# Implementation of DIFFERENT-Gate using VHDL

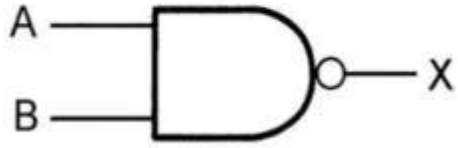
## ▪ OBJECTIVE:

To implement different gate using VHDL and verify its waveform in Xilinx.

## ▪ INTRODUCTION:

### 1. NAND GATE:

$$X = \overline{A.B}$$

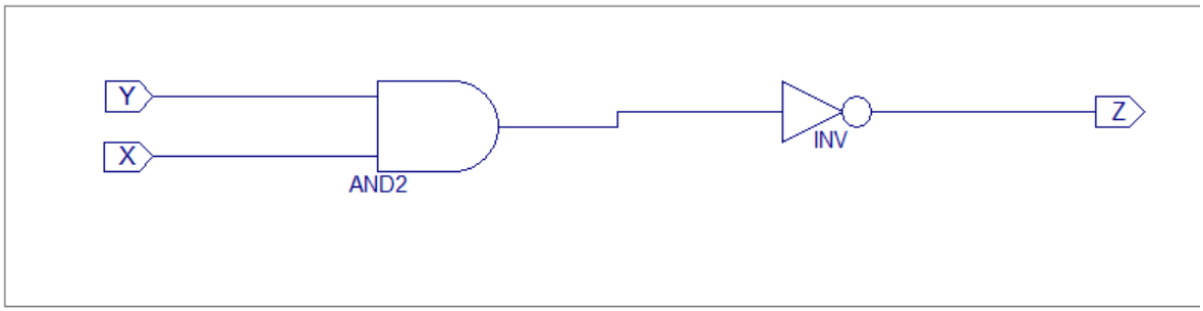


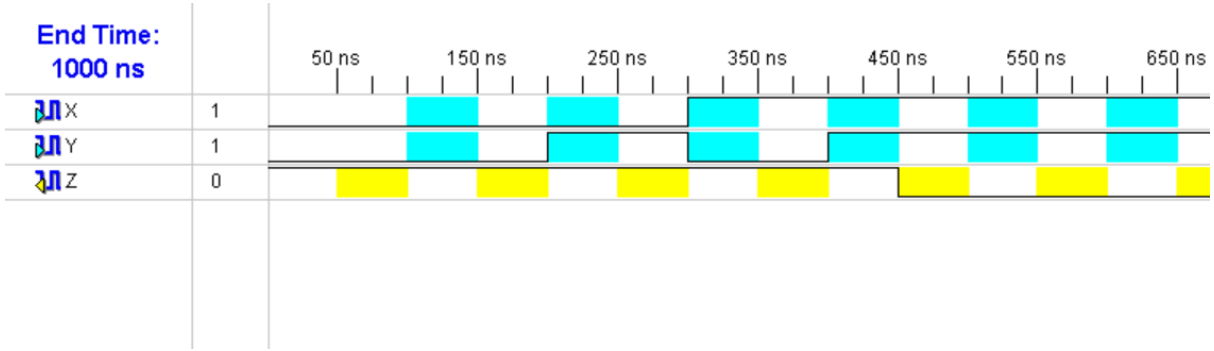

| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

▪ **Source Code:**

```
29
30  entity OHO is
31      Port ( X : in  STD_LOGIC;
32             Y : in  STD_LOGIC;
33             Z : out  STD_LOGIC);
34  end OHO;
35
36  architecture Behavioral of OHO is
37
38  Begin
39     Process (X, Y)
40        Begin
41           If (X = '1' and Y = '1') then
42              Z <= '0';
43           Else
44              Z <= '1';
45        End if;
46     End process;
47
48  End Behavioral;
```

- **Output:**



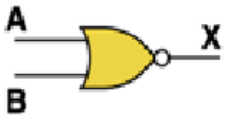




## 2. NOR GATE:

**Boolean Expression**

$X = (A + B)'$

**Logic Diagram Symbol**



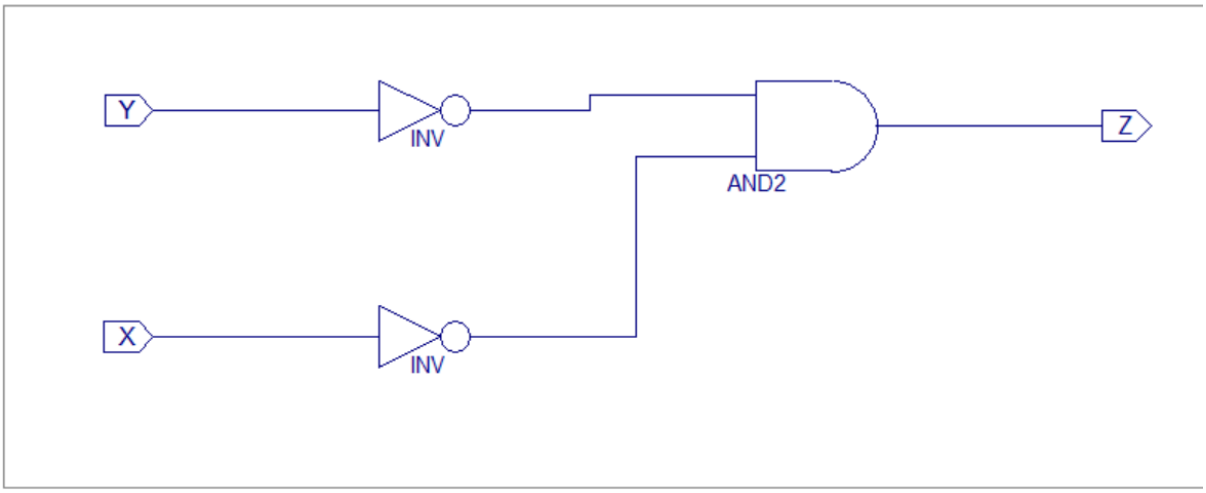

**Truth Table**

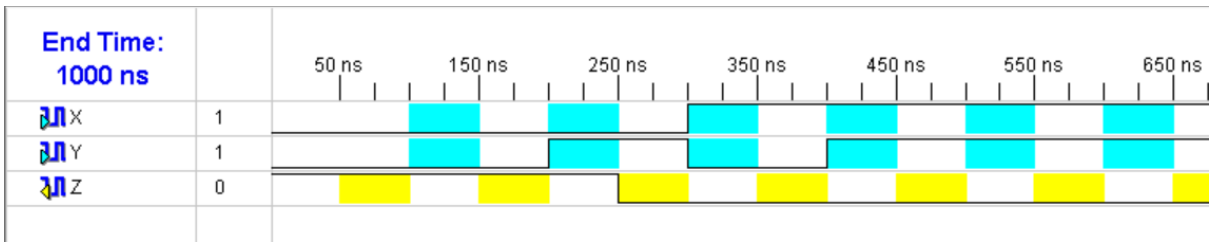
| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

- **Source Code:**

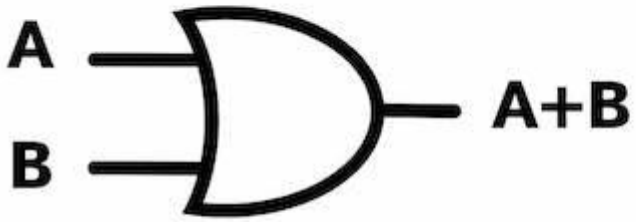
```
29
30  entity OHO is
31      Port ( X : in  STD_LOGIC;
32             Y : in  STD_LOGIC;
33             Z : out  STD_LOGIC);
34  end OHO;
35
36  architecture Behavioral of OHO is
37
38  Begin
39     Process (X, Y)
40        Begin
41           If (X = '0' and Y = '0') then
42              Z <= '1';
43           Else
44              Z <= '0';
45        End if;
46     End process;
47
48  End Behavioral;
```

- **Output:**



## 3. OR GATE:



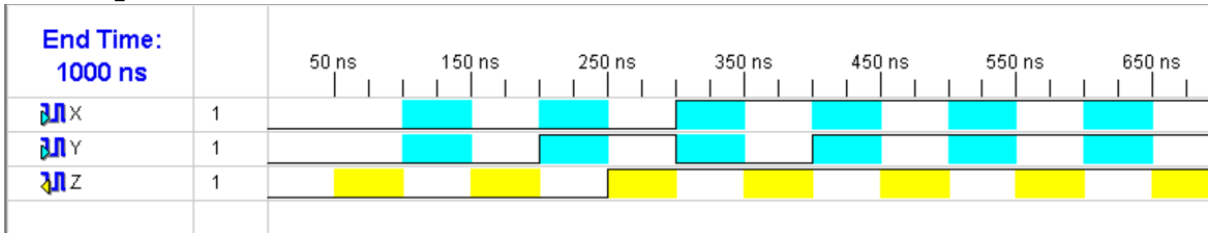

2 input OR gate

| A | B | A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

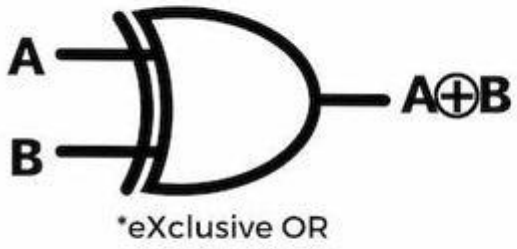
- **Source Code:**

```
29
30  entity OHO is
31      Port ( X : in  STD_LOGIC;
32             Y : in  STD_LOGIC;
33             Z : out  STD_LOGIC);
34  end OHO;
35
36  architecture Behavioral of OHO is
37
38  Begin
39     Process (X, Y)
40        Begin
41           If (X = '0' and Y = '0') then
42              Z <= '0';
43           Else
44              Z <= '1';
45        End if;
46     End process;
47
48  End Behavioral;
```

- **Output:**



## 4. X-OR GATE:



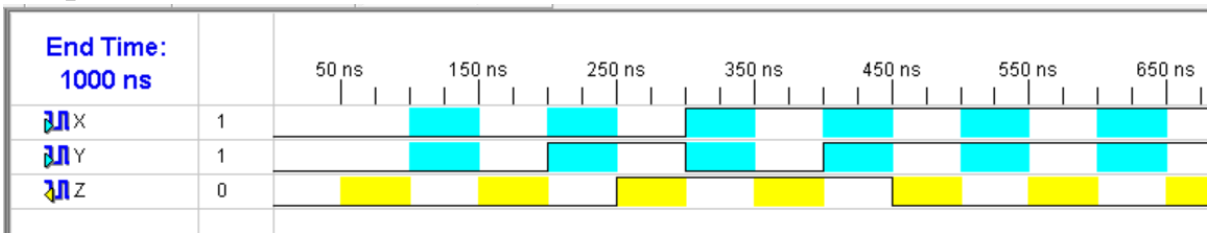

2 input XOR gate

| A | B | A⊕B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

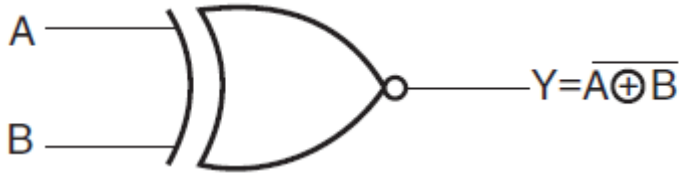
- **Source Code:**

```
29
30  entity OHO is
31      Port ( X : in  STD_LOGIC;
32             Y : in  STD_LOGIC;
33             Z : out  STD_LOGIC);
34  end OHO;
35
36  architecture Behavioral of OHO is
37
38  Begin
39     Process (X, Y)
40        Begin
41           If (X = '1' and Y = '1') then
42              Z <= '0';
43           Elsif (X = '0' and Y = '0') then
44              Z <= '0';
45           Else
46              Z <= '1';
47        End if;
48     End process;
```

- **Output:**



## 5. X-NOR GATE:



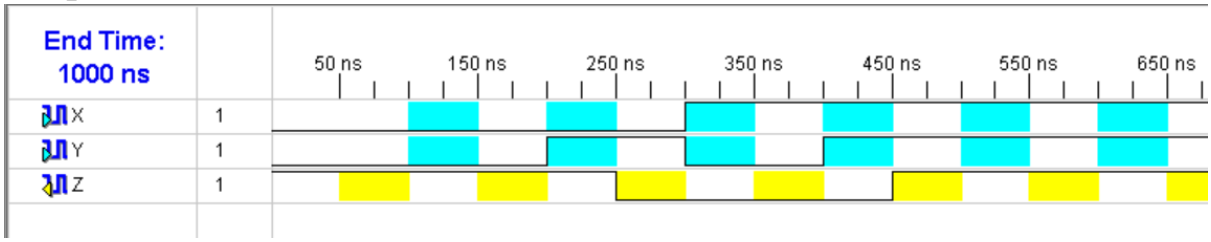

$$Y = (\overline{A \oplus B}) = (A.B + \overline{A}.\overline{B})$$

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- **Source Code:**

```
29
30  entity OHO is
31      Port ( X : in  STD_LOGIC;
32             Y : in  STD_LOGIC;
33             Z : out  STD_LOGIC);
34  end OHO;
35
36  architecture Behavioral of OHO is
37
38  Begin
39     Process (X, Y)
40        Begin
41           If (X = '1' and Y = '1') then
42              Z <= '1';
43           Elsif (X = '0' and Y = '0') then
44              Z <= '1';
45           Else
46              Z <= '0';
47        End if;
48     End process;
```

- **Output:**



- ## DISCUSSION:

  The implementation of the DIFFERENT-Gate using VHDL on the Xilinx platform was successful. After overcoming some initial syntax errors, the VHDL code was corrected, and the gate functioned as expected during simulation. The project helped me better understand VHDL and FPGA design.

  **FPGA Design** refers to the process of creating digital circuits and systems using Field-Programmable Gate Arrays (FPGAs). An FPGA is a type of integrated circuit that can be programmed or configured to perform specific tasks. Unlike traditional fixed-function hardware, FPGAs allow designers to program custom logic and functionality.

- ## CONCLUSION:

  The DIFFERENT-Gate was successfully implemented and simulated. This project enhanced my skills in VHDL coding and FPGA design, providing valuable hands-on experience with embedded systems.