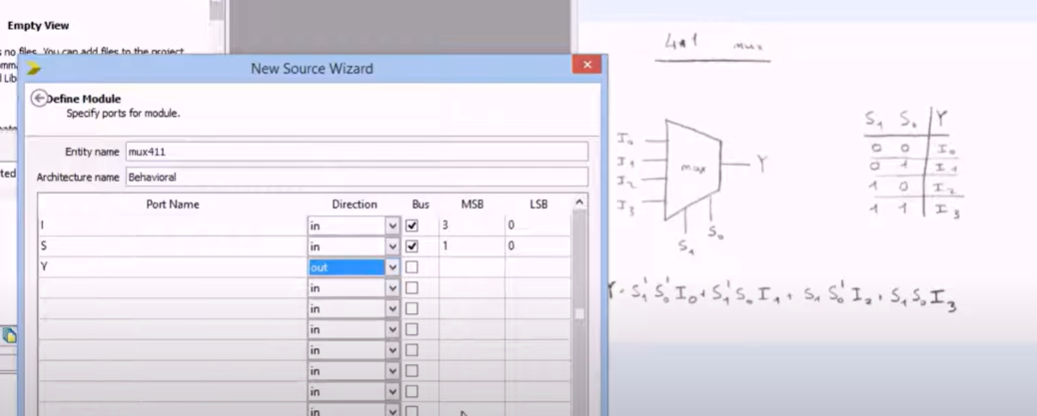
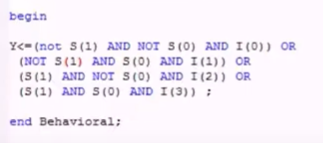
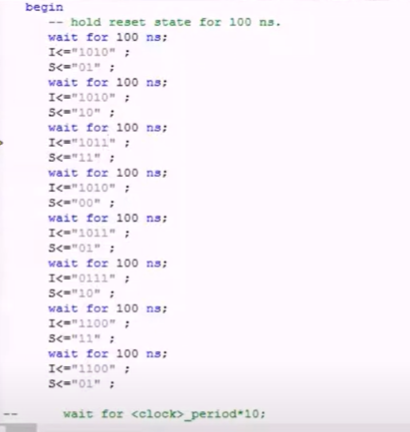
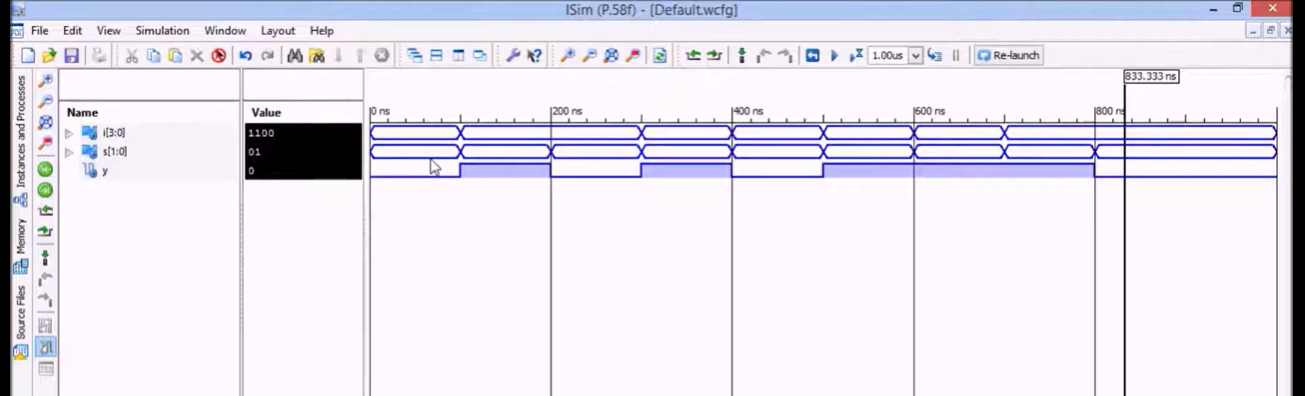
**Method 1: 4X1 mux**





**TEST BENCH**





**Method 1: 4x1 mux**

I0,I1,I2,I3 ---in

s--in MSB=1 LSB= 0 select

y---out

entity muxlab is

Port ( I0 : in STD\_LOGIC;

I1 : in STD\_LOGIC;

I2 : in STD\_LOGIC;

I3 : in STD\_LOGIC;

S : in STD\_LOGIC\_VECTOR (1 downto 0);

Y : out STD\_LOGIC);

end muxlab;

architecture Behavioral of muxlab is

begin

Y<= I0 when s="00" else

I1 when s="01" else

I2 when s="10" else

I3;

end Behavioral;

**simulation:**

I0<='1'; I1<='0'; I2<='0'; I3<='1';

s<="00"; wait for 100ns;

s<="01"; wait for 100ns;

s<="10"; wait for 100ns;

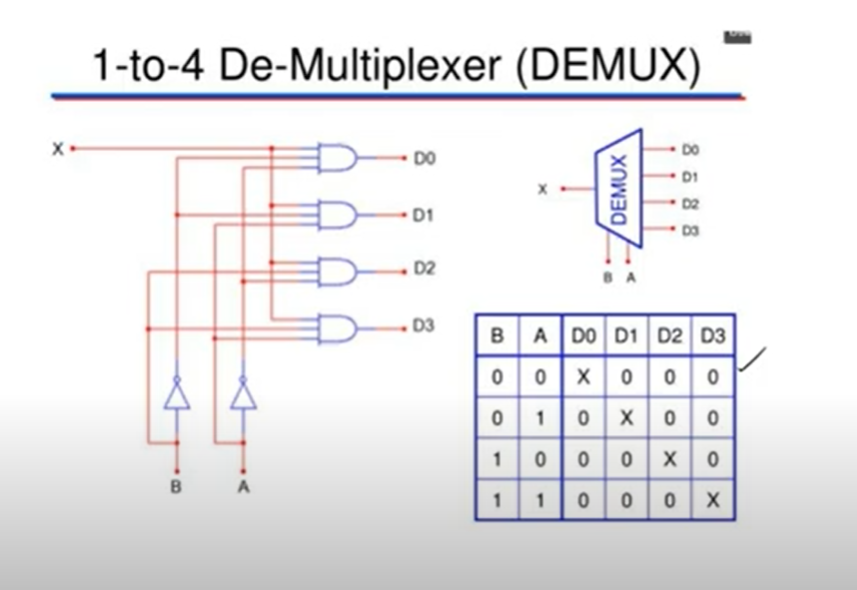
s<="11"; wait for 100ns;

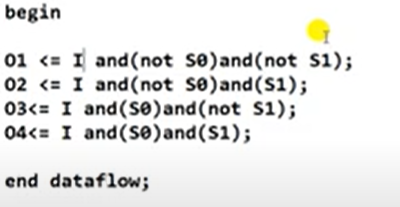
--wait;

end process;

END;

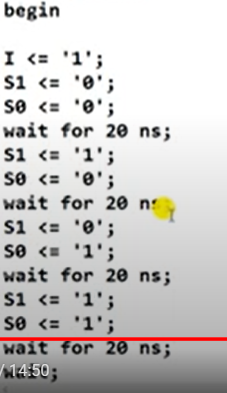
**1X4 Demux**





Here , I=X, S0=A, S1=B

O=D,



entity demuxlab is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

I : in STD\_LOGIC;

O : out STD\_LOGIC\_VECTOR (3 downto 0));

end demuxlab;

architecture Behavioral of demuxlab is

begin

O (0) <= I and ( not A) and ( not B);

O (1) <= I and ( not A) and ( B);

O (2) <= I and ( A) and ( not B);

O (3) <= I and ( A ) and ( B );

end Behavioral;

**simulation**

begin

I <= '1';

A <='0';

B <='0';

wait for 100 ns;

A <='0';

B <='1';

wait for 100 ns;

A <='1';

B <='0';

wait for 100 ns;

A <='1';

B <='1';

wait for 100 ns;

--wait;

end process;

END;