



16x4 Encoder

VLSI Systems

Assignment-2 Annexure-II

PREPARED BY

Priyank Lohariwal

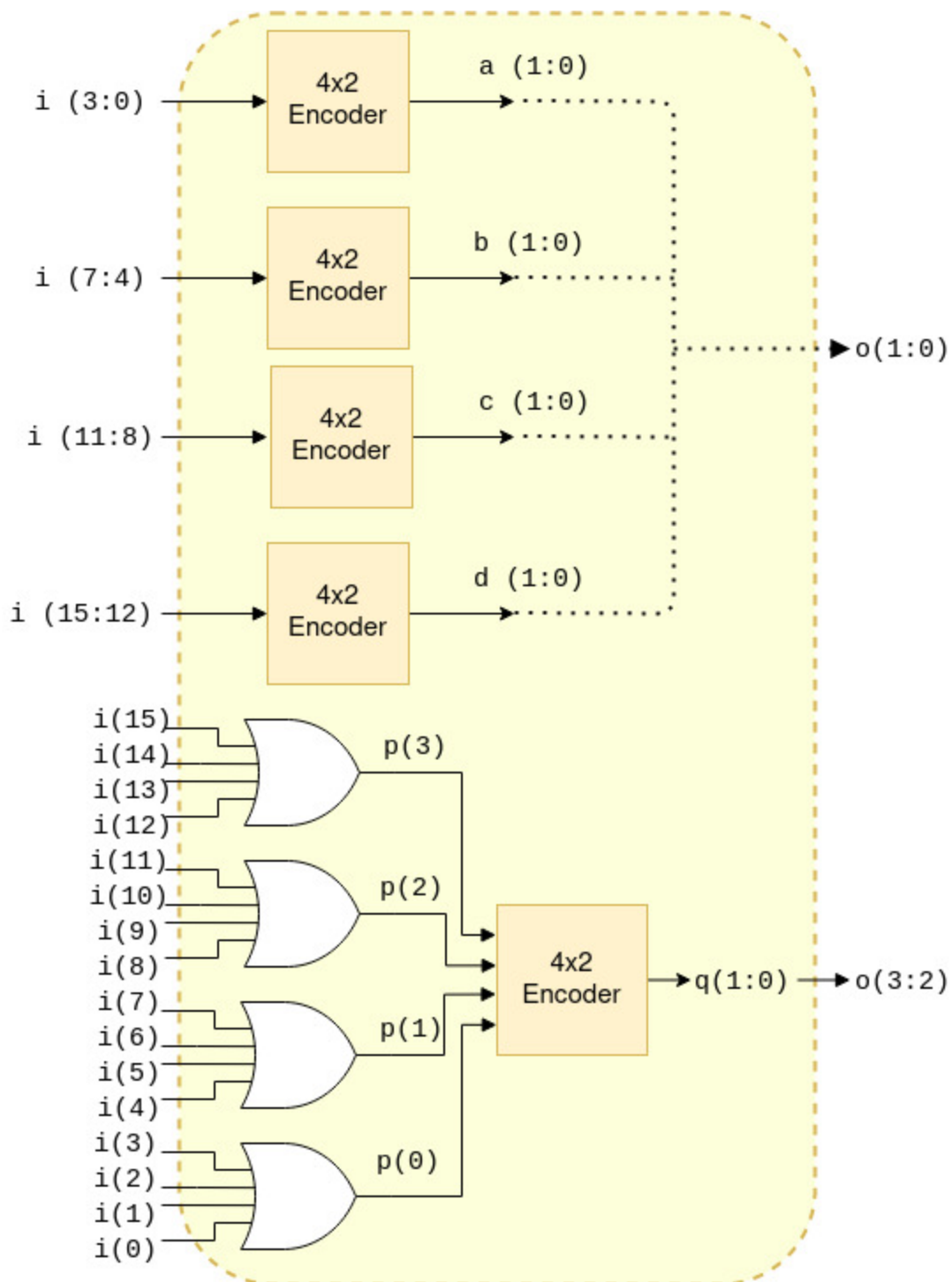
BCSE-IV

001710501055

Jadavpur University

[illegible]

Circuit Diagram



Code

1. Using functions of 4x2 encoders

```
architecture Behavioral of ass2_annex2a is
function mux4x2(x: in std_logic_vector) return std_logic_vector is
    variable a: std_logic_vector(1 downto 0);
begin
    if x = "0001" then
        a := "00";
    elsif x = "0010" then
        a := "01";
    elsif x = "0100" then
        a := "10";
    elsif x = "1000" then
        a := "11";
    else
        a := "ZZ";
    end if;
    return a;
end function;

begin

    p1: process(X)
        variable j,k: integer;
        variable a: std_logic_vector(7 downto 0);
        variable q: std_logic_vector(1 downto 0);
        variable p: std_logic_vector(3 downto 0);

    begin
        for k in 0 to 3 loop
            a(2*k+1 downto 2*k) := mux4x2(X(4*k+3 downto 4*k));
            p(k) := '0';
            for j in 0 to 3 loop
                p(k) := p(k) or X(4*k + j);
            end loop;
        end loop;

        q := mux4x2(p);
        if q = "ZZ" then
            Y <= "ZZZZ";
        else
            for k in 0 to 3 loop
```

```

        if p(k) = '1' then
            if a(2*k+1 downto 2*k) = "ZZ" then
                Y <= "ZZZZ";
            else
                Y <= q & a(2*k+1 downto 2*k);
            end if;
        end if;
    end loop;
end if;
end process;
end Behavioral;

```

2. Using procedures of 4x2 encoders

```

architecture Behavioral of ass2_annex2b is
    procedure mux4x2(x:in std_logic_vector; y:out std_logic_vector) is
        variable a: std_logic_vector(1 downto 0);
    begin
        if x = "0001" then
            a := "00";
        elsif x = "0010" then
            a := "01";
        elsif x = "0100" then
            a := "10";
        elsif x = "1000" then
            a := "11";
        else
            a := "ZZ";
        end if;
        y := a;
    end procedure;
end Behavioral;

```

```
begin
```

```

p1: process(X)
    variable j,k: integer;
    variable a: std_logic_vector(7 downto 0);
    variable q: std_logic_vector(1 downto 0);
    variable p: std_logic_vector(3 downto 0);
begin
    for k in 0 to 3 loop
        proc1: mux4x2(X(4*k+3 downto 4*k), a(2*k+1 downto 2*k));
        p(k) := '0';
        for j in 0 to 3 loop

```

```

        p(k) := p(k) or X(4*k+j);
    end loop;
end loop;
proc2: mux4x2(p, q);
if q = "ZZ" then
    Y <= "ZZZZ";
else
    for k in 0 to 3 loop
        if p(k) = '1' then
            if a(2*k+1 downto 2*k) = "ZZ" then
                Y <= "ZZZZ";
            else
                Y <= q & a(2*k+1 downto 2*k);
            end if;
        end if;
    end loop;
end if;
end process;
end Behavioral;

```

Test Bench

```

ARCHITECTURE behavior OF ass2_annex1_test_bench IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT ass2_annex1
    PORT (
        X : IN  std_logic_vector(15 downto 0);
        Y : OUT std_logic_vector(3  downto 0)
    );
    END COMPONENT;

    --Inputs
    signal X : std_logic_vector(15 downto 0) := (others => '0');

    --Outputs
    signal Y : std_logic_vector(3  downto 0);
BEGIN
    -- Instantiate the Unit Under Test (UUT)
    uut: ass2_annex1 PORT MAP (
        X => X,
        Y => Y
    );

```

```

-- Stimulus process
stim_proc: process
begin
    X <= "0000000000000000";
    wait for 1 ps;
    for i in 0 to 15 loop
        X(i) <= '1';
        wait for 1ps;
        X(i) <= '0';
    end loop;
end process;
END;

```

Timing Diagram

