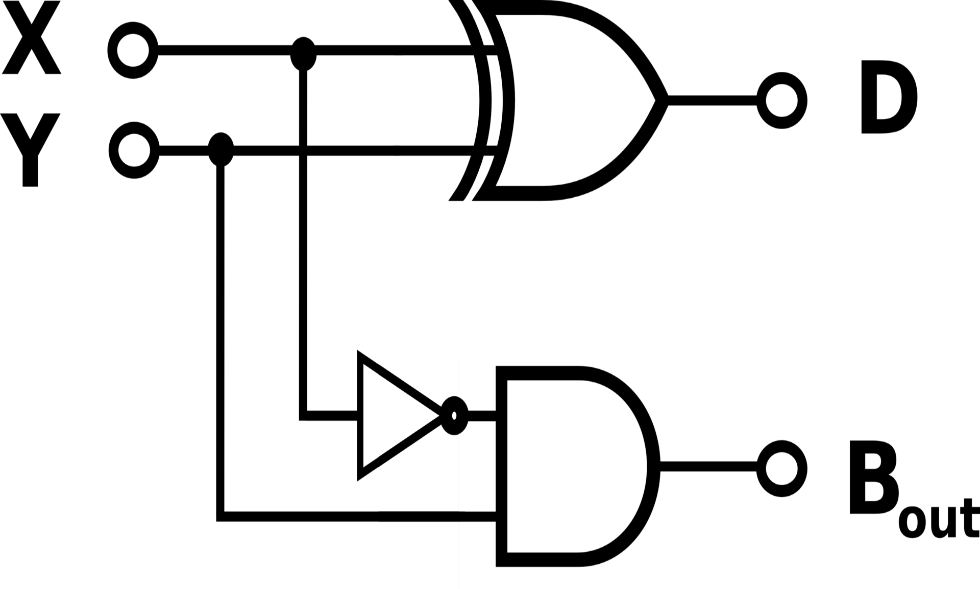
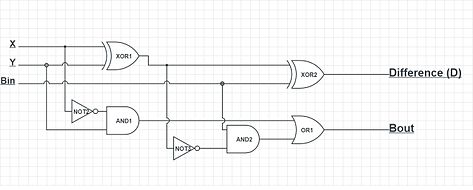
EXPERIMENT 2

**AIM:** To design and simulate Half Subtractor and full Subtractor using VHDL in Modelism.

**THEORY:** The **half subtractor** is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, the minuend X {\displaystyle X} and subtrahend Y {\displaystyle Y} and two outputs the difference D {\displaystyle D} and borrow out B out {\displaystyle B\_{\text{out}}} . his is an important distinction to make since subtraction itself is not commutative.

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | **Outputs** | |
| **X** | **Y** | **D** | **BOUT** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **Y** | **BIN** | **D** | **BOUT** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

The **full subtractor** is a combinational circuit which is used to perform subtraction of three input bits: the minuend XX {\displaystyle X} , subtrahend Y Y {\displaystyle Y} , and borrow in BINB in {\displaystyle B\_{\text{in}}} . The full subtractor generates two output bits: the difference D {\displaystyle D} and borrow out BOUT B out {\displaystyle B\_{\text{out}}} .BIN B in {\displaystyle B\_{\text{in}}} is set when the previous digit is borrowed from X {\displaystyle X} X.

**VHDL CODE: FOR FULL SUBTRACTOR**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY FULL\_SUBTRACTOR IS

port(A,B,BORROW\_IN: in std\_logic ; DIFF,BORROW: out std\_logic);

END ENTITY FULL\_SUBTRACTOR;

--

ARCHITECTURE FULL\_SUBTRACTOR\_DATAFLOW OF FULL\_SUBTRACTOR IS

SIGNAL ABAR: std\_logic;

BEGIN

DIFF<= A XOR B XOR BORROW\_IN;

ABAR<= not A;

BORROW<= (ABAR AND BORROW\_IN)OR(ABAR AND B)OR(B AND BORROW\_IN);

END ARCHITECTURE FULL\_SUBTRACTOR\_DATAFLOW;

ARCHITECTURE FULL\_SUBTRACTOR\_BEHAV OF FULL\_SUBTRACTOR IS

SIGNAL S:std\_logic\_vector(2 downto 0);

BEGIN

S(2)<=A;S(1)<=B;S(0)<=BORROW\_IN;

PROCESS(S)

BEGIN

if (S = "001" or S = "010" or S = "111") then

DIFF<='1'; BORROW<='1';

else if (S = "011") then

DIFF<='0'; BORROW<='1';

else if (S = "100") then

DIFF<='1'; BORROW<='0';

else

DIFF<='0'; BORROW<='0';

end if;

end if;

end if;

END PROCESS;

END ARCHITECTURE FULL\_SUBTRACTOR\_BEHAV;

ARCHITECTURE FULL\_SUBTRACTOR\_STRUCTURAL OF FULL\_SUBTRACTOR IS

component AND\_GATE

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

end component;

component XOR\_GATE

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

end component;

component OR\_GATE

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

end component;

component NOT\_GATE

port(A:in std\_logic;Z: out std\_logic);

end component;

SIGNAL var: std\_logic\_vector(4 downto 0);

BEGIN

X1: XOR\_GATE port map (A,B,var(0));

X2: XOR\_GATE port map (BORROW\_IN,var(0),DIFF);

N1: NOT\_GATE port map (var(0),var(1));

A1: AND\_GATE port map (var(1),BORROW\_IN,var(2));

N2: NOT\_GATE port map (A,var(3));

A2: AND\_GATE port map (B,var(3),var(4));

O1: OR\_GATE port map (var(2),var(4),BORROW);

END ARCHITECTURE FULL\_SUBTRACTOR\_STRUCTURAL;

**FOR HALF SUBTRACTOR**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY HALF\_SUBTRACTOR IS

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

END ENTITY HALF\_SUBTRACTOR;

--

ARCHITECTURE HALF\_SUBTRACTOR\_BEHAV OF HALF\_SUBTRACTOR IS

SIGNAL s:std\_logic\_vector(1 downto 0);

BEGIN

s(1)<=A;

s(0)<=B;

PROCESS(s)

BEGIN

if (S = "00" or S = "11") then

DIFF<='0'; BORROW<='0';

else if (S = "01") then

DIFF<='1'; BORROW<='1';

else

DIFF<='1'; BORROW<='0';

end if;

end if;

END PROCESS;

END ARCHITECTURE HALF\_SUBTRACTOR\_BEHAV;

ARCHITECTURE HALF\_SUBTRACTOR\_DATAFLOW OF HALF\_SUBTRACTOR IS

BEGIN

DIFF<=A XOR B;

BORROW<=( NOT A) AND B;

END ARCHITECTURE HALF\_SUBTRACTOR\_DATAFLOW;

ARCHITECTURE HALF\_SUBTRACTOR\_STRUCTURAL OF HALF\_SUBTRACTOR IS

component AND\_GATE

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

end component;

component XOR\_GATE

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

end component;

component NOT\_GATE

port(A:in std\_logic;Z: out std\_logic);

end component;

SIGNAL ANOT:std\_logic;

BEGIN

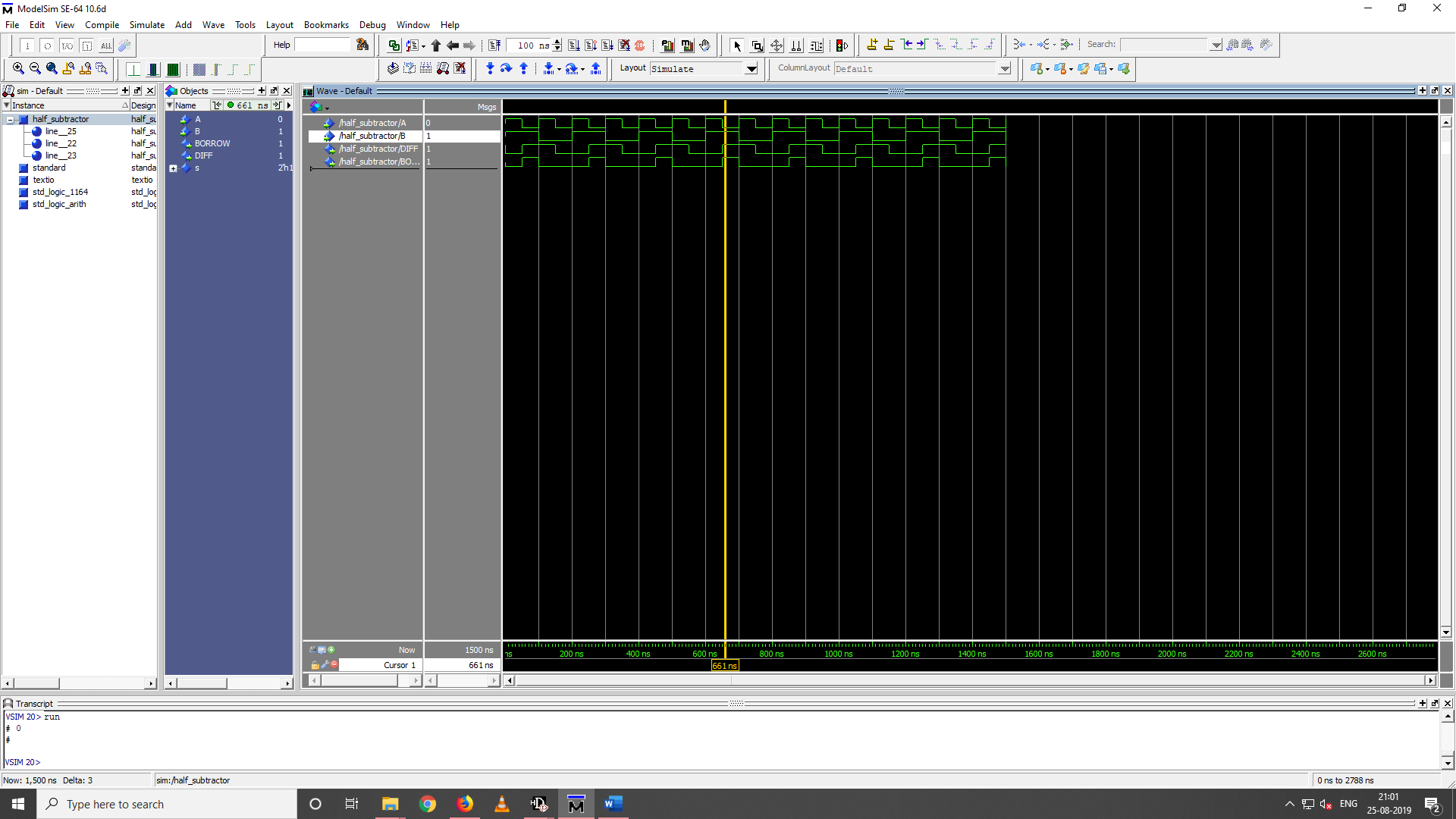
A1: AND\_GATE port map(A,B,DIFF);

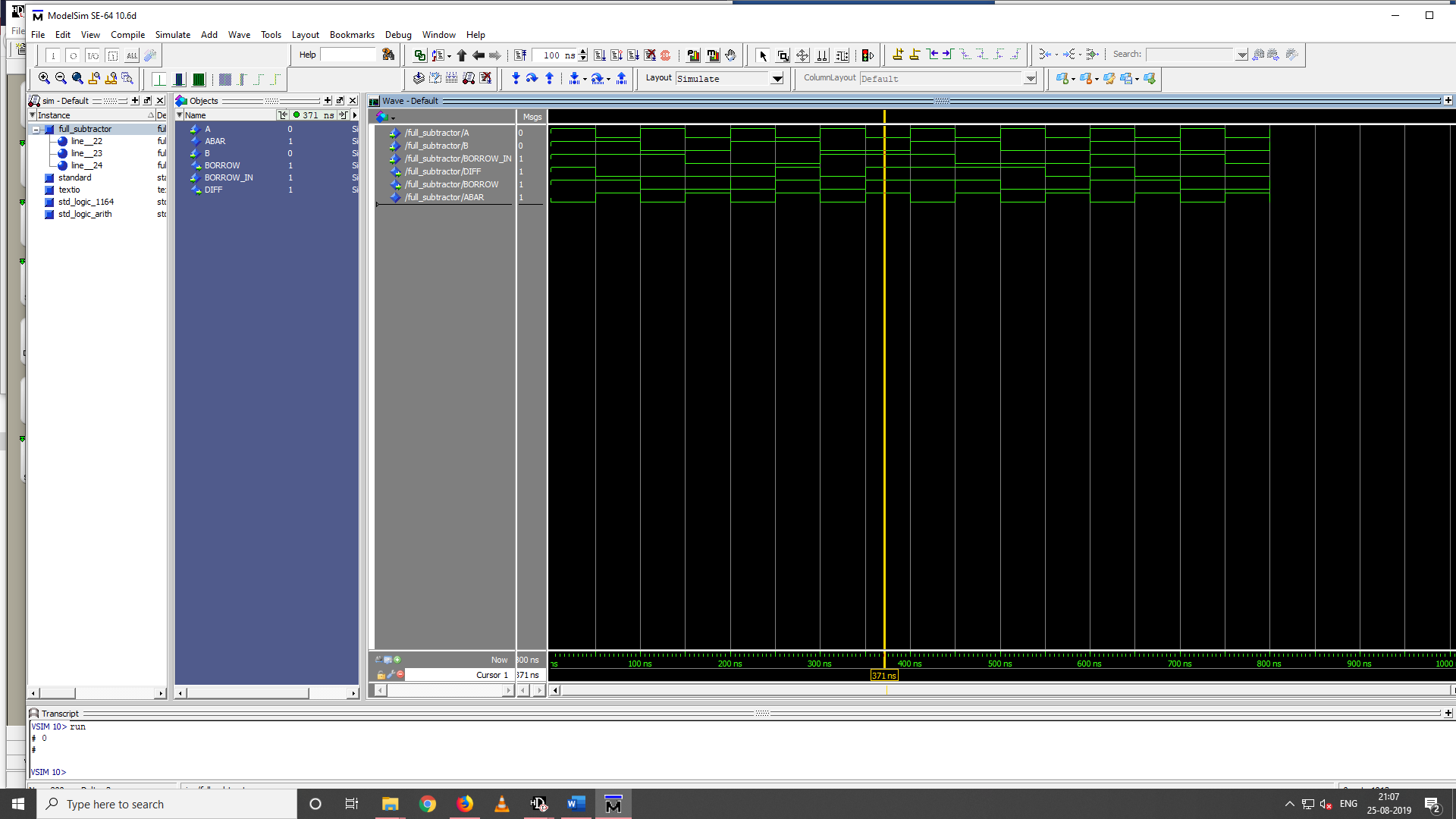
N1: NOT\_GATE port map(A,ANOT);

X1: XOR\_GATE port map(ANOT,B,BORROW);

END ARCHITECTURE HALF\_SUBTRACTOR\_STRUCTURAL;

**OUTPUT:**



**HALF SUBTRACTOR**

**FULL SUBTRACTOR**

**RESULT: Half-subtractor and Full-subtractor are simulated using model sim and HDL**

**Designer.**