**COA**

**MINI PROJECT**

**By: G.Saiprasad 114A1012**

**Shweta Shekhar 116A1072**

**Sunil Pillai 116A1058**

**Aim:** To implement 64 bit addition and subtraction using VHDL

**Theory:**

VHDL is a Hardware Description Language used for modelling digital systems made of interconnection of components. The complexity of the digital system being modelled may vary from that of simple gate to a complete electronic circuit/system. VHDL is an acronym for VHSIC Hardware Description Language and VHSIC is an acronym for Very High Speed Integrated Circuits. This language was first introduced in 1981 for the Department of Defence (DoD) under the VHSIC program.

**Describing a Design**

In VHDL an entity is used to describe a hardware module. An entity can be described using,

* Entity declaration
* Architecture
* Configuration
* Package declaration
* Package body

**Entity Declaration**

It defines the names, input output signals and modes of a hardware module.

**Syntax** −

An entity declaration should start with ‘entity’ and end with ‘end’ keywords. The direction will be input, output or inout.

|  |  |
| --- | --- |
| In | Port can be read |
| Out | Port can be written |
| Inout | Port can be read and written |
| Buffer | Port can be read and written, it can have only one source. |

**Architecture** −

Architecture can be described using structural, dataflow, behavioural or mixed style.

**Syntax** −

architecture architecture\_name of entity\_name

architecture\_declarative\_part;

begin

Statements;

end architecture\_name;

Here, we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the ‘begin’ and ‘end’ keyword. Architecture declarative part may contain variables, constants, or component declaration.

In our program **library** **IEEE** is imported under which the **package LOGIC\_1164** **&** **LOGIC\_UNSIGNED** is used.

An **entity bitadd** is created in which **two ports** of **in type a, b** and **two ports** of **out type c, d** are set. **Input** ports **a** & **b** are of **Vector** type each having 65 bits **(64 downto 0)**.

Similarly, **Output** ports **c** & **d** are of **Vector** type automatically having 65 bits each.

An **architecture add** is created for the **entity add** which does the following operations:

* Addition (**a + b)** is carried out and result is stored in bit **c**.
* Subtraction (**a – b)** is carried out and result is stored in bit **d**.

**Program:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity bitadd is

Port (

a : in STD\_LOGIC\_VECTOR (64 downto 0) := "00000000000000000000000000000000000000000000000000000000000000000";

b : in STD\_LOGIC\_VECTOR (64 downto 0) := "00000000000000000000000000000000000000000000000000000000000000000";

c: out STD\_LOGIC\_VECTOR (64 downto 0) ;

d: out STD\_LOGIC\_VECTOR (64 downto 0) );

end bitadd;

architecture add of bitadd is

begin

c <= a + b;

d <= a − b;

end add;

**Example:** To demonstrate (1234567898765432109)10 + (9876543210123456789)10

& (1234567898765432109)10 −(9876543210123456789)10

→

(1234567898765432109)10

= (1000100100010000100001111011010000001000000111001010100101101)2

(9876543210123456789)10

= (1000100100010000100001111011100010110000001101000111000100010101)2

(1000100100010000100001111011010000001000000111001010100101101)2

+ (1000100100010000100001111011100010110000001101000111000100010101)2

= (01001101000110010100110001010111100110001001110000000011001000010)2

= (11111111108888888898)10

(1000100100010000100001111011010000001000000111001010100101101)2

− (1000100100010000100001111011100010110000001101000111000100010101)2

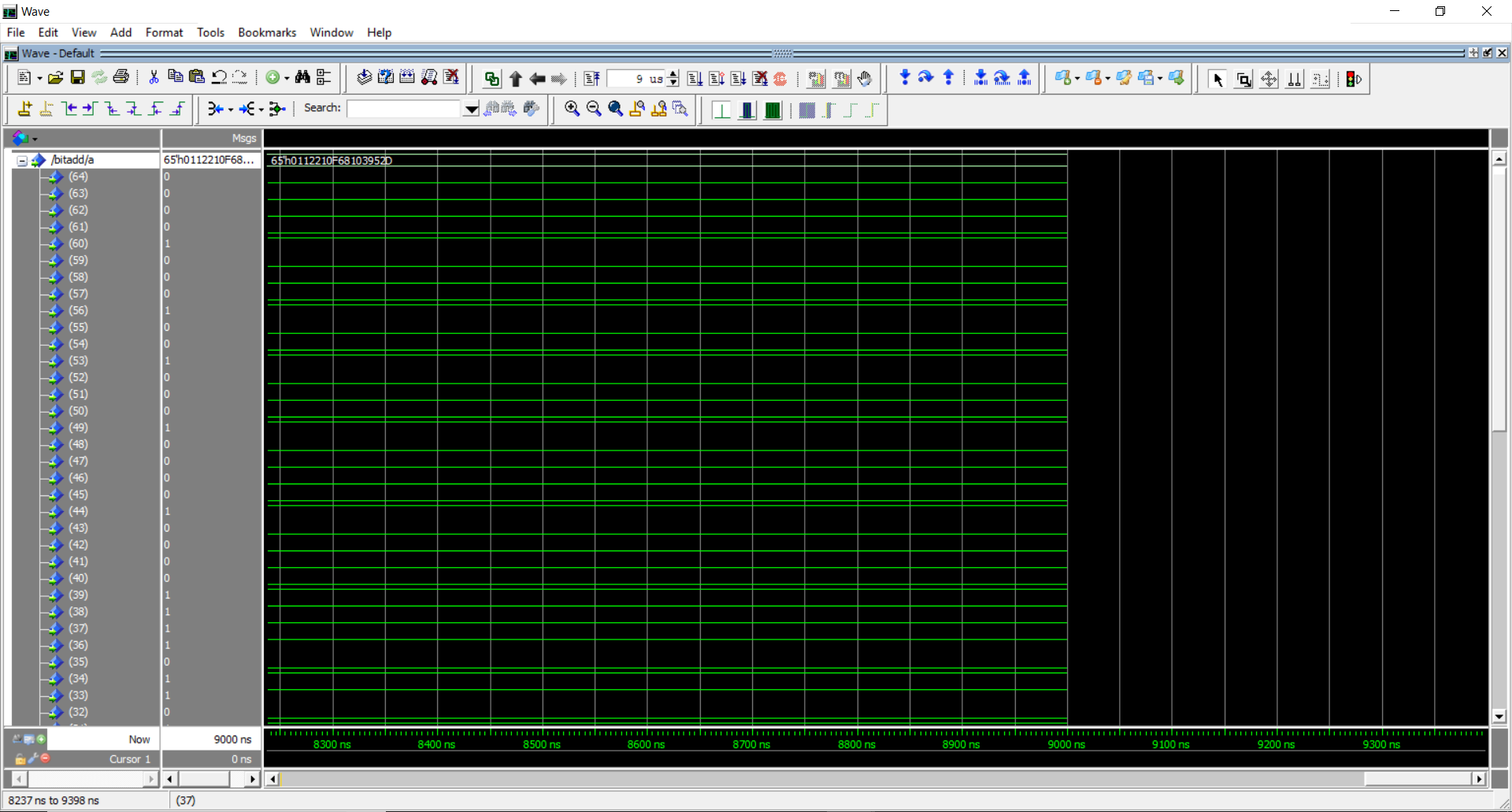
= (11000100000010001100010010011110111010000110011110010010000011000)2

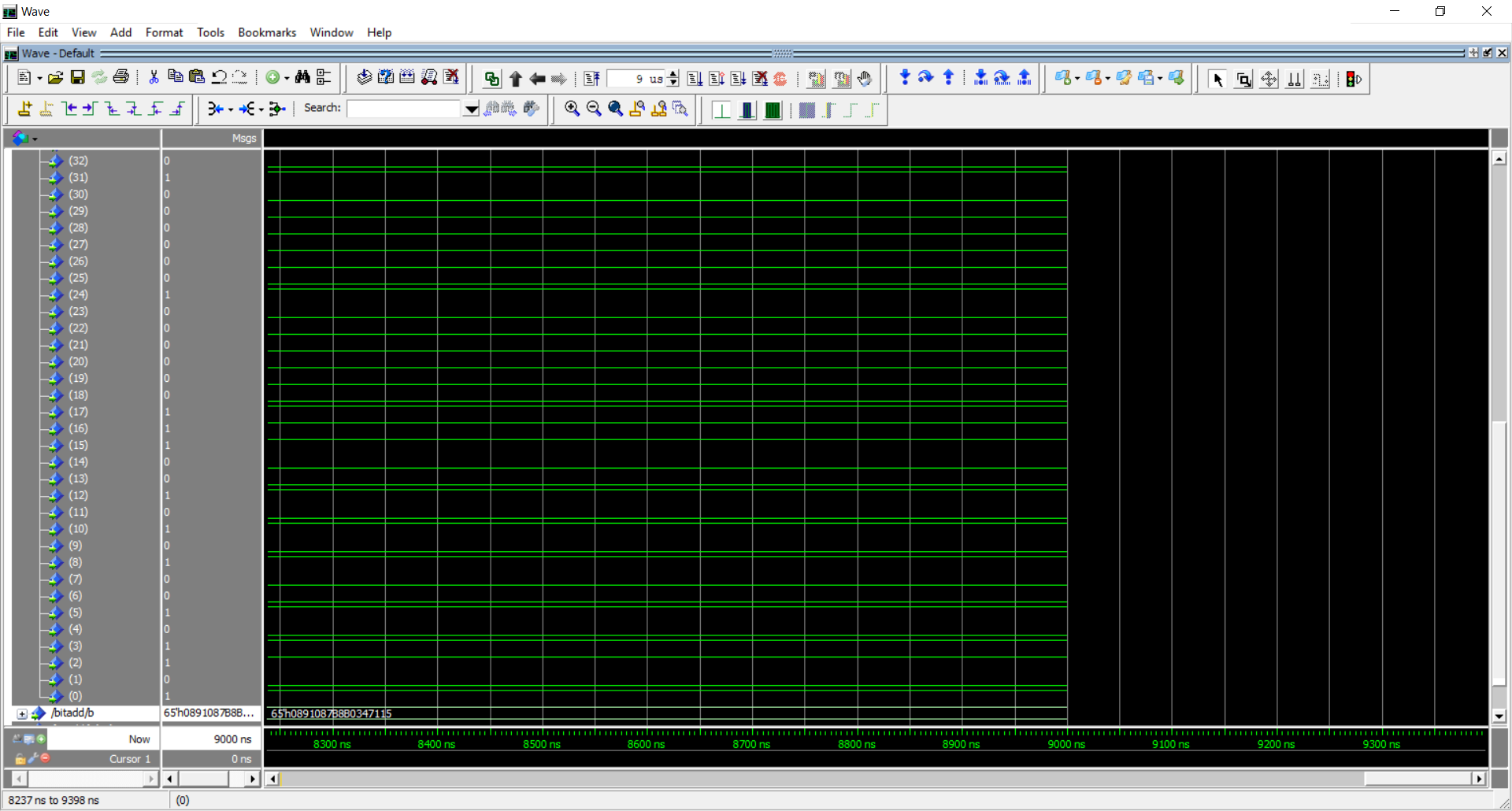
(In Signed 2’s Complement)

= (-8641975311358024680)

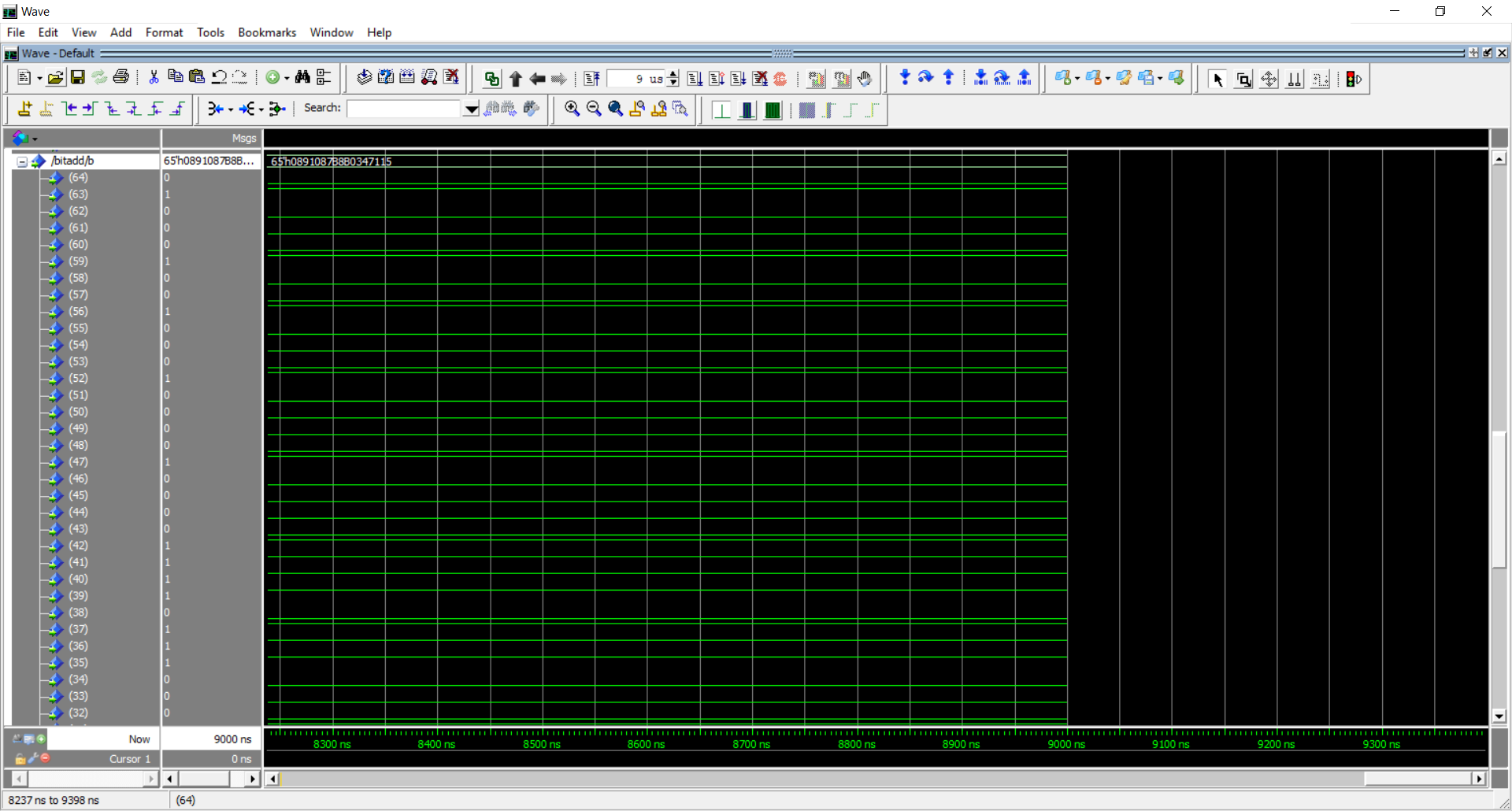
**Output:**

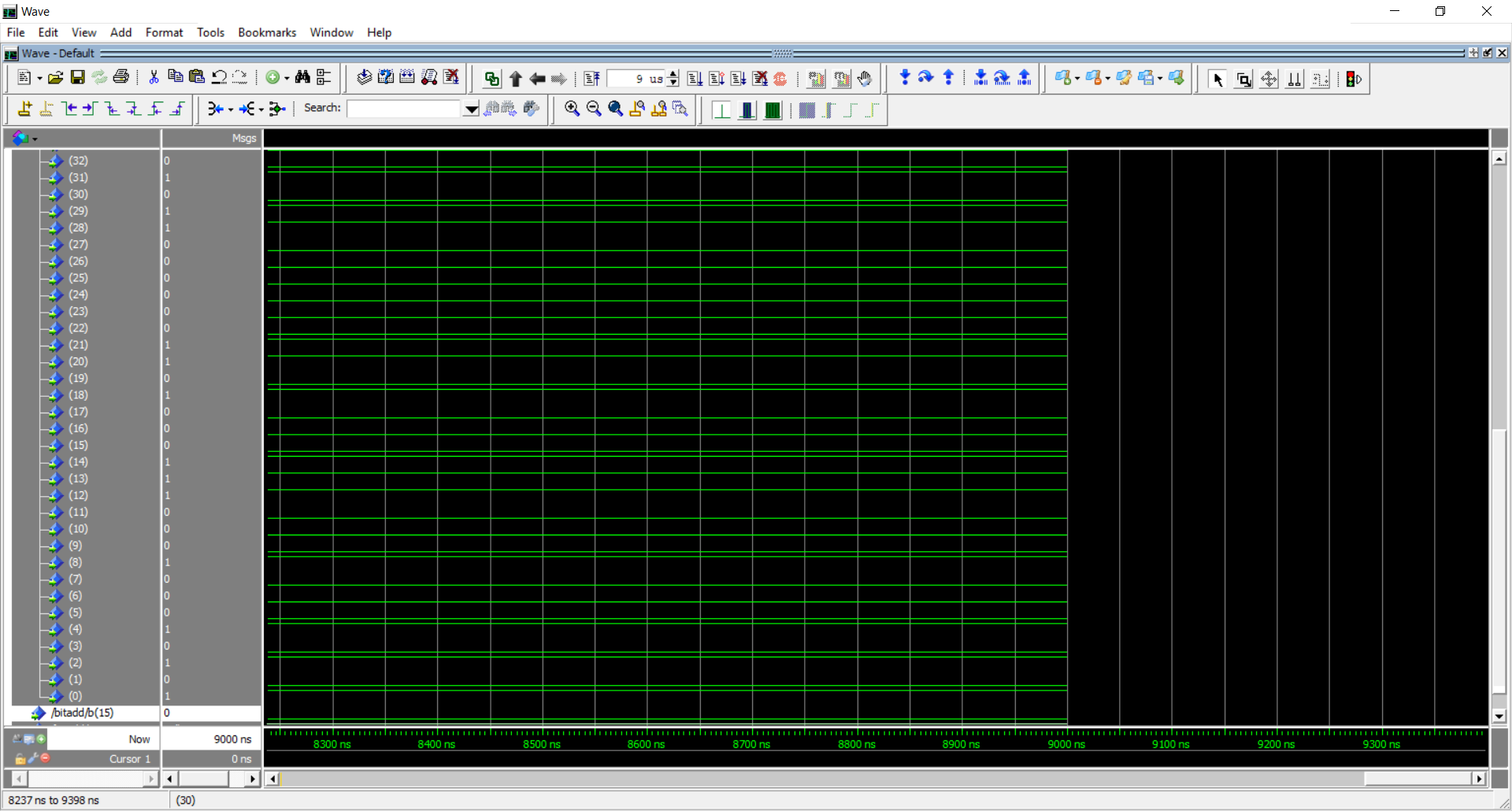
**Input bit a:**



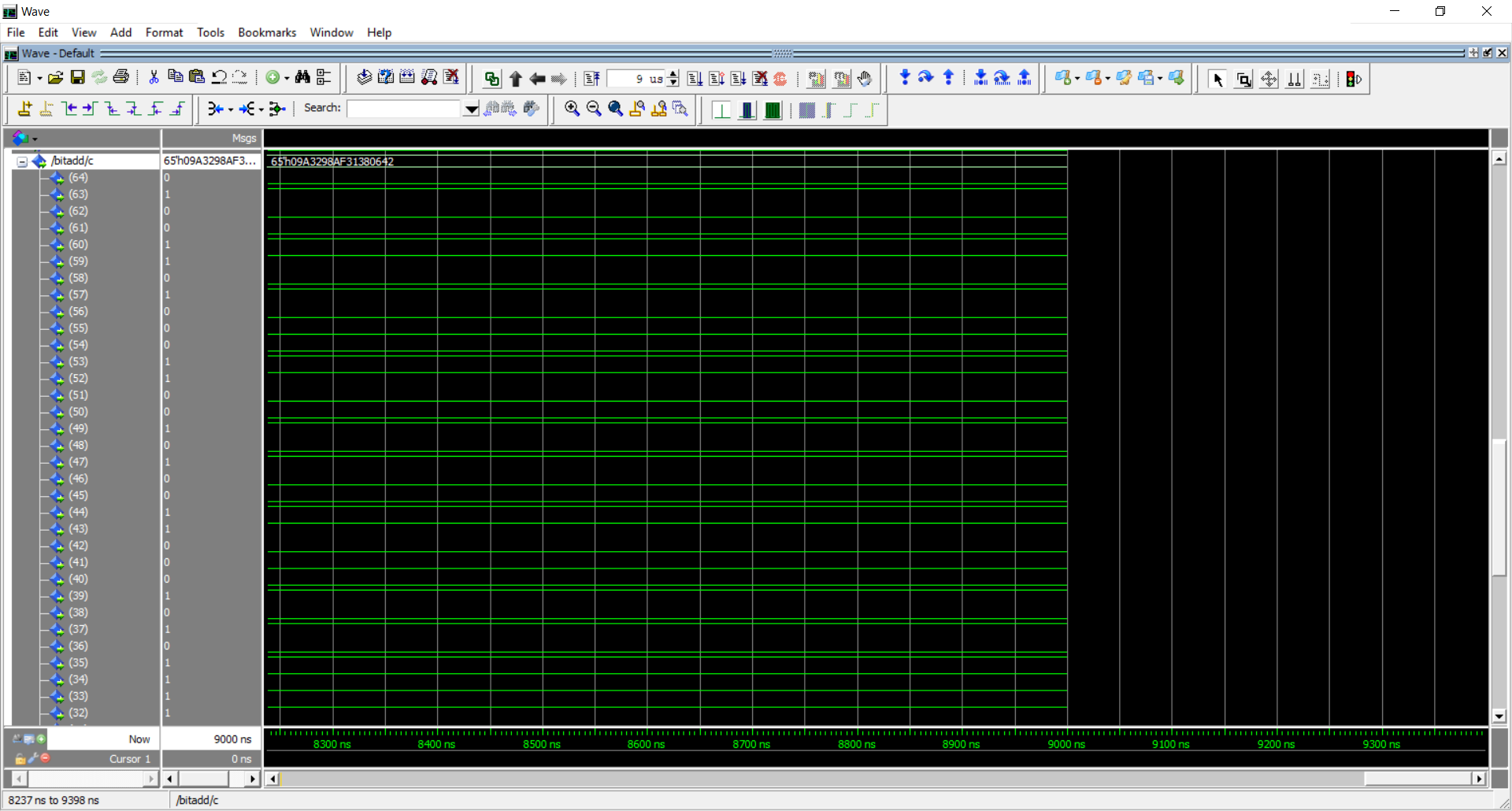


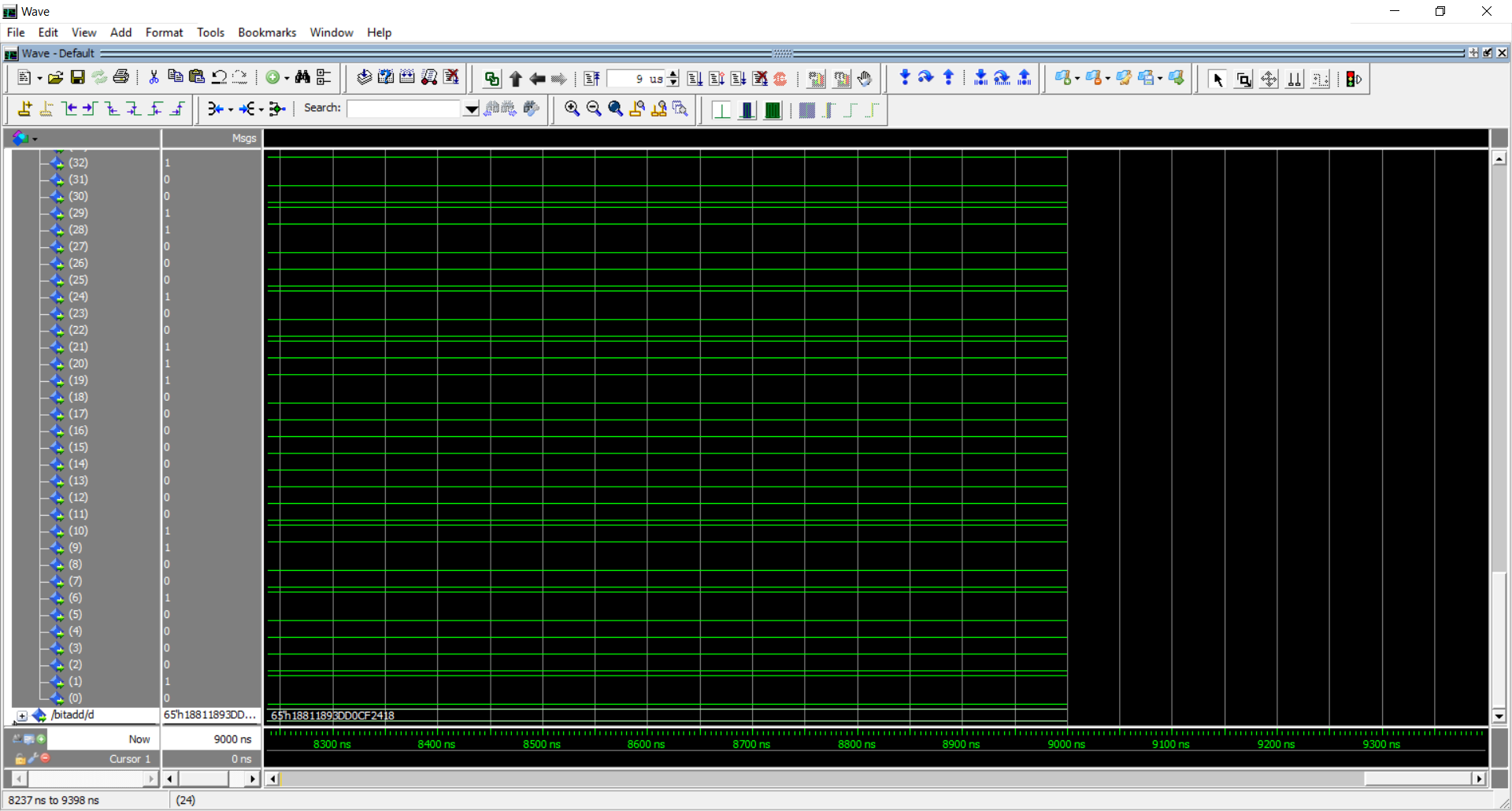
**Input bit b:**





**Output bit c (a+b):**





**Output bit d (a-b):**

