**VHDL Code for SR FlipFlop:-**

|  |  |
| --- | --- |
|  | library ieee;  use ieee. std\_logic\_1164.all;  use ieee. std\_logic\_arith.all;  use ieee. std\_logic\_unsigned.all;    entity SR\_FF is  PORT( S,R,CLOCK: in std\_logic;  Q, QBAR: out std\_logic);  end SR\_FF;    Architecture behavioral of SR\_FF is  begin  PROCESS(CLOCK)  variable tmp: std\_logic;  begin  if(CLOCK='1' and CLOCK'EVENT) then  if(S='0' and R='0')then  tmp:=tmp;  elsif(S='1' and R='1')then  tmp:='Z';  elsif(S='0' and R='1')then  tmp:='0';  else  tmp:='1';  end if;  end if;  Q <= tmp;  QBAR <= not tmp;  end PROCESS;  end behavioral;  C:\Users\vedan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\80ADAA98.tmp |

**VHDL Code for D FlipFlop:-**

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity D\_FF is

PORT( D,CLOCK: in std\_logic;

Q: out std\_logic);

end D\_FF;

architecture behavioral of D\_FF is

begin

process(CLOCK)

begin

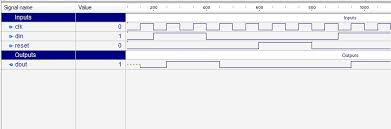
if(CLOCK='1' and CLOCK'EVENT) then

Q <= D;

end if;

end process;

end behavioral;



|  |  |  |  |
| --- | --- | --- | --- |
| **VHDL Code for JK FlipFlop:-**   |  |  | | --- | --- | |  | library ieee;  use ieee. std\_logic\_1164.all;  use ieee. std\_logic\_arith.all;  use ieee. std\_logic\_unsigned.all;    entity JK\_FF is  PORT( J,K,CLOCK: in std\_logic;  Q, QB: out std\_logic);  end JK\_FF;    Architecture behavioral of JK\_FF is  begin  PROCESS(CLOCK)  variable TMP: std\_logic;  begin  if(CLOCK='1' and CLOCK'EVENT) then  if(J='0' and K='0')then  TMP:=TMP;  elsif(J='1' and K='1')then  TMP:= not TMP;  elsif(J='0' and K='1')then  TMP:='0';  else  TMP:='1';  end if;  end if;  Q<=TMP;  Q <=not TMP;  end PROCESS;  end behavioral;  **C:\Users\vedan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\64EB9A8C.tmp** |  VHDL Code for T FlipFlop:- library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;    entity T\_FF is  port( T: in std\_logic;  Clock: in std\_logic;  Q: out std\_logic);  end T\_FF;    architecture Behavioral of T\_FF is  signal tmp: std\_logic;  begin  process (Clock)  begin  if Clock'event and Clock='1' then    if T='0' then  tmp <= tmp;  elsif T='1' then  tmp <= not (tmp);  end if;  end if;  end process;  Q <= tmp;  end Behavioral;  C:\Users\vedan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\57BF9564.tmp  **VHDL CODE FOR 4 BIT Up Down COUNTER:-**  up-down-counter  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity counter is  port(C, CLR, up\_down : in std\_logic;  Q : out std\_logic\_vector(3 downto 0));  end counter;  architecture archi of counter is  signal tmp: std\_logic\_vector(3 downto 0);  begin  process (C, CLR)  begin  if (CLR='1') then  tmp <= "0000";  elsif (C'event and C='1') then  if (up\_down='1') then  tmp <= tmp + 1;  else  tmp <= tmp - 1;  end if;  end if;  end process;  Q <= tmp;  end archi;  end process;  Q <= tmp;  end archi;  end process;  Q <= tmp;  end archi; | library ieee;  use ieee. std\_logic\_1164.all;  use ieee. std\_logic\_arith.all;  use ieee. std\_logic\_unsigned.all;    entity D\_FF is  PORT( D,CLOCK: in std\_logic;  Q: out std\_logic);  end D\_FF;    architecture behavioral of D\_FF is  begin  process(CLOCK)  begin  if(CLOCK='1' and CLOCK'EVENT) then  Q <= D;  end if;  end process;  end behavioral; |