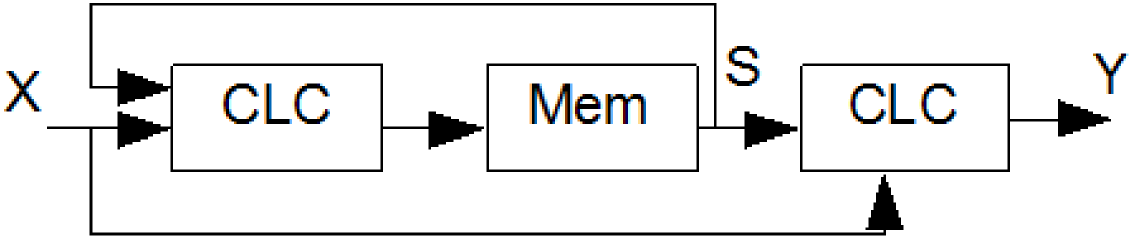
Sequential Logic Circuits (SLC)

- A SLS is a circuit in which outputs depend not only on its current inputs but also on the past sequence of inputs.
- All SLC have feedback loops, which presence required to take into consideration the time variable.
- SLC contain CLC and memory elements - latches and flip-flops.
- The state of a SLC in moment t is determined according to the values on the outputs of the memory elements. If a CLS contains n memory elements – it has 2^n states.

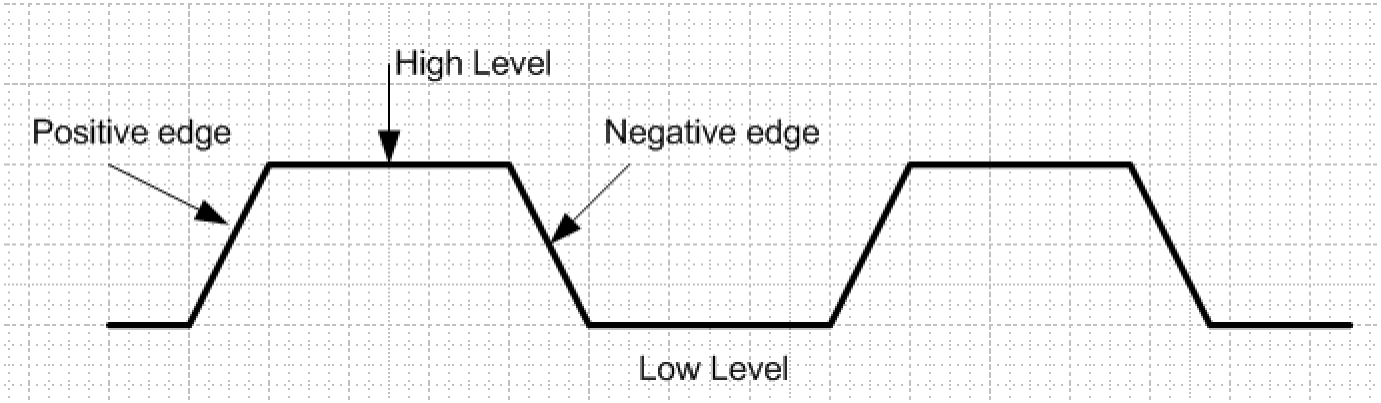


$$Y(t) = \lambda[S(t), X(t)]$$
$$S(t+1) = \varphi[S(t), X(t)]$$

Latches and Flip-Flops

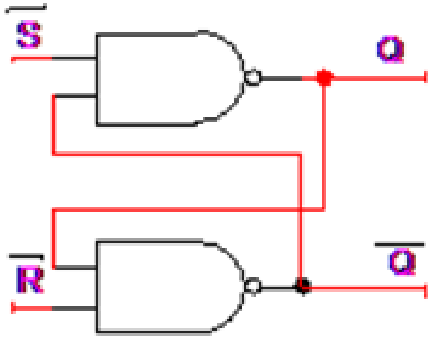
Latches and Flip-Flops are the basic building blocks of most sequential circuits. They are designed as a feedback sequential circuit using individual logic gates and feedback loops.

- A latch is **level triggered.**
- A flip-flop is **edge triggered.**



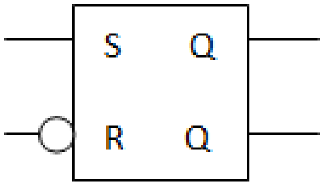
$\bar{S}\ \bar{R}$ Latch (set-reset)

An $\bar{S}\bar{R}$ (set-reset) latch based on NAND (NOR) gates:

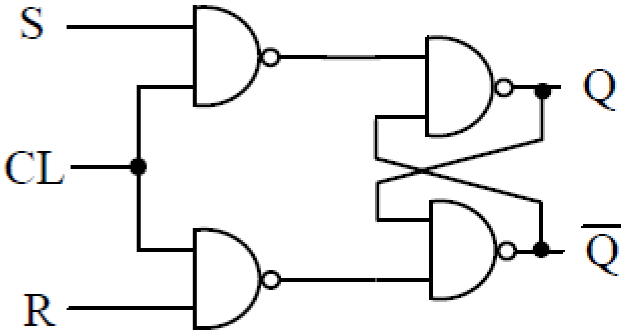


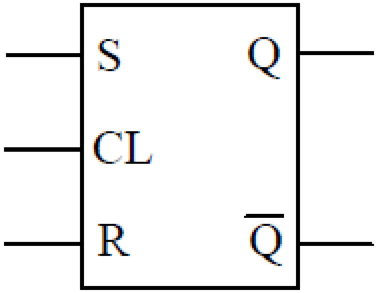
Transition Table

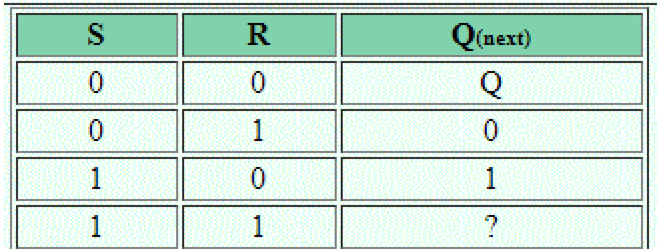
S	R	Q	$\bar{Q}$	
0	0	1	1	not alowed
0	1	1	0	set
1	0	0	1	reset
1	1	Q	$\bar{Q}$	hold



Clocked SR latch

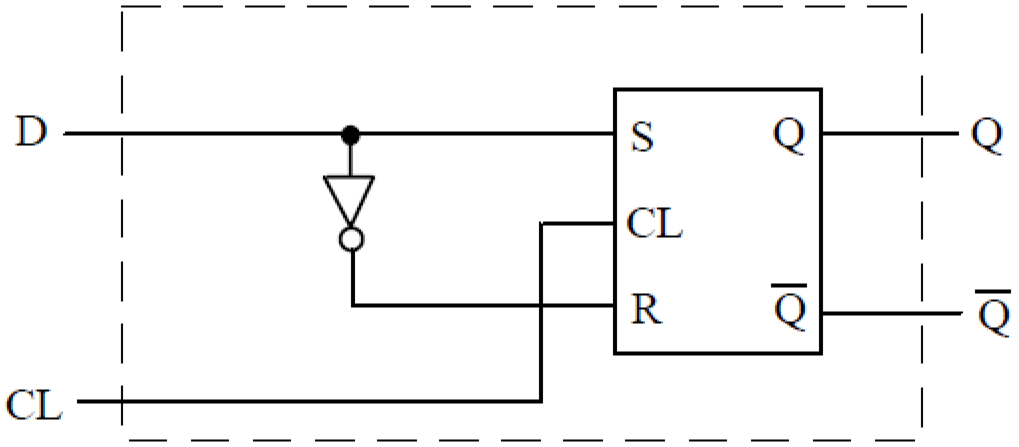


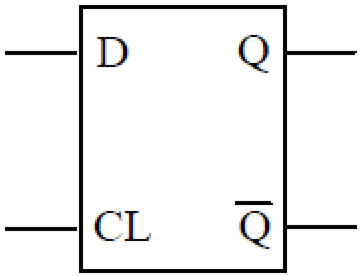


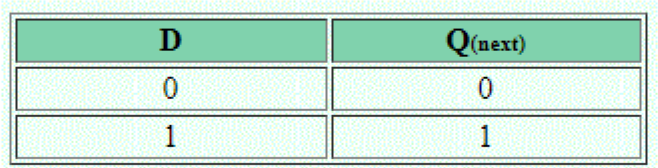


S	R	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	?

D Latch (delay)

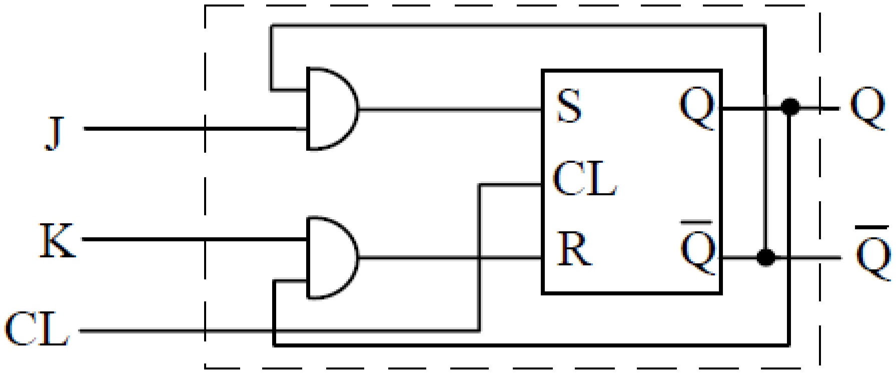


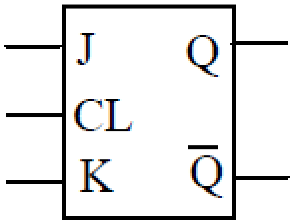


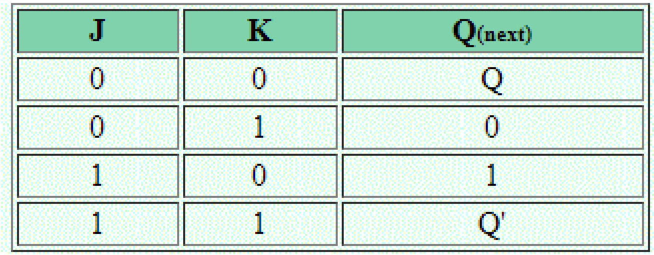


D	Q(next)
0	0
1	1

JK Latch

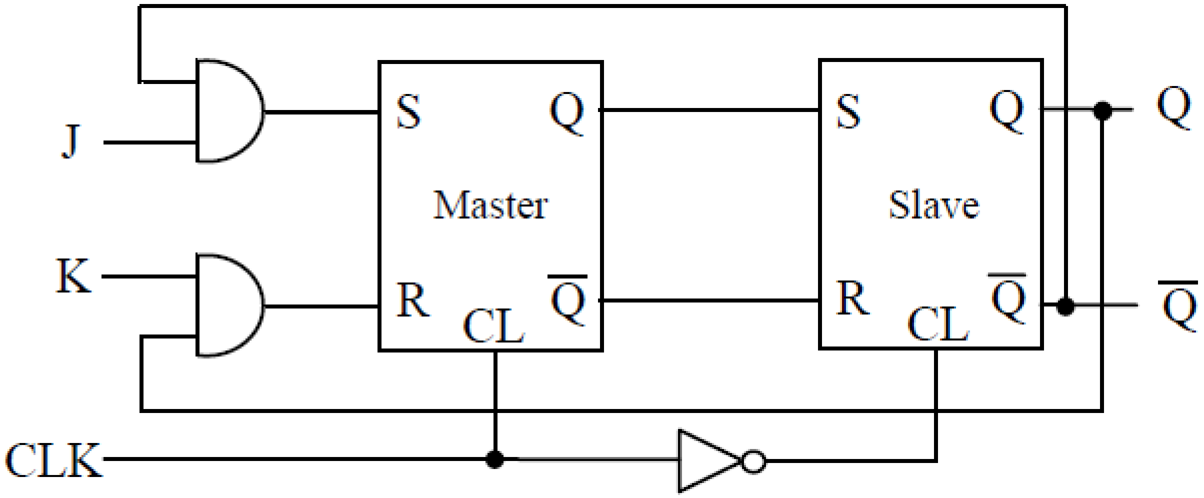


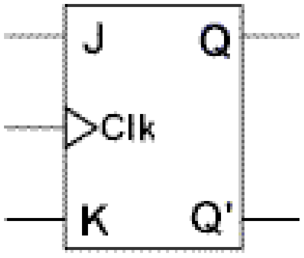


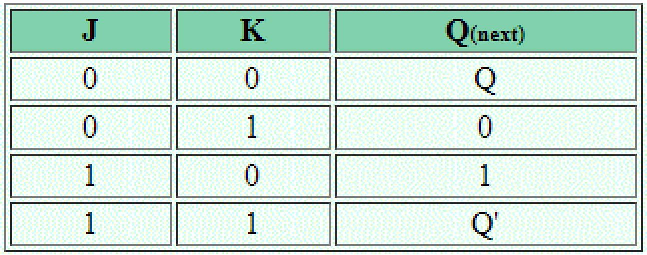


J	K	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	Q'

JK Flip-Flop

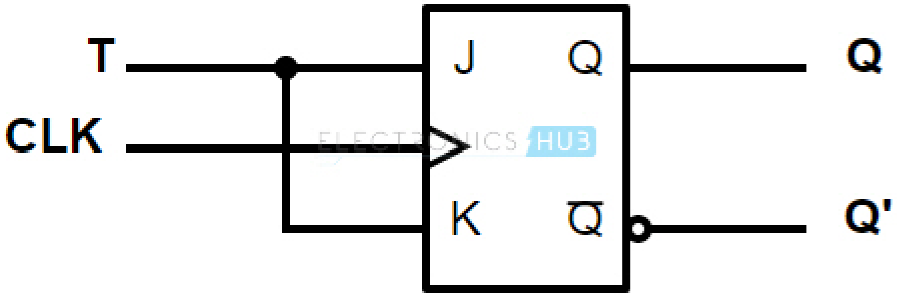


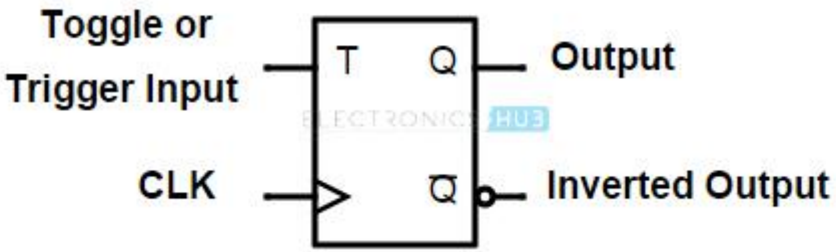


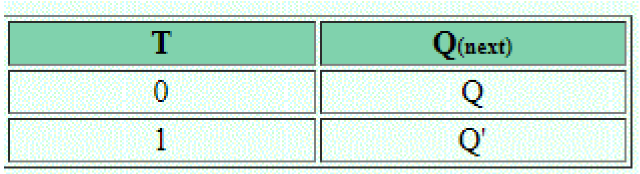


J	K	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	Q'

T Flip-Flop (toggle)

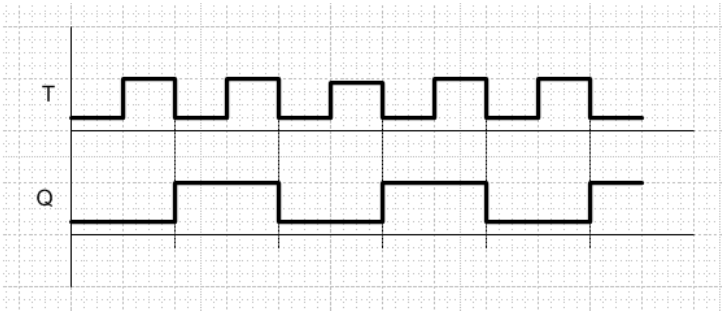






T	Q(next)
0	Q
1	Q'

T flip-flops are used in counters and frequency dividers because the main property of the T flip-flop is: dividing
the input frequency by 2.



Registers

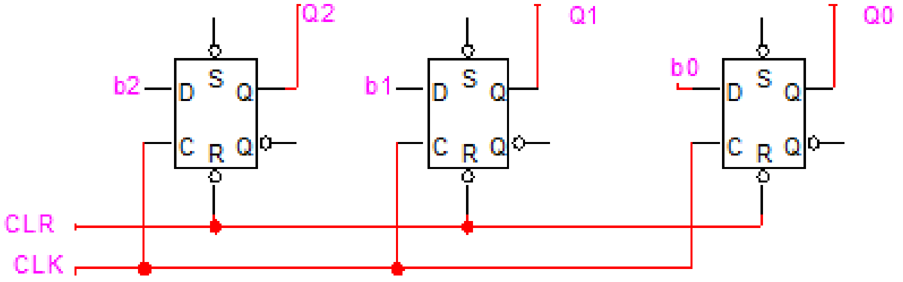
Registers are often used to store and to process binary information. They contain flip-flops, controlled by combinational circuits.

Common micro-operations in registers:
HOLD
SHIFT RIGHT
SHIFT LEFT
LOAD

Registers can be:
Serial in
Serial out
Parallel in
Parallel out
SISO, SIPO, PISO, PIPO

Parallel registers

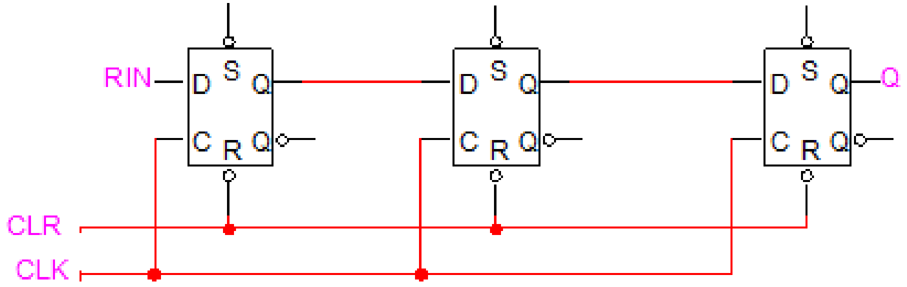
Binary words can be writing or reading in such registers at the same time. The main function of them to store binary information.



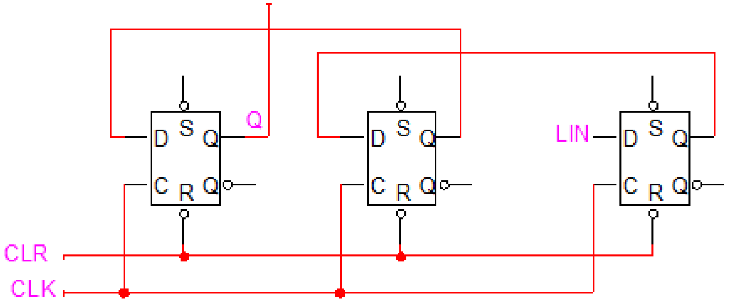
Serial registers

In these registers binary words can be written only through serial inputs LIN or RIN. The binary word can be read through serial output Q. Such registers are used to shift binary words.

Shift right serial register

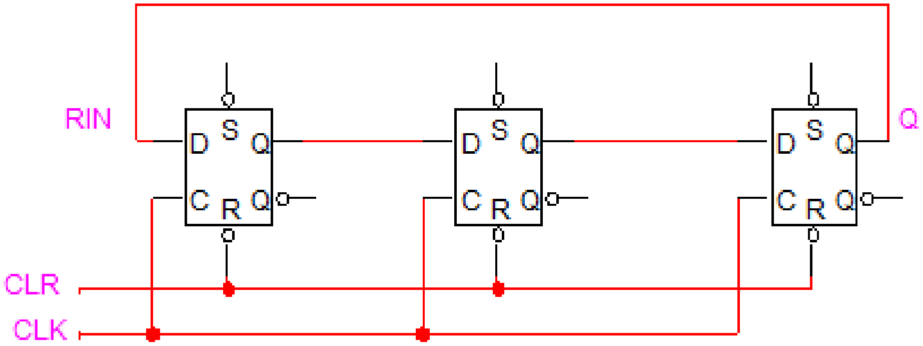


Shift left serial register



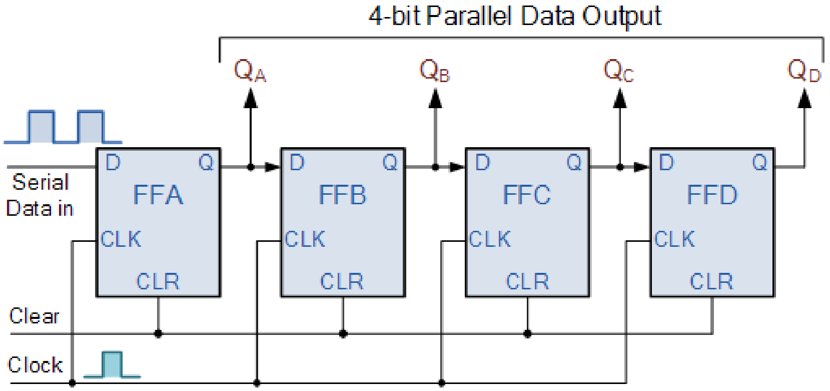
The RIN specifies a new bit to be shifted into one end at each clock tick. This bit appears at the serial output after n clock ticks. Thus an n-bit serial-in, serial-out shift register can be used to delay a signal by n clock ticks

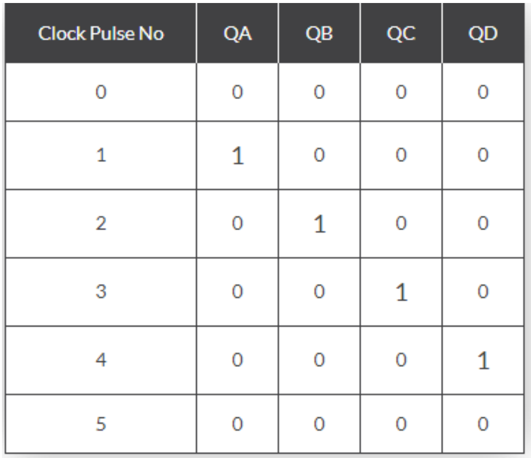
- The register that can shift only in one direction is called a **unidirectional shift register**.
- The register that can shift in either of two directions – left or right is called a **bidirectional shift register.**
- Shift left on n-bit is equivalent with multiplication on 2^n. Shift right on n-bit is equivalent with division on 2^n.
- D shift register can be combined with a combinational logic to form a state machine whose state diagram is cyclic.
- Such a circuit is called a **shift-register counter (ring counter)**. Unlike a binary counter a shift-register counter does not count in an ascending or descending binary sequence, but it is useful in many control application.



Serial-in to Parallel-out (SIPO) Shift Register

4-bit Serial-in to Parallel-out Shift Register





Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

The operation is as follows. Lets assume that all the flip-flops have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D 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 Q_A will be set to logic “1” with all the other outputs still remaining at logic “0”. Assume now that the DATA input pin of FFA has returned 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 Q_B HIGH to logic “1” as its input D has the logic “1” level on it from Q_A. The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at Q_A.

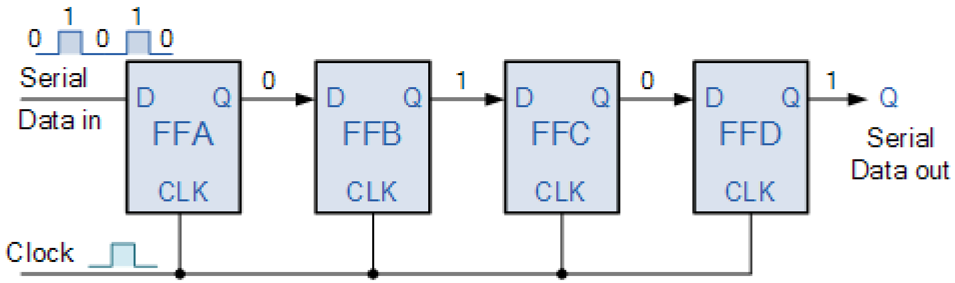
When the third clock pulse arrives this logic “1” value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level “0” because the input to FFA has remained constant at logic level “0”.

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 Q_A to Q_D.

Then the data has been converted from a serial data input signal to a parallel data output.

Serial-in to Serial-out (SISO) Shift Register

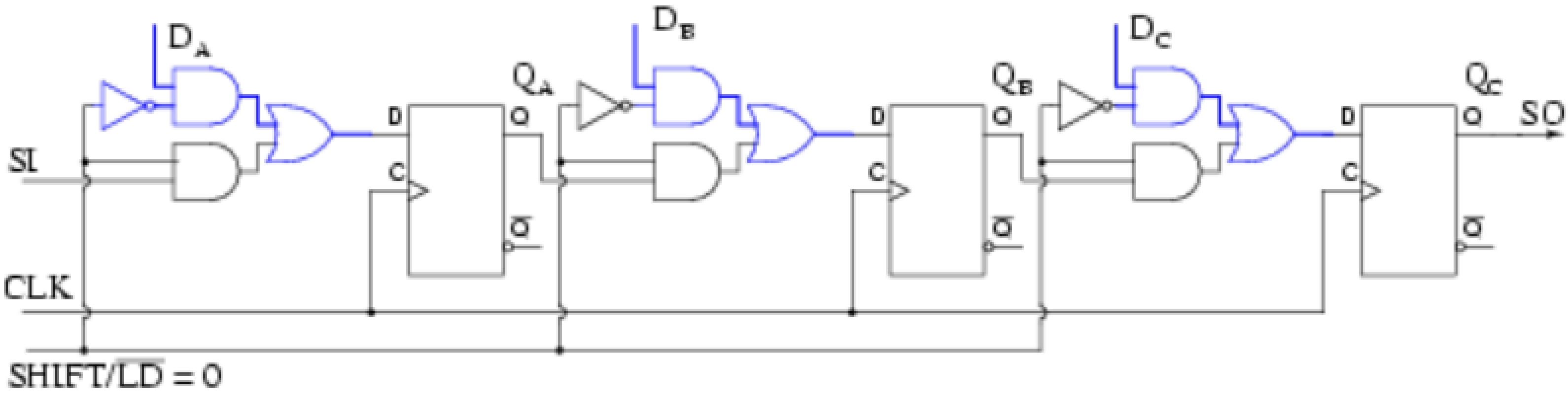
4-bit Serial-in to Serial-out Shift Register



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 Q_A to Q_D, 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) which determines what enters the left hand flip-flop, the serial output (SO) which 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.

Parallel-in/Serial-out register (PISO)

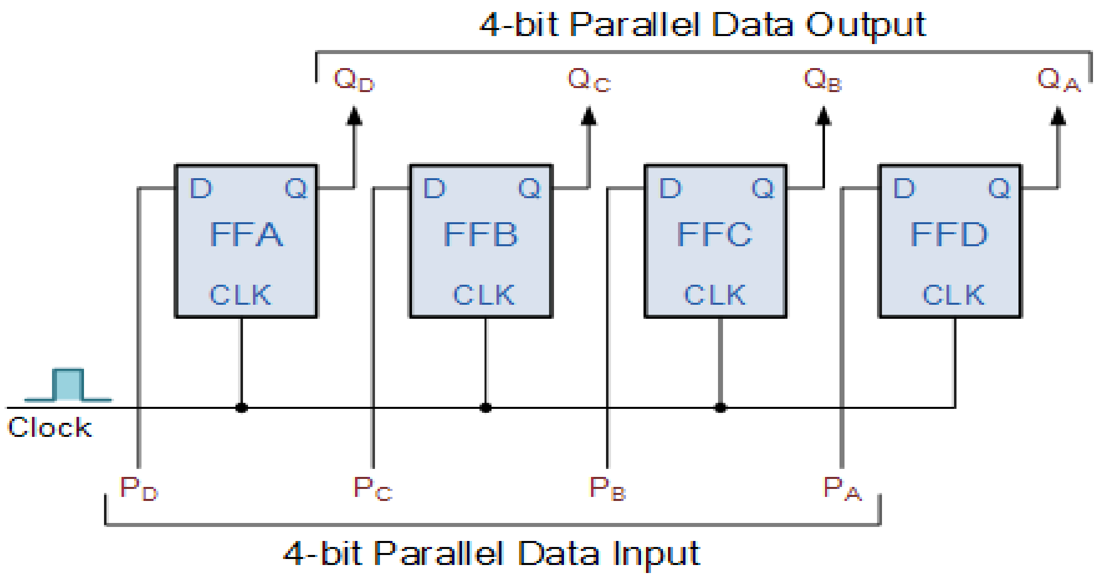


The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above. The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins P_A to P_D of the register. The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at P_A to P_D.

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.

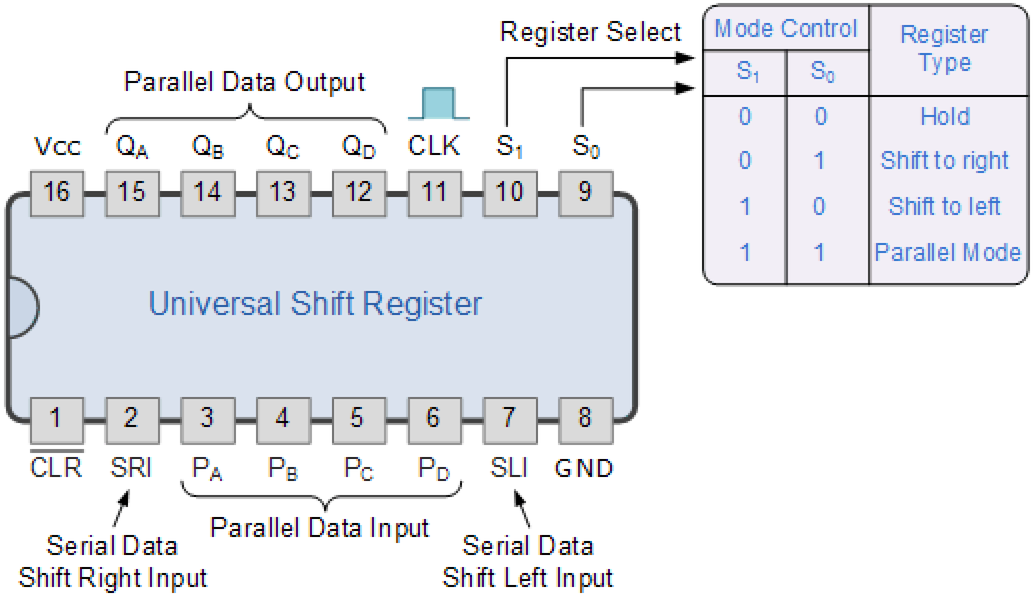
Parallel-in to Parallel-out (PIPO) Shift Register

4-bit Parallel-in to Parallel-out 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 P_A to P_D and then transferred together directly to their respective output pins Q_A to Q_D by the same clock pulse. Then one clock pulse loads and unloads the register. This arrangement for parallel loading and unloading is shown below.

4-bit Universal Shift Register 74LS194



Mode Control		Register Type
S_1	S_0	
0	0	Hold
0	1	Shift to right
1	0	Shift to left
1	1	Parallel Mode

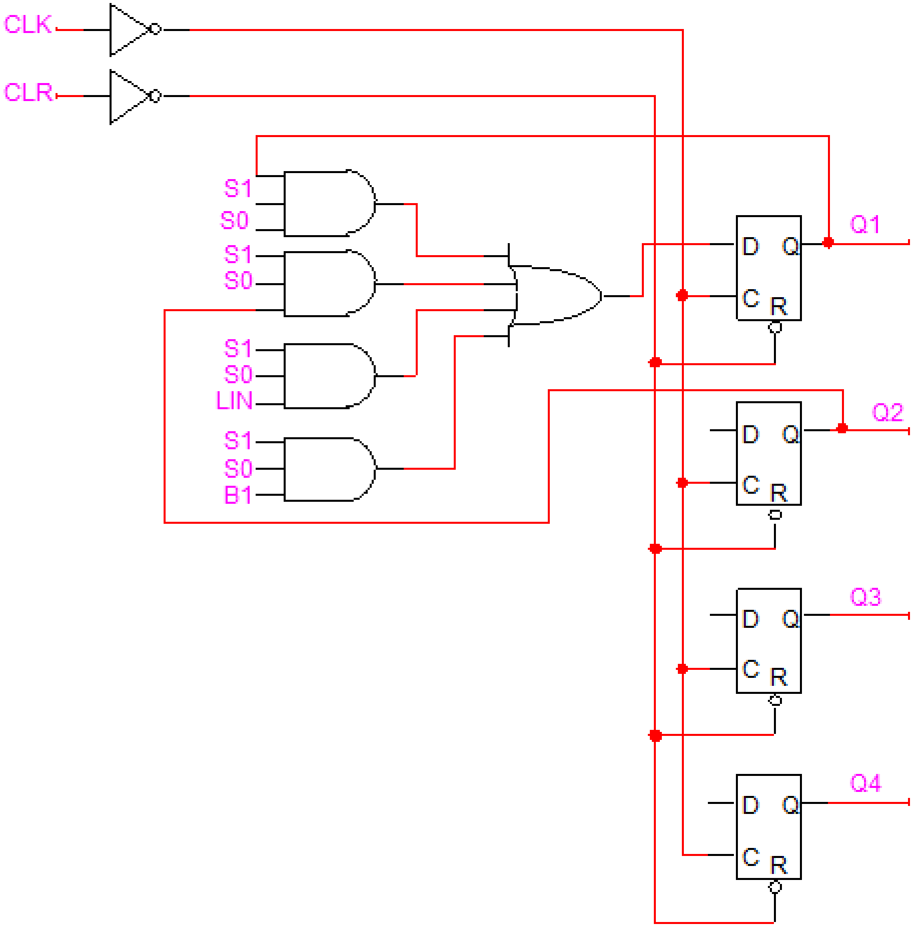
Universal shift registers are very useful digital devices. They can be configured to respond to operations that require some form of temporary memory storage or for the delay of information such as the SISO or PIPO configuration modes or transfer data from one point to another in either a serial or parallel format. Universal shift registers are frequently used in arithmetic operations to shift data to the left or right for multiplication or division.

A transition table for 74-194

Functions	Inputs					
	S1	S2	$Q4^*$	$Q3^*$	$Q2^*$	$Q1^*$
Hold	0	0	Q4	Q3	Q2	Q1
Shift-right	0	1	RIN	Q4	Q3	Q2
Shift-left	1	0	Q3	Q2	Q1	LIN
Load	1	1	B4	B3	B2	B1

This functions table is highly compressed since it does not contain columns for most of the inputs B4 – B1, RIN, LIN, S1, and S0. In this case we can have 2^8 possible combinations -> a 256 row table.

$D4 = S_1S_0 * Q_4 \text{ v } S_1S_0 \text{ RIN v } S_1S_0 * Q_3 \text{ v } S_1S_0 * B_4$
$D3 = S_1S_0 * Q_3 \text{ v } S_1S_0 \, Q_4 \text{ v } S_1S_0 * Q_2 \text{ v } S_1S_0 * B_3$
$D2 = S_1S_0 * Q_2 \text{ v } S_1S_0 \, Q_3 \text{ v } S_1S_0 * Q_1 \text{ v } S_1S_0 * B_2$
$D1 = S_1S_0 * Q_1 \text{ v } S_1S_0 \, Q_2 \text{ v } S_1S_0 * \text{LIN v } S_1S_0 * B_1$


CLK
CLR
S1
S0
S1
S0
S1
S0
LIN
S1
S0
B1
D
Q
C
R
Q1
D
Q
C
R
Q2
D
Q
C
R
Q3
D
Q
C
R
Q4

Counters

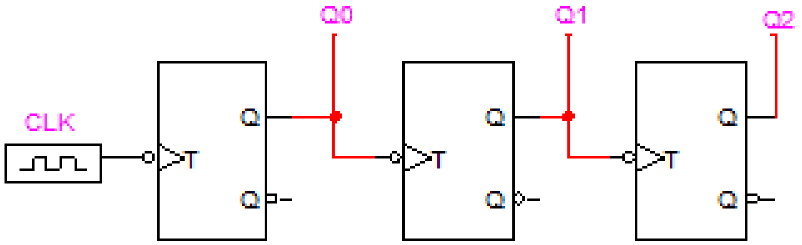
- The name counter is generally used for any clocked sequential circuit whose state diagram contains a single cycle.
- The **modulus** of a counter is the number of states in the cycle. A counter with m states is called a **modulo-m counter** or sometimes, a **divide-by-m counter**.
- A counter with **non-power-of-2** modulus has extra states that are not used in normal operation.
- The most commonly used counter is an **n-bit binary counter**. Such a counter has n flip-flops and 2^n states.
- Counter can be **ascending** and **descending**, **synchronous** and **asynchronous (ripple).**

Ripple Counters

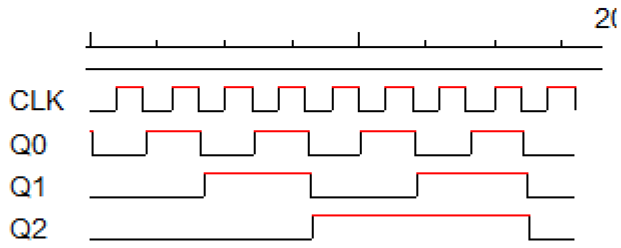
An n-bit binary counter can be designed with just n flip-flops and no other components for any value of n.

The counter is called the ripple counter because the carry information ripples from the less significant bits to the more significant bits, one bit at a time.

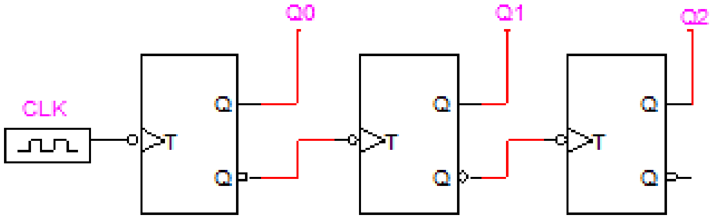
An 3-bit ascending ripple counter



Timing diagram for the ascending counter



An 3-bit descending ripple counter



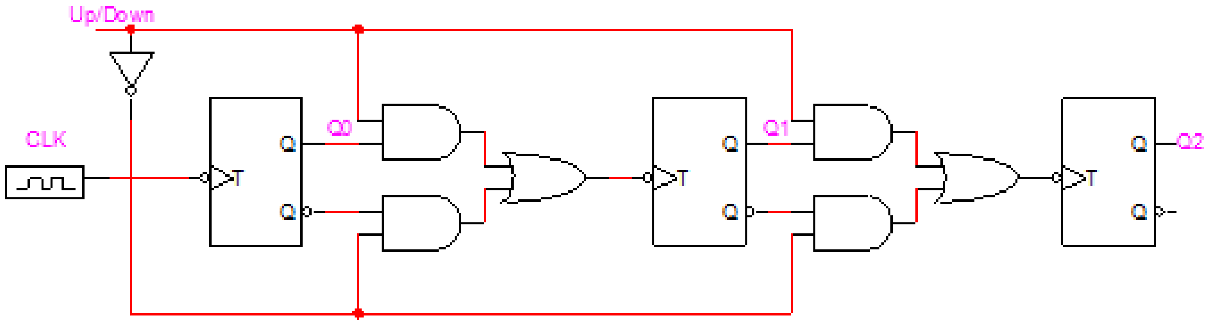
T flip-flop changes state on every falling edge of its clock input. Thus each bit of the counter toggles if and only if the immediately preceding bit changes from 1 to 0

Bidirectional ripple counters

The counter that can count in either of two directions (ascending and descending) is called a bidirectional counter.

Such a counter has a control signal to determine the direction of counting.

3-bit bidirectional ripple counter

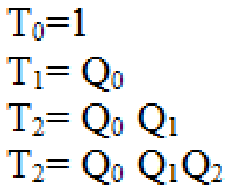


Synchronous Counters

Although a ripple counter requires fewer components that any other type of binary counter, it does so at a price-it is slower that any other type of binary counter. In the worst case, when the most significant bit must change, the output is not valid until time n*t_{TQ}, where t_{TQ} is the propagation delay from input to output of a T flip-flop.

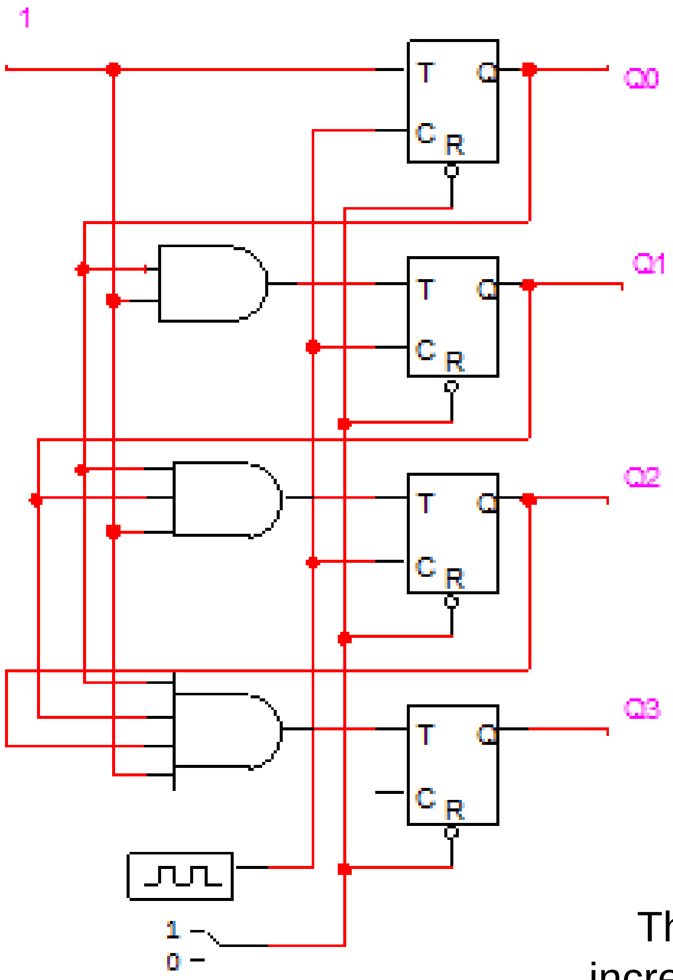
A synchronous counter connects all of its flip-flops clock inputs to the same common clock signal, so that all flip-flops outputs change at the same time, after only t_{TQ} ns of delay. This requires the use of T flip-flops with clock inputs; the output toggles on the falling edge of CLK if and only if T=1.

Toggle equations:



T_0=1
T_1= Q_0
T_2= Q_0 Q_1
T_2= Q_0 Q_1Q_2

A synchronous 4-bit binary counter with parallel enable logic



$T_0=1$

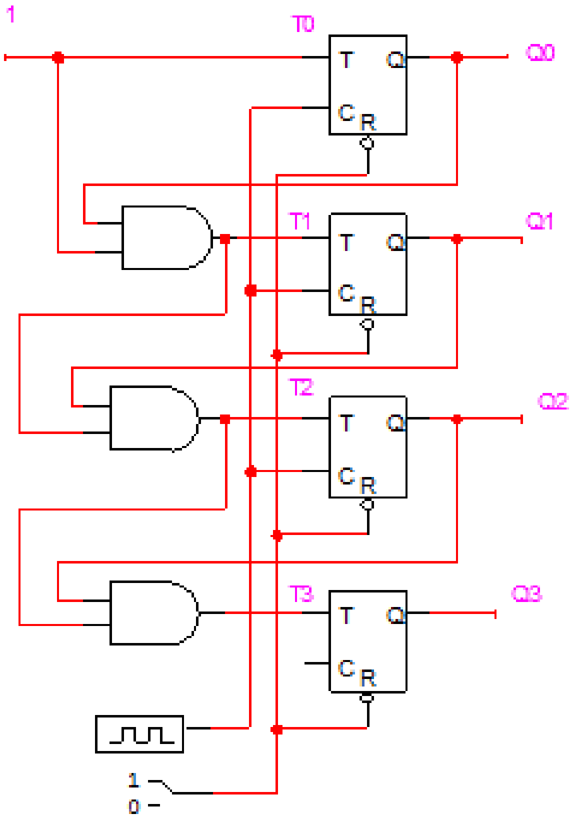
$T_1= Q_0$

$T_2= Q_0\ Q_1$

$T_2= Q_0\ Q_1Q_2$

This is the fastest binary counter structure with increasing the number of flip-flops, the number of inputs to the AND gates are also increased.

A synchronous 4-bit binary counter with serial enable logic.



T_0=1

T_1= Q_0 T_0=Q_0

T_2= T_1 Q_1

T_2= T_2 Q_2

This circuit uses only 2-input AND gates but is not so fast

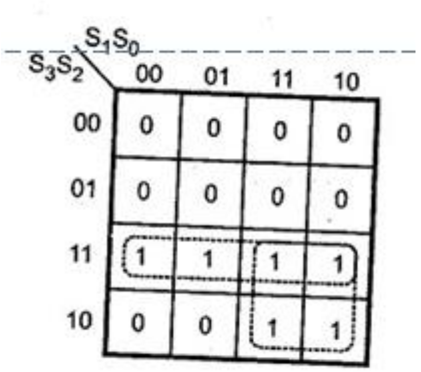
A non-power-of-2 counters

$2^{n-1}<m<2^n$

A ripple counter with a non-power-of -2 modulus has an additional logic gate the output of which generates a flip-flps reset signal.

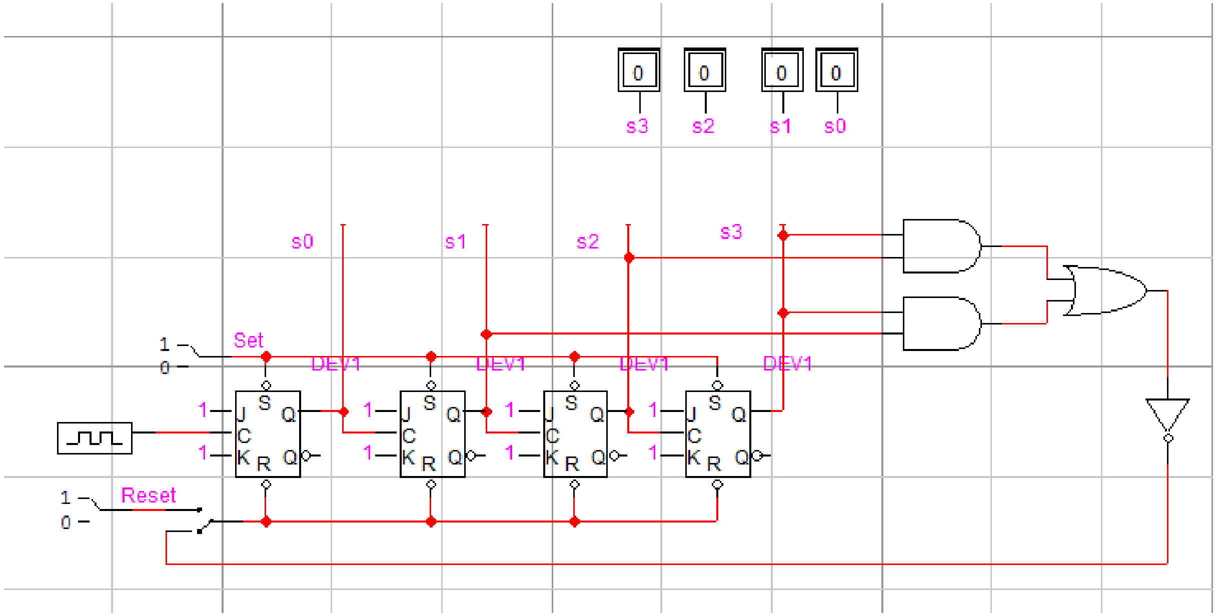
Ex: m=10

Valid states are 0-9, non valid states 10-15.



$S_3S_2 \backslash S_1S_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$Y = S_3S_2 + S_3S_1$$


0
0
0
0
s3
s2
s1
s0
s0
s1
s2
s3
Set
1
0
DEV1
DEV1
DEV1
DEV1
J
S
Q
C
K
R
Q
1
Reset
1
0