Multiplexers

VLSI Systems
Assignment-6

PREPARED BY

Priyank Lohariwal

BCSE-IV

001710501055

Jadavpur University

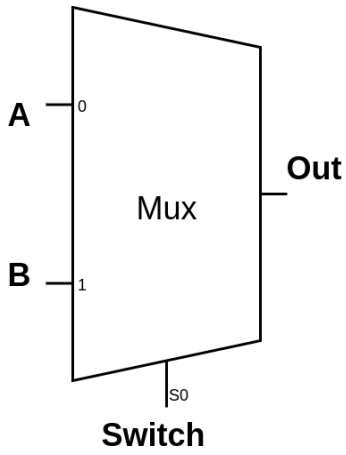
Description

Designing the following multiplexers

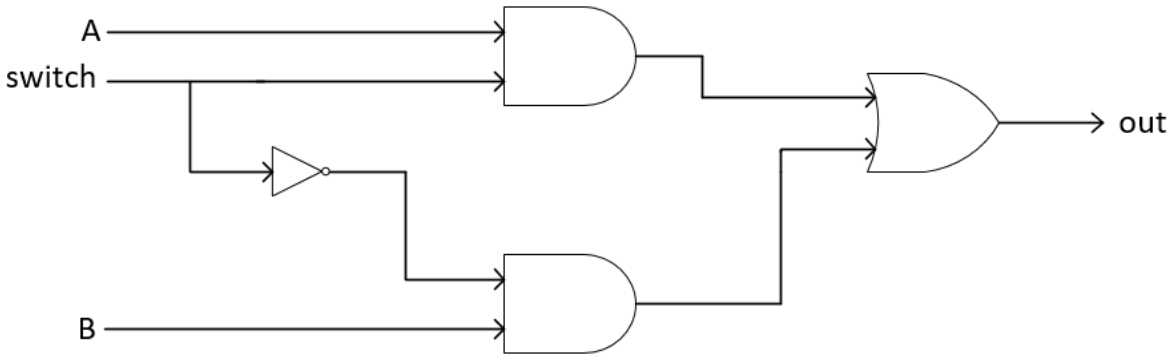
1. 2x1 mux using gate level modelling
2. 4x1 mux using gate level modelling
3. 8x1 mux using 4x1 and 2x1 multiplexers
4. 16x1 mux using 8x1 and 2x1 multiplexers
5. 16x1 mux using 4x1 multiplexers

2x1 Multiplexer

Block Diagram



Circuit Diagram



Truth Table

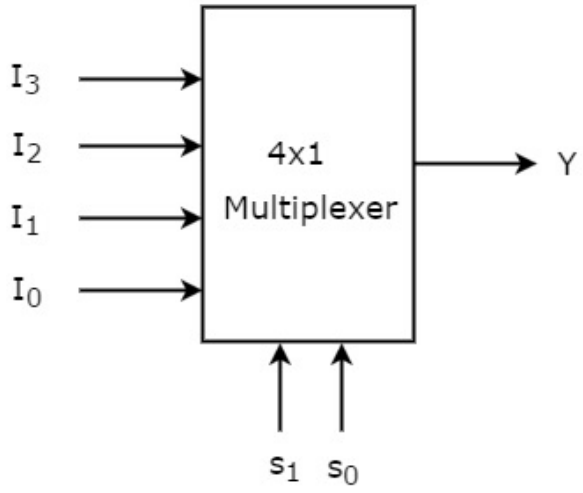
Inputs			Output
s	**A**	**B**	**Y**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Code

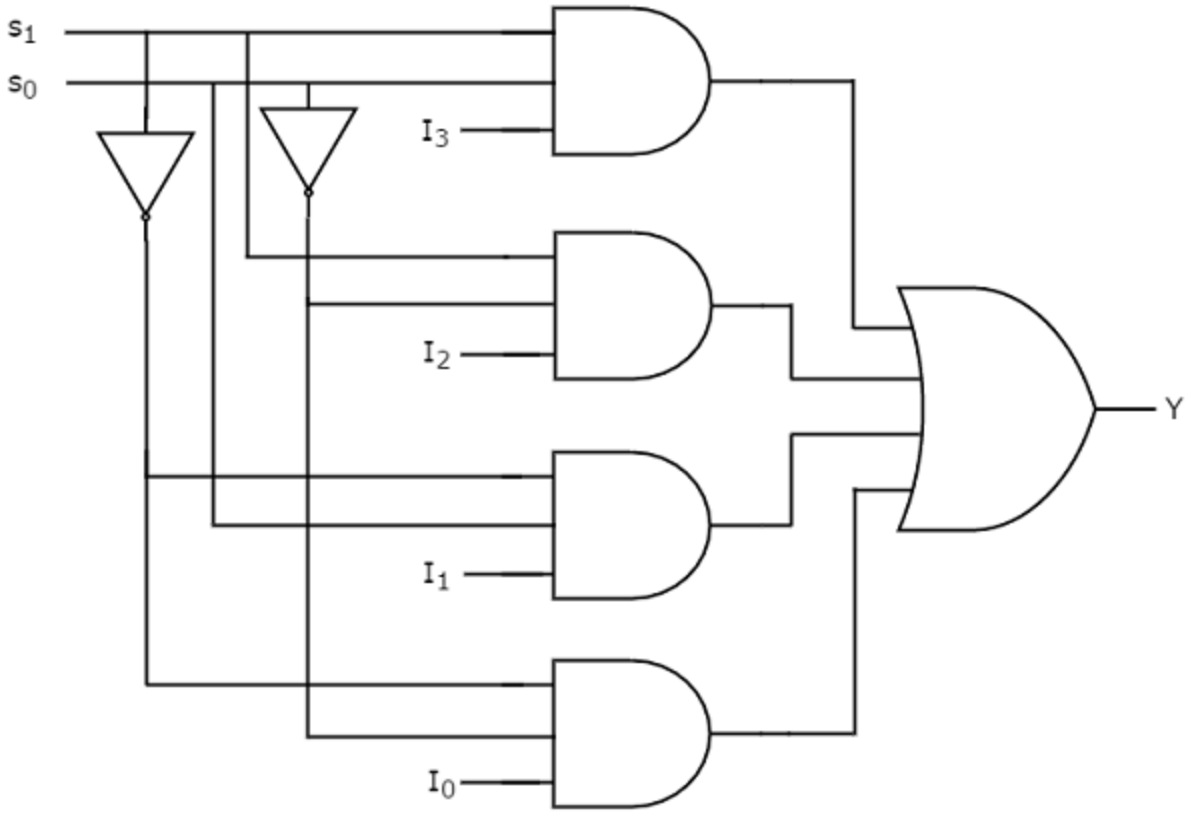
```
architecture Behavioral of mux2x1 is
   signal a,b: std_logic;
begin
   a <= (X(0) nand s);
   b <= (X(1) nand (not s));

   Y <= a nand b;
end Behavioral;
```

4x1 Multiplexer

Block Diagram



Circuit Diagram



Truth Table

Inputs						Output
S1	**S0**	**I3**	**I2**	**I1**	**I0**	**Y**
0	0	x	x	x	1	1
0	1	x	x	1	x	1
1	0	x	1	x	x	1
1	1	1	x	x	x	1

Code

Implementation

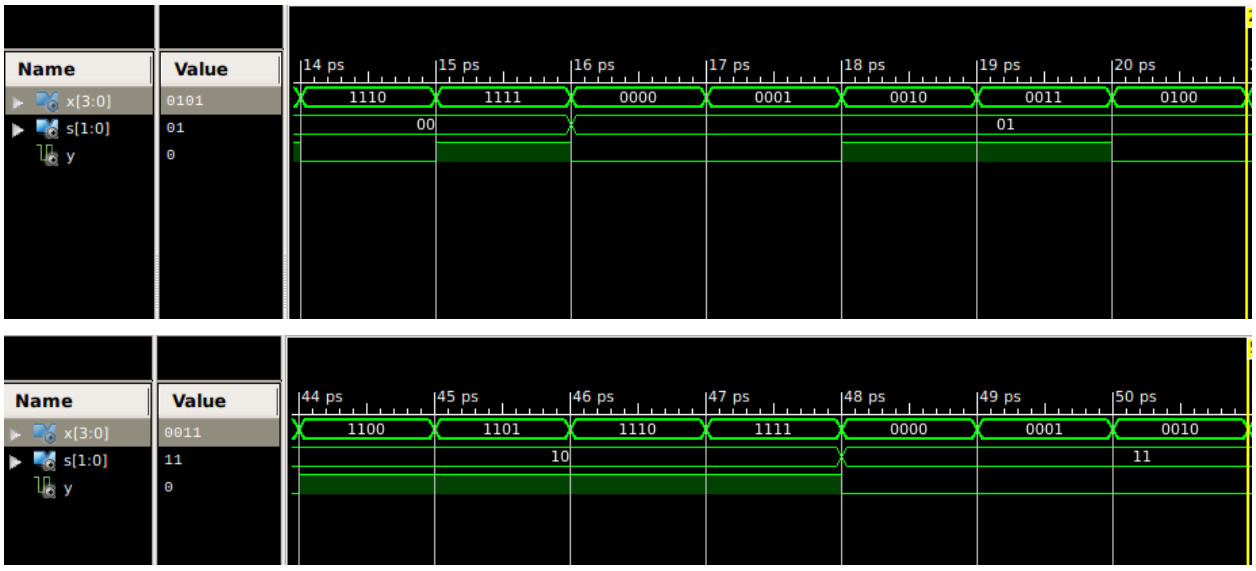
```
architecture Behavioral of mux4x1 is
   signal t1, t2, t3, t4: std_logic;
begin
   t1 <= (X(0) and (not s(0)) and (not s(1)));
   t2 <= (X(1) and s(0) and (not s(1)));
   t3 <= (X(2) and (not s(0)) and s(1));
   t4 <= (X(3) and s(0) and s(1));

   y <= (t1 or t2 or t3 or t4);
end Behavioral;
```

Test Bench

```
ARCHITECTURE behavior OF mux4x1_test_bench IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT mux4x1
   PORT(
        X : IN  std_logic_vector(3 downto 0);
        s : IN  std_logic_vector(1 downto 0);
        Y : OUT  std_logic
       );
   END COMPONENT;


  --Inputs
  signal X : std_logic_vector(3 downto 0) := (others => '0');
  signal s : std_logic_vector(1 downto 0) := (others => '0');

  --Outputs
  signal Y : std_logic;
```

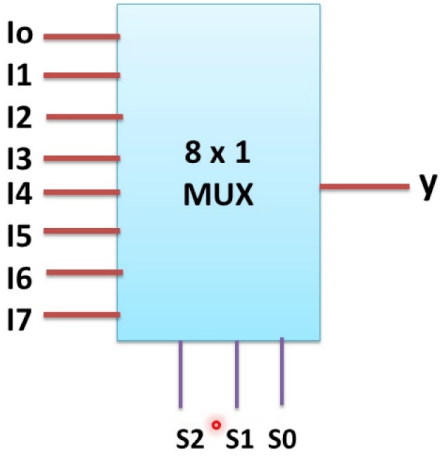
```
BEGIN
  -- Instantiate the Unit Under Test (UUT)
  uut: mux4x1 PORT MAP (
         X => X,
         s => s,
         Y => Y
       );

  stim_proc: process
     variable s_bin: std_logic_vector(1 downto 0);
     variable binary: std_logic_vector(3 downto 0);
  begin
     for i in 0 to 3 loop
        proc_s: dec_to_bin_proc(i, 2, s_bin);
        s <= s_bin;
        for k in 0 to 15 loop
           proc: dec_to_bin_proc(k, 4, binary);
           X <= binary;
           wait for 1ps;
        end loop;
     end loop;
  end process;
END;
```

Timing Diagram

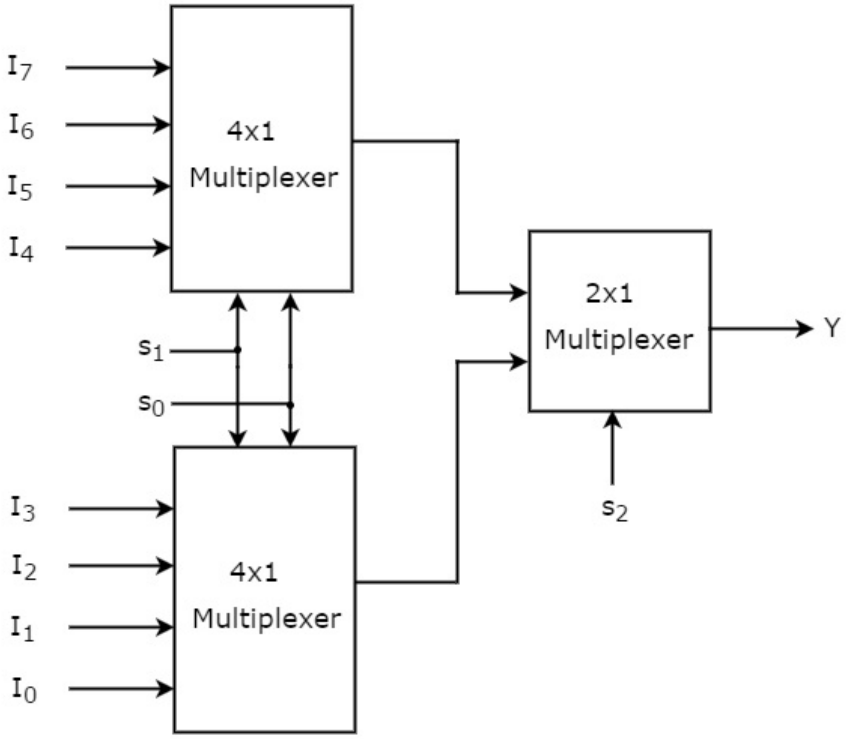


8x1 Multiplexer

Block Diagram



Circuit Diagram



Truth Table

Inputs			Output
S2	**S1**	**S0**	**Y**
0	0	0	**I0**
0	0	1	**I1**
0	1	0	**I2**
0	1	1	**I3**
1	0	0	**I4**
1	0	1	**I5**
1	1	0	**I6**
1	1	1	**I7**

Code

Implementation

```
architecture Behavioral of mux8x1 is

COMPONENT mux4x1
   PORT(
        X : IN  std_logic_vector(3 downto 0);
        s : IN  std_logic_vector(1 downto 0);
           Y : OUT  std_logic
       );
   END COMPONENT;

COMPONENT mux2x1
   PORT(
        X : IN  std_logic_vector(1 downto 0);
        s : IN  std_logic;
           Y : OUT  std_logic
       );
   END COMPONENT;
   signal t: std_logic_vector(1 downto 0);
begin
      c1: mux4x1 port map(X(7 downto 4), s(1 downto 0), t(0));
      c2: mux4x1 port map(X(3 downto 0), s(1 downto 0), t(1));
      C3: mux2x1 port map(t, s(2), Y);

end Behavioral;
```

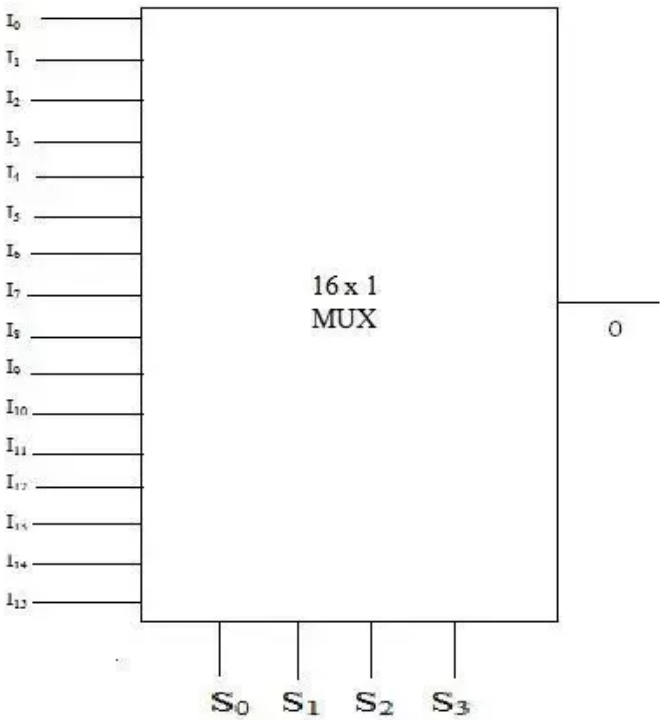
Test Bench

```
ARCHITECTURE behavior OF mux8x1_test_bench IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT mux8x1
   PORT(
        X : IN  std_logic_vector(7 downto 0);
        s : IN  std_logic_vector(2 downto 0);
        Y : OUT  std_logic
       );
   END COMPONENT;


  --Inputs
  signal X : std_logic_vector(7 downto 0) := (others => '0');
  signal s : std_logic_vector(2 downto 0) := (others => '0');
  --Outputs
  signal Y : std_logic;
BEGIN
  -- Instantiate the Unit Under Test (UUT)
  uut: mux8x1 PORT MAP (
         X => X,
         s => s,
         Y => Y
       );
  -- Stimulus process
  stim_proc: process
     variable s_bin: std_logic_vector(2 downto 0);
     variable binary: std_logic_vector(7 downto 0);
  begin
     for i in 0 to 7 loop
        proc_s: dec_to_bin_proc(i, 3, s_bin);
        s <= s_bin;
        for k in 0 to 127 loop
           proc: dec_to_bin_proc(k, 8, binary);
           X <= binary;
           wait for 1ps;
        end loop;
     end loop;
  end process;
END;
```

Timing Diagram

16x1 Multiplexer

Block Diagram

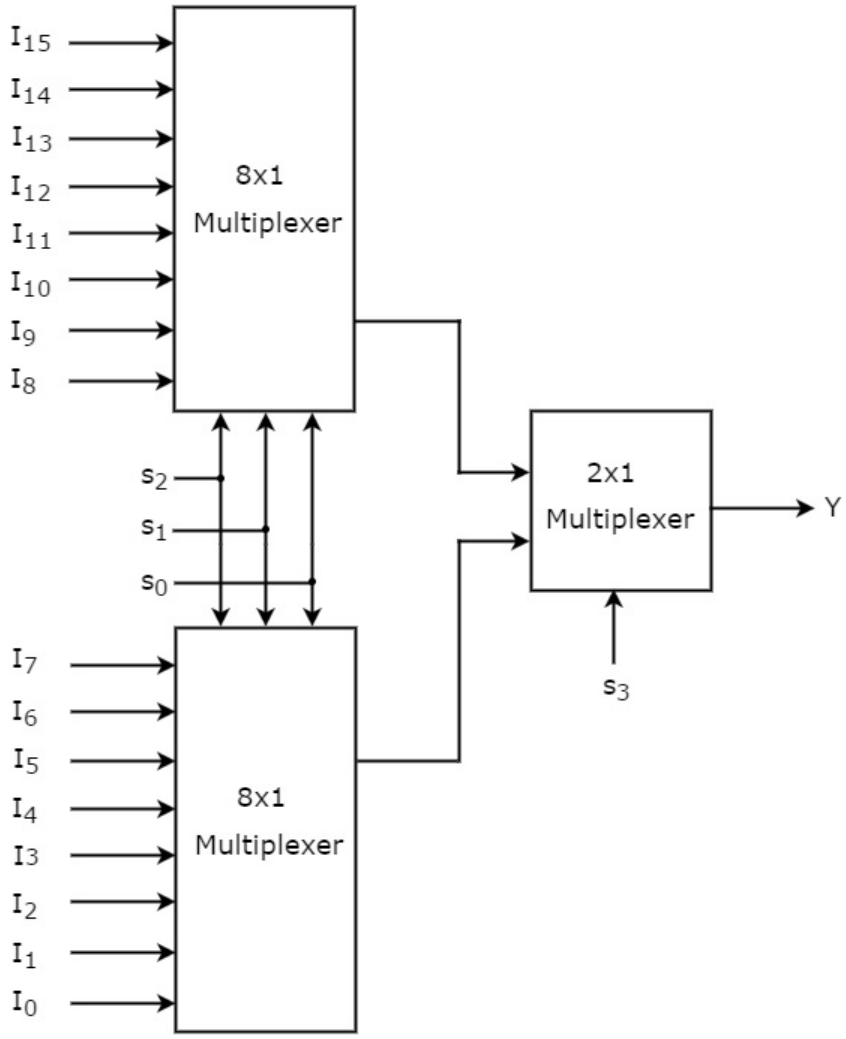


Truth Table

Inputs				Output
S3	**S2**	**S1**	**S0**	**Y**
0	0	0	0	**I0**
0	0	0	1	**I1**
0	0	1	0	**I2**
0	0	1	1	**I3**
0	1	0	0	**I4**
0	1	0	1	**I5**
0	1	1	0	**I6**
0	1	1	1	**I7**
1	0	0	0	**I8**
1	0	0	1	**I9**
1	0	1	0	**I10**
1	0	1	1	**I11**
1	1	0	0	**I12**
1	1	0	1	**I13**
1	1	1	0	**I14**
1	1	1	1	**I15**

Using 8x1 and 2x1 Multiplexers

Circuit Diagram

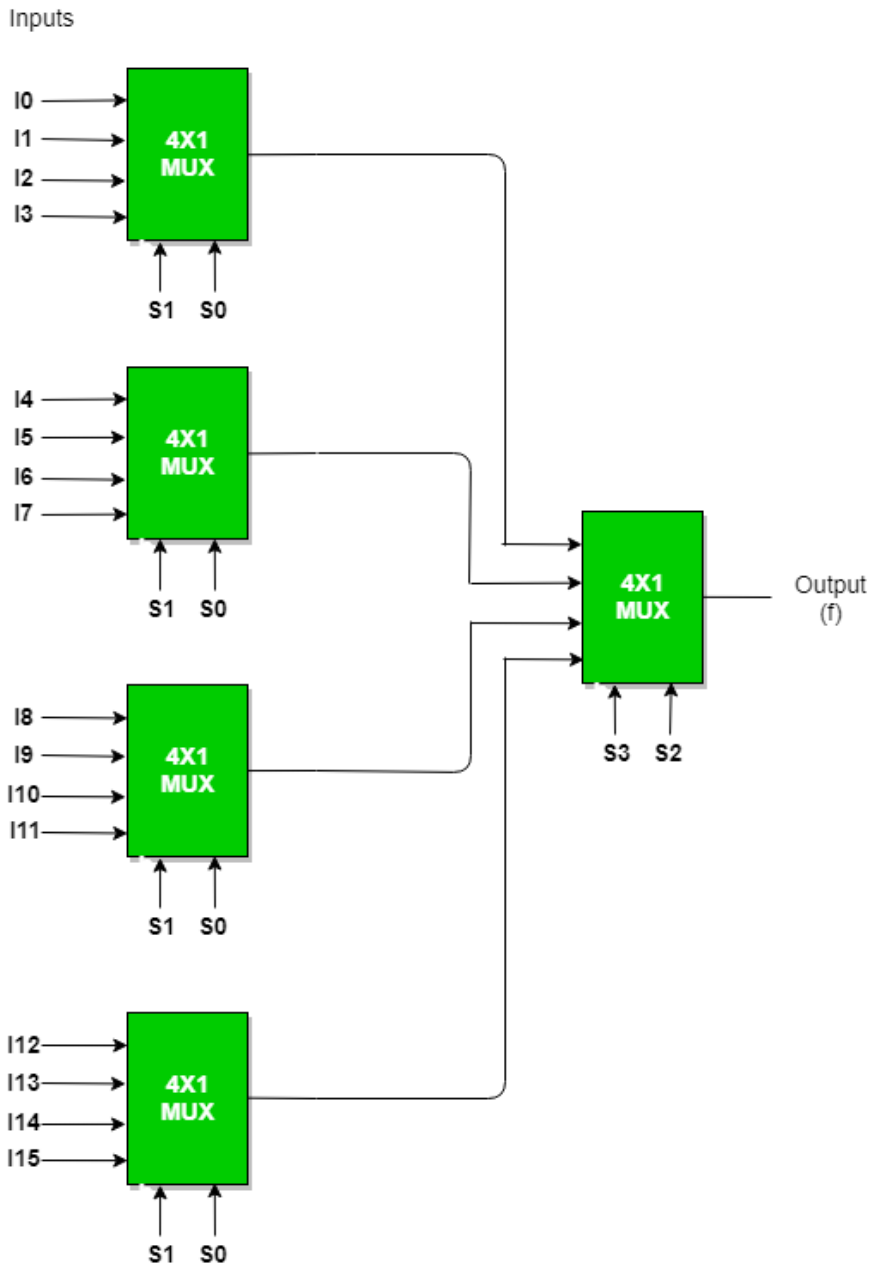


Code

```
entity mux16x1 is
   Port ( X : in  STD_LOGIC_VECTOR (15 downto 0);
          s : in  STD_LOGIC_VECTOR (3 downto 0);
          Y : out  STD_LOGIC);
end mux16x1;

architecture Behavioral of mux16x1 is
COMPONENT mux8x1
   PORT(
        X : IN  std_logic_vector(7 downto 0);
        s : IN  std_logic_vector(2 downto 0);
           Y : OUT  std_logic
       );
   END COMPONENT;

COMPONENT mux2x1
   PORT(
        X : IN  std_logic_vector(1 downto 0);
        s : IN  std_logic;
           Y : OUT  std_logic
       );
   END COMPONENT;

   signal t: std_logic_vector(1 downto 0);
begin

c1: mux8x1 port map(X(15 downto 8), s(2 downto 0), t(0));
c2: mux8x1 port map(X(7 downto 0), s(2 downto 0), t(1));
c3: mux2x1 port map(t, s(3), Y);

end Behavioral;
```

Using 4x1 Multiplexers

Circuit Diagram



Code

```
architecture Behavioral of mux16x1b is
COMPONENT mux4x1
   PORT(
        X : IN  std_logic_vector(3 downto 0);
        s : IN  std_logic_vector(1 downto 0);
           Y : OUT  std_logic
       );
   END COMPONENT;

   signal t: std_logic_vector(3 downto 0);
begin
   c1: mux4x1 port map(X(15 downto 12), s(1 downto 0), t(3));
   c2: mux4x1 port map(X(11 downto 8), s(1 downto 0), t(2));
   c3: mux4x1 port map(X(7 downto 4), s(1 downto 0), t(1));
   c4: mux4x1 port map(X(3 downto 0), s(1 downto 0), t(0));
   c5: mux4x1 port map(t, s(3 downto 2), Y);
end Behavioral;
```

Test Bench

```
 ARCHITECTURE behavior OF mux16x1_test_bench IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT mux16x1b
   PORT(
        X : IN  std_logic_vector(15 downto 0);
        s : IN  std_logic_vector(3 downto 0);
        Y : OUT  std_logic
       );
   END COMPONENT;
  --Inputs
  signal X : std_logic_vector(15 downto 0) := (others => '0');
  signal s : std_logic_vector(3 downto 0) := (others => '0');
 --Outputs
  signal Y : std_logic;
BEGIN
 -- Instantiate the Unit Under Test (UUT)
  uut: mux16x1b PORT MAP (
         X => X,
         s => s,
         Y => Y
       );
```

```
    -- Stimulus process
    stim_proc: process
     variable s_bin: std_logic_vector(3 downto 0);
     variable binary: std_logic_vector(15 downto 0);
    begin
     for i in 0 to 15 loop
       proc_s: dec_to_bin_proc(i, 4, s_bin);
       s <= s_bin;
       for k in 0 to 256 loop
         proc: dec_to_bin_proc(k, 16, binary);
         X <= binary;
         wait for 1ps;
       end loop;
     end loop;

    end process;

END;
```

Timing Diagram