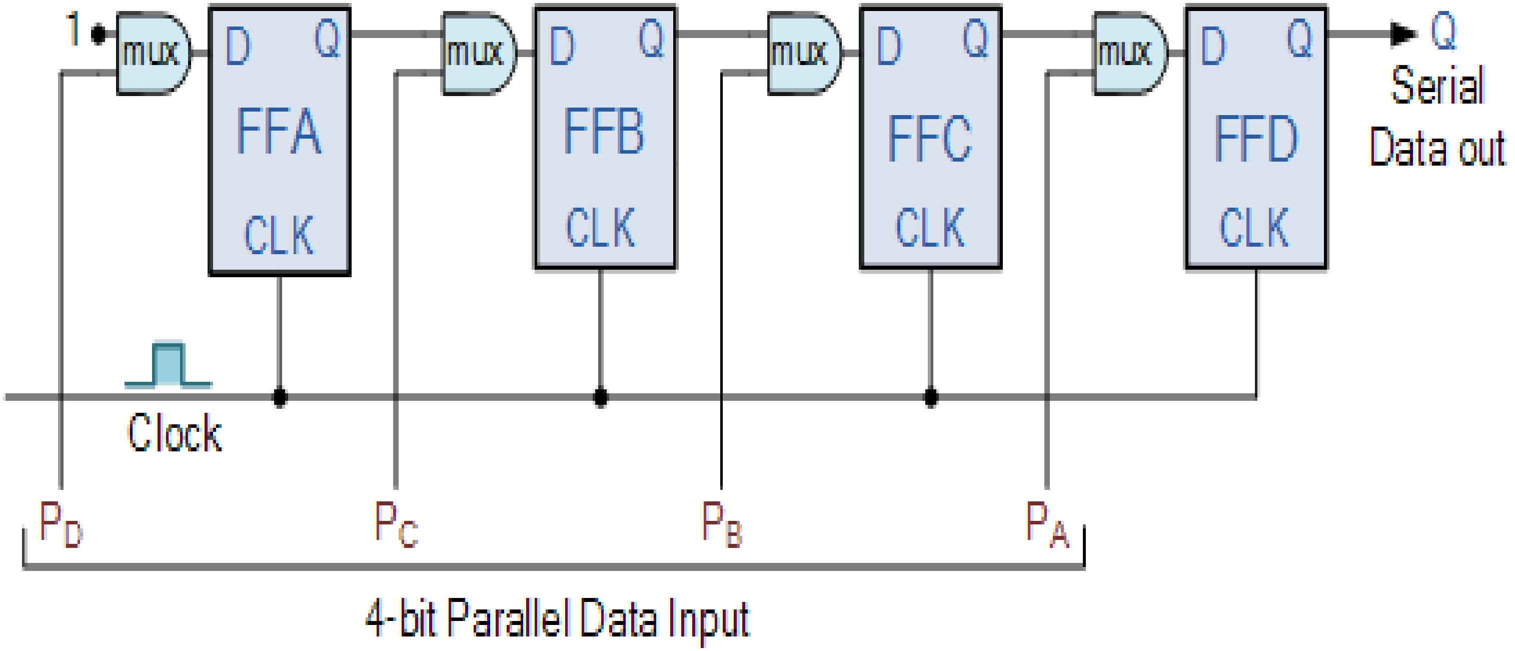


Sequential Circuit

- 4-bit Parallel-in to Serial-out (PISO) Shift Register :
  - The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above.
  - The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins PA to PD of the register.
  - The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at PA to PD.
  - This data is outputted one bit at a time on each clock cycle in a serial format.
  - It is important to note that with this type of data register a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data.

Sequential Circuit

- 4-bit Parallel-in to Serial-out (PISO) Shift Register :
  - As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream.

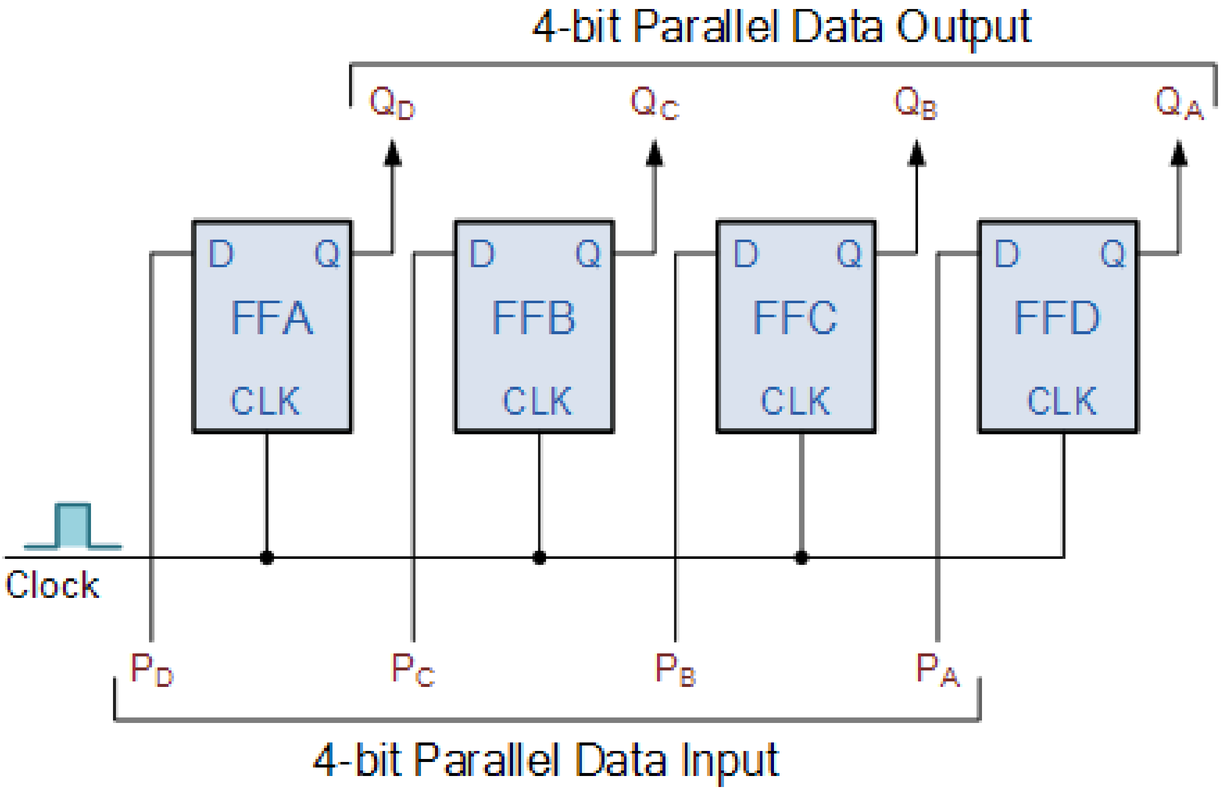


Sequential Circuit

- 4-bit Parallel-in to Parallel-out (PIPO) Shift Register:
  - The final mode of operation is the Parallel-in to Parallel-out Shift Register.
  - This type of shift register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above.
  - The data is presented in a parallel format to the parallel input pins PA to PD and then transferred together directly to their respective output pins QA to QD by the same clock pulse.
  - Then one clock pulse loads and unloads the register.
  - The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enters the flip-flop, the parallel output (PO) and the sequencing clock signal (Clk).

Sequential Circuit

- 4-bit Parallel-in to Parallel-out (PIPO) Shift Register:
  - This arrangement for parallel loading and unloading is shown below.

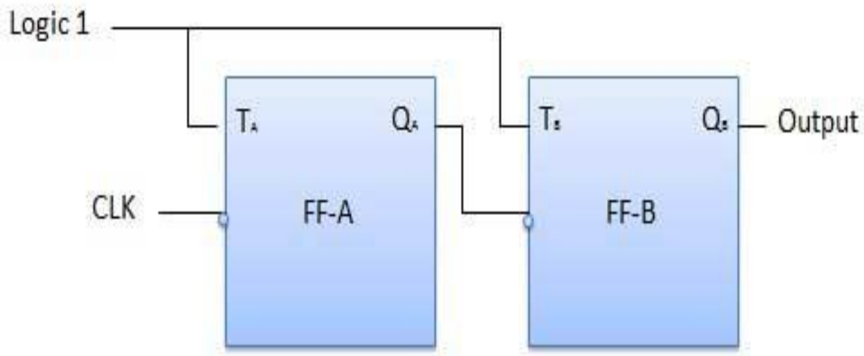


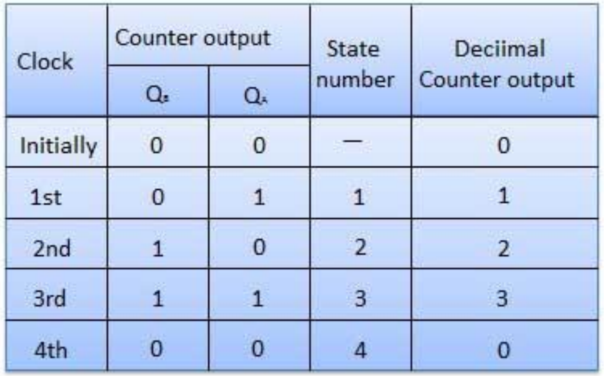
SEQUENTIAL CIRCUIT

- Counters:
  - Counter is a sequential circuit.
  - A digital circuit which is used for a counting pulses is known counter.
  - Counter is the widest application of flip-flops.
  - It is a group of flip-flops with a clock signal applied.
- Counters are of two types.
  - Asynchronous or ripple counters.
  - Synchronous counters.

Sequential Circuit

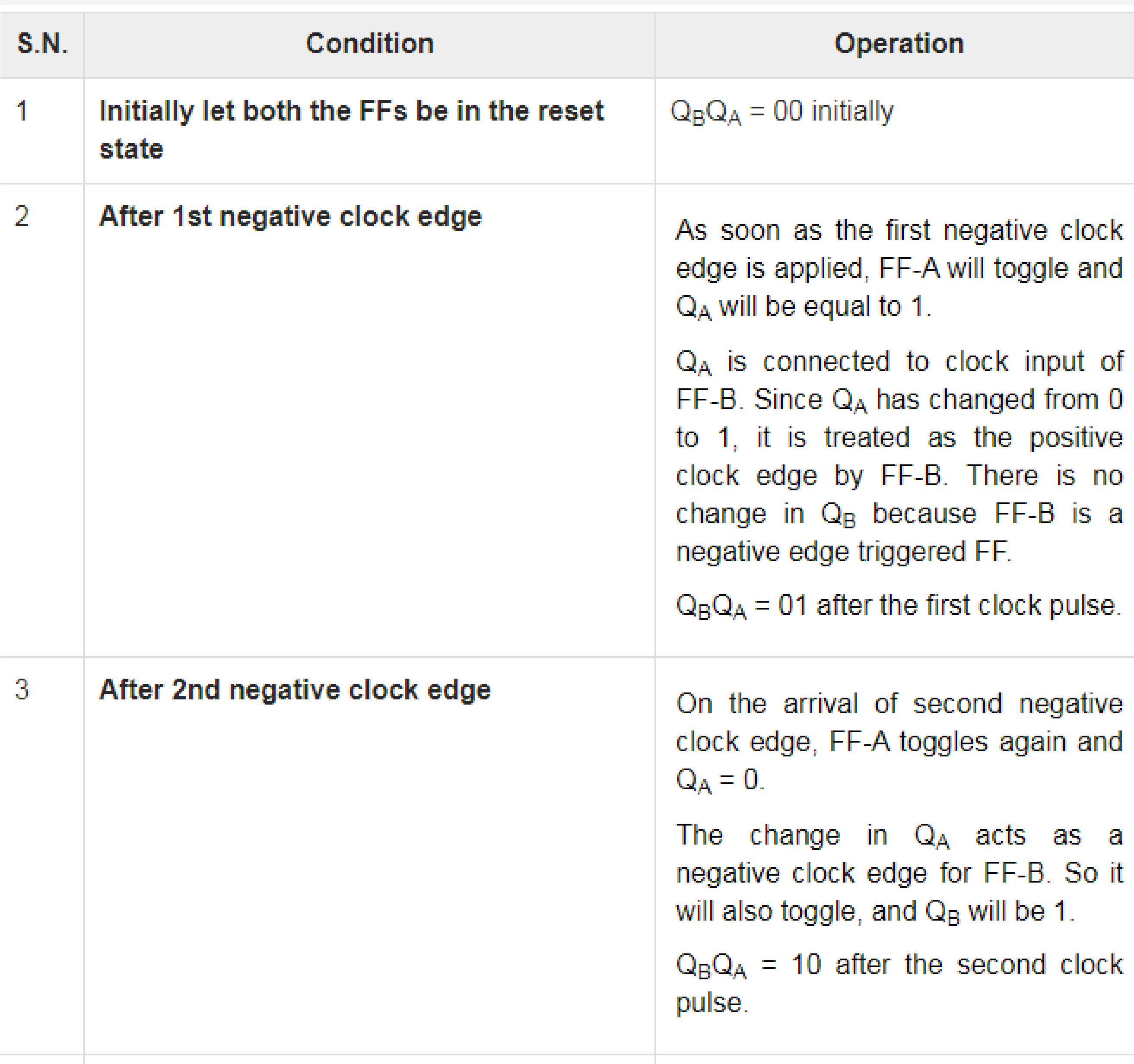
- Asynchronous or ripple counters:
  - The logic diagram of a 2-bit ripple up counter is shown in figure. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q_A output is applied to the clock input of the next flip-flop i.e. FF-B.





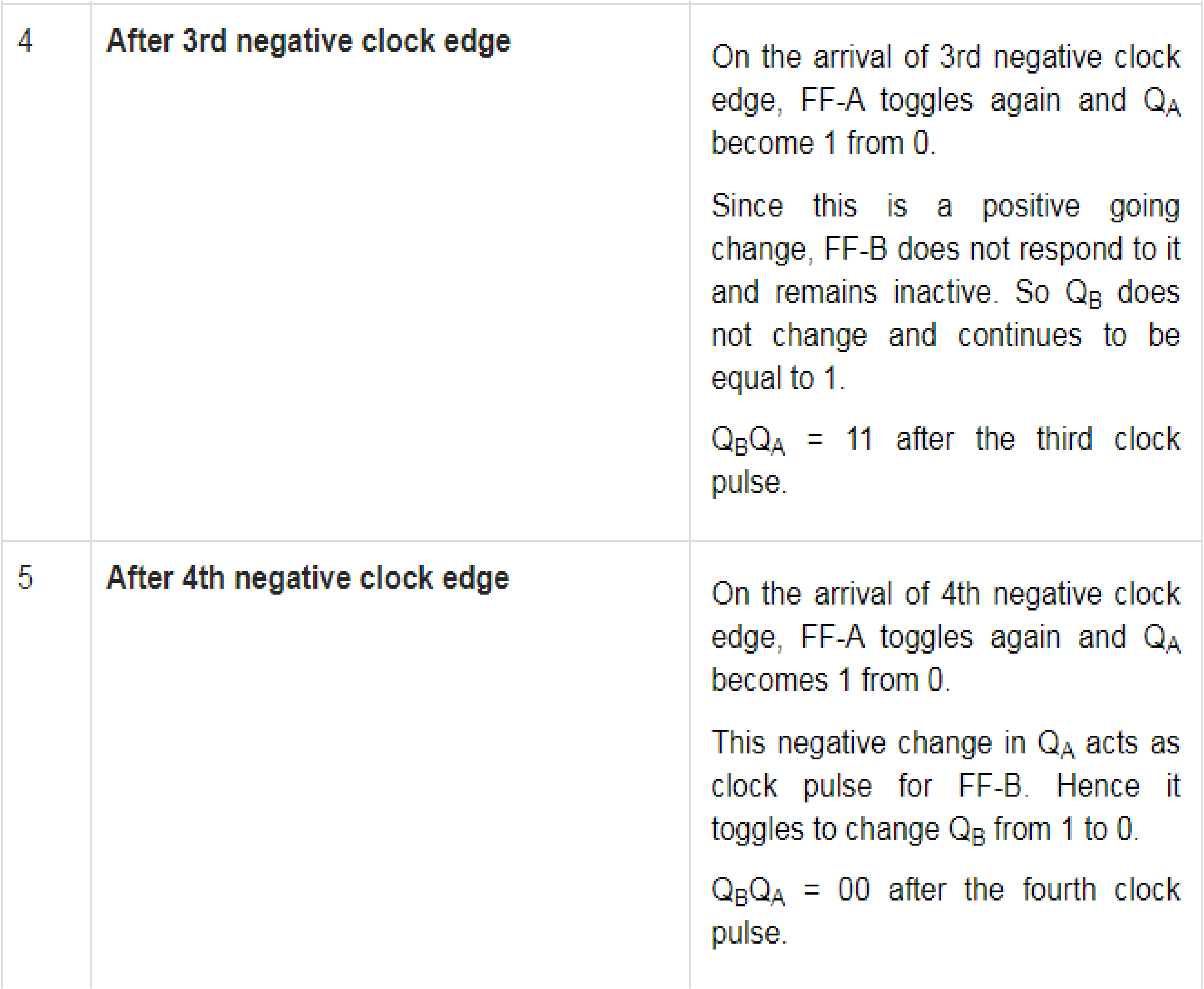
Clock	Counter output		State number	Deciimal Counter output
	Q_B	Q_A		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

SEQUENTIAL CIRCUIT



S.N.	Condition	Operation
1	**Initially let both the FFs be in the reset state**	$Q_B Q_A$ = 00 initially
2	**After 1st negative clock edge**	As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will be equal to 1. Q_A is connected to clock input of FF-B. Since Q_A has changed from 0 to 1, it is treated as the positive clock edge by FF-B. There is no change in Q_B because FF-B is a negative edge triggered FF. $Q_B Q_A$ = 01 after the first clock pulse.
3	**After 2nd negative clock edge**	On the arrival of second negative clock edge, FF-A toggles again and Q_A = 0. The change in Q_A acts as a negative clock edge for FF-B. So it will also toggle, and Q_B will be 1. $Q_B Q_A$ = 10 after the second clock pulse.

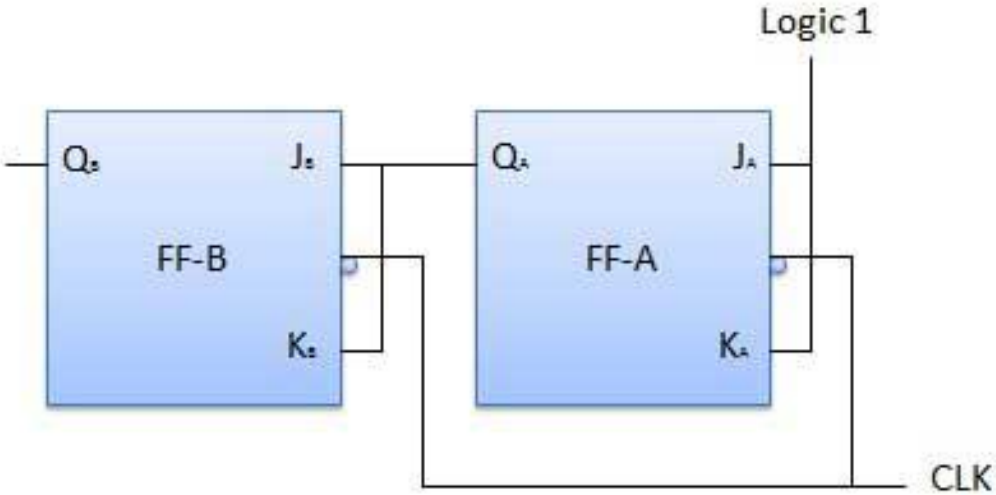
SEQUENTIAL CIRCUIT



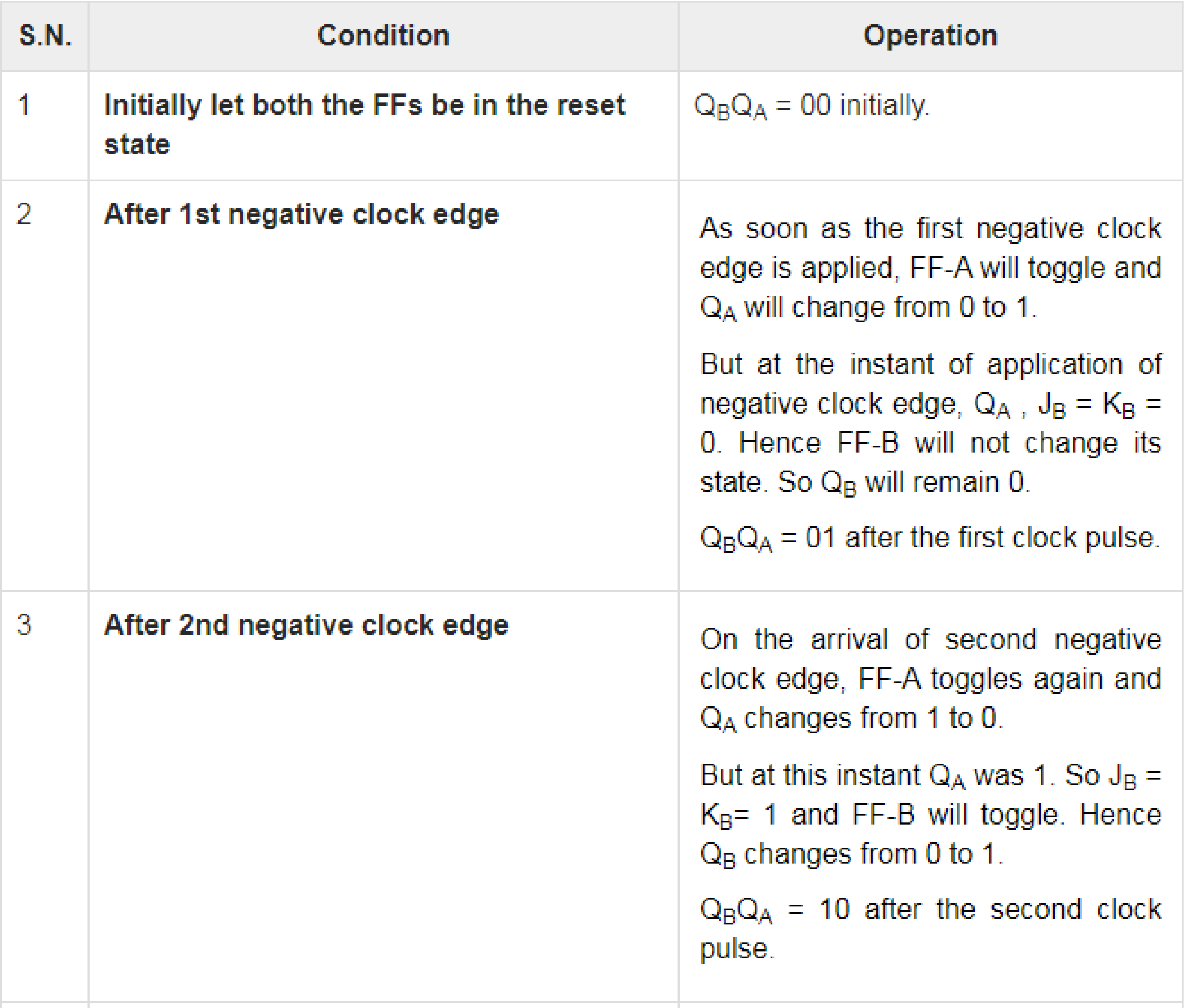
4	**After 3rd negative clock edge**	On the arrival of 3rd negative clock edge, FF-A toggles again and Q_A become 1 from 0. Since this is a positive going change, FF-B does not respond to it and remains inactive. So Q_B does not change and continues to be equal to 1. Q_BQ_A = 11 after the third clock pulse.
5	**After 4th negative clock edge**	On the arrival of 4th negative clock edge, FF-A toggles again and Q_A becomes 1 from 0. This negative change in Q_A acts as clock pulse for FF-B. Hence it toggles to change Q_B from 1 to 0. Q_BQ_A = 00 after the fourth clock pulse.

Sequential Circuit

- Synchronous counters
  - If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.
- 2-bit Synchronous up counter
  - The J_A and K_A inputs of FF-A are tied to logic 1.
  - So FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A.

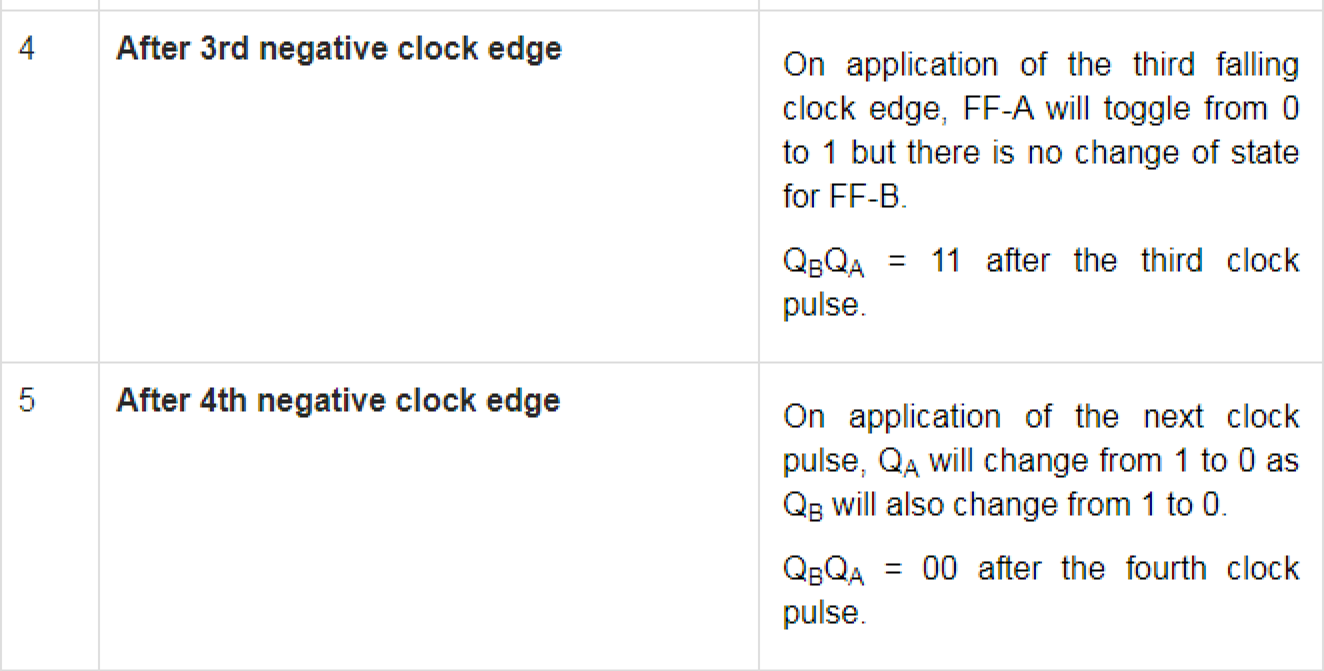


SEQUENTIAL CIRCUIT



S.N.	Condition	Operation
1	**Initially let both the FFs be in the reset state**	Q_BQ_A = 00 initially.
2	**After 1st negative clock edge**	As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will change from 0 to 1. But at the instant of application of negative clock edge, Q_A , J_B = K_B = 0. Hence FF-B will not change its state. So Q_B will remain 0. Q_BQ_A = 01 after the first clock pulse.
3	**After 2nd negative clock edge**	On the arrival of second negative clock edge, FF-A toggles again and Q_A changes from 1 to 0. But at this instant Q_A was 1. So J_B = K_B= 1 and FF-B will toggle. Hence Q_B changes from 0 to 1. Q_BQ_A = 10 after the second clock pulse.

SEQUENTIAL CIRCUIT



4	**After 3rd negative clock edge**	On application of the third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B. Q_BQ_A = 11 after the third clock pulse.
5	**After 4th negative clock edge**	On application of the next clock pulse, Q_A will change from 1 to 0 as Q_B will also change from 1 to 0. Q_BQ_A = 00 after the fourth clock pulse.

Thank you