**Computer Hardware Experiments**

Lab\_01: Basic operation of ALU

**Name : 李泽宇 Student\_ID: 1709853W-I011-0036**

Part 1：Design a one bit full adder

1.source code:

—————————————————————————————library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity adder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Ci : in STD\_LOGIC;

S : out STD\_LOGIC;

Co : out STD\_LOGIC);

end adder;

architecture Behavioral of adder is

begin

S <= A xor B xor Ci;

Co <= (A and B) or (Ci and (A xor B));

end Behavioral;

1. simulation waveform：

A.

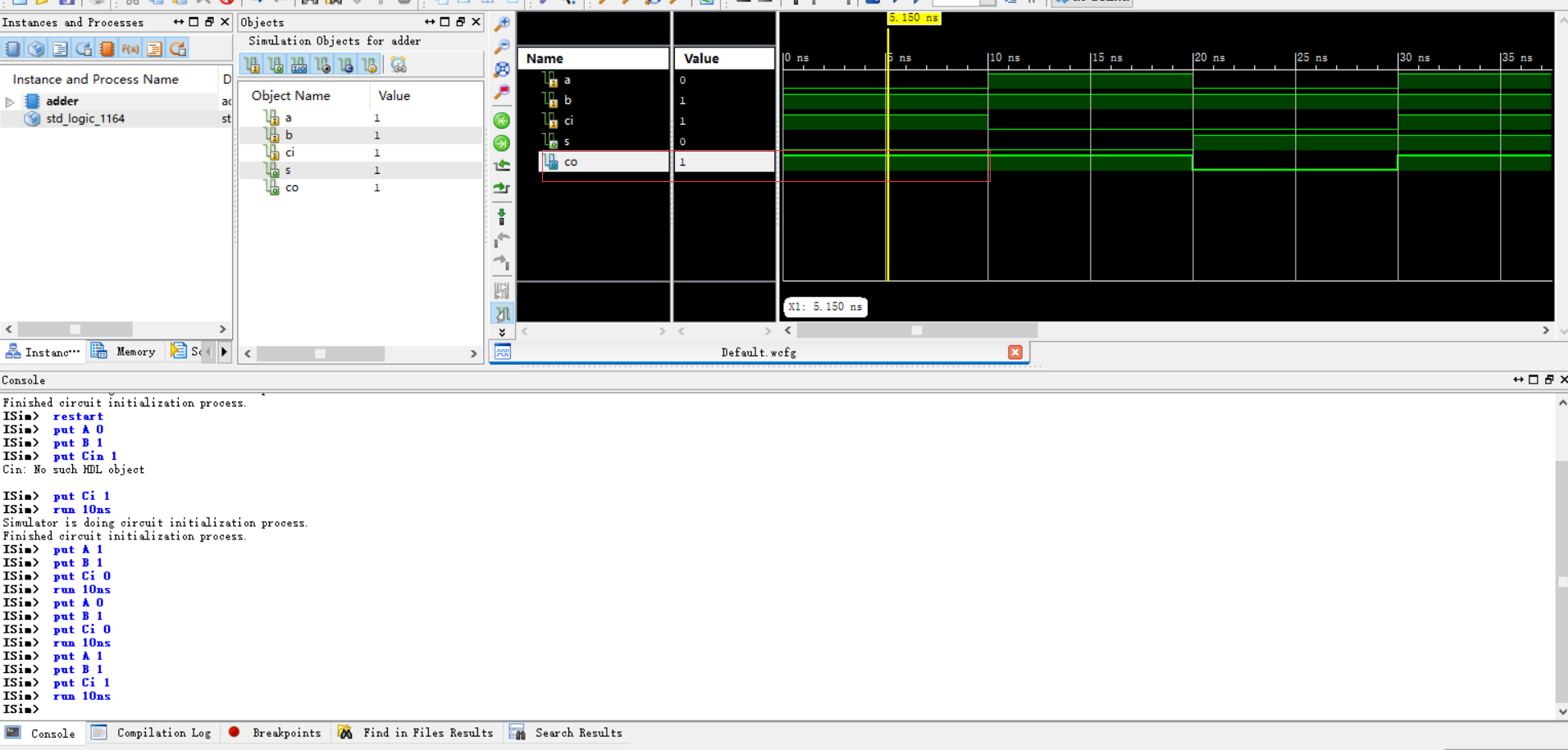
ISim > restart

ISim > put A 0

ISim > put B 1

ISim > put Cin 1

ISim > run 10ns



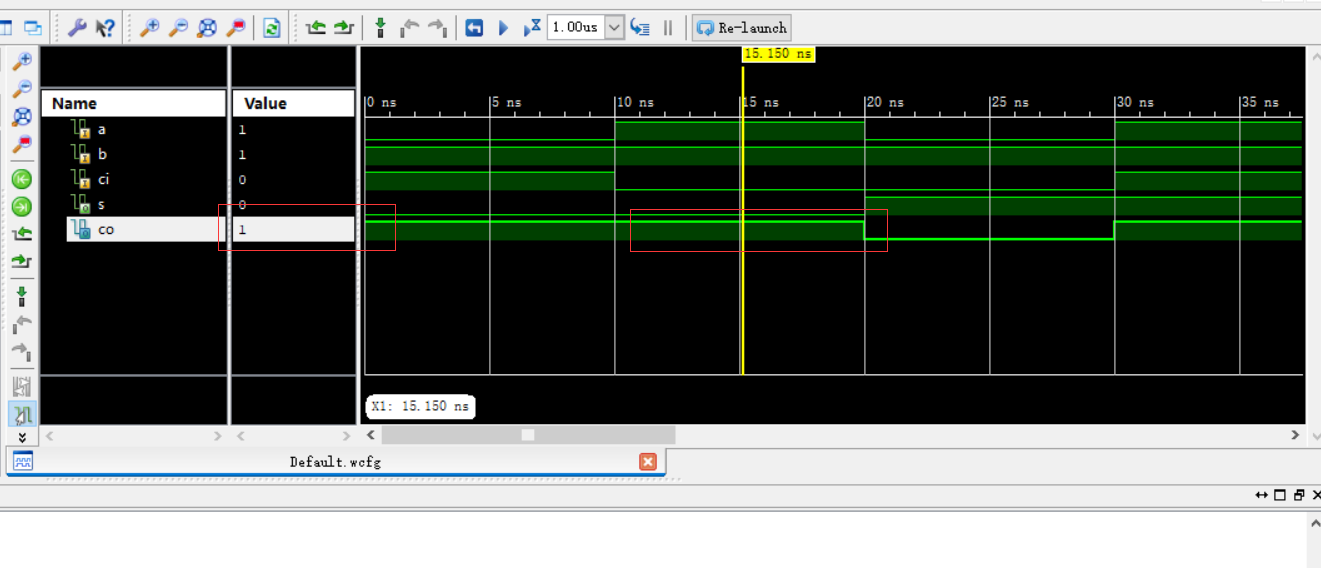
B.

ISim > put A 1

ISim > put B 1

ISim > put Cin 0

ISim > run 10ns



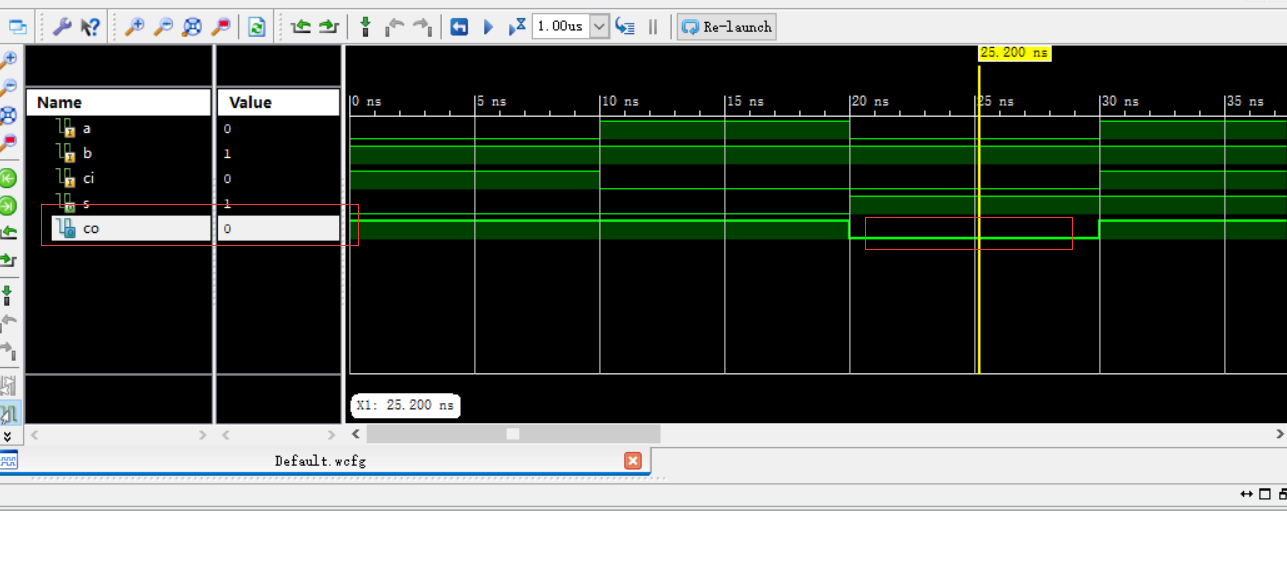
C.

ISim > put A 0

ISim > put B 1

ISim > put Cin 0

ISim > run 10ns



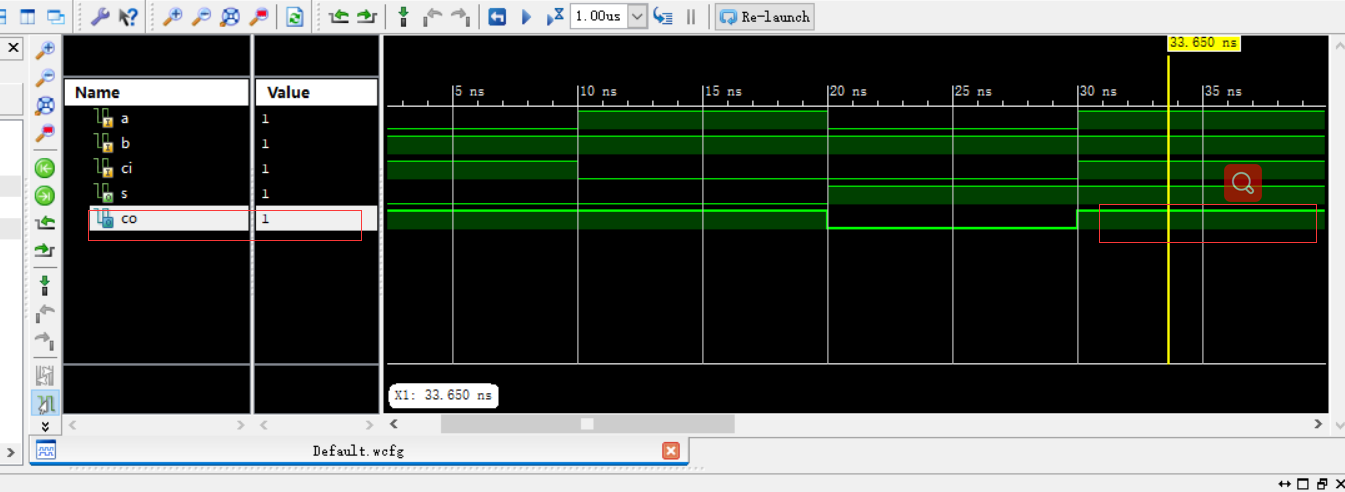
D.ISim > put A 1

ISim > put B 1

ISim > put Cin 1

ISim > run 10ns

ISim >



Part2：Design a 16 bit Multiplier.

1.source code:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity multiplier is

Port ( A : in STD\_LOGIC\_VECTOR (15 downto 0);

B : in STD\_LOGIC\_VECTOR (15 downto 0);

P : out STD\_LOGIC\_VECTOR (31 downto 0));

end multiplier;

architecture Behavioral of multiplier is

begin

P <= A \* B;

end Behavioral;

2.simulation waveform：

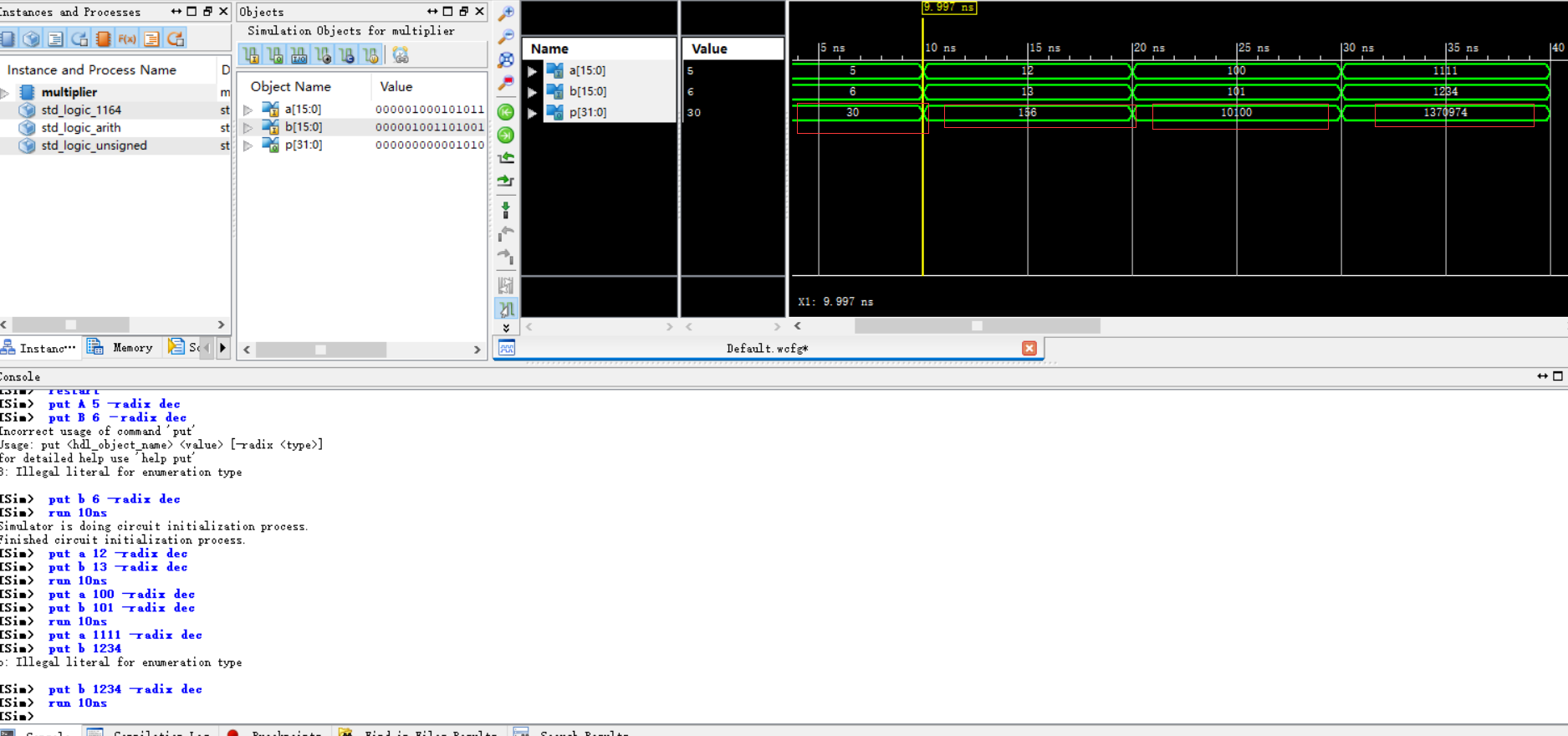
A.

ISim > restart

ISim > put A 5 –radix dec

ISim > put B 6 –radix dec

ISim > run 10ns

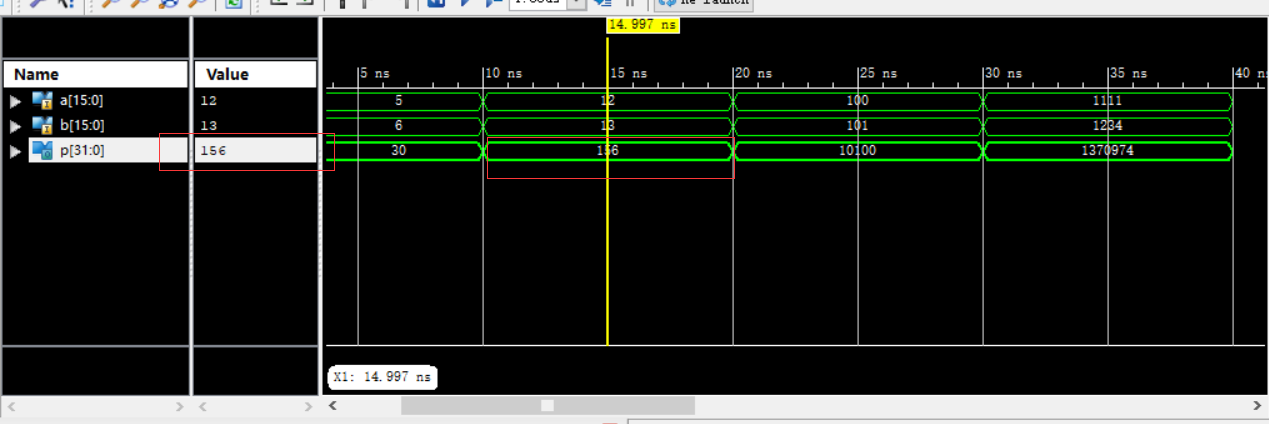


B.

ISim > put A 12 –radix dec

ISim > put B 13 –radix dec

ISim > run 10ns

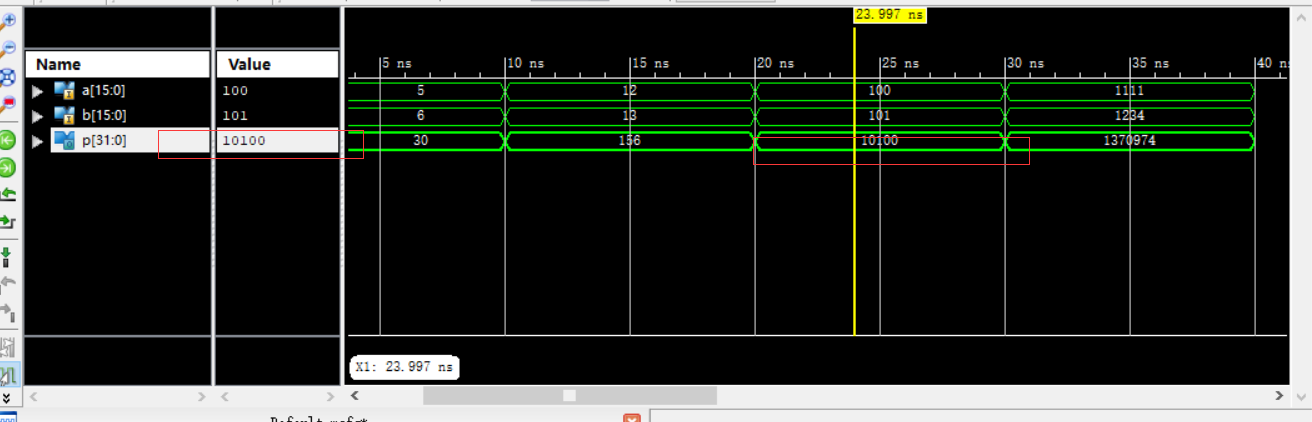


C.

ISim > put A 100 –radix dec

ISim > put B 101 –radix dec

ISim > run 10ns



D.

ISim > put A 1111 –radix dec

ISim > put B 1234 –radix dec

ISim > run 10ns

ISim >

