**Experiment-1**

**Aim:**

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

a. OR gate

b. Half adder

c. 4:1 mux

d. Full adder

**Software required:**

1.Quartus II

**VHDL code:**

**Code a:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity or\_gate is

Port(A,B: in bit;

C: out bit);

end or\_gate;

architecture dataflow of or\_gate is

begin

c <= a or b;

end dataflow;

**Code b:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity half\_adder is

Port(A,B: in bit;

S,C: out bit);

end half\_adder;

Architecture dataflow of half\_adder is

begin

s <= ((not a) and b) or (a and (not b));

c <= a and b;

end dataflow;

**Code c:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity mux is

Port(I: in std\_logic\_vector(3 downto 0);

S: in std\_logic\_vector(1 downto 0);

Y: out std\_logic);

end mux;

architecture dataflow of mux is

begin

y <= ((not s(0)) and (not s(1)) and i(0)) or (s(0) and (not s(1)) and i(1)) or ((not s(0)) and s(1) and i(2)) or (s(0) and s(1) and i(3));

end dataflow;

**Code d:**

library ieee;

use ieee.std\_logic\_1164.all;

Entity full\_adder is

Port(A,B,Ci: in bit;

S,Co: out bit);

end full\_adder;

architecture dataflow of full\_adder is

begin

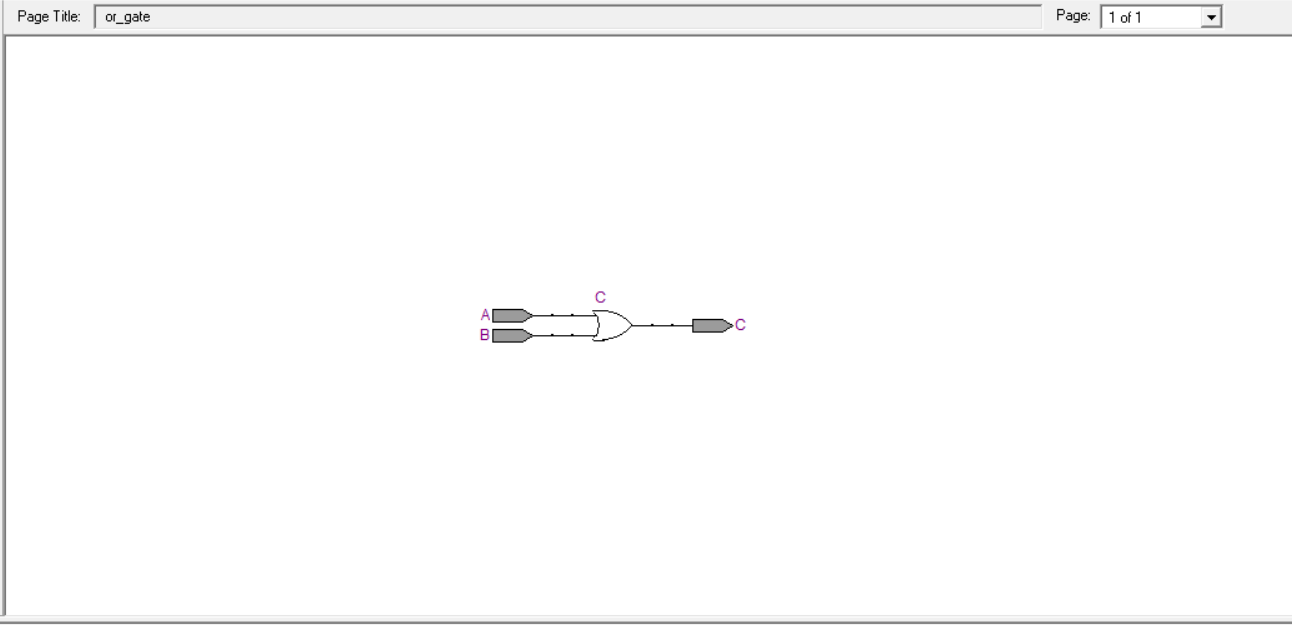
s <= ((not a) and (not b) and ci) or ((not a) and b and (not ci)) or (a and b and ci) or (a and (not b) and (not ci));

co <= (b and ci) or (a and ci) or (a and b);

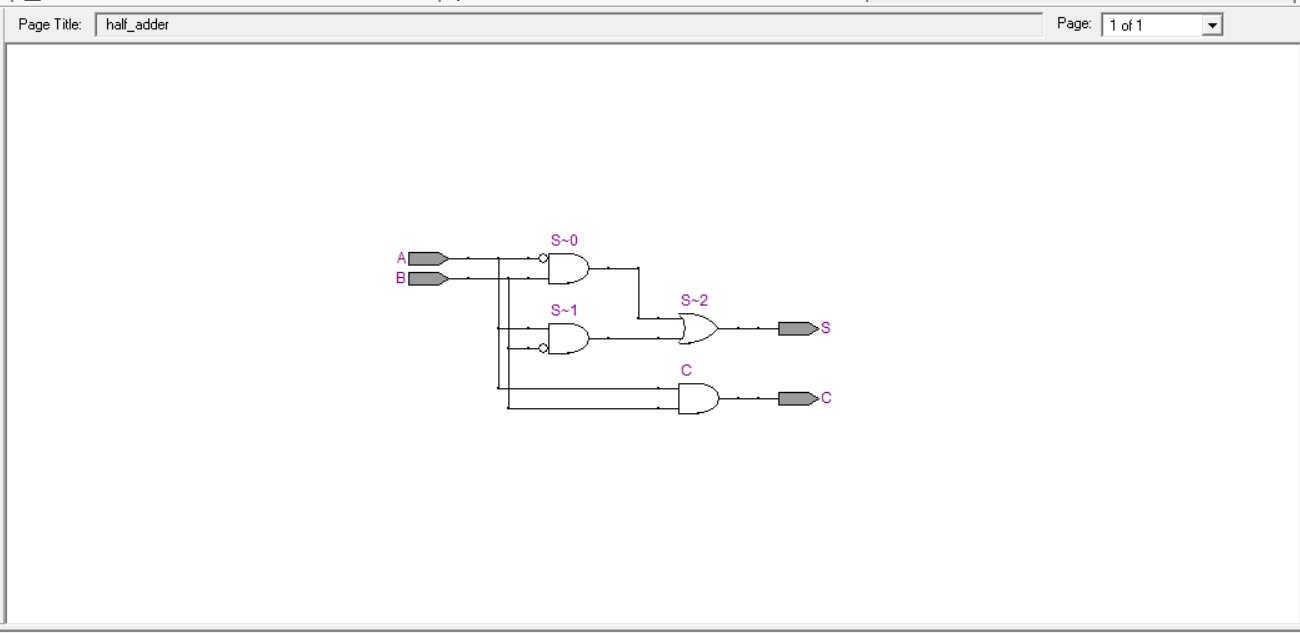
end dataflow;

**RTL schematic:**

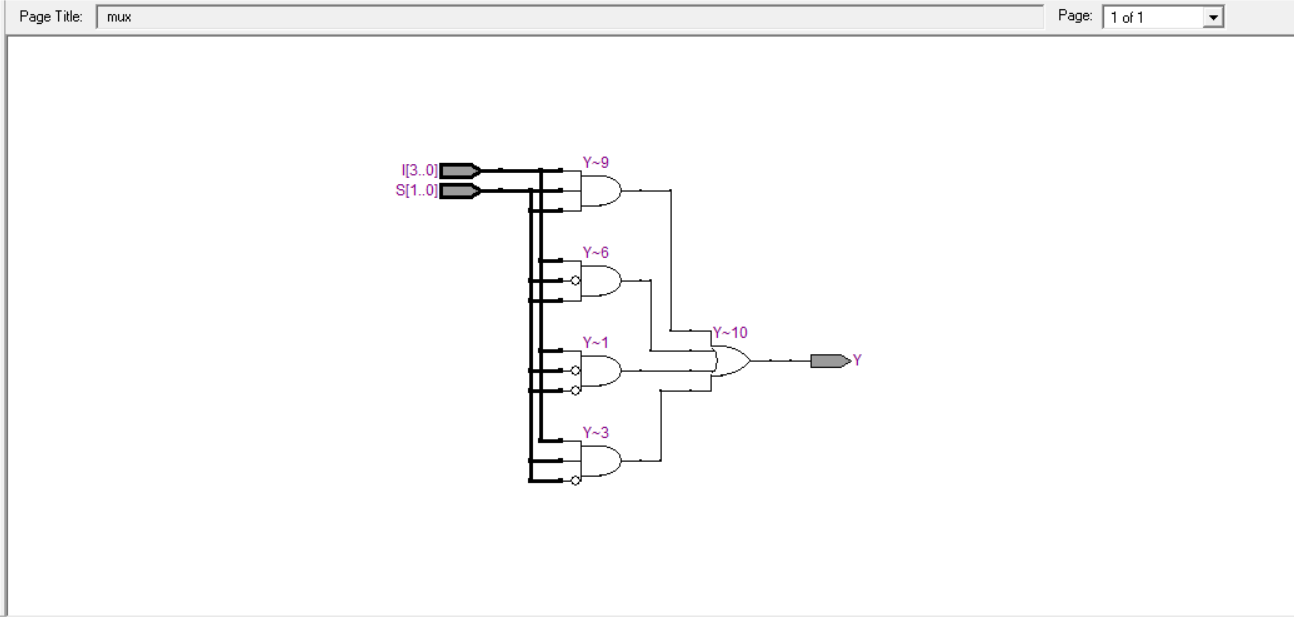
**RTL a:**



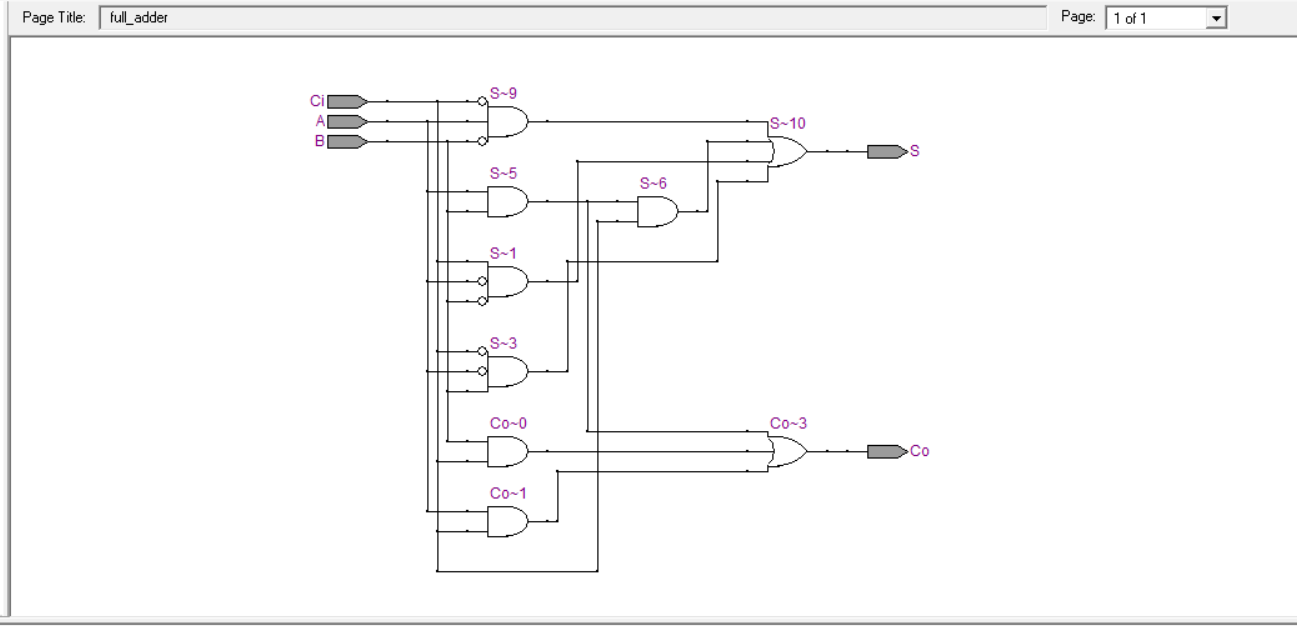
**RTL b:**



**RTL c:**

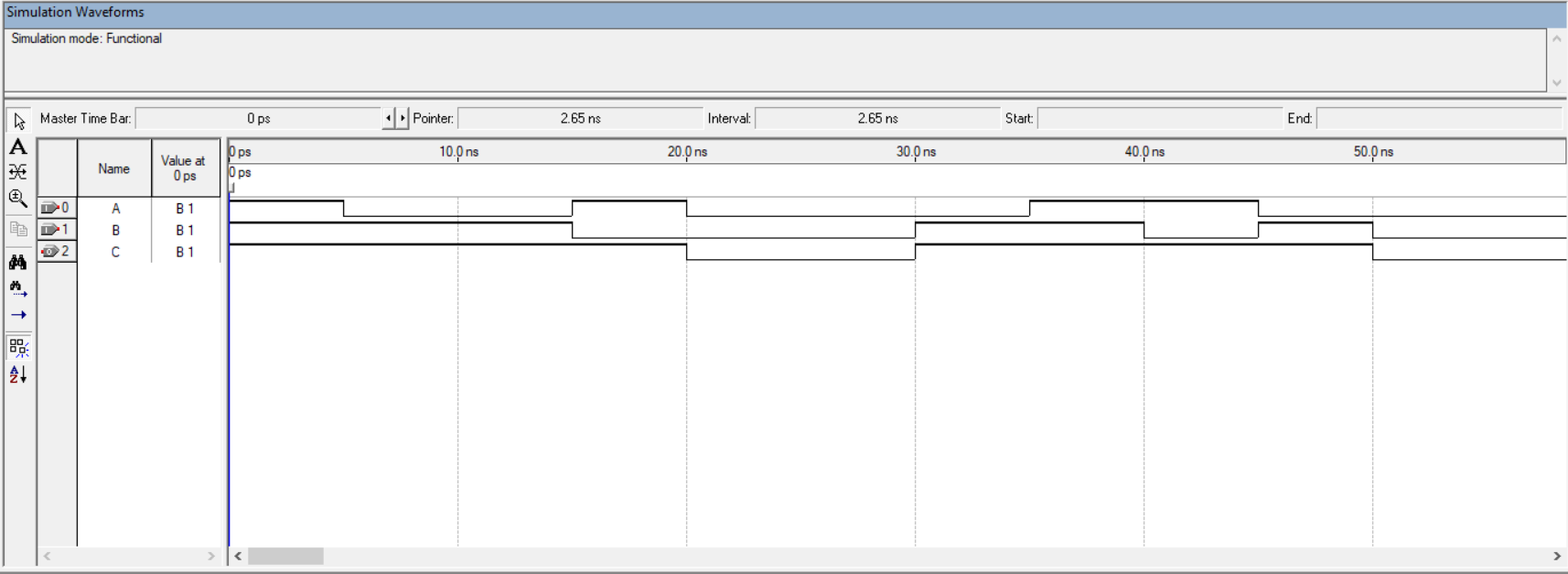


**RTL d:**

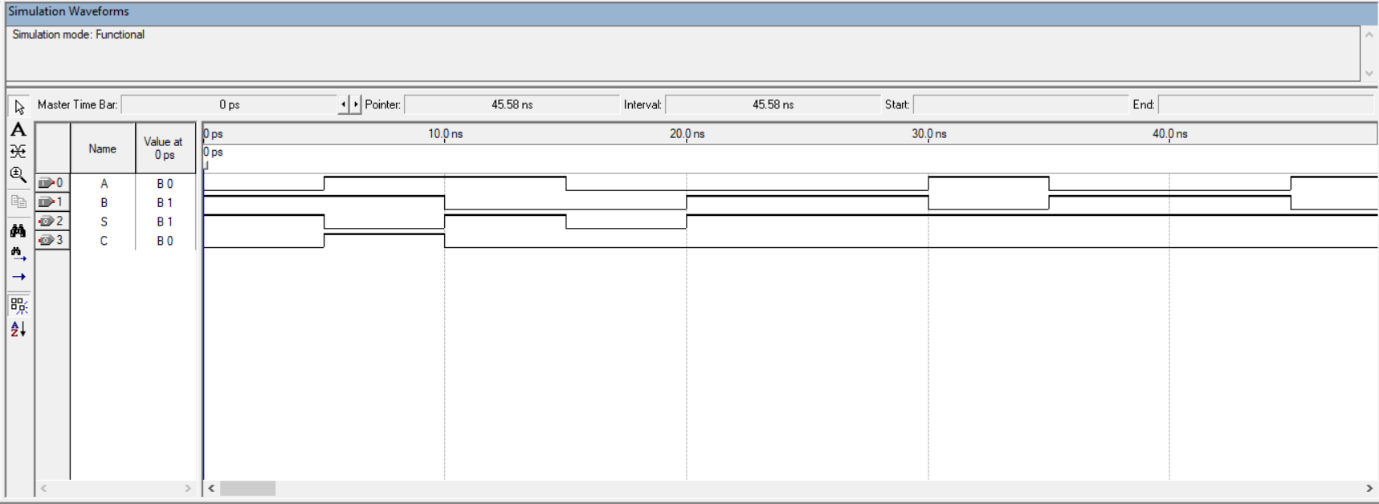


**Waveforms:**

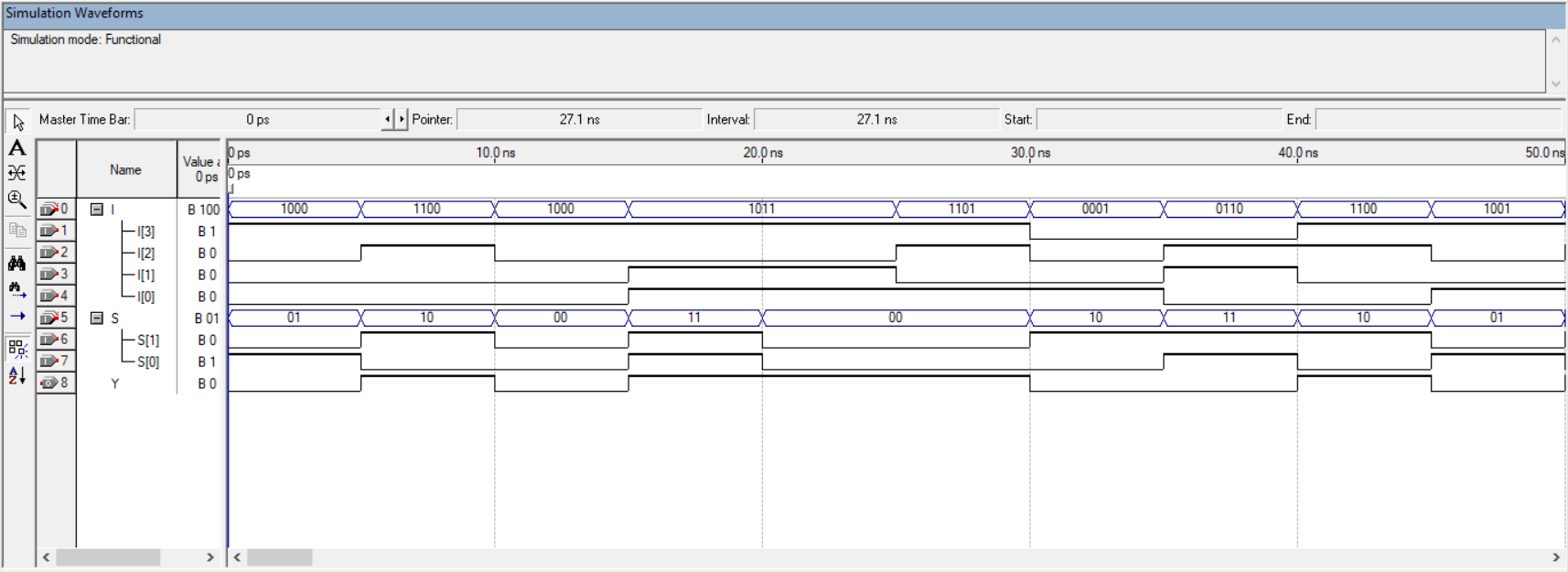
**Waveform a:**



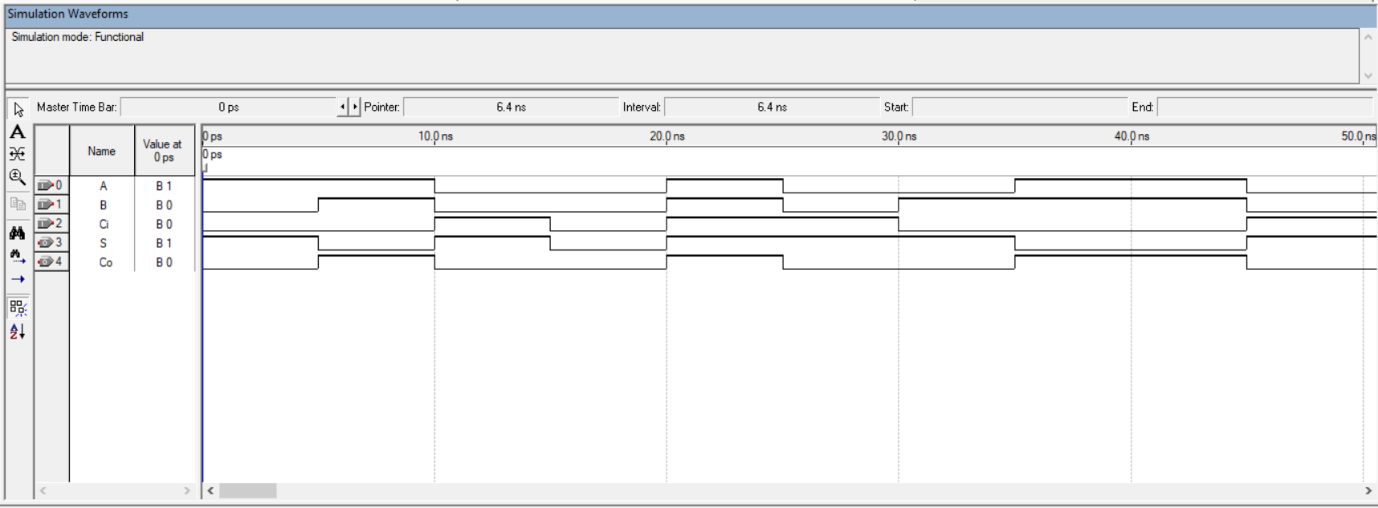
**Waveform b:**



**Waveform c:**



**Waveform d:**



**Result:**

Hence the VHDL codes were written and the simulation was successful for the given bsic combinational logic.

Waveforms and RTL schematic were recorded and verified.