INSTITUTE OF ENGINEERING AND TECHNOLOGY DAVV , INDORE

CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

Session (2020-21)

4 BIT LEFT SHIFT REGISTER

Submitted to:-

Dr. Vaibhav Nema

Submitted by :-

Vishal Jaiswal

E&I (7th Sem)

Roll no.- 17E7058

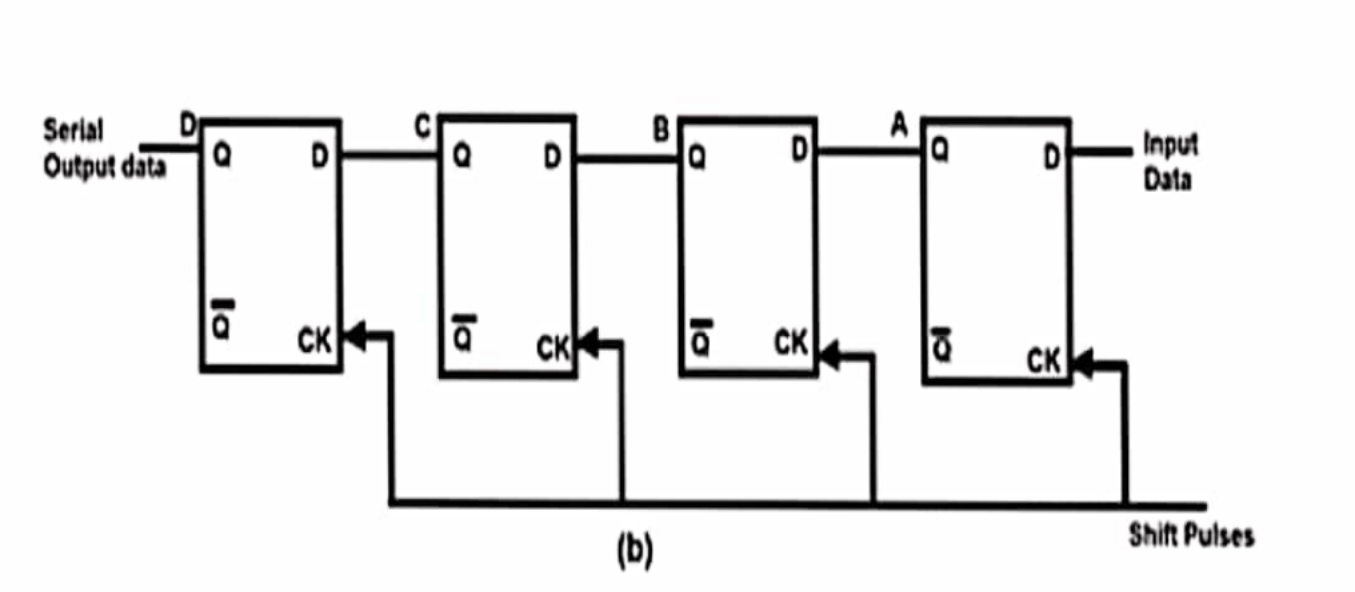
Enroll. No. - DE17249

Register :- A Register is a collection of flip flops. A flip flop is used to store single bit digital data. For storing a large number of bits, the storage capacity is increased by grouping more than one flip flops.

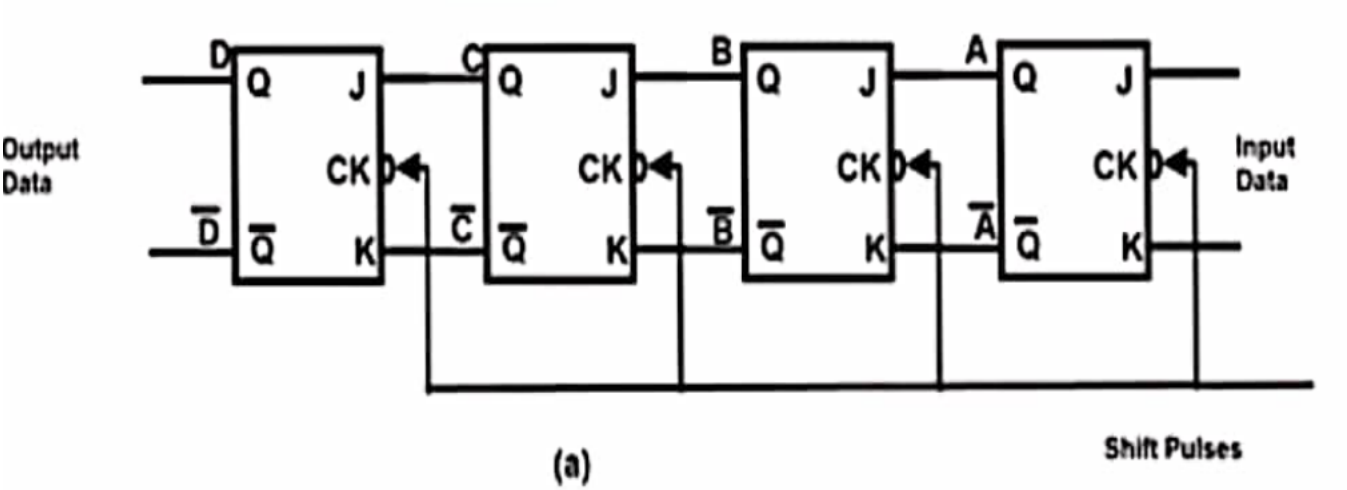
Shift Registers :- Shift Registers are sequential logic circuits, capable of storage and transfer of data. They are made up of Flip Flops which are connected in such a way that the output of one flip flop could serve as the input of the other flip-flop, depending on the type of shift registers being created. D-Flip Flop shift Register

An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data. ... The registers which will shift the bits to left are called “Shift left registers”.

LEFT SHIFT REGISTER USING D-FLIPFLOP :--



LEFT SHIFT REGISTER USING J-K FLIPFLOP :--



> Serial data 1111 provide as input

Table 1				
OPERATION OF SHIFT-LEFT REGISTER				
Shift Pulse	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1

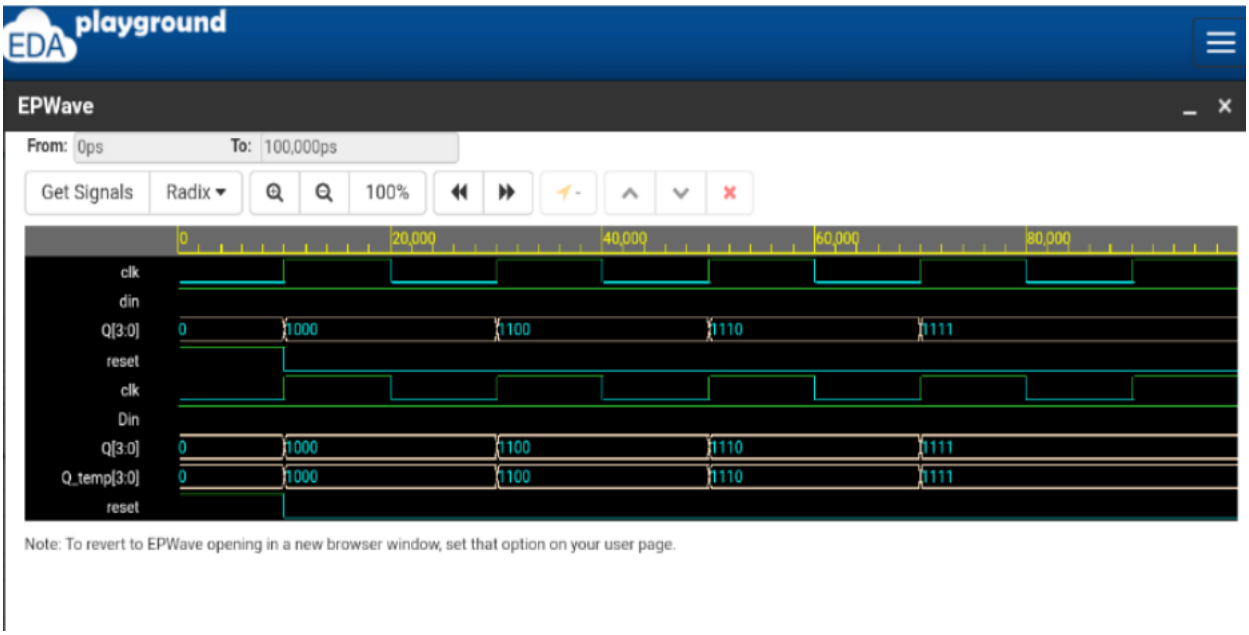
design.vhd

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity ShiftReg4bit_beh is
 Port ( Din,clk,reset : in STD_LOGIC;
 Q : out STD_LOGIC_VECTOR (3 downto 0));

end ShiftReg4bit_beh;
architecture Behavioral of ShiftReg4bit_beh is
signal Q_temp: std_logic_vector(3 downto 0);
begin
Process(clk,reset)
begin
if(reset='1')then
 Q_temp<="0000";
elsif(clk'event and clk='1')then
 Q_temp<= Din & Q_temp(3 downto 1);
end if;
end process;
Q <= Q_temp;
end Behavioral;
```

testbench.vhd

VHDL Testbench

```
library IEEE;
use IEEE.std_logic_1164.all;

ENTITY test_ShiftReg4bits IS
END test_ShiftReg4bits;
ARCHITECTURE
test_ShiftReg4bits OF
test_ShiftReg4bits IS
 COMPONENT ShiftReg4bit_beh
 PORT(din,clk,reset : IN
std_logic;
 Q : OUT std_logic_vector(3
downto 0));
 END COMPONENT;
 --Inputs
 signal din : std_logic :=
'0';
 signal clk : std_logic :=
'0';
 signal reset : std_logic :=
'0';
 --Outputs
 signal Q :
std_logic_vector(3 downto 0);
BEGIN
 uut: ShiftReg4bit_beh PORT
MAP (din,clk,reset,Q);
 clk_process :process
 begin
clk <= '0';
wait for 10ns;
clk <= '1';
wait for 10ns;
 end process;
 stim_proc: process
 begin
 Reset <= '1'; Din<='1';
 wait for 10ns;
 Reset <= '0';
 wait;
 end process;
END;
```


EDA playground
EPWave
From: 0ps
To: 100,000ps
Get Signals
Radix
100%
0
20,000
40,000
60,000
80,000
clk
din
Q[3:0]
reset
clk
Din
Q[3:0]
Q_temp[3:0]
reset
0
1000
1100
1110
1111
Note: To revert to EPWave opening in a new browser window, set that option on your user page.