Assignment 3

IT451 : COA LAB

ANINDYA KUNDU
IT, 4th Semester
ID: 510817020 (Hx-19)

25-03-2019

Question 1

Design the following types of code converters in a single module using switch case statement.

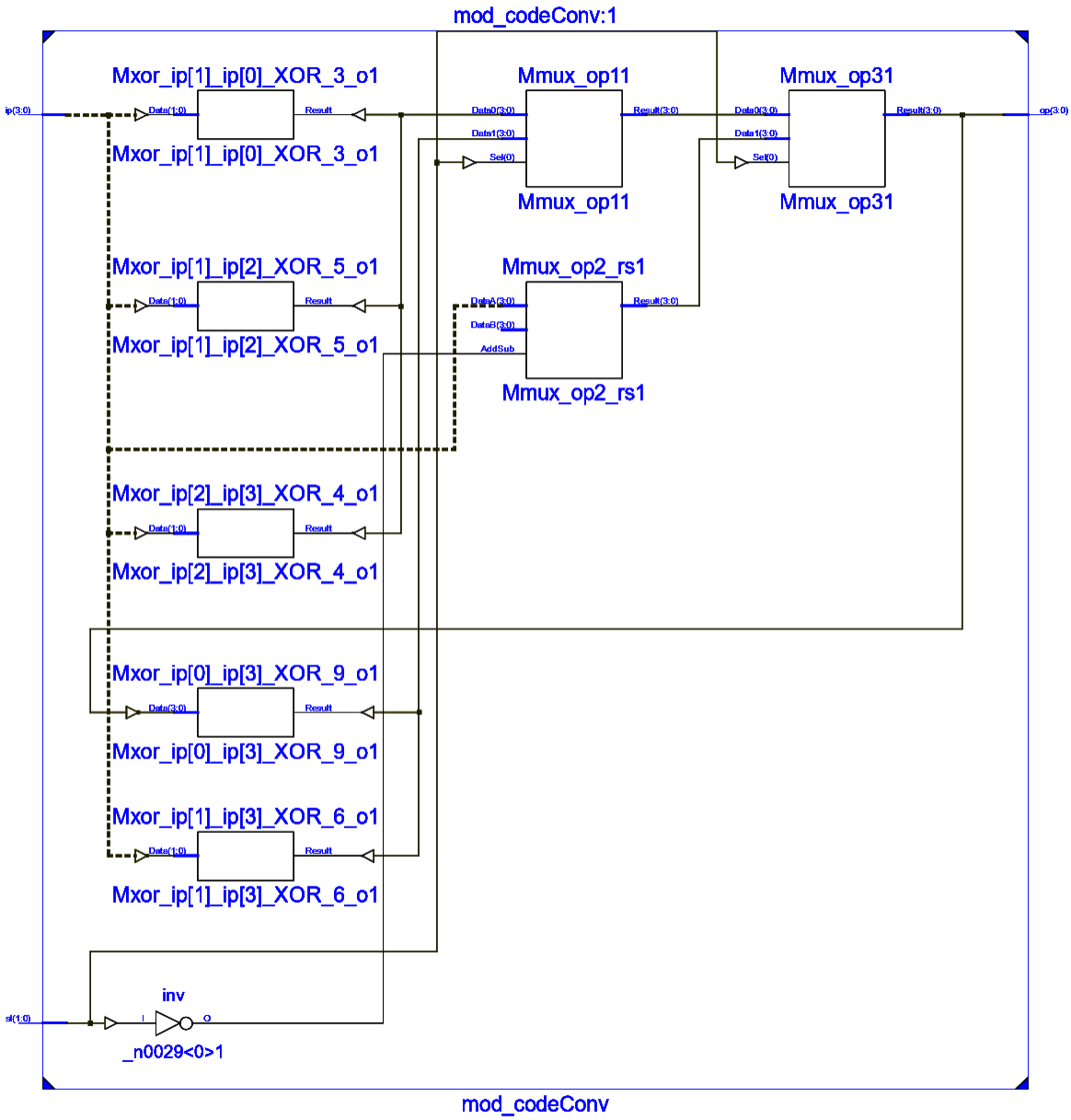
a. Binary to Gray
b. Gray to Binary
c. Binary to Excess-3
d. Excess-3 to Binary

VHDL Module: *mod_codeConv.vhd*

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

entity mod_codeConv is
    Port ( sl : in  STD_LOGIC_VECTOR (1 downto 0);
           ip : in  STD_LOGIC_VECTOR (3 downto 0);
           op : out STD_LOGIC_VECTOR (3 downto 0));
end mod_codeConv;

architecture Dataflow of mod_codeConv is
      signal s : STD_LOGIC_VECTOR (3 downto 0);

begin

process(sl, ip)
begin
      case sl is
            when "00" =>
                  s(3) <= ip(3);
                  s(2) <= ip(3) xor ip(2);
                  s(1) <= ip(2) xor ip(1);
                  s(0) <= ip(1) xor ip(0);
            when "01" =>
                  s(3) <= ip(3);
                  s(2) <= ip(2) xor ip(3);
                  s(1) <= ip(1) xor ip(2) xor ip(3);
                  s(0) <= ip(0) xor s(1) xor ip(2) xor ip(3);
            when "10" =>
                  s <= STD_LOGIC_VECTOR (unsigned(ip) + 3);
            when "11" =>
                  s <= STD_LOGIC_VECTOR (unsigned(ip) - 3);
            when others =>
      end case;
end process;
```

```
        op <= s;

end Dataflow;
```

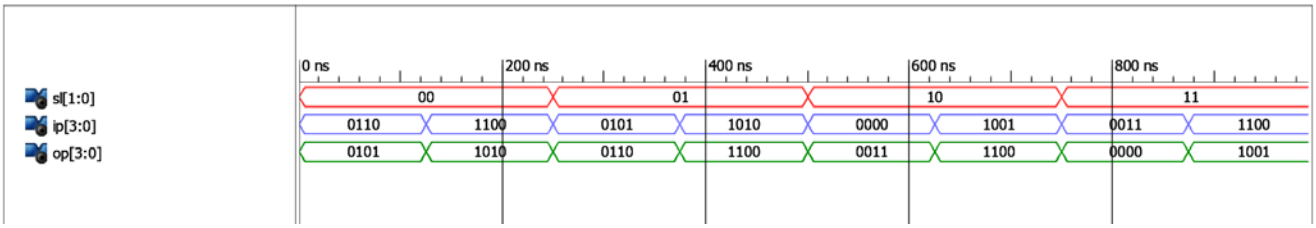
RTL Schematic:



VHDL Test Bench: *tb_codeConv.vhd*

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

ENTITY tb_codeConv IS
END tb_codeConv;

ARCHITECTURE behavior OF tb_codeConv IS
    COMPONENT mod_codeConv
    PORT(
         sl : IN  std_logic_vector(1 downto 0);
         ip : IN  std_logic_vector(3 downto 0);
         op : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;

   signal sl : std_logic_vector(1 downto 0) := (others => '0');
   signal ip : std_logic_vector(3 downto 0) := (others => '0');

   signal op : std_logic_vector(3 downto 0);

BEGIN
   uut: mod_codeConv PORT MAP (
          sl => sl,
          ip => ip,
          op => op
        );

   stim_proc: process
   begin
      sl <= "00", "01" after 250 ns, "10" after 500 ns, "11" after 750ns;
      ip <= "0110"; wait for 125ns; ip <= "1100"; wait for 125ns;
      ip <= "0101"; wait for 125ns; ip <= "1010"; wait for 125ns;
      ip <= "0000"; wait for 125ns; ip <= "1001"; wait for 125ns;
      ip <= "0011"; wait for 125ns; ip <= "1100"; wait for 125ns;
      wait;
   end process;
END;
```

Simulation:


0 ns
200 ns
400 ns
600 ns
800 ns
sl[1:0]
ip[3:0]
op[3:0]
00
01
10
11
0110
1100
0101
1010
0000
1001
0011
1100
0101
1010
0110
1100
0011
1100
0000
1001

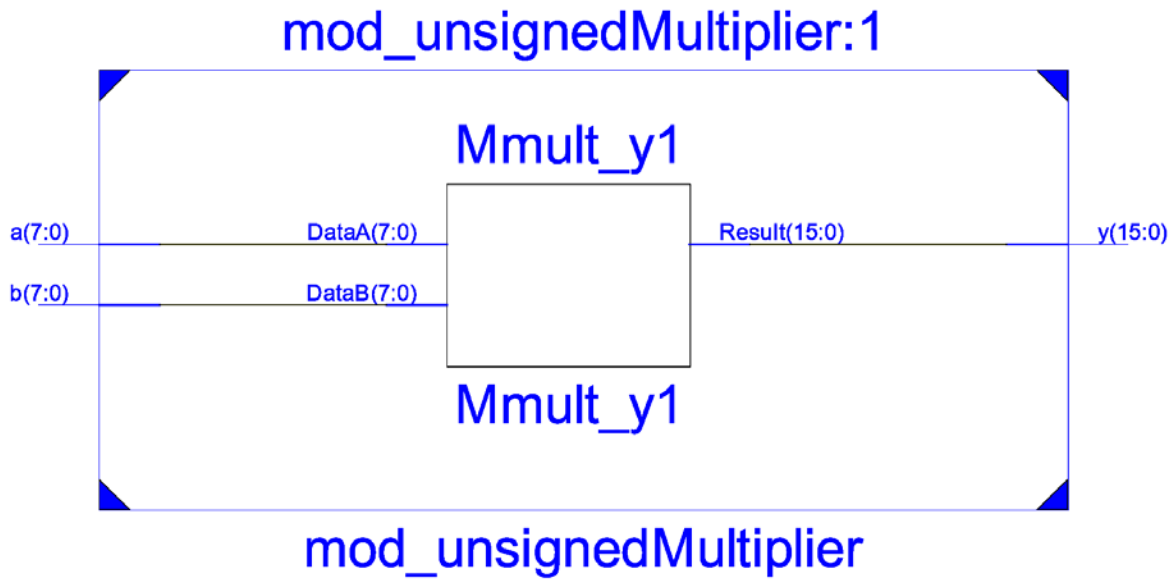
Question 2

Design and simulate 8 bit unsigned binary multiplier. Synthesize it for Spartan 6. Report device utilization.

VHDL Module: *mod_unsignedMultiplier.vhd*

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

entity mod_unsignedMultiplier is
    Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
           b : in  STD_LOGIC_VECTOR (7 downto 0);
           y : out STD_LOGIC_VECTOR (15 downto 0));
end mod_unsignedMultiplier;

architecture Behavioral of mod_unsignedMultiplier is

begin
      y <= STD_LOGIC_VECTOR(unsigned(a) * unsigned(b));
end Behavioral;
```

RTL Schematic:



Device Utilisation Summary:

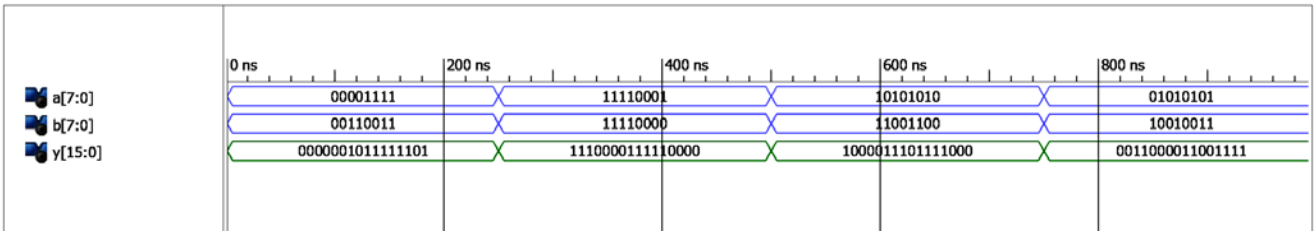
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of bonded IOBs	32	296	10%
Number of DSP48A1s	1	58	1%

VHDL Test Bench: *tb_unsignedMultiplier.vhd*

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

ENTITY tb_unsignedMultiplier IS
END tb_unsignedMultiplier;

ARCHITECTURE behavior OF tb_unsignedMultiplier IS
    COMPONENT mod_unsignedMultiplier
    PORT(
         a : IN  std_logic_vector(7 downto 0);
         b : IN  std_logic_vector(7 downto 0);
         y : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;

   signal a : std_logic_vector(7 downto 0) := (others => '0');
   signal b : std_logic_vector(7 downto 0) := (others => '0');
   signal y : std_logic_vector(15 downto 0);

BEGIN
   uut: mod_unsignedMultiplier PORT MAP (
          a => a,
          b => b,
          y => y
        );

   proc_a: process
   begin
      a <= "00001111", "11110001" after 250 ns, "10101010" after 500 ns, "01010101"
      after 750 ns;
      wait;
   end process;

   proc_b: process
   begin
      b <= "00110011", "11110000" after 250 ns, "11001100" after 500 ns, "10010011"
      after 750 ns;
```

```
        wait;
    end process;
END;
```

Simulation:



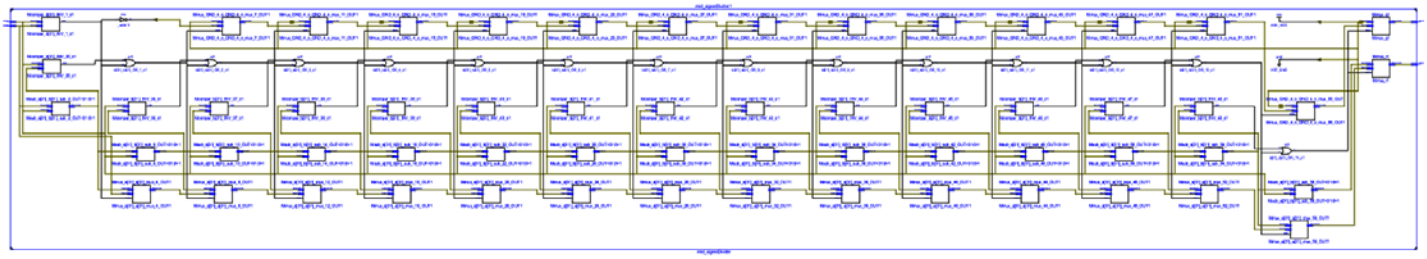
Question 3

Design and simulate a 32 bit divider. Synthesize it for Spartan 6. Report device utilization.

VHDL Module: *mod_signedDivider.vhd*

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

entity mod_unsignedDivider is
    Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
           b : in  STD_LOGIC_VECTOR (3 downto 0);
           q : out  STD_LOGIC_VECTOR (3 downto 0);
           r : out  STD_LOGIC_VECTOR (3 downto 0));
end mod_unsignedDivider;

architecture Behavioral of mod_unsignedDivider is
      signal au, bu, qu, ru : signed (3 downto 0);

begin

process (a, b) is
    variable cnt  : STD_LOGIC_VECTOR (31 downto 0);
    variable tem  : STD_LOGIC_VECTOR (31 downto 0);

begin
      cnt   := "00000000000000000000000000000000";
      tem   := a;

      while (tem >= b and cnt < "11111111111111111111111111111111") loop
            tem   := STD_LOGIC_VECTOR(signed(tem) - signed(b));
            cnt   := STD_LOGIC_VECTOR(signed(cnt) + 1);
      end loop;

      q <= cnt;
      r <= tem;
end process;

end Behavioral;
```

RTL Schematic:



Device Utilisation Summary:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	1247	27288	4%
Number of fully used LUT-FF pairs	0	1247	0%
Number of bonded IOBs	128	296	43%

VHDL Test Bench: *tb_signedDivider.vhd*

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

ENTITY tb_unsignedDivider IS
END tb_unsignedDivider;

ARCHITECTURE behavior OF tb_unsignedDivider IS
    COMPONENT mod_unsignedDivider
    PORT(
         a : IN  std_logic_vector(3 downto 0);
         b : IN  std_logic_vector(3 downto 0);
         q : OUT std_logic_vector(3 downto 0);
         r : OUT std_logic_vector(3 downto 0)
        );
    END COMPONENT;

   signal a : std_logic_vector(3 downto 0) := (others => '0');
   signal b : std_logic_vector(3 downto 0) := (others => '0');
   signal q : std_logic_vector(3 downto 0);
   signal r : std_logic_vector(3 downto 0);

BEGIN
   uut: mod_unsignedDivider PORT MAP (
          a => a,
          b => b,
          q => q,
          r => r
```

```
        );

    proc_a: process
    begin
       a <= "00000000000000000000000000000111";
       wait for 333 ns;
       a <= "11110000111111110000111100000011";
       wait for 333 ns;
       a <= "00000000000000000111111111001000";
       wait;
    end process;

    proc_b: process
    begin
       b <= "00000000000000000000000000000010";
       wait for 333 ns;
       b <= "11110000111111110000111100000011";
       wait for 333 ns;
       b <= "11000000000000000000000000000001";
       wait;
    end process;
END;
```

Simulation: