**Experiment-2**

**Aim:**

1. To write and simulate the VHDL codes for the following basic combinational logic using dataflow modelling.

1. 3:8 Decoder
2. 4 bit comparator
3. 4 bit ALU with 8 possible operators (bits)
4. 4 bit ALU with 8 possible operators (integers)

**Software required:**

1.Quartus II

**VHDL code:**

**Code A:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity decoder\_3\_8 is

port (I: in std\_logic\_vector(2 downto 0);

En: in std\_logic;

y: out std\_logic\_vector(7 downto 0));

end decoder\_3\_8;

Architecture behave of decoder\_3\_8 is

begin

y<="00000001" when I = "000" else

"00000010" when I = "001" else

"00000100" when I = "010" else

"00001000" when I = "011" else

"00010000" when I = "100" else

"00100000" when I = "101" else

"01000000" when I = "110" else

"10000000" when I = "111" else

"00000000";

end behave ;

**Code B:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity comparator\_4\_bit is

port (A,B: in std\_logic\_vector(3 downto 0);

y: out std\_logic\_vector(2 downto 0));

end comparator\_4\_bit;

Architecture behave of comparator\_4\_bit is

begin

y <= "100" when A=B else

"010" when A>B else

"001" when A<B else

"000";

end behave ;

**Code C:**

library ieee;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_1164.all;

Entity alu\_4\_bit is

port (A,B: in std\_logic\_vector(3 downto 0);

s: in std\_logic\_vector(2 downto 0);

y: out std\_logic\_vector(3 downto 0));

end alu\_4\_bit;

Architecture dataflow of alu\_4\_bit is

begin

with s select

y <= A + B when "000",

A - B when "001",

A and B when "010",

A or B when "011",

A xor B when "100",

A nand B when "101",

A nor B when "110",

A xnor B when "111",

"0000" when others;

end dataflow ;

**Code D:**

library ieee;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_1164.all;

Entity alu\_4\_bit\_int is

port (A,B: in integer range 0 to 3;

s: in std\_logic\_vector(2 downto 0);

y: out integer range 0 to 3);

end alu\_4\_bit\_int;

Architecture dataflow of alu\_4\_bit\_int is

begin

with s select

y <= A + B when "000",

A - B when "001",

abs (A - B) when "010",

A \* B when "011",

A / B when "100",

A mod B when "101",

A \*\* 2 when "110",

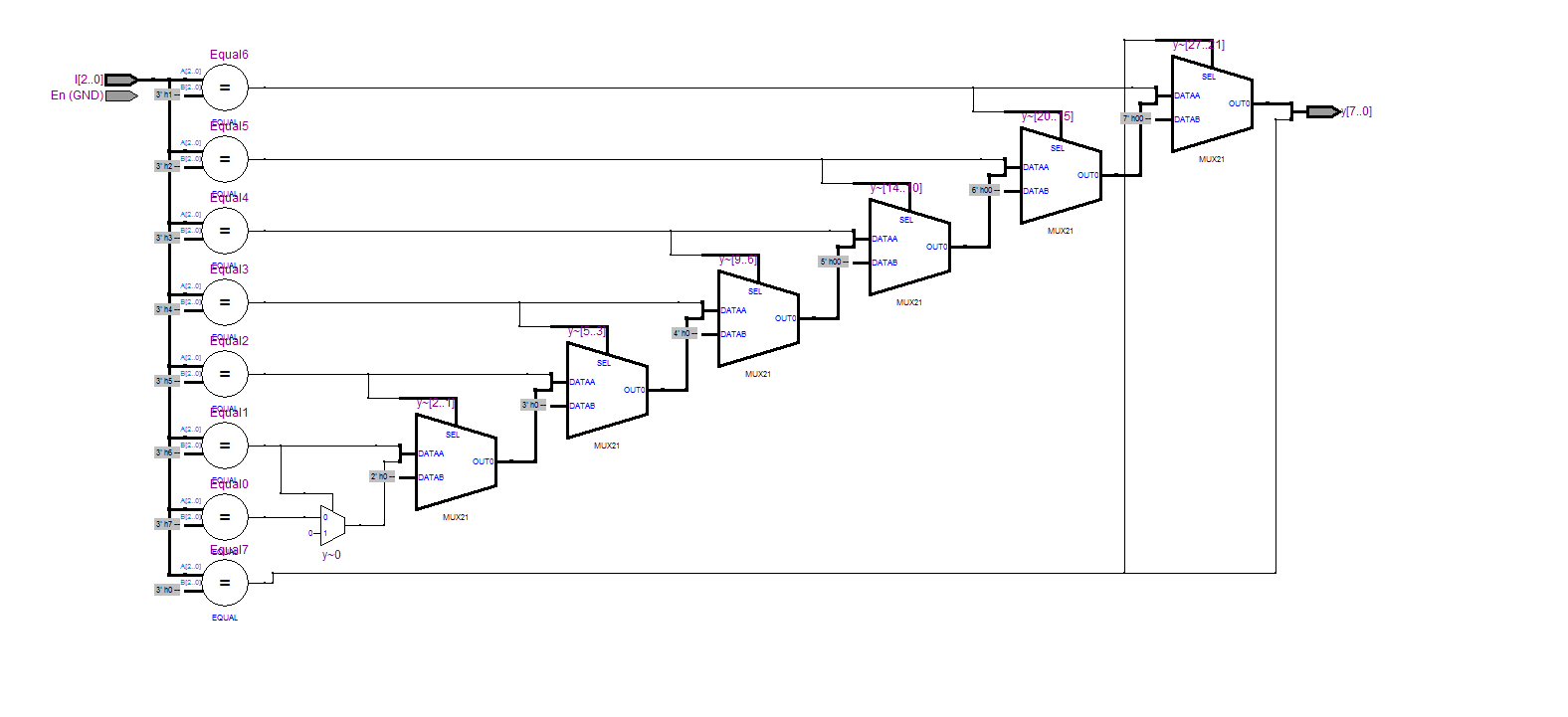
2 \*\* B when "111",

0 when others;

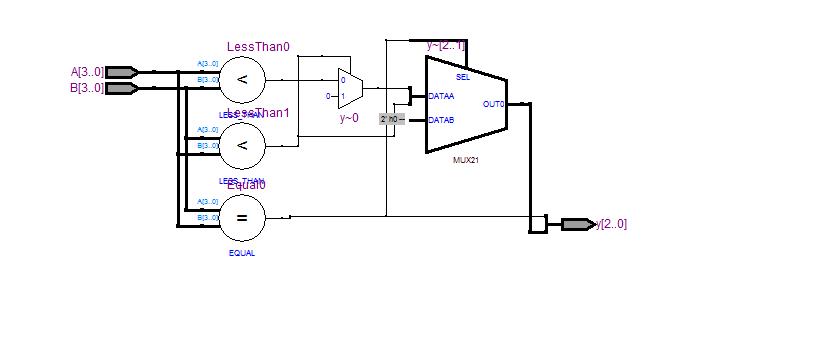
end dataflow ;

**RTL schematic:**

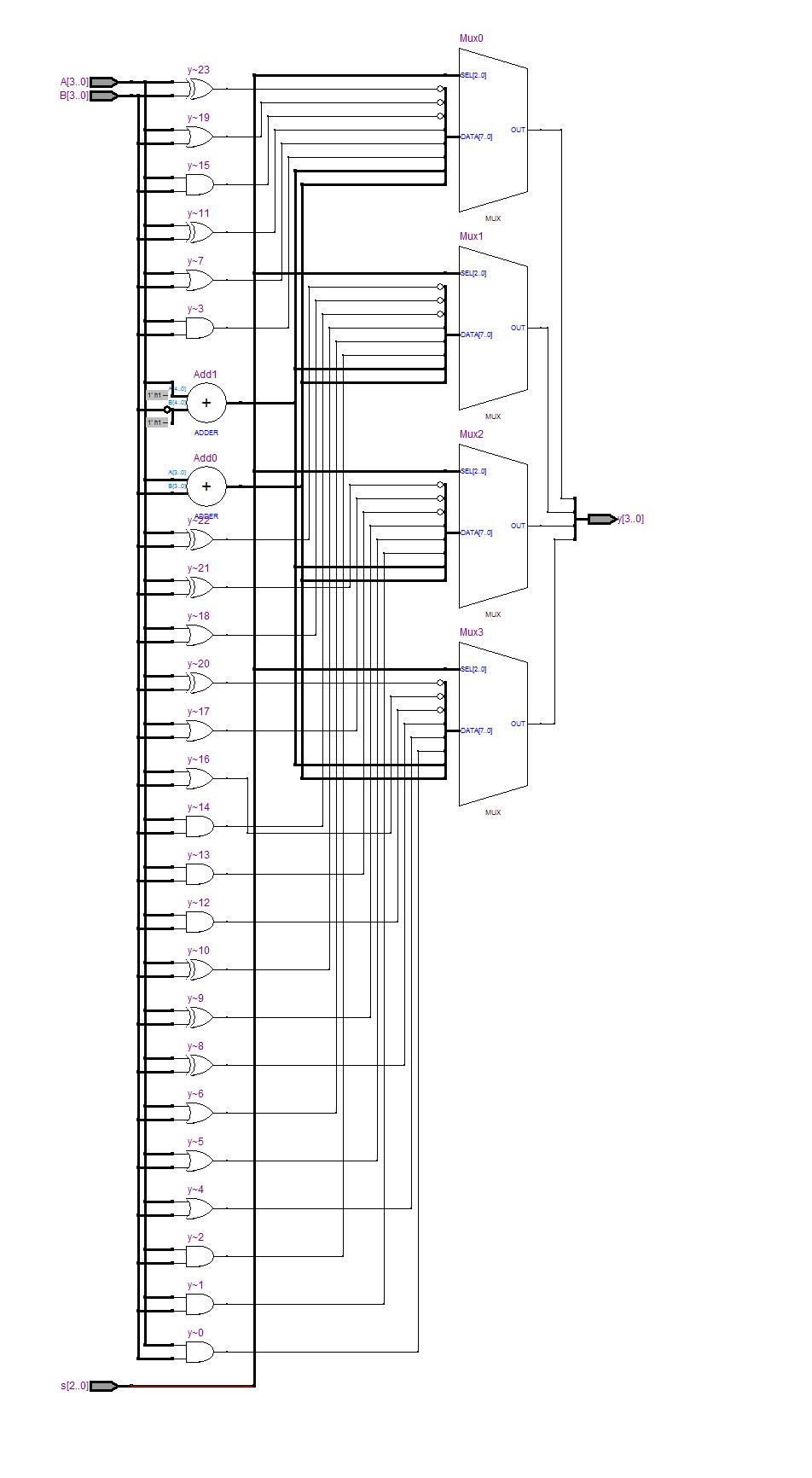
**RTL a:**



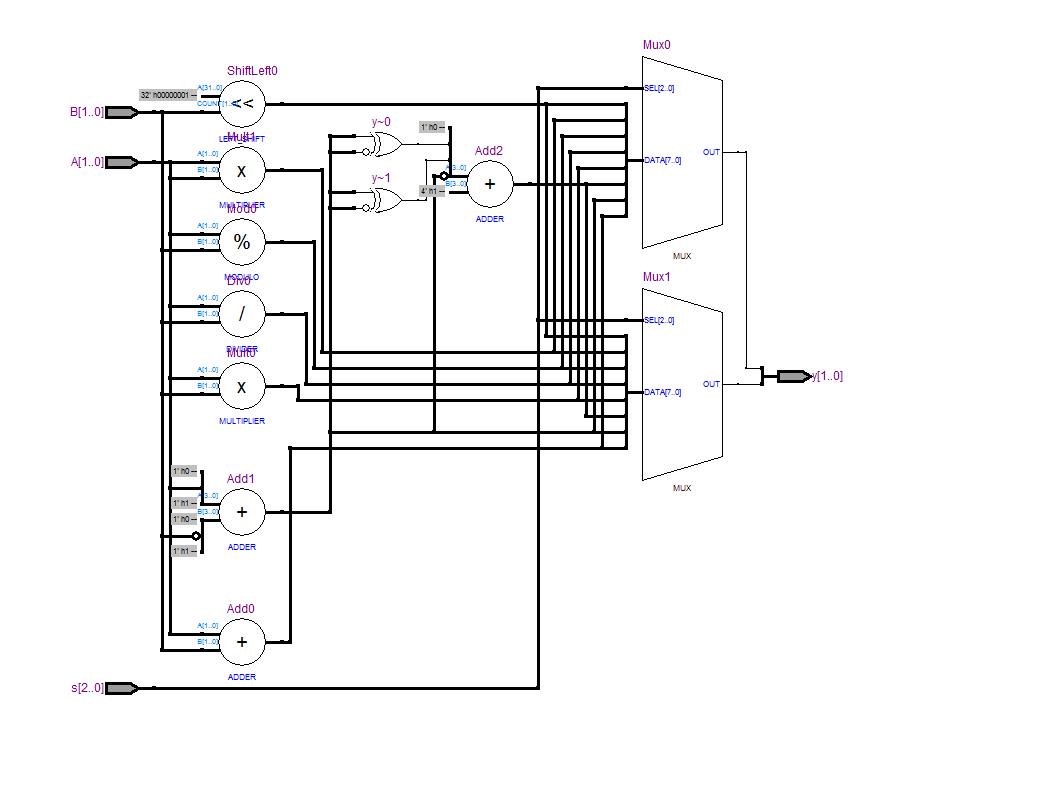
**RTL b:**

****

**RTL c:**

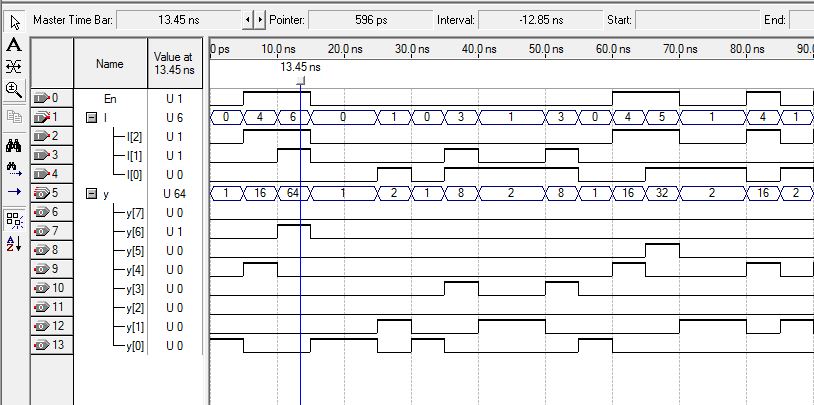
****

**RTL d:**

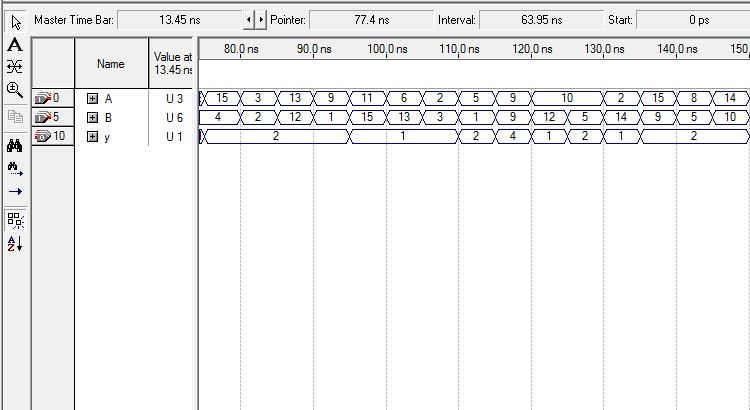
****

**Waveforms:**

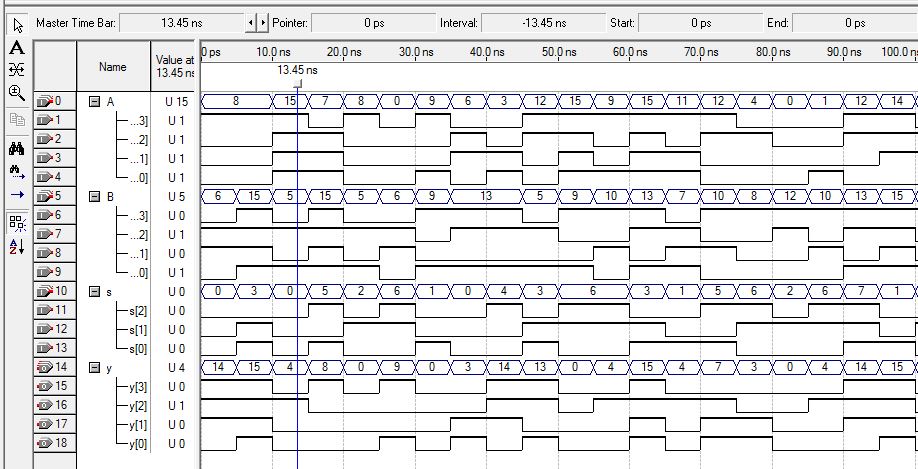
**Waveform A:**

****

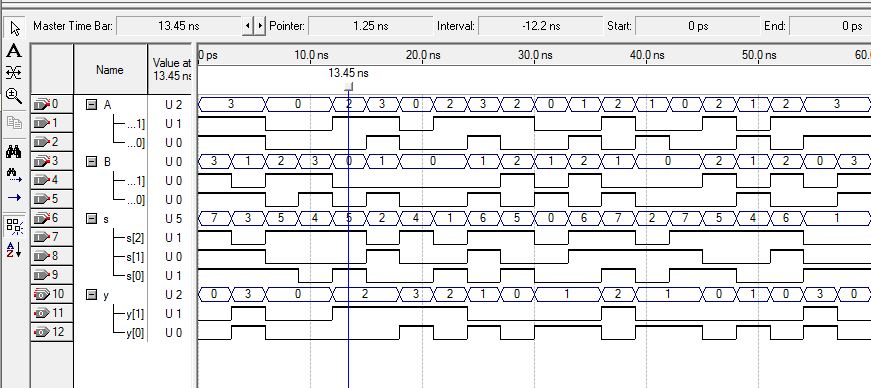
**Waveform B:**

****

**Waveform C:**

****

**Waveform D:**



**Result:**

Hence the VHDL codes were written and the simulation was successful for the given basic combinational logic. Waveforms and RTL schematic were recorded and verified.