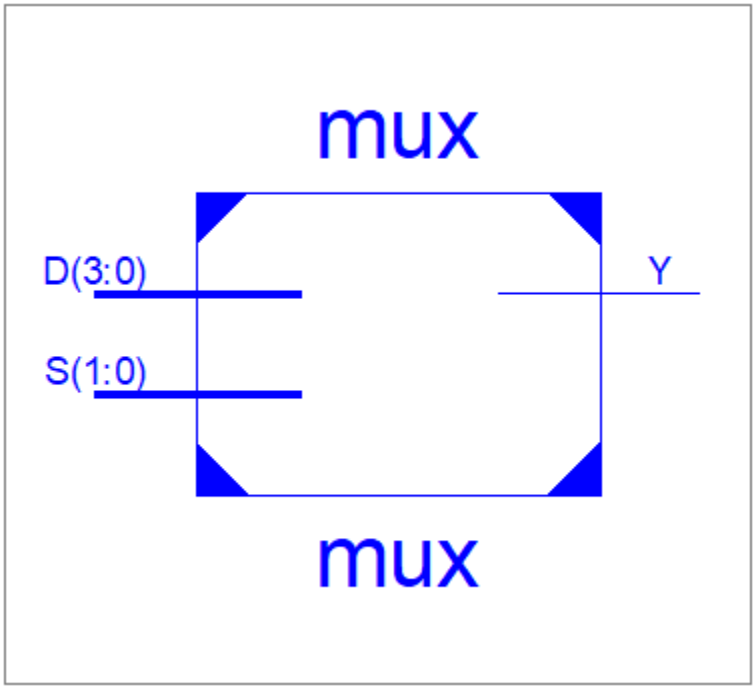
4:1 MUX

Code for MUX Module:

```
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mux is
    Port ( D : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC;
           S : in  STD_LOGIC_VECTOR (1 downto 0));
end mux;

architecture mux_arch of mux is

begin
-- S, D are passed inside the sensitivity list
process (S, D)
begin
      if (S="00") then
      Y<= D(0);
```

```
        elsif(S="01") then
        Y<=D(1);
        elsif(S="10") then
        Y<=D(2);
        else
        Y<=D(3);
        end if;

end process;

end mux_arch;
```

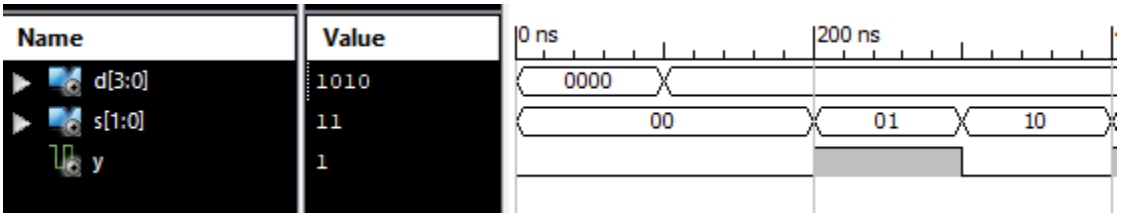
RTL Schematic:



Time Bench:

```
stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;
            D<="1010";
            S<="00";
            wait for 100 ns;
            S<="01";
            wait for 100 ns;
            S<="10";
            wait for 100 ns;
            S<="11";

--       wait for <clock>_period*10;

      -- insert stimulus here

      wait;
   end process;
```

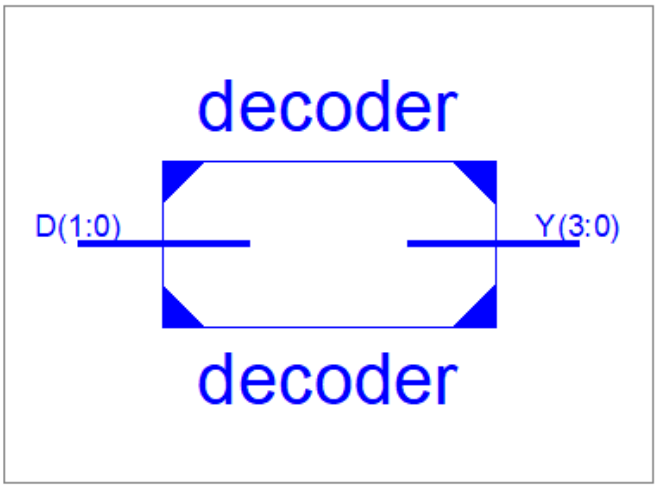
TIMING DIAGRAM:



2:4 Line Decoder Using Case:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity two_four_decoder is
    Port ( D : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end two_four_decoder;

architecture decoder_arch of two_four_decoder is

begin

process (D)
begin

case D is
when "00"=>
      Y<="0001";
when "01"=>
```

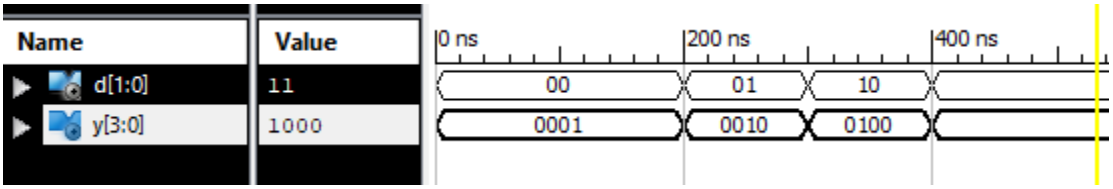
```
      Y<="0010";
when "10=>
      Y<="0100";
when others=>
      Y<="1000";
end case;
end process;

end decoder_arch;
```



Code for Decoder Test Bench

```
stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;

            D<="00";
```

```
            wait for 100 ns;
            D<="01";
            wait for 100 ns;
            D<="10";
            wait for 100 ns;
            D<="11";

--      wait for <clock>_period*10;

      -- insert stimulus here

      wait;
   end process;
```

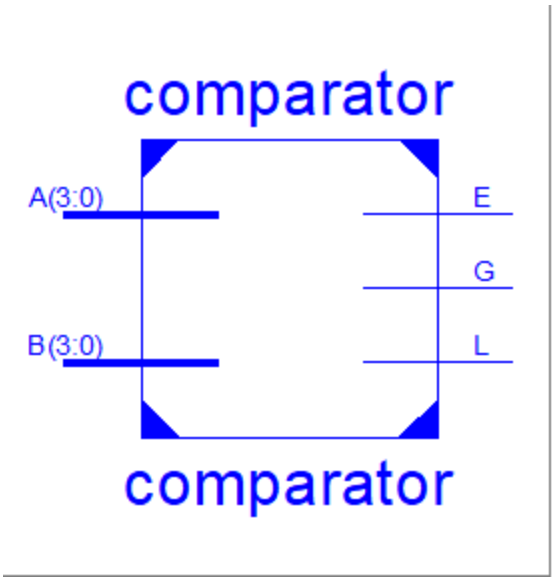


4-bit Magnitude Comparator

Code for 4-bit magnitude comparator

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity comparator is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
           L : out  STD_LOGIC;
           E : out  STD_LOGIC;
           G : out  STD_LOGIC);
end comparator;

architecture comparator_arch of comparator is

begin
```

```
process (A, B)
begin

if(A>B) then
G<='1';
L<='0';
E<='0';

elsif(A<B) then
G<='0';
L<='1';
E<='0';

else
G<='0';
L<='0';
E<='1';

end comparator_arch;
```

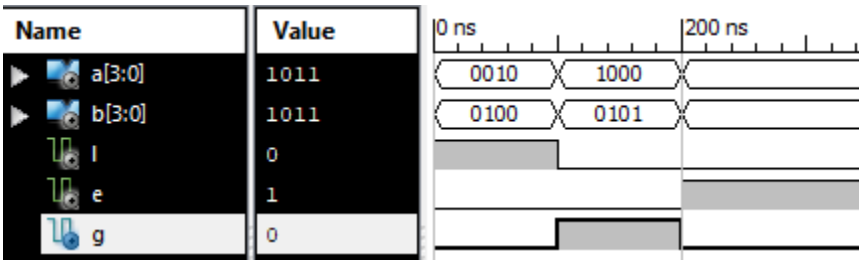


Code for Test Bench 4-bit Comparator:

```
stim_proc: process
   begin
      -- hold reset state for 100 ns.

            A<="0010";
            B<="0100";

      wait for 100 ns;
            A<="1000";
            B<="0101";

            wait for 100 ns;
```

```
            A<="1011";
            B<="1011";

--      wait for <clock>_period*10;

      -- insert stimulus here

      wait;
   end process;
```

TIMING Diagram:



JK Flip-Flop:

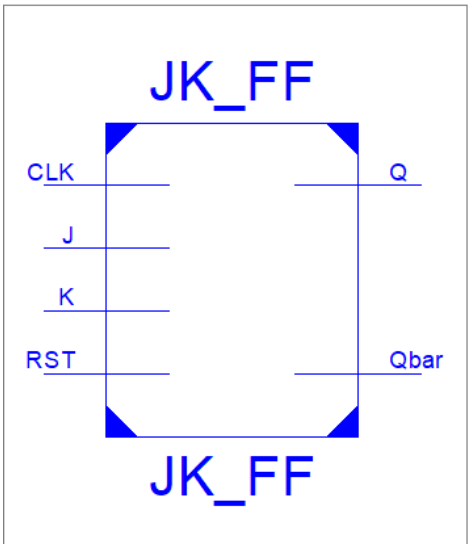
Code for JK VHDL module:

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity JK_FF is
    Port ( J : in  STD_LOGIC;
           K : in  STD_LOGIC;
           CLK : in  STD_LOGIC;
           RST : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           Qbar : out  STD_LOGIC);
end JK_FF;

architecture JK_FF_arch of JK_FF is
signal temp: STD_LOGIC :='0';
begin

process(RST, CLK, J, K)
begin
      if(RST='0') then
      temp<='0';
--clk’ event is for triggering event clock?
      elsif (CLK='1' and CLK' Event) then
```

```
            if(J='0' and K='0') then
            temp<=temp;
            elsif(J='0' and K='1') then
            temp<='0';
            elsif(J='1' and K='1') then
            temp<='1';
            else
            temp<= not temp;

            end if;

      end if;

Q<=temp;
Qbar<=not temp;


end process;

end JK_FF_arch;
```

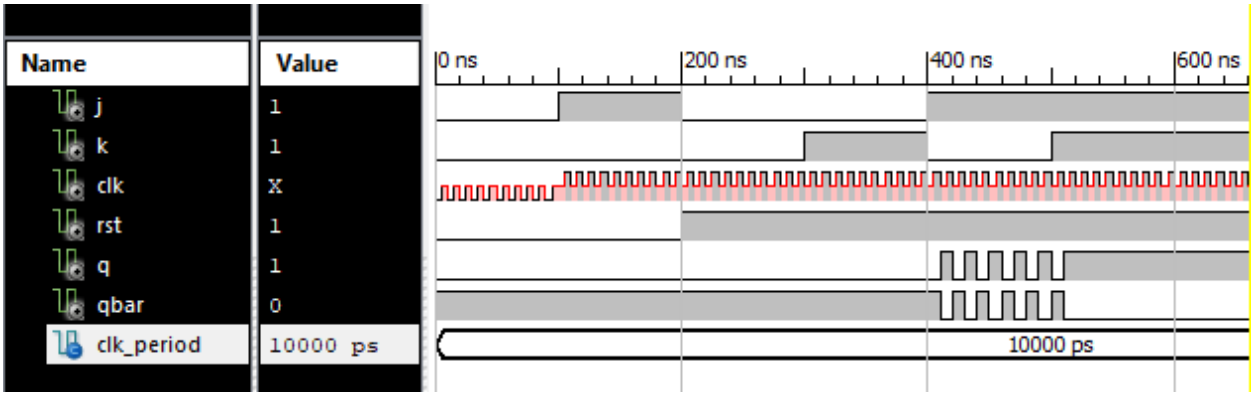
RTL Schematic:



JK Flip Flop Test Bench Code:

```
stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;


            RST<='0';
            CLK<='1';
            J<='1';
            K<='0';

            wait for 100 ns;

            RST<='1';
```

```
        CLK<='1';
        J<='0';
        K<='0';

        wait for 100 ns;

        J<='0';
        K<='1';

        wait for 100 ns;

        J<='1';
        K<='0';

        wait for 100 ns;

        J<='1';
        K<='1';


   -- insert stimulus here

   wait;
end process;
```



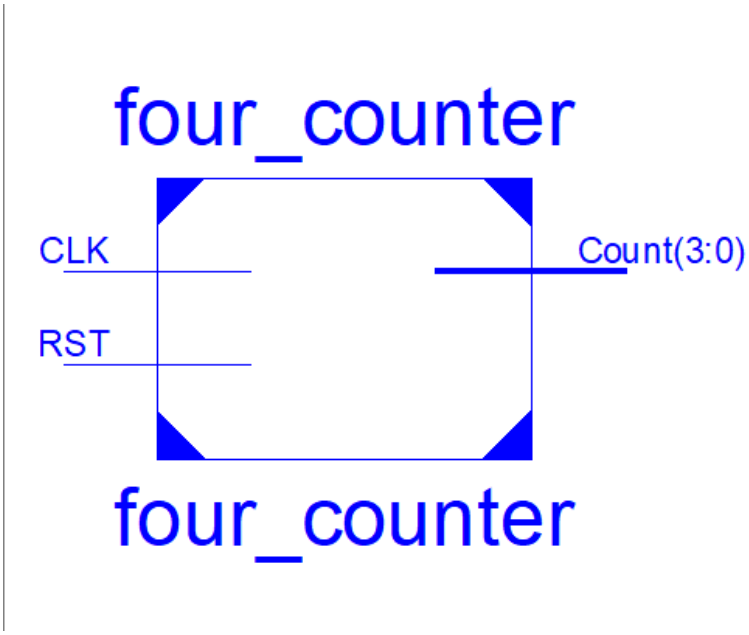
4 bit counter:

Code for Counter VHDL Module:

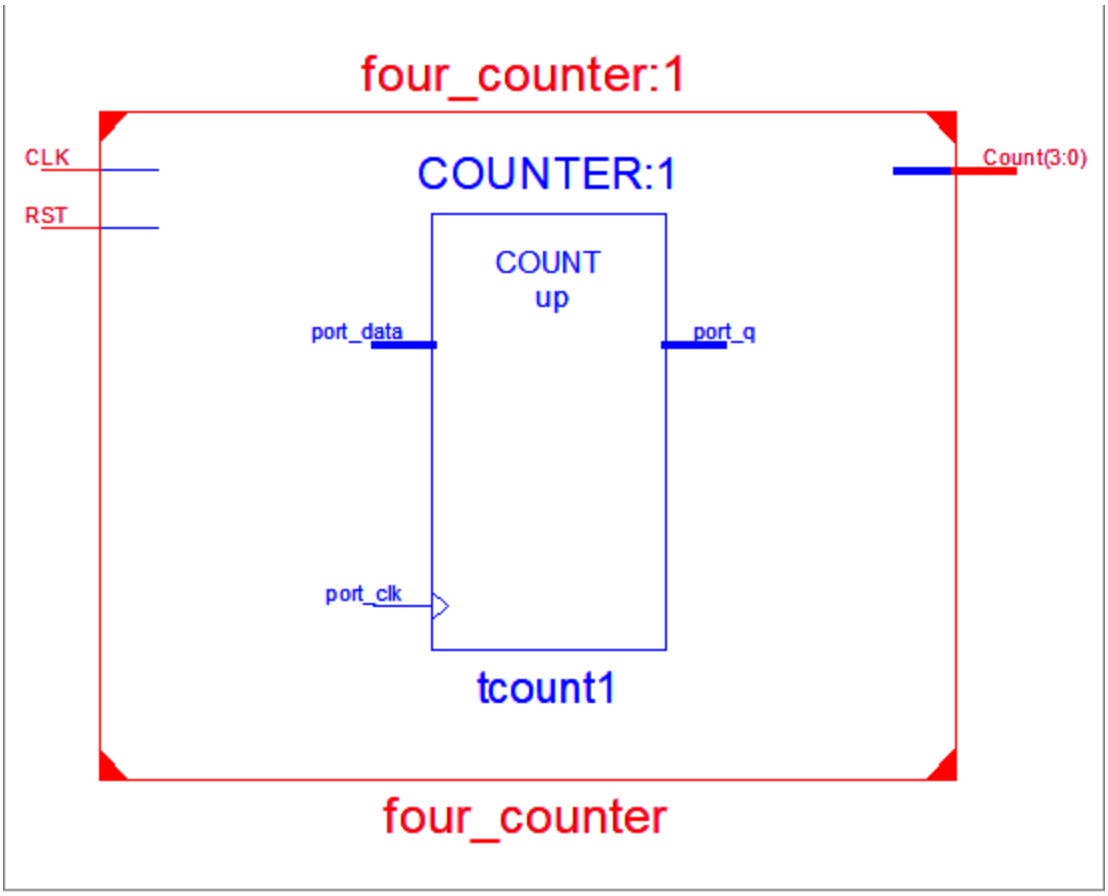
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
```

```
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity four_counter is
    Port ( CLK : in  STD_LOGIC;
           RST : in  STD_LOGIC;
           Count : out  STD_LOGIC_VECTOR (3 downto 0));
end four_counter;

architecture four_counter_arch of four_counter is
signal tcount: std_logic_vector(3 downto 0):="0000";
begin

process(CLK, RST)
begin

if(RST='0') then
      tcount<="0000";
elsif (CLK='1' and CLK' Event) then
      tcount<=tcount+1;

end if;
Count<=tcount;
end process;



end four_counter_arch;
```

RTL Schematic:



Or



Code for 4-bit Counter Test Bench

```
stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;
```

```
        RST<='0';
              CLK<='1';

              wait for 100 ns;

              RST<='1';
              CLK<='1';
        -- insert stimulus here

        wait;
    end process;
```

Timing Diagram: