USN □□□□□□□□□□

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 2 – January 2022

Sub:	Analog and Digital Electronics				Sub Code:	18CS33	Branch:	ISE	
Date:	25/1/2022	Duration:	90 min’s	Max Marks: 50		Sem/Sec:	III / A, B and C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Explain the structure of VHDL Program. Write VHDL Code for 4-bit parallel adder.	10	CO5	L2
2	Construct SR latch using NOR gate. Show how SR latch can be used for switch de-bouncing.	10	CO4	L3
3	a. With an example, explain the syntax of conditional signal assignment statement in VHDL.	4	CO5	L2
	b. Construct binary counter which counts from 000 to 111 using three T flip flops.	6	CO4	L3
4	Draw the logic diagram of Maser Slave JK flip flop using NAND gates and explain the working with suitable timing diagram	10	CO4	L2
5	Derive the characteristics equations for D, T, SR, JK FLIP-FLOPS	10	CO4	L1
6	With neat sketch, explain the working principle of SISO & PIPO shift register.	10	CO4	L2

Faculty Signature CCI Signature HOD Signature

USN □□□□□□□□□□

CELEBRATING 25 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 2 – January 2022

Sub:	Analog and Digital Electronics				Sub Code:	18CS33	Branch:	ISE	
Date:	25/1/2022	Duration:	90 min’s	Max Marks: 50		Sem/Sec:	III / A, B and C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Explain the structure of VHDL Program. Write VHDL Code for 4-bit parallel adder.	10	CO5	L2
2	Construct SR latch using NOR gate. Show how SR latch can be used for switch de-bouncing.	10	CO4	L3
3	a. With an example, explain the syntax of conditional signal assignment statement in VHDL.	4	CO5	L2
	b. Construct binary counter which counts from 000 to 111 using three T flip flops.	6	CO4	L3
4	Draw the logic diagram of Maser Slave JK flip flop using NAND gates and explain the working with suitable timing diagram	10	CO4	L2
5	Derive the characteristics equations for D, T, SR, JK FLIP-FLOPS	10	CO4	L1
6	With neat sketch, explain the working principle of SISO & PIPO shift register.	10	CO4	L2

Faculty Signature CCI Signature HOD Signature

USN □□□□□□□□□□

Internal Assessment Test 2 – January 2022

Sub:	Analog and Digital Electronics				Sub Code:	18CS33	Branch:	ISE	
Date:	25/1/2022	Duration:	90 min’s	Max Marks:	50	Sem/Sec:	III / A, B and C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Explain the structure of VHDL Program. Write VHDL Code for 4-bit parallel adder..	10	CO5	L2

Solution:

When we describe a system in VHDL, we must specify an entity and an architecture at the top level, and also specify an entity and architecture for each of the component modules that are part of the system. Each entity declaration includes a list of interface signals that can be used to connect to other

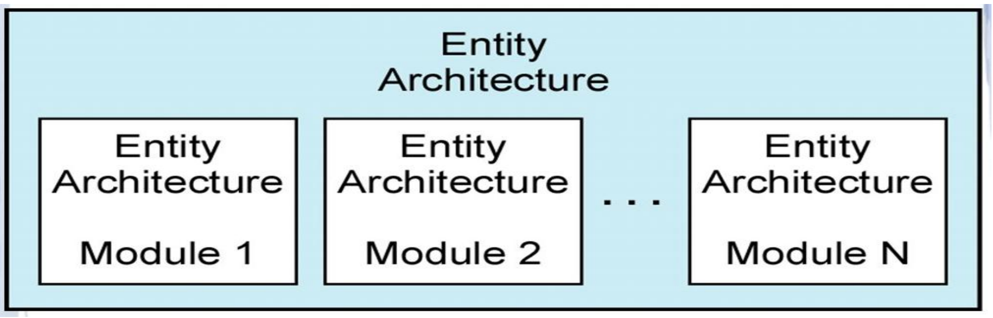
modules or to the outside world. We will use entity declarations of the form:

entity entity-name **is** [**port**(interface-signal-declaration);] **end** [**entity**] [entity-name];

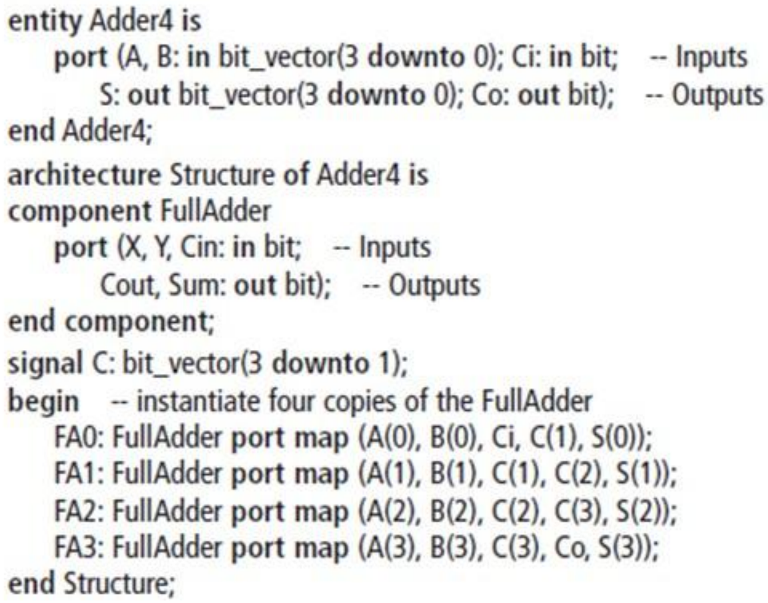
The items enclosed in square brackets are optional. The interface-signal-declaration normally has the following form:

list-of-interface-signals: mode type [: _ initial-value]

{; list-of-interface-signals: mode type [: _ initial-value]};



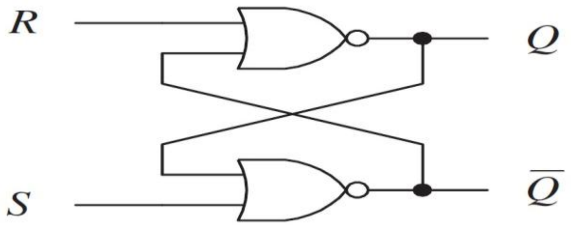
VHDL Program Structure



```
entity Adder4 is
    port (A, B: in bit_vector(3 downto 0); Ci: in bit;    -- Inputs
        S: out bit_vector(3 downto 0); Co: out bit);    -- Outputs
end Adder4;
architecture Structure of Adder4 is
component FullAdder
    port (X, Y, Cin: in bit;    -- Inputs
        Cout, Sum: out bit);    -- Outputs
end component;
signal C: bit_vector(3 downto 1);
begin    -- instantiate four copies of the FullAdder
    FA0: FullAdder port map (A(0), B(0), Ci, C(1), S(0));
    FA1: FullAdder port map (A(1), B(1), C(1), C(2), S(1));
    FA2: FullAdder port map (A(2), B(2), C(2), C(3), S(2));
    FA3: FullAdder port map (A(3), B(3), C(3), Co, S(3));
end Structure;
```

2	Construct SR latch using NOR gate. Show how SR latch can be used for switch de-bouncing.	10	CO4	L3

Solution:



From this, the flip-flop has two inputs: R and S and two outputs: Q and Q and it is clear from the representation that the outputs are complementary to each other. Let us try to analyze the different possibilities of inputs and their corresponding outputs.

An important point to note here is that for a NOR gate, Logic ‘1’ is a dominating input and if any one of its input is Logic ‘1’ (HIGH), then the output is Logic ‘0’ (LOW), irrespective of the other input. With this in mind, let us analyze the above circuit.

Case 1: R = 0 and S = 0

In the first case, the inputs of both the NOR gates are Logic ‘0’. As neither of them are dominating inputs, they have no effect on the output. So, the output retains their previous states i.e., there is no change in the output. This condition is called as Hold Condition or No Change Condition.

Case 2: R = 0 and S = 1

In this case, the ‘S’ input is 1, which means the output of the NOR Gate B will become 0. As a result, both the inputs of NOR Gate A become 0 and hence the output of the NOR Gate A and thus the value of Q is 1 (HIGH). As ‘1’ at input S makes the output to switch to one of its stable states and sets it to ‘1’, the S input is known as SET input.

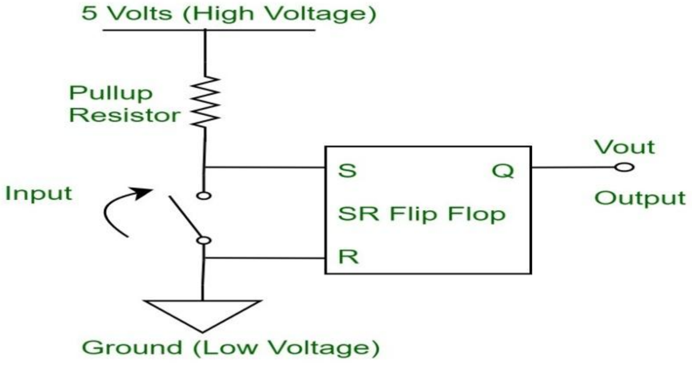
Case 3: R = 1 and S = 0

In this case, the ‘R’ input is 1, which means the output of the NOR Gate A will become 0 i.e., Q is 0 (LOW). As a result, both the inputs of NOR Gate B become 0 and hence the output of the NOR Gate B is 1 (HIGH). As ‘1’ at input R makes the output to switch to one of its stable states and resets it to ‘0’, the R input is known as RESET input.

Case 4: R = 1 and S = 1

This input condition is forbidden as it forces outputs of both NOR Gates to become 0, which is a violation of complementary outputs. Even if this input condition is applied, if the next inputs become R = 0 and S = 0 (hold condition), then it causes a ‘race condition’ between the NOR Gates, which causes an unstable or unpredictable state at the output.

Hence, the input condition R = 1 and S = 1 is simply not used.



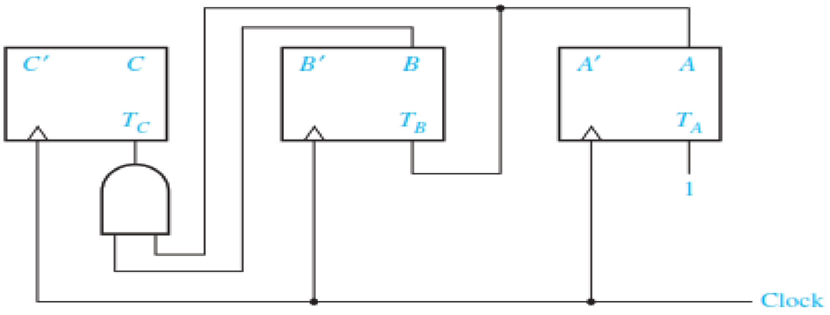
Switch Debouncing

Use of S-R Flip Flop Latch circuit. The circuit when introduced in the output part of the switch, it will retain the voltage level of the input as the output state. Thus, latching to the input, when change in state is introduced. This method is useful, but adds to the bulkiness of the simple circuit.

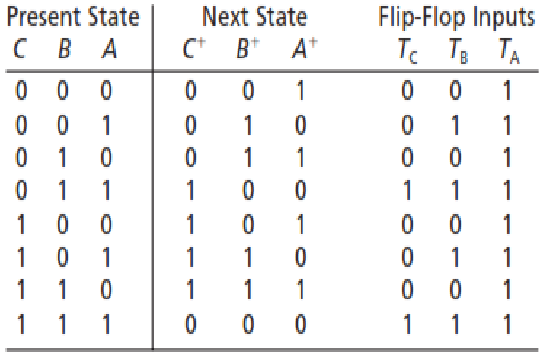
3 a. With an example, explain the syntax of conditional signal assignment statement in VHDL. 4 CO5 L2

```
signal_name <= expression_1 when condition_1 else
               expression_2 when condition_2 else
               expression_3 ;
architecture COND of BRANCH is
begin
   Z <= A when X > 5 else
        B when X < 5 else
        C;
end COND;
```

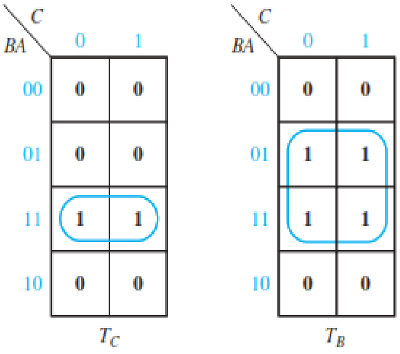
b. Construct binary counter which counts from 000 to 111 using three T flip flops. 6 CO4 L3



Binary Counter



Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

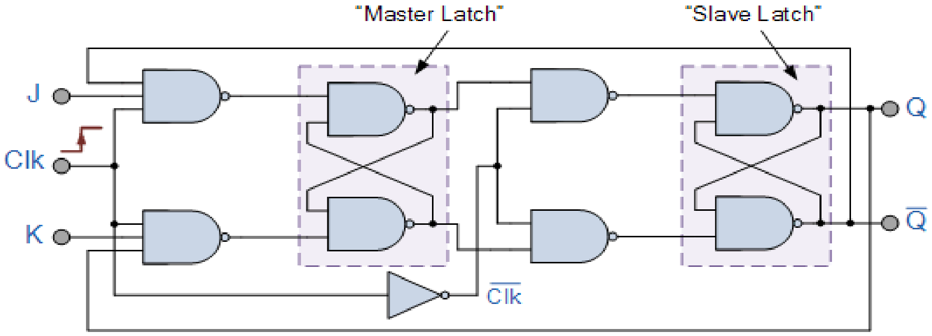


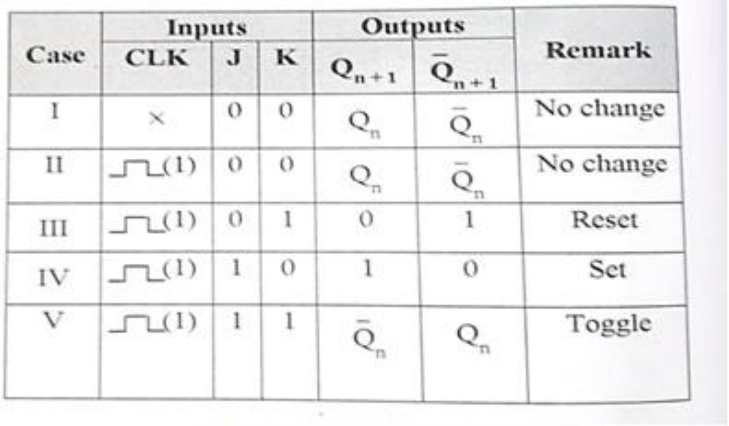
Truth Table and K Map

TC, TB, and TA are now derived from the table as functions of C, B, and A. By inspection, T_A= 1. Using K-maps for TC and TB, we will get T_C = BA and T_B = A.

4 Draw the logic diagram of Maser Slave JK flip flop using NAND gates and explain the working with suitable timing diagram 10 CO4 L2

Solution:





Case	Inputs			Outputs		Remark
	CLK	J	K	Q_{n+1}	$\bar{Q}_{n+1}$	
I	×	0	0	Q_n	$\bar{Q}_n$	No change
II	⎍(1)	0	0	Q_n	$\bar{Q}_n$	No change
III	⎍(1)	0	1	0	1	Reset
IV	⎍(1)	1	0	1	0	Set
V	⎍(1)	1	1	$\bar{Q}_n$	Q_n	Toggle

Truth table of Master slave JK FF

When the clock pulse is true, the slave flip flop will be in the isolated state, and the system's state may be affected by the J and K inputs. The "slave" remains isolated until the CP is 1. When the CP set to 0, the master flip-flop passes the information to the slave flip flop to obtain the output.

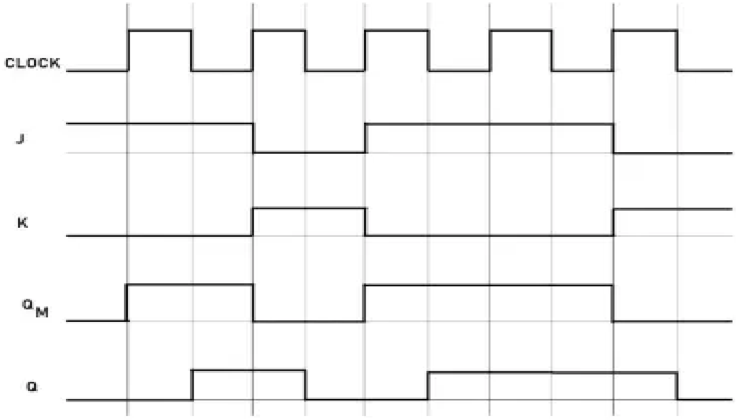
The master flip flop responds first from the slave because the master flip flop is the positive level trigger, and the slave flip flop is the negative level trigger.

The output Q'=1 of the master flip flop is passed to the slave flip flop as an input K when the input J set to 0 and K set to 1. The clock forces the slave flip flop to work as reset, and then the slave copies the master flip flop.

When J=1, and K=0, the output Q=1 is passed to the J input of the slave. The clock's negative transition sets the slave and copies the master.

The master flip flop toggles on the clock's positive transition when the inputs J and K set to 1. At that time, the slave flip flop toggles on the clock's negative transition.

The flip flop will be disabled, and Q remains unchanged when both the inputs of the JK flip flop set to 0



5	Derive the characteristics equations for D, T, SR, JK FLIP-FLOPS	10	CO4	L1

SR Flip flop

1. Truth Table for SR flip-flop

S	R	Q_n	Q_{n+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	invalid	
1	1	invalid	

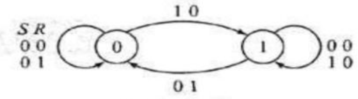
4. K-map Simplification

$S\backslash RQ_n$	00	01	11	10
0	0^{0}	1^{1}	0^{3}	0^{2}
1	1^{4}	1^{5}	X^{7}	X^{6}

$D = S + \bar{R}Q_n$

. Excitation Table for SR flip-flop

Outputs		Inputs	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



(a) *SR* flip-flop

D-Flipflop

1. Truth Table for D Flip-Flop

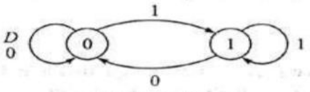
Input	Outputs	
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

$Q_n\backslash D$	0	1
0	0	1
1	0	1

(b) $Q_{n+1} = D$

Excitation Table for D Flip Flop

Outputs		Input
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1



(b) *D* flip-flop

T-flipflop

Truth Table for T Flip Flop

Input	Outputs	
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

4. K-map Simplification

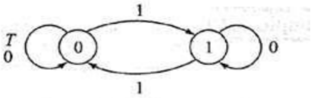
$T\backslash Q_n$	0	1
0	0^{0}	1^{1}
1	1^{2}	0^{3}

$D = T\bar{Q}_n + \bar{T}Q_n$

$= T \oplus Q_n$

Excitation Table for T Flip Flop

Outputs		Input
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0



(d) *T* flip-flop

. K-map Simplification

$J\backslash KQ_n$	00	01	11	10
0	0^{0}	1^{1}	0^{3}	0^{2}
1	1^{4}	1^{5}	0^{7}	1^{6}

$D = J\bar{Q}_n + \bar{K}Q_n$

JK-Flipflop

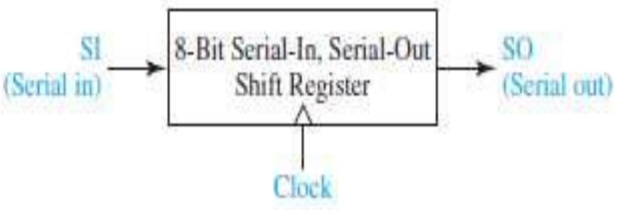
Truth Table for JK Flip-Flop

Inputs		Outputs	
J	K	Q_n	Q_{n+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

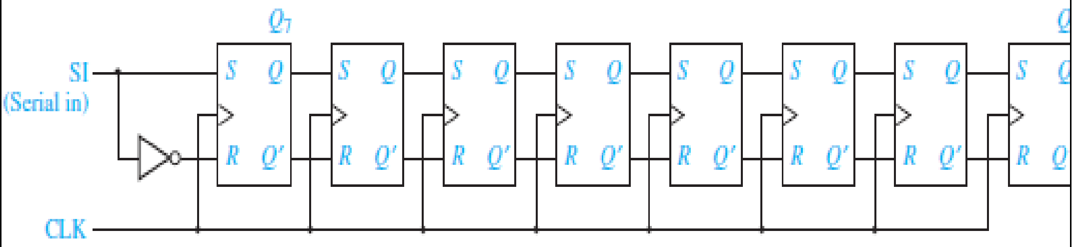
6	With neat sketch, explain the working principle of SISO & PIPO shift register.	10	CO4	L2

Solution:

8-Bit Serial-in, Serial-out Shift Register



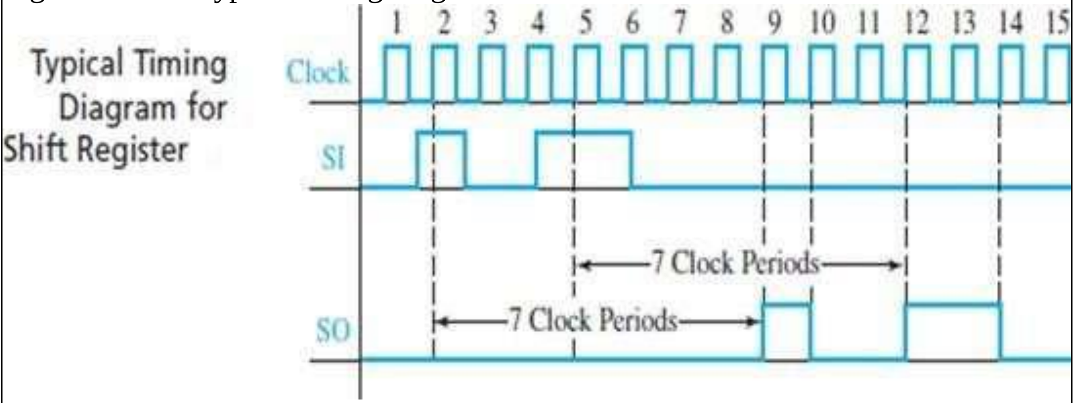
(a) Block diagram



(b) Logic diagram

The following Figure
illustrates an 8-bit serial-in, serial-out shift register. Serial in means that data is shifted into the first flipflop
one bit at a time, and the flip-flops cannot be loaded in parallel. Serial out means that data can only
be read out of the last flip-flop and the outputs from the other flip-flops are not connected to terminals of
the integrated circuit.

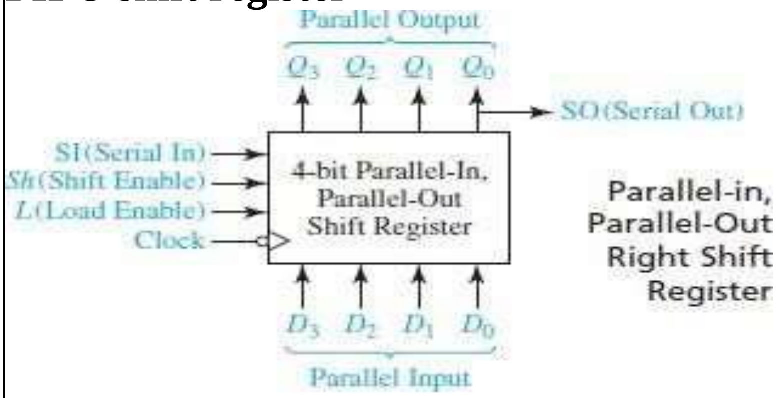
The inputs to the first flip-flop are $S = SI$ and $R = SI'$. Thus, if $SI = 1$, a 1 is shifted into the register when it is clocked, and if $SI = 0$, a 0 is shifted in. The following Figure shows a typical timing diagram.



The following Figure (a) shows a 4-bit parallel-in, parallel-out shift register. Parallel-in implies that all
four bits can be loaded at the same time, and parallel-out implies that all bits can be read out at the same
time. The shift register has two control inputs, shift enable (Sh) and load enable (L). If $Sh = 1$ (and $L = 1$
or $L = 0$), clocking the register causes the serial input (SI) to be shifted into the first flip-flop, while the
data in flip-flops Q3,Q2, and Q1 are shifted right. If $Sh = 0$ and $L = 1$, clocking the shift register will cause the four data inputs (D3, D2, D1, D0) to be loaded in parallel into the flip-flops. If $Sh = L = 0$,

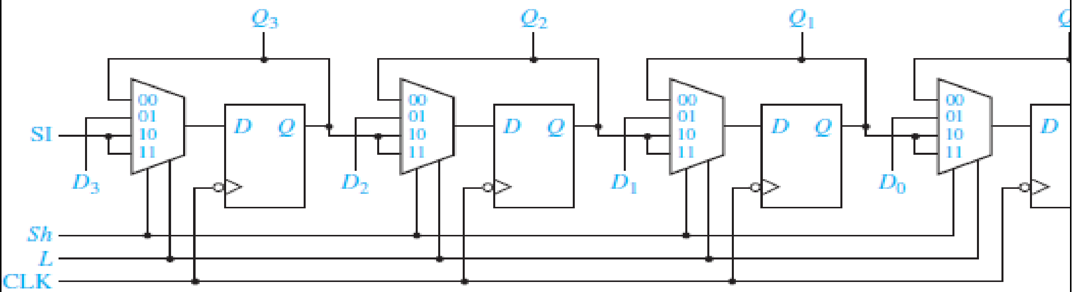
clocking the register causes no change of state.

PIPO shift register



Parallel-in, Parallel-Out Right Shift Register

(a) Block diagram



(b) Implementation using flip-flops and MUXes

The following Table summarizes the operation of this shift register. All state changes occur immediately
following the falling edge of the clock.

Shift Register Operation

Inputs		Next State				
Sh (Shift)	*L* (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	Action
0	0	Q_3	Q_2	Q_1	Q_0	No change
0	1	D_3	D_2	D_1	D_0	Load
1	X	SI	Q_3	Q_2	Q_1	Right shift

The shift register can be implemented using MUXes and D flip-flops, as shown in the above Figure (b).

For the first flip-flop, when *Sh* = *L* = *0*, the flip-flop Q3 output is selected by the MUX, so Q3$_+$ = Q3 and
no state change occurs. When *Sh* = *0* and *L* = *1*, the data input D3 is selected and loaded into the flip-flop.
When *Sh* = *1* and *L* = *0* or *1*, *SI* is selected and loaded into the flip-flop. The second MUX selects Q2,
D2, or Q3, etc. **The next-state equations for the** flip-flops are

$$Q_3^+ = Sh'\cdot L'\cdot Q_3 + Sh'\cdot L\cdot D_3 + Sh\cdot \mathrm{SI}$$

$$Q_2^+ = Sh'\cdot L'\cdot Q_2 + Sh'\cdot L\cdot D_2 + Sh\cdot Q_3$$

$$Q_1^+ = Sh'\cdot L'\cdot Q_1 + Sh'\cdot L\cdot D_1 + Sh\cdot Q_2$$

$$Q_0^+ = Sh'\cdot L'\cdot Q_0 + Sh'\cdot L\cdot D_0 + Sh\cdot Q_1$$