INSTITUTE OF ENGINEERING & TECHNOLOGY

Devi Ahilya Vishwavidyalaya

INDORE

Department of E&I ENGINEERING

SESSION: 2017-2021

Subject: VLSI

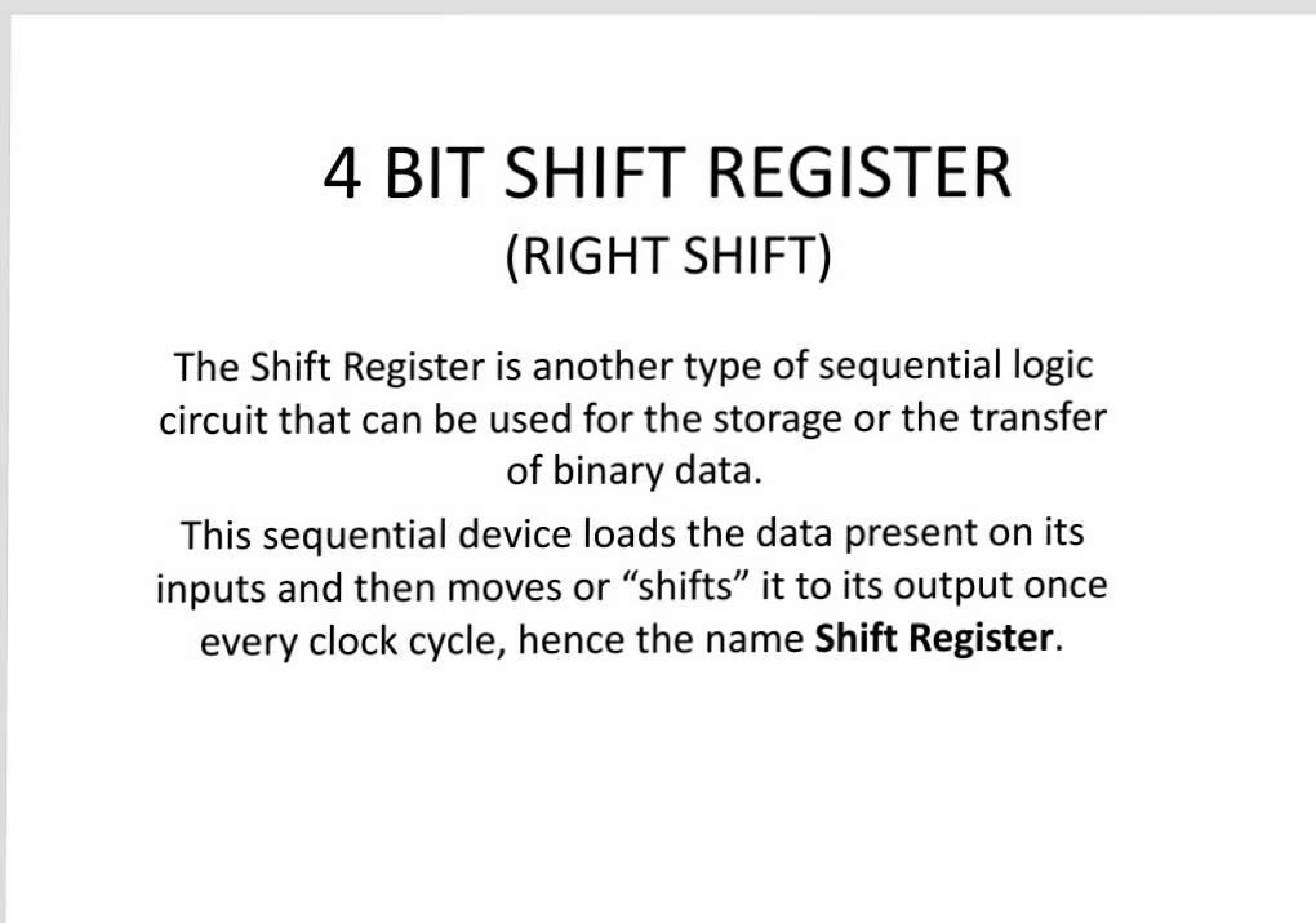
17E7050

SUBMITTED TO:

Vaibhav Neema sir

SUBMITTED BY:

siddhi sugandhi



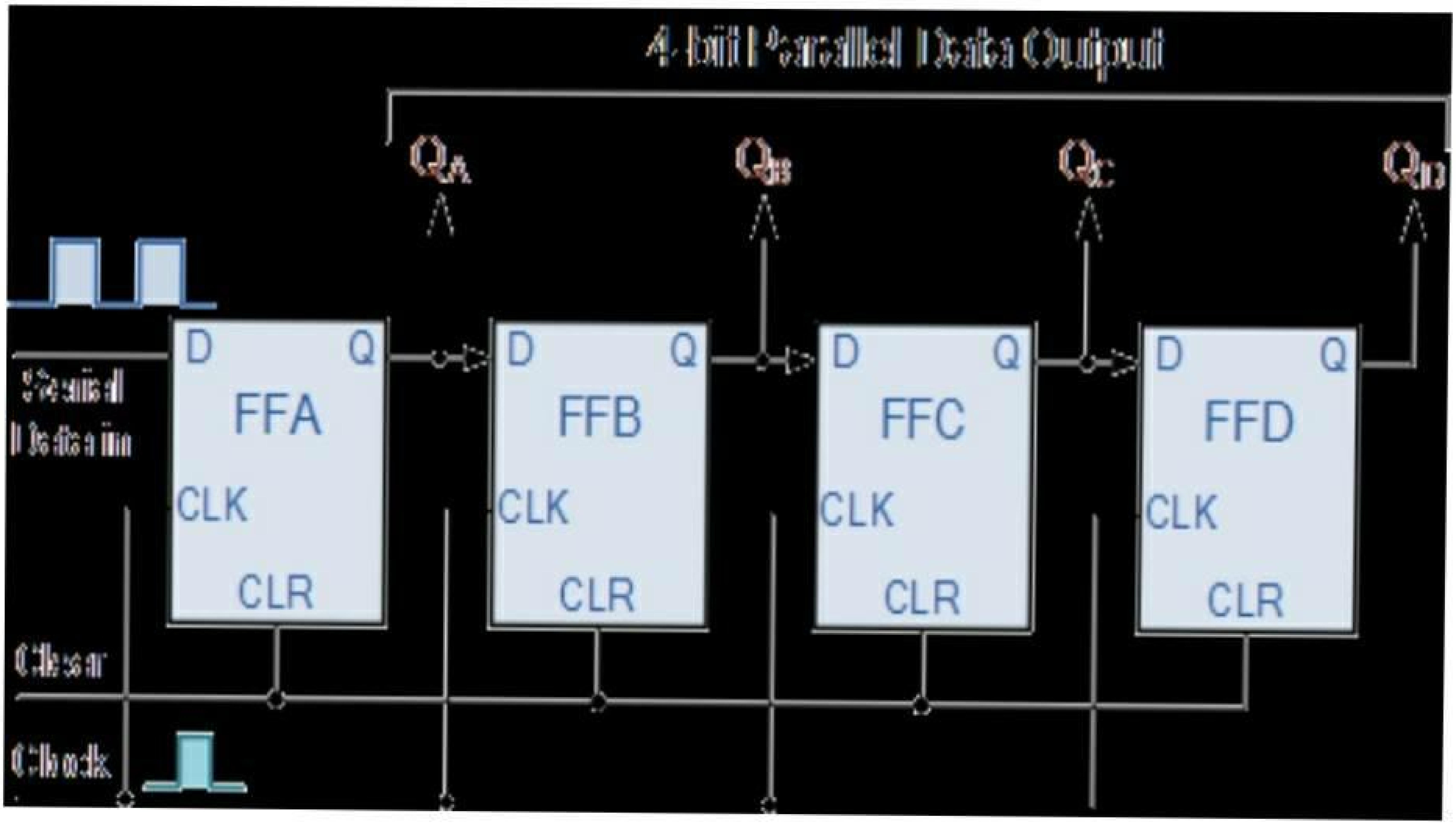
4 BIT SHIFT REGISTER

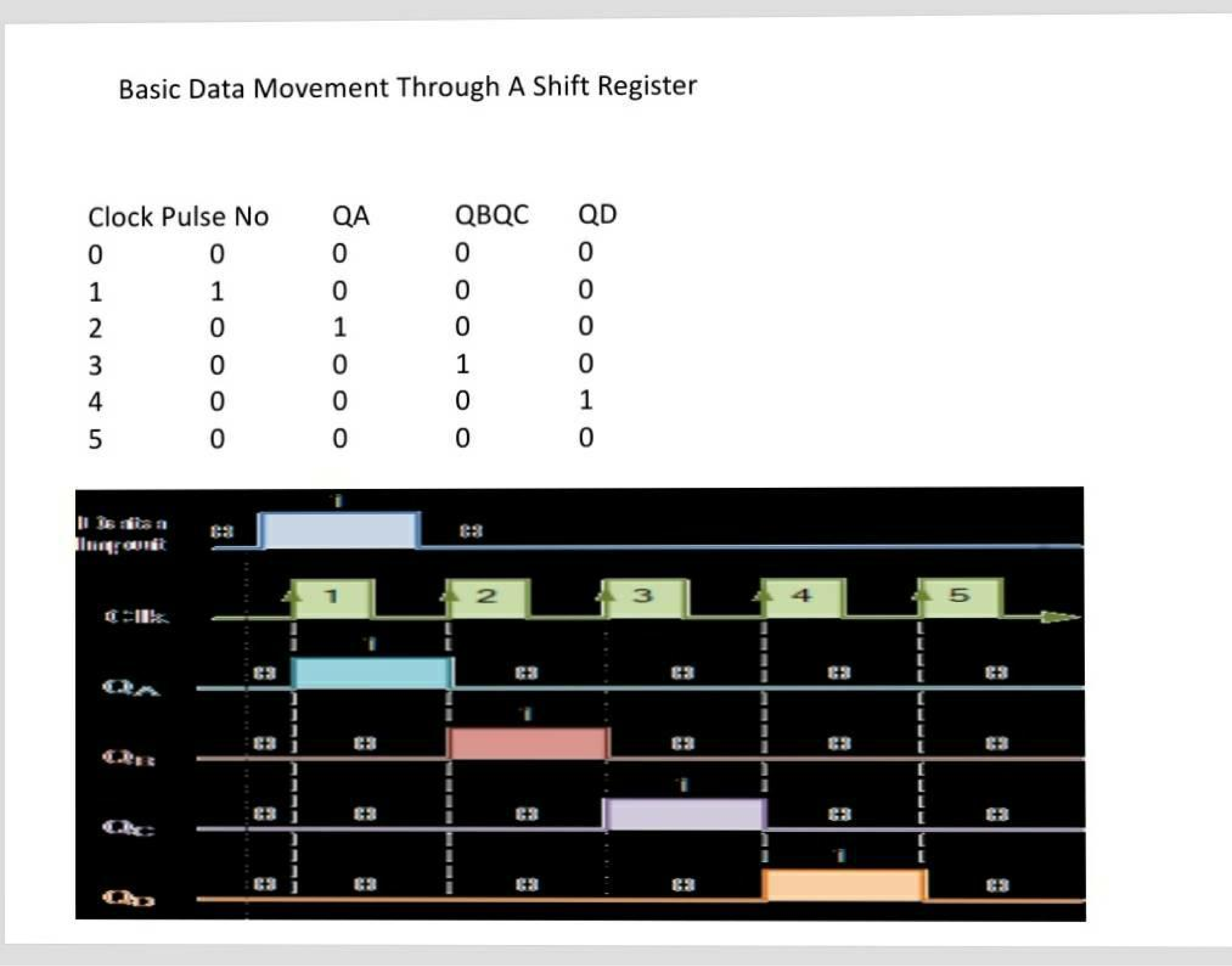
(RIGHT SHIFT)

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**.

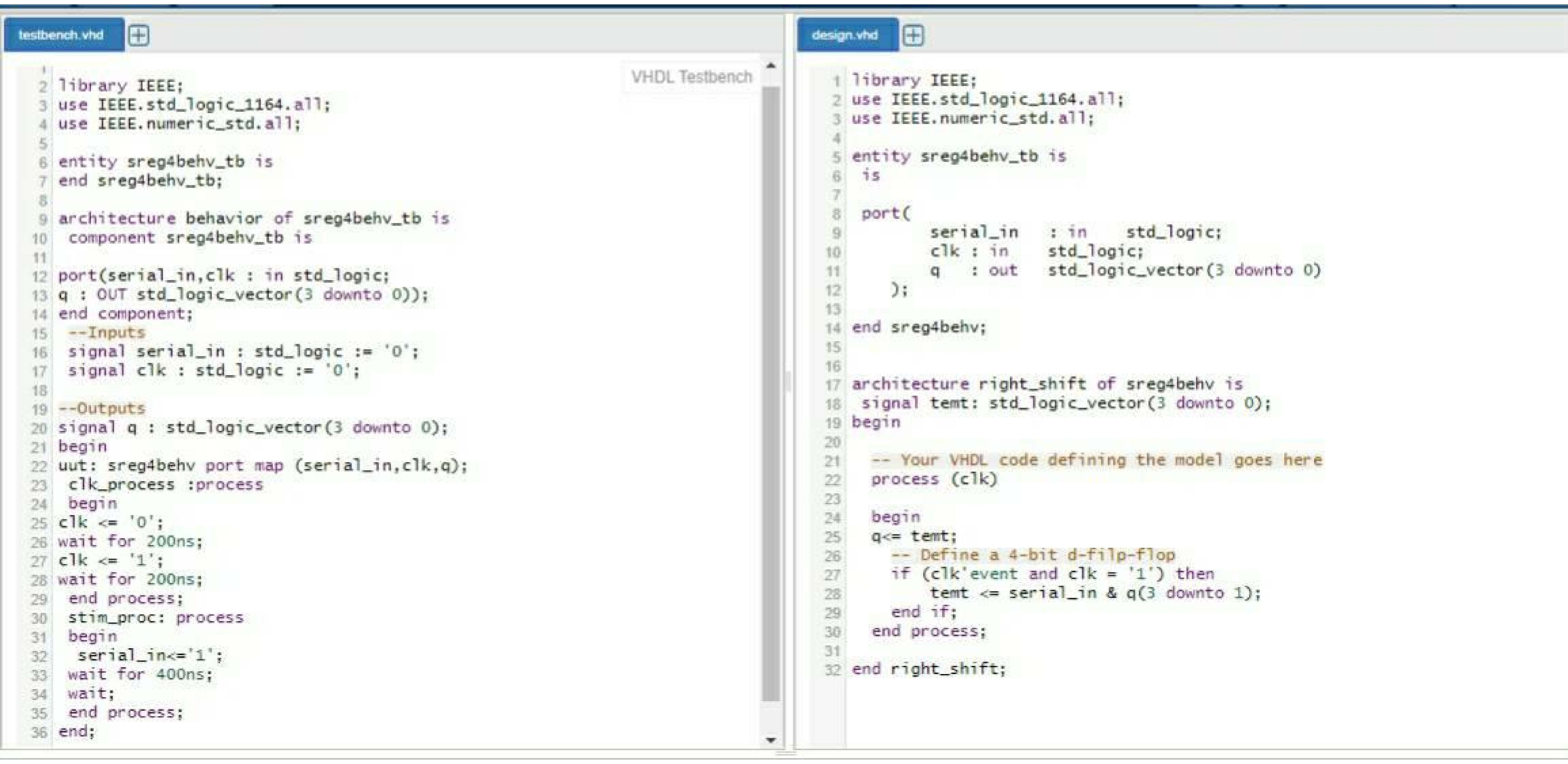
4-bit Serial-in to Parallel-out Shift Register





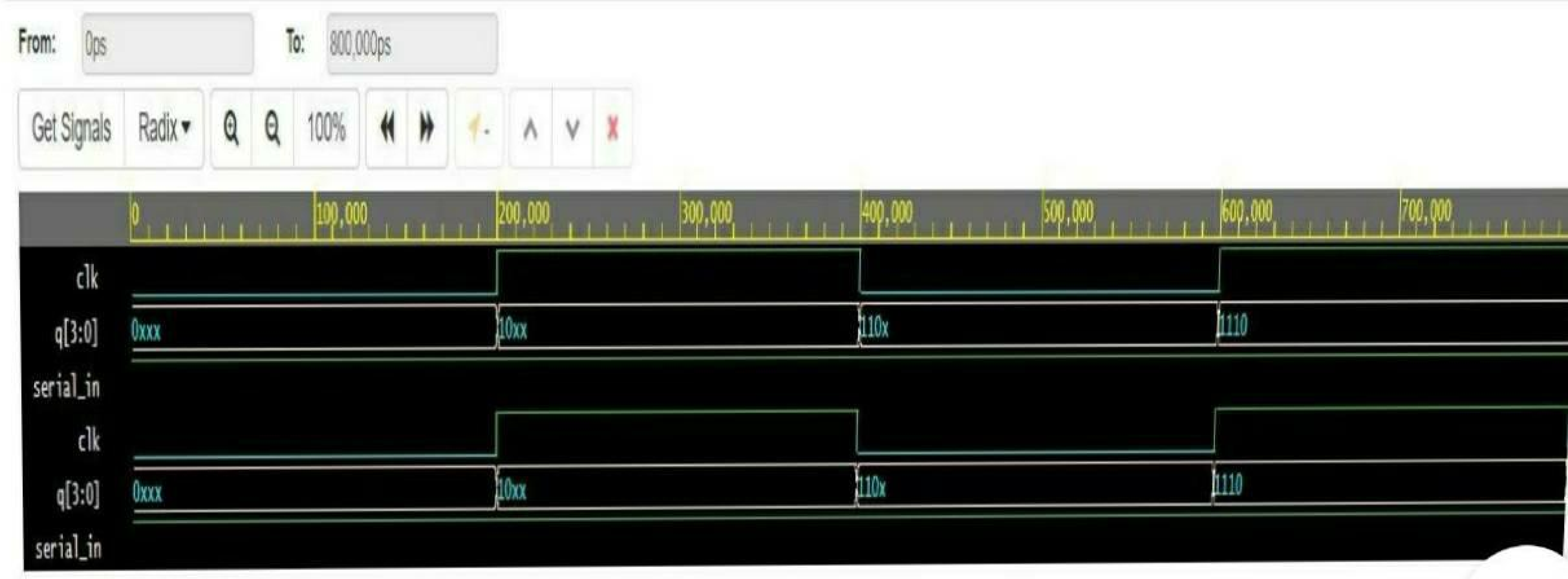
Basic Data Movement Through A Shift Register

Clock Pulse No		QA	QBQC	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



```vhdl

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

entity sreg4behv_tb is
end sreg4behv_tb;

architecture behavior of sreg4behv_tb is
 component sreg4behv_tb is

port(serial_in,clk : in std_logic;
q : OUT std_logic_vector(3 downto 0));
end component;
 --Inputs
 signal serial_in : std_logic := '0';
 signal clk : std_logic := '0';

--Outputs
signal q : std_logic_vector(3 downto 0);
begin
uut: sreg4behv port map (serial_in,clk,q);
 clk_process :process
 begin
clk <= '0';
wait for 200ns;
clk <= '1';
wait for 200ns;
 end process;
 stim_proc: process
 begin
  serial_in<='1';
 wait for 400ns;
 wait;
 end process;
end;
```

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

entity sreg4behv_tb is
 is

 port(
        serial_in   : in    std_logic;
        clk : in    std_logic;
        q   : out   std_logic_vector(3 downto 0)
    );

end sreg4behv;


architecture right_shift of sreg4behv is
 signal temt: std_logic_vector(3 downto 0);
begin

  -- Your VHDL code defining the model goes here
  process (clk)

  begin
  q<= temt;
    -- Define a 4-bit d-filp-flop
    if (clk'event and clk = '1') then
        temt <= serial_in & q(3 downto 1);
    end if;
  end process;

end right_shift;
```



From: 0ps To: 800,000ps