**EXPERIMENT 1**

**AIM**: To simulate basic Logic gates using Model Sim Software.

**Theory**:To simulate basic gates first we need to know about the software and logic gate boolean tables so that we can check if there are any errors with our simulations or not.

There are basicly 7 logic gates :

1. AND
2. OR
3. NAND
4. NOR
5. NOT
6. XOR
7. XNOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **NO.** | **TYPE** | **SYMBOL** | **BOOLEAN ALGEBRA** | **TRUTH TABLE** |
| **1.** | AND | AND symbol | A.B | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | |
| **2.** | OR | OR symbol | A+B | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 1 | |
| **3.** | NOT | NOT symbol |  | |  |  | | --- | --- | | A | Q | | 0 | 1 | | 1 | 0 | |
| **4.** | NOR | NOR symbol |  | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 1 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 0 | |
| **5.** | NAND | NAND symbol |  | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 1 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 0 | |
| **6.** | XOR | XOR symbol |  | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 0 | |
| **7.** | XNOR | XNOR symbol |  | |  |  |  | | --- | --- | --- | | A | B | Q | | 0 | 0 | 1 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | |

So above is the table for all the logic gates.

All the gates simulated below are made using universal logic gate NAND.

**VHDL CODE:**

**//CODE FOR AND GATE**

ENTITY AND\_GATE IS

port (A,B:in std\_logic; Q:out std\_logic);

END ENTITY AND\_GATE;

--

ARCHITECTURE AND\_GATE\_DATAFLOW OF AND\_GATE IS

SIGNAL temp1:std\_logic;

BEGIN

temp1<= A nand B;

Q<=not temp1;

END ARCHITECTURE AND\_GATE\_DATAFLOW;

**//CODE FOR OR GATE**

ENTITY OR\_GATE IS

port (A,B:in std\_logic; Q:out std\_logic);

END ENTITY OR\_GATE;

--

ARCHITECTURE OR\_GATE\_DATAFLOW OF OR\_GATE IS

SIGNAL Temp1,Temp2:std\_logic;

BEGIN

Temp1<=A nand A;

Temp2<=B nand B;

Q<=Temp1 nand Temp2;

END ARCHITECTURE OR\_GATE\_DATAFLOW;

**//CODE FOR NOT GATE**

ENTITY NOT\_GATE IS

port(A:in std\_logic;Q:out Std\_logic);

END ENTITY NOT\_GATE;

--

ARCHITECTURE NOT\_GATE\_DATAFLOW OF NOT\_GATE IS

BEGIN

Q<=A nand A;

END ARCHITECTURE NOT\_GATE\_DATAFLOW;

**//CODE FOR NAND GATE**

ENTITY NAND\_GATE IS

port (A,B:in std\_logic; Q:out std\_logic);

END ENTITY NAND\_GATE;

--

ARCHITECTURE NAND\_GATE\_DATAFLOW OF NAND\_GATE IS

BEGIN

Q<= A nand B;

END ARCHITECTURE NAND\_GATE\_DATAFLOW;

**//CODE FOR NOR GATE**

ENTITY NOR\_GATE IS

port (A,B:in std\_logic; Q:out std\_logic);

END ENTITY NOR\_GATE;

--

ARCHITECTURE NOR\_GATE\_DATAFLOW OF NOR\_GATE IS

SIGNAL Temp1,Temp2,Temp3,Temp4:std\_logic;

BEGIN

Temp1<=A nand A;

Temp2<=B nand B;

Temp3<=Temp1 nand Temp2;

Q<= Temp3 nand Temp3;

END ARCHITECTURE NOR\_GATE\_DATAFLOW;

**//CODE FOR XOR GATE**

ENTITY XOR\_GATE IS

port(A,B:in std\_logic;Q:out std\_logic);

END ENTITY XOR\_GATE;

--

ARCHITECTURE XOR\_GATE\_DATAFLOW OF XOR\_GATE IS

SIGNAL Temp1,Temp2,Temp3:std\_logic;

BEGIN

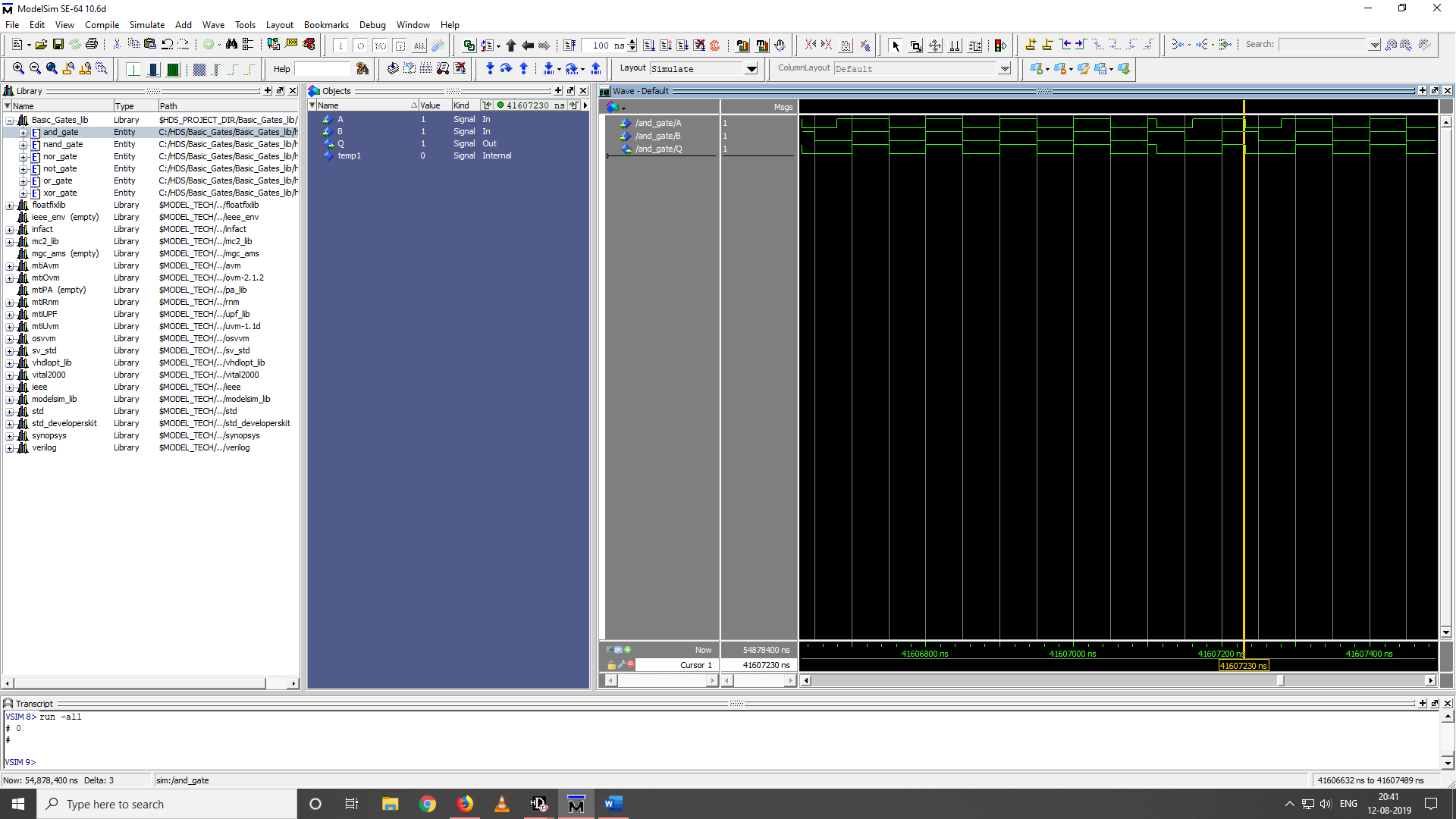
Temp1<=A nand B;

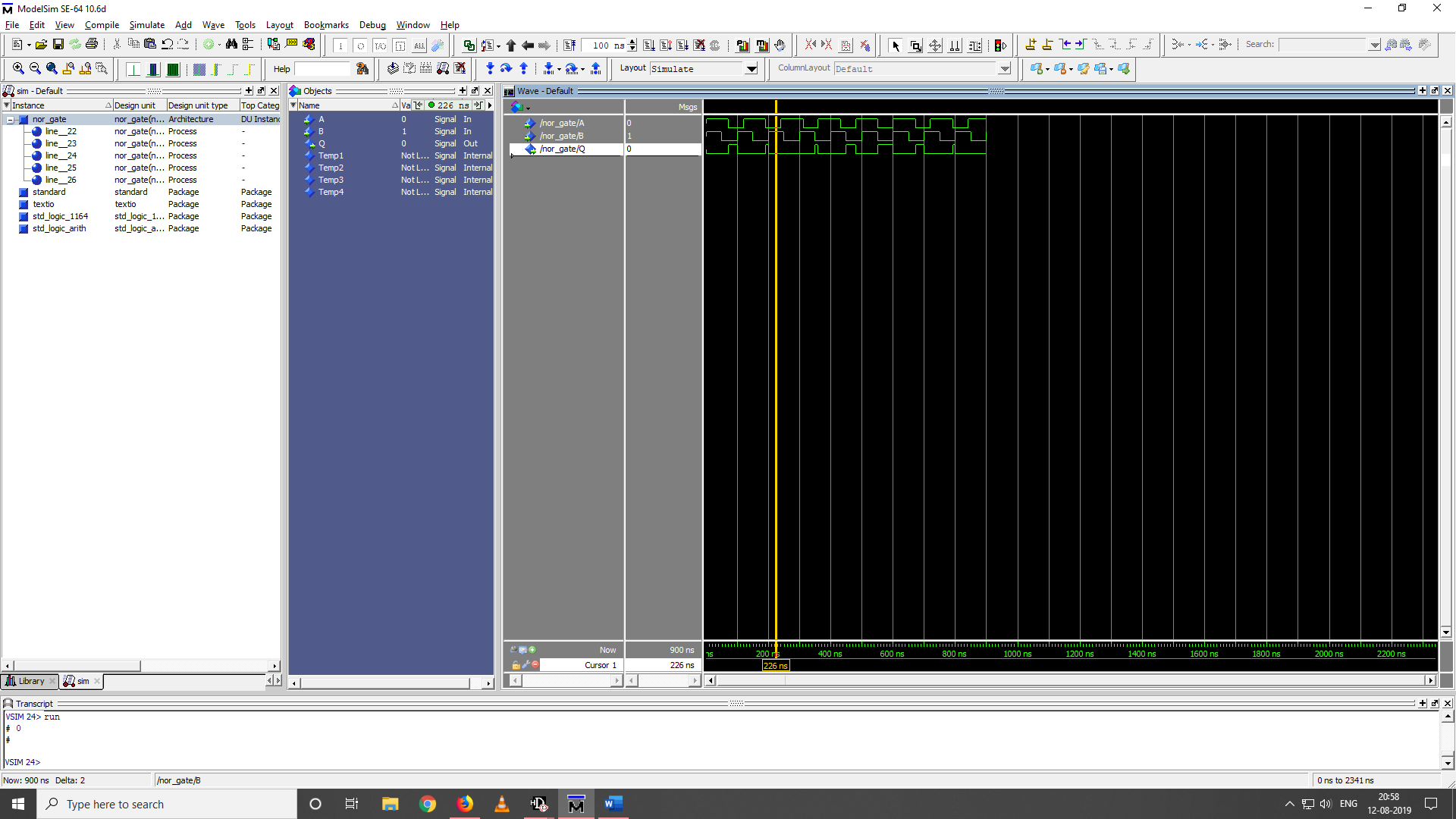
Temp2<=A nand Temp1;

Temp3<=B nand Temp2;

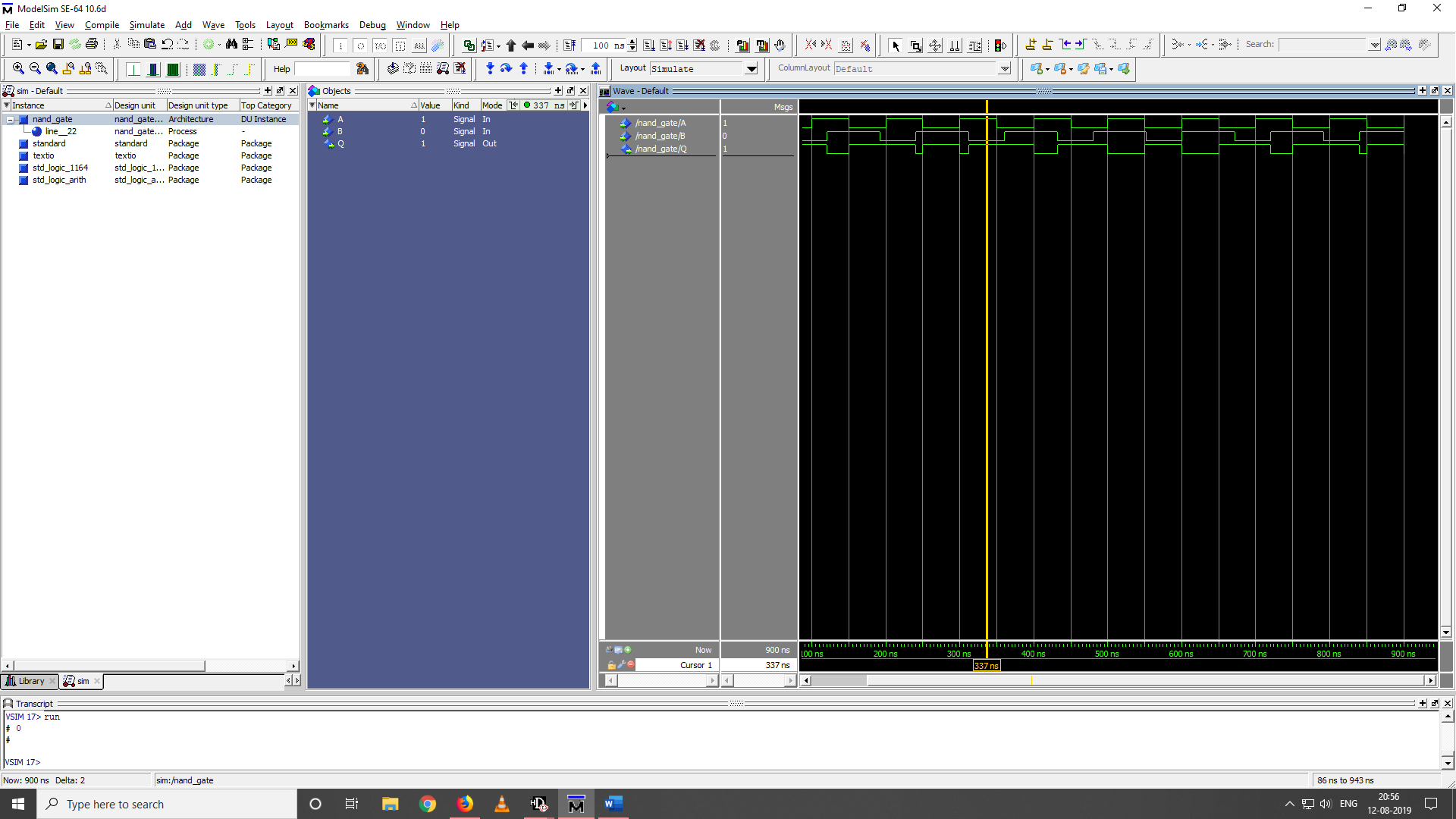
Q<=Temp2 nand Temp3;

END ARCHITECTURE XOR\_GATE\_DATAFLOW;

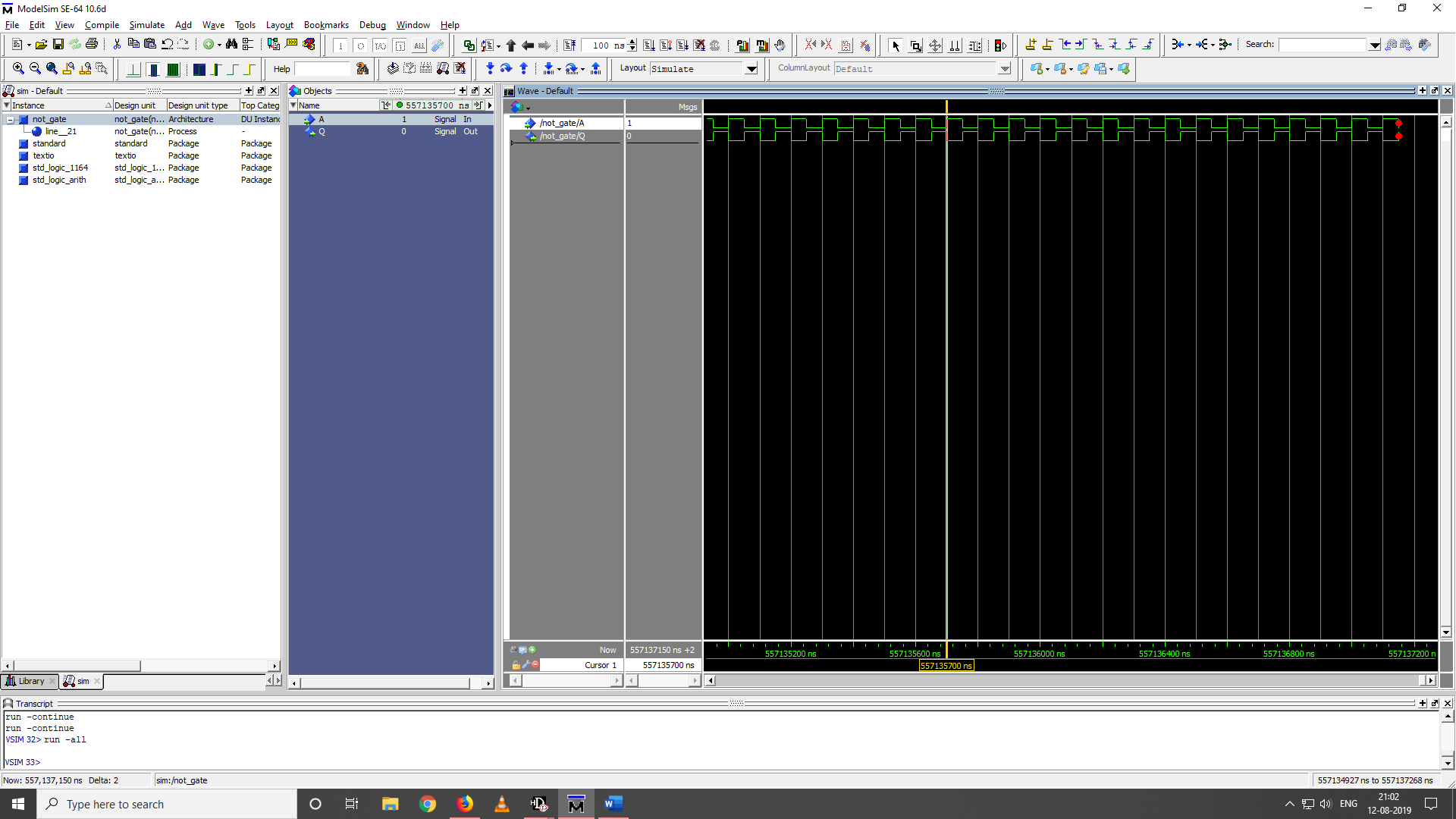
**OUTPUTS:**

**AND GATE**

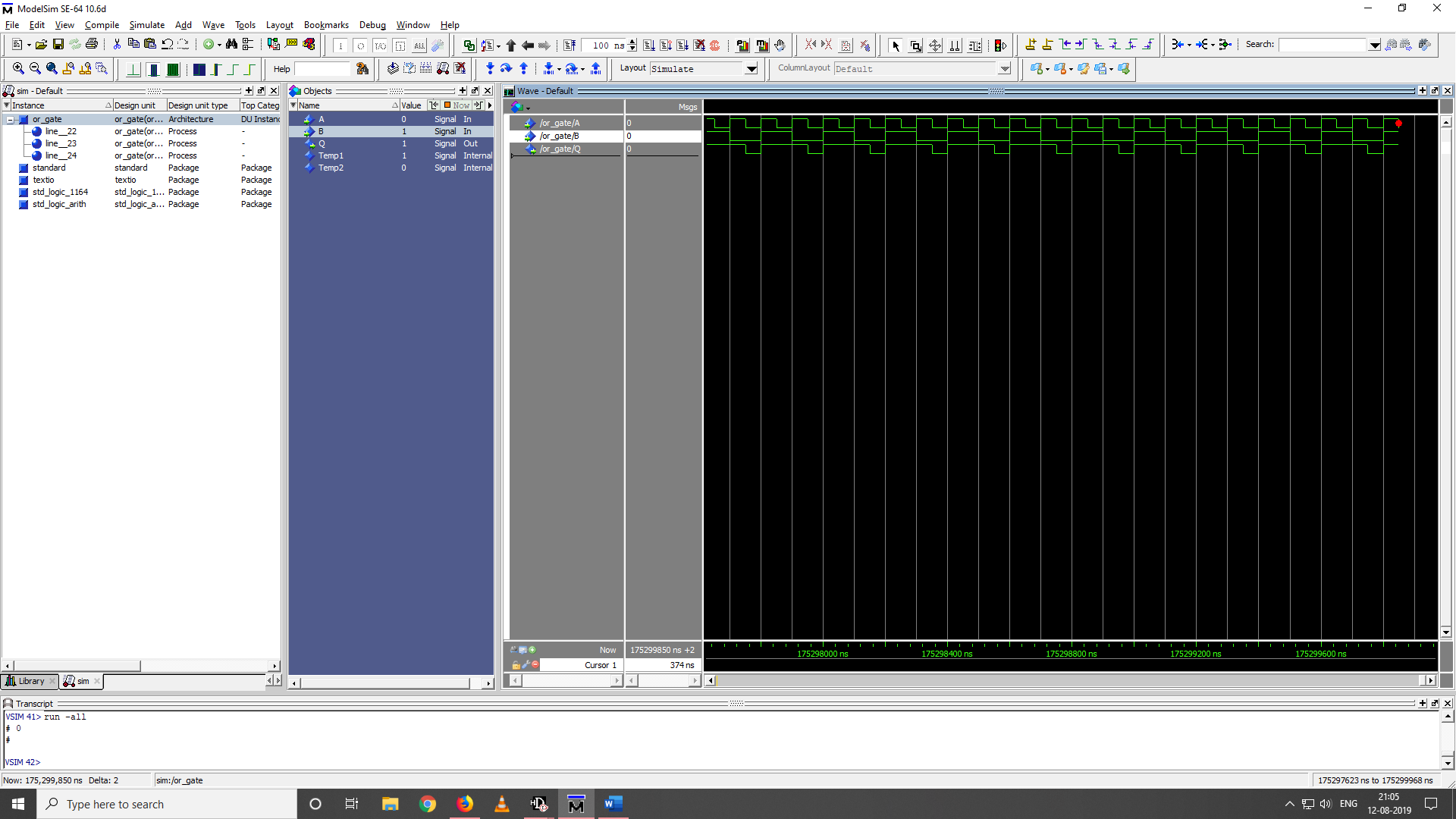
**NOR GATE**



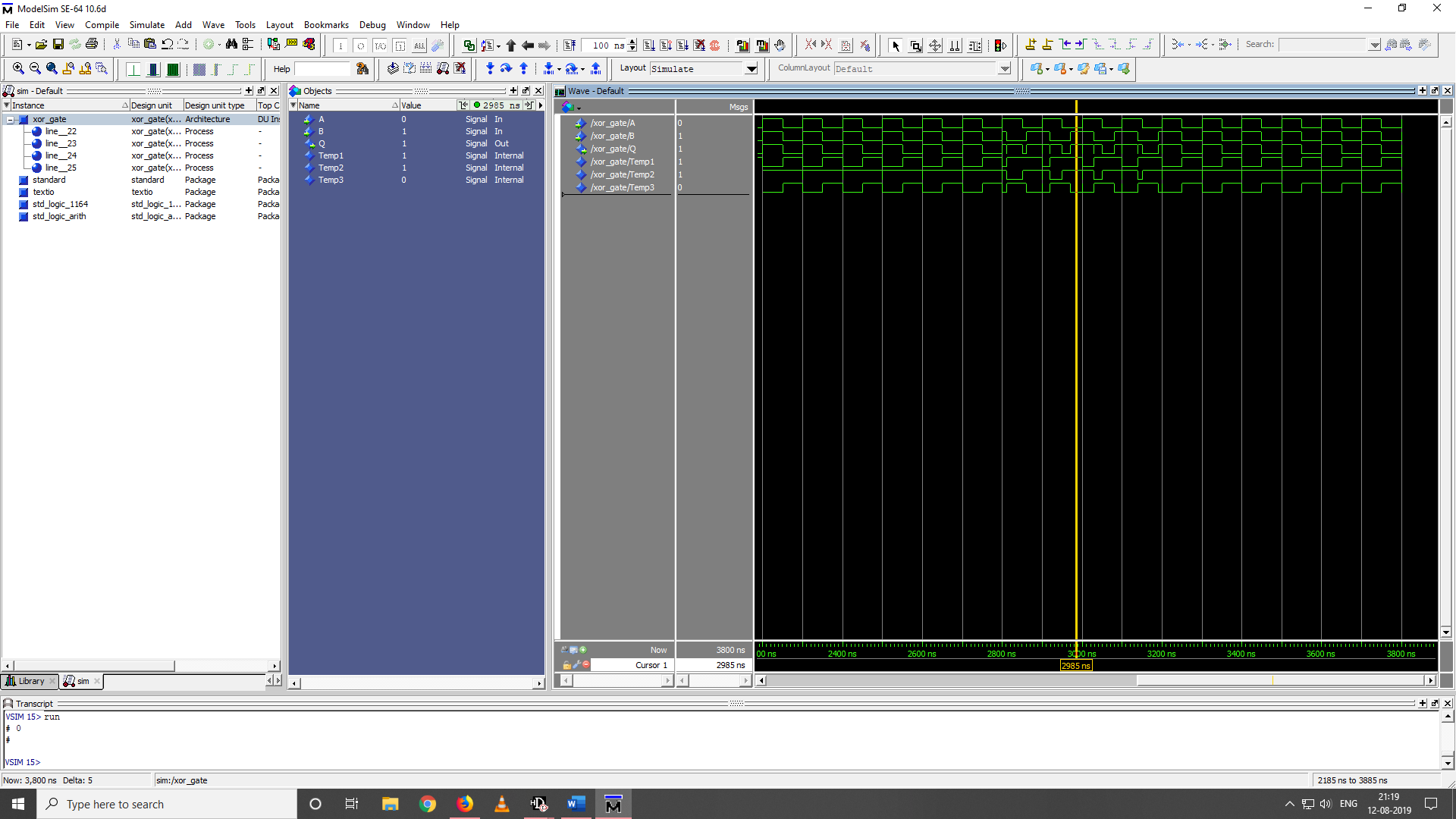
**NAND GATE**



**NOT GATE**



**OR GATE**

**XOR GATE**

**Result: All the logic gates are simulated using MODELSIM Software.**