VLSI
Assignment 3

Priyank Lohariwal
BCSE IV
Roll-001710501055

1 Describtion

- Design a 1x2 decoder
  - using gate level modelling
  - using behavioral modelling
    - using if else statement
    - using case statement
    - using when else statement
    - using select when statement
- Design a 2x4 decoder
  - using gate level modelling
  - using behavioral modelling
    - using if else statement
    - using case statement
    - using when else statement
    - using select when statement
- Design a 3x8 decoder
  - using gate level modelling
  - using behavioral modelling
    - using if else statement
    - using case statement
    - using when else statement
    - using select when statement
- Design a 3x8 decoder using 2x4 decoder and 1x2 decoder by component instantiation
- Design a 4x16 decoder using 2x4 decoder only by component instantiation

2 Block Diagram

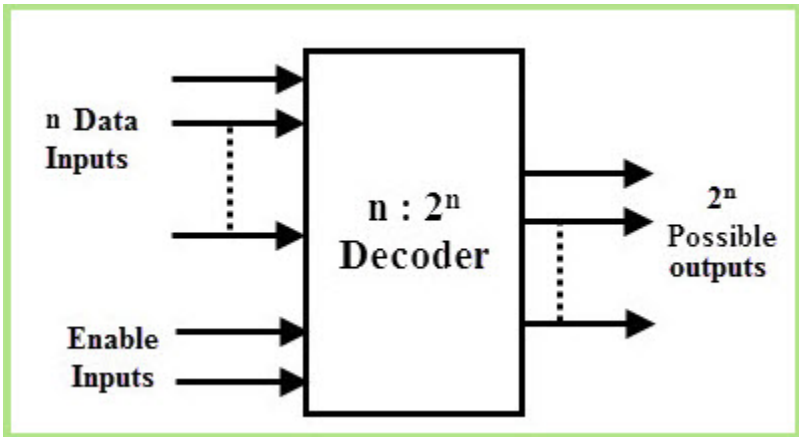


Figure 1: Decoder block diagram

3 1x2 Decoder

3.1 Truth Table

X	Y(1)	Y(0)
0	0	1
1	1	0

3.2 Circuit Diagram

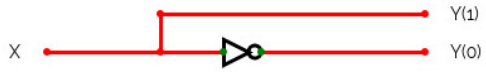


Figure 2: 1x2 circuit diagram

3.3 Code

3.3.1 Using gate level modelling

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

entity a3_1a is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end a3_1a;

architecture Behavioral of a3_1a is

begin
        Y(1) <= X;
        Y(0) <= not X;
end Behavioral;
```

3.3.2 Using if else statement

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

entity a3_1ba is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end a3_1ba;

architecture Behavioral of a3_1ba is

begin
        p1 : process(X)
        begin
                if X = '0' then
                        Y <= "01";
                elsif X = '1' then
                        Y <= "10";
                end if;
        end process;

end Behavioral;
```

3.3.3 Using case statement

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

```
entity a3_1bb is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end a3_1bb;

architecture Behavioral of a3_1bb is

begin

        p1 : process(X)
        begin
                case X is
                        when '1' => Y <= "01";
                        when '0' => Y <= "10";
                        when others => Y <= "ZZ";
                end case;
        end process;


end Behavioral;
```

3.3.4 Using when else statement

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

entity a3_1bc is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end a3_1bc;

architecture Behavioral of a3_1bc is

begin

        Y <=          "01" when X = '0' else
                      "10" when X = '1' else
                      "ZZ";

end Behavioral;
```

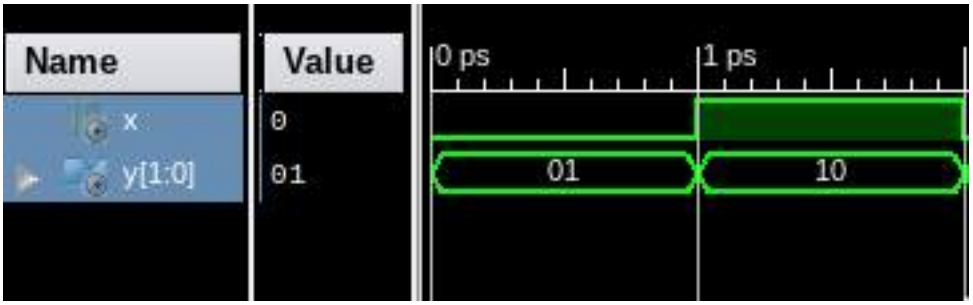
3.3.5 Using select when statement

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

entity a3_1bd is
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end a3_1bd;

architecture Behavioral of a3_1bd is

begin

        with X select Y <=
                "01" when '0',
                "10" when '1',
                "ZZ" when others;

end Behavioral;
```

3.4 Test Bench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_a3_1a IS
END tb_a3_1a;

ARCHITECTURE behavior OF tb_a3_1a IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT a3_1a
    PORT(
         X : IN  std_logic;
         Y : OUT  std_logic_vector(1 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal X : std_logic := '0';

         --Outputs
   signal Y : std_logic_vector(1 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: a3_1a PORT MAP (
          X => X,
          Y => Y
        );

   -- Stimulus process
   stim_proc: process
   begin
                X <= '0';
                wait for 1 ps;
                X <= '1';
                wait for 1 ps;
   end process;

END;
```

3.5 Timing diagram



4 2x4 Decoder

4.1 Truth Table

X(1)	X(0)	Y(3)	Y(2)	Y(1)	Y(0)
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

4.2 Circuit Diagram

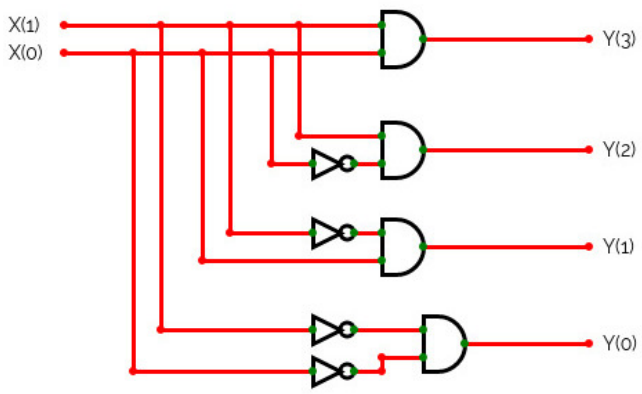


Figure 3: 2x4 circuit diagram

4.3 Code

4.3.1 Using gate level modelling

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

entity a3_2_1a is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
                           E : in STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end a3_2_1a;

architecture Behavioral of a3_2_1a is

begin

        Y(3) <= E and (X(1) and X(0));
        Y(2) <= E and (X(1) and not X(0));
        Y(1) <= E and (not X(1) and X(0));
        Y(0) <= E and (not X(1) and not X(0));

end Behavioral;
```

4.3.2 Using if else statement

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

```
entity a3_2_1ba is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end a3_2_1ba;

architecture Behavioral of a3_2_1ba is

begin

        p1 : process(X)
        begin
                if X = "00" then Y <= "0001";
                elsif X = "01" then Y <= "0010";
                elsif X = "10" then Y <= "0100";
                elsif X = "11" then Y <= "1000";
                end if;
        end process;

end Behavioral;
```

4.3.3 Using case statement

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

entity a3_2_1bb is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end a3_2_1bb;

architecture Behavioral of a3_2_1bb is

begin
        p1: process(X)
        begin
        case X is
                when "00" => Y <= "0001";
                when "01" => Y <= "0010";
                when "10" => Y <= "0100";
                when "11" => Y <= "1000";
                when others => Y <= "ZZZZ";
        end case;
        end process;

end Behavioral;
```

4.3.4 Using when else statement

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

entity a3_2_1bc is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end a3_2_1bc;

architecture Behavioral of a3_2_1bc is

begin

        Y <=         "0001" when X = "00" else
```

```
                        "0010" when X = "01" else
                        "0100" when X = "10" else
                        "1000" when X = "11" else
                        "ZZZZ";


end Behavioral;
```

4.3.5 Using select when statement

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

entity a3_2_1bd is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end a3_2_1bd;

architecture Behavioral of a3_2_1bd is

begin

        with X select Y <=
                "0001" when "00",
                "0010" when "01",
                "0100" when "10",
                "1000" when "11",
                "ZZZZ" when others;

end Behavioral;
```

4.4 Test Bench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_a3_2_1a IS
END tb_a3_2_1a;

ARCHITECTURE behavior OF tb_a3_2_1a IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT a3_2_1a
    PORT(
         X : IN  std_logic_vector(1 downto 0);
                        E : in STD_LOGIC;
         Y : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;


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

         --Outputs
   signal Y : std_logic_vector(3 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name
```

```
BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: a3_2_1a PORT MAP (
          X => X,
                          E => E,
          Y => Y
        );


   -- Stimulus process
   stim_proc: process
   begin
                E <= '1';

                X <= "00";
                wait for 1 ps;
                X <= "01";
                wait for 1 ps;
                X <= "10";
                wait for 1 ps;
                X <= "11";
                wait for 1 ps;
   end process;

END;
```

4.5 Timing diagram

5 3x8 Decoder

5.1 Truth Table

X(2)	X(1)	X(0)	Y(7)	Y(6)	Y(5)	Y(4)	Y(3)	Y(2)	Y(1)	Y(0)
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

5.2 Circuit Diagram

5.3 Code

5.3.1 Using gate level modelling

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

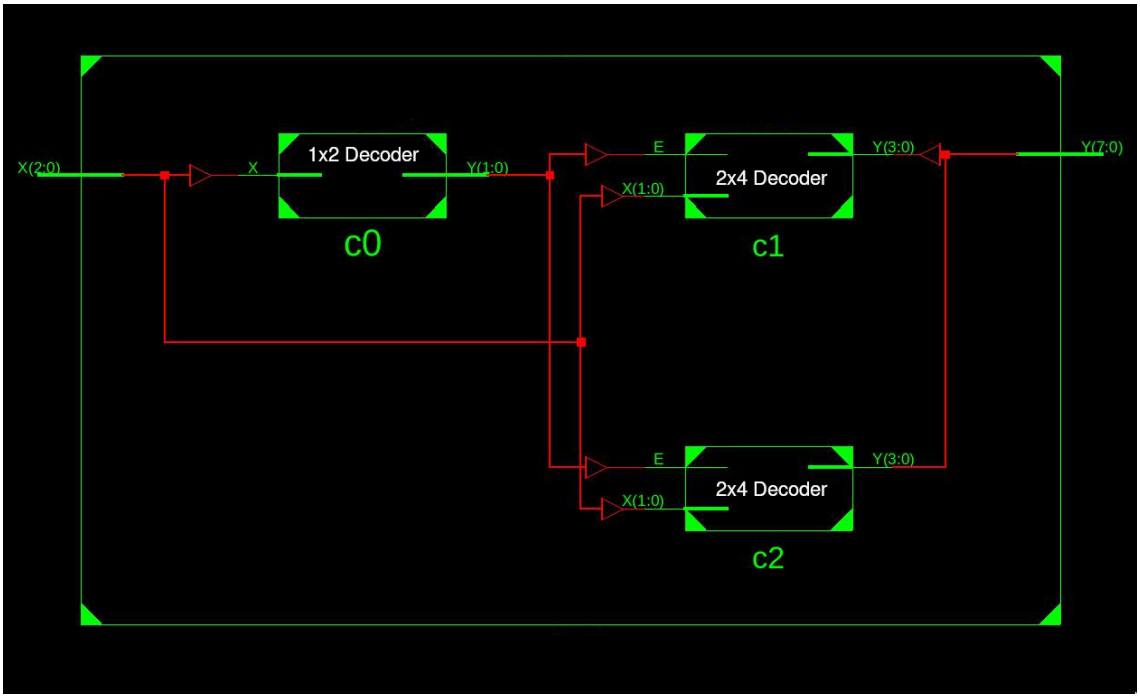


Figure 4: 3x8 circuit diagram

```
entity a3_2_2a is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_2_2a;

architecture Behavioral of a3_2_2a is

begin

        Y(7) <= X(2) and X(1) and X(0);
        Y(6) <= X(2) and X(1) and not X(0);
        Y(5) <= X(2) and not X(1) and X(0);
        Y(4) <= X(2) and not X(1) and not X(0);
        Y(3) <= not X(2) and X(1) and X(0);
        Y(2) <= not X(2) and X(1) and not X(0);
        Y(1) <= not X(2) and not X(1) and X(0);
        Y(0) <= not X(2) and not X(1) and not X(0);


end Behavioral;
```

5.3.2 Using if else statement

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

entity a3_2_2ba is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_2_2ba;

architecture Behavioral of a3_2_2ba is

begin
```

```
        p1:process(X)
        begin
        if X = "000" then Y <= "00000001";
        elsif X = "001" then Y <= "00000010";
        elsif X = "010" then Y <= "00000100";
        elsif X = "011" then Y <= "00001000";
        elsif X = "100" then Y <= "00010000";
        elsif X = "101" then Y <= "00100000";
        elsif X = "110" then Y <= "01000000";
        elsif X = "111" then Y <= "10000000";
        else Y <= "ZZZZZZZZ";
        end if;
        end process;

end Behavioral;
```

5.3.3 Using case statement

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

entity a3_2_2bb is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_2_2bb;

architecture Behavioral of a3_2_2bb is

begin
        p1 : process(X)
        begin
                case X is
                        when "000" => Y <= "00000001";
                        when "001" => Y <= "00000010";
                        when "010" => Y <= "00000100";
                        when "011" => Y <= "00001000";
                        when "100" => Y <= "00010000";
                        when "101" => Y <= "00100000";
                        when "110" => Y <= "01000000";
                        when "111" => Y <= "10000000";
                        when others => Y <= "ZZZZZZZZ";
                end case;
        end process;


end Behavioral;
```

5.3.4 Using when else statement

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

entity a3_2_2bc is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_2_2bc;

architecture Behavioral of a3_2_2bc is

begin
```

```
        Y <=         "00000001" when X = "000" else
                        "00000010" when x = "001" else
                        "00000100" when X = "010" else
                        "00001000" when X = "011" else
                        "00010000" when X = "100" else
                        "00100000" when X = "101" else
                        "01000000" when X = "110" else
                        "10000000" when X = "111" else
                        "ZZZZZZZZ";
end Behavioral;
```

5.3.5 Using select when statement

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

entity a3_2_2bd is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_2_2bd;

architecture Behavioral of a3_2_2bd is

begin

        with X select Y <=
                "00000001" when "000",
                "00000010" when "001",
                "00000100" when "010",
                "00001000" when "011",
                "00010000" when "100",
                "00100000" when "101",
                "01000000" when "110",
                "10000000" when "111",
                "ZZZZZZZZ" when others;

end Behavioral;
```

5.3.6 Using 2x4 decoder and 1x2 decoder by component instantiation

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

entity a3_3 is
    Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
           Y : out  STD_LOGIC_VECTOR (7 downto 0));
end a3_3;

architecture Behavioral of a3_3 is


component a3_2_1a is
    PORT(
         X : IN  std_logic_vector(1 downto 0);
                        E : in STD_LOGIC;
         Y : OUT  std_logic_vector(3 downto 0)
        );
end component;

component a3_1a is
```

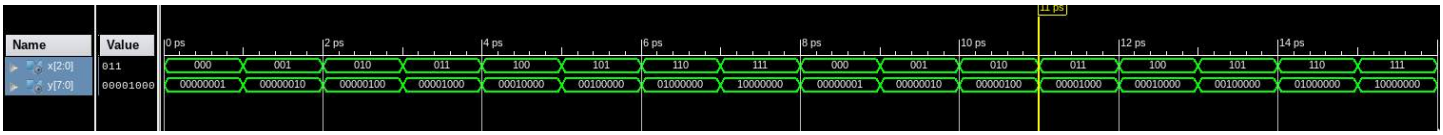
```
    Port ( X : in  STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (1 downto 0));
end component;

signal v1,v2 : std_logic_vector (3 downto 0);
signal e1    : std_logic_vector (1 downto 0);

begin

        c0 : a3_1a port map(X(2), e1);
        c1 : a3_2_1a port map(X(1 downto 0), e1(0), v1);
        c2 : a3_2_1a port map(X(1 downto 0), e1(1), v2);

        Y <= v2 & v1;

end Behavioral;
```

5.4 Test Bench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_a3_2_2a IS
END tb_a3_2_2a;

ARCHITECTURE behavior OF tb_a3_2_2a IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT a3_2_2a
    PORT(
         X : IN  std_logic_vector(2 downto 0);
         Y : OUT  std_logic_vector(7 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal X : std_logic_vector(2 downto 0) := (others => '0');

         --Outputs
   signal Y : std_logic_vector(7 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

BEGIN

        -- Instantiate the Unit Under Test (UUT)
   uut: a3_2_2a PORT MAP (
          X => X,
          Y => Y
        );



   -- Stimulus process
   stim_proc: process
   begin
               X <= "000"; wait for 1 ps;
               X <= "001"; wait for 1 ps;
```

```
                X <= "010"; wait for 1 ps;
                X <= "011"; wait for 1 ps;
                X <= "100"; wait for 1 ps;
                X <= "101"; wait for 1 ps;
                X <= "110"; wait for 1 ps;
                X <= "111"; wait for 1 ps;


   end process;

END;
```

5.5 Timing diagram



6 4x16 Decoder

6.1 Truth Table

X(3 - 0)	Y(15 - 0)
0000	0000000000000001
0001	0000000000000010
0010	0000000000000100
0011	0000000000001000
0100	0000000000010000
0101	0000000000100000
0110	0000000001000000
0111	0000000010000000
1000	0000000100000000
1001	0000001000000000
1010	0000010000000000
1011	0000100000000000
1100	0001000000000000
1101	0010000000000000
1110	0100000000000000
1111	1000000000000000

6.2 Circuit Diagram

6.3 Code

6.3.1 Using 2x4 decoder only by component instantiation

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

entity a3_4 is
    Port ( X : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (15 downto 0));
end a3_4;

architecture Behavioral of a3_4 is

component a3_2_1a is
    Port ( X : in  STD_LOGIC_VECTOR (1 downto 0);
                        E : in STD_LOGIC;
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
```

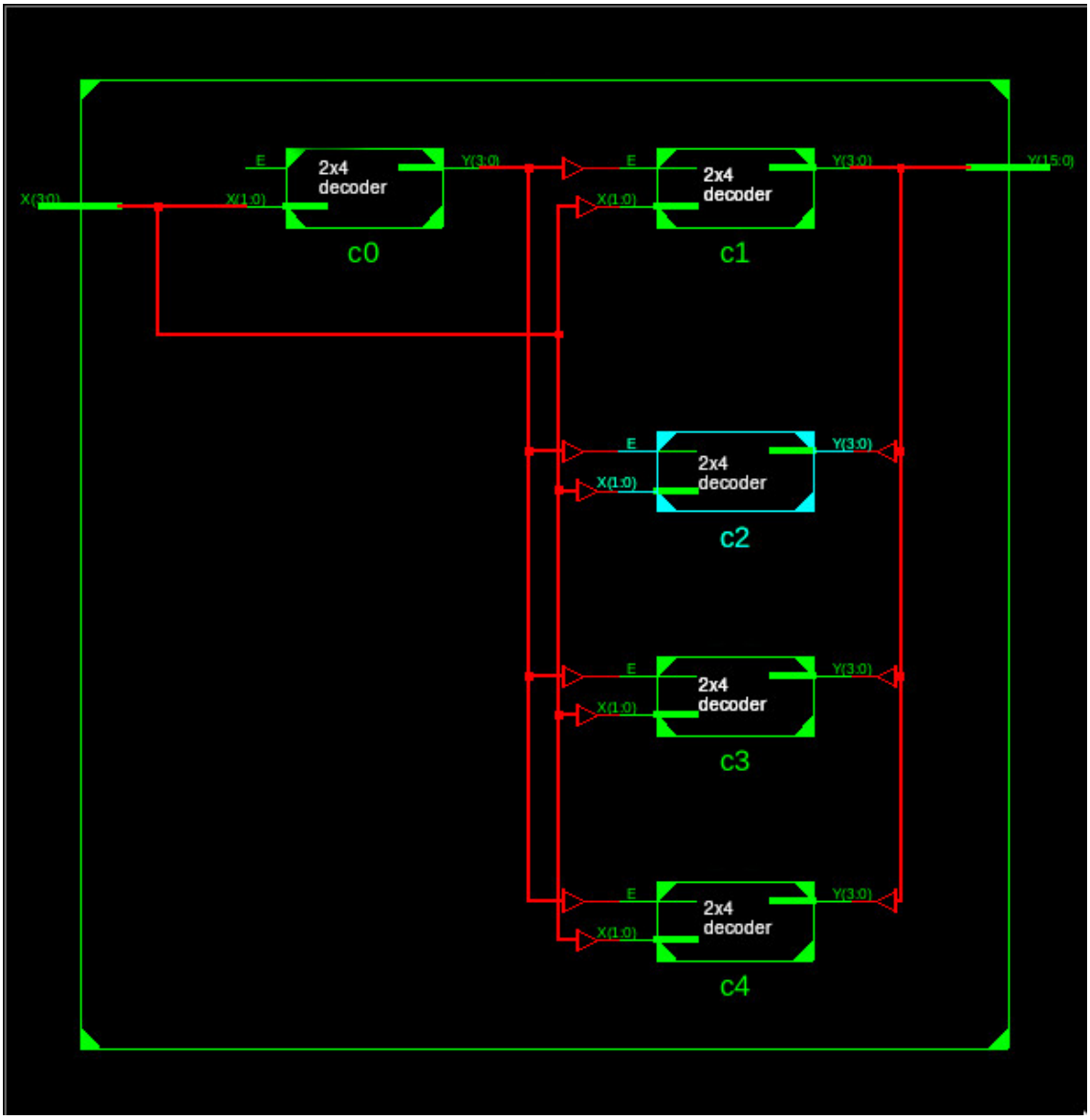


Figure 5: 4x16 circuit diagram

```
end component;

signal e1,v0,v1,v2,v3: std_logic_vector(3 downto 0);

begin

                c0 : a3_2_1a port map(X(3 downto 2), '1', e1);
                c1 : a3_2_1a port map(X(1 downto 0), e1(3), v3);
                c2 : a3_2_1a port map(X(1 downto 0), e1(2), v2);
                c3 : a3_2_1a port map(X(1 downto 0), e1(1), v1);
                c4 : a3_2_1a port map(X(1 downto 0), e1(0), v0);

                Y <= v3 & v2 & v1 & v0;

end Behavioral;
```

6.4 Test Bench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
```

```
ENTITY tb_a3_4 IS
END tb_a3_4;

ARCHITECTURE behavior OF tb_a3_4 IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT a3_4
    PORT(
         X : IN  std_logic_vector(3 downto 0);
         Y : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


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

         --Outputs
   signal Y : std_logic_vector(15 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name


BEGIN

         -- Instantiate the Unit Under Test (UUT)
   uut: a3_4 PORT MAP (
          X => X,
          Y => Y
        );


   -- Stimulus process
   stim_proc: process
   begin
                X <= "0000";
                wait for 1 ps;
                X <= "0001";
                wait for 1 ps;
                X <= "0010";
                wait for 1 ps;
                X <= "0011";
                wait for 1 ps;
                X <= "0100";
                wait for 1 ps;
                X <= "0101";
                wait for 1 ps;
                X <= "0110";
                wait for 1 ps;
                X <= "0111";
                wait for 1 ps;

                X <= "1000";
                wait for 1 ps;
                X <= "1001";
                wait for 1 ps;
                X <= "1010";
                wait for 1 ps;
```

```
                X <= "1011";
                wait for 1 ps;
                X <= "1100";
                wait for 1 ps;
                X <= "1101";
                wait for 1 ps;
                X <= "1110";
                wait for 1 ps;
                X <= "1111";
                wait for 1 ps;
    end process;

END;
```

6.5 Timing diagram