
# HERITAGE INSTITUTE OF TECHNOLOGY

**Paper Name -** Computer Organisation and Architechture

**Paper Code -** CSE2252

**Team Members -**

Chinki Raj - 2361021

Vivek Kumar - 2361023

Asmita Banerjee - 2361024

Kishalay Kamar - 2361025

Aritra Hutait - 2361026

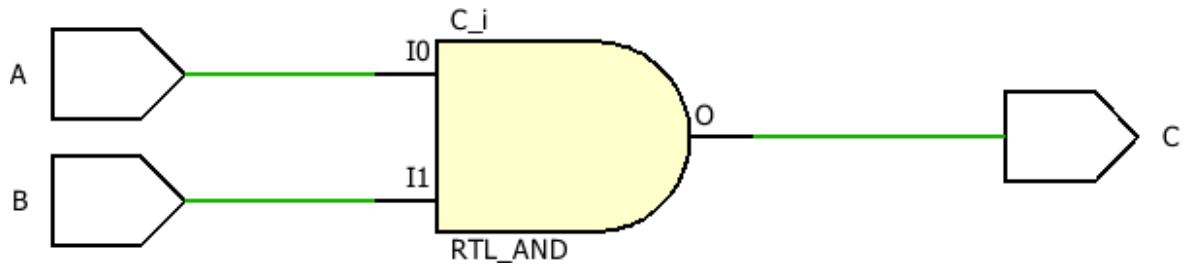
# Table of Contents

# AND Gate Dataflow Model

## VHD Code:

```
entity AND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end AND_DF;
architecture Dataflow of AND_DF is
begin

C <= A AND B;

end Dataflow;
```

## RTL Diagram

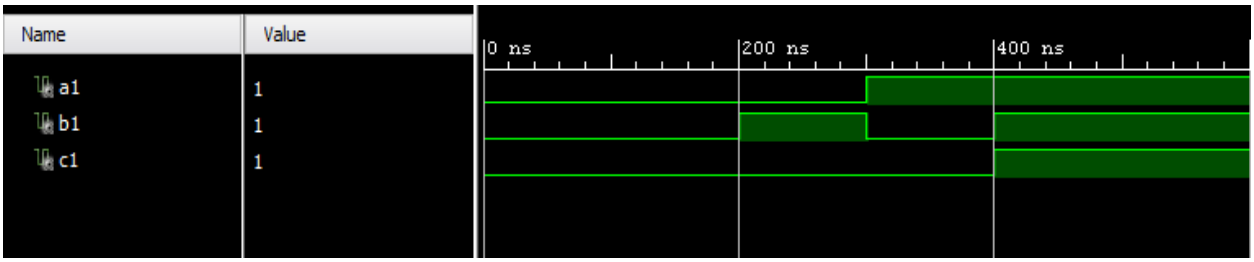


# TBW Code:

```
entity AND_DF_TBW is
-- Port ( );
end AND_DF_TBW;

architecture Dataflow of AND_DF_TBW is
component AND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin

UUT: AND_DF Port map(A=>a1, B=>b1, c=>c1);
stim_proc: process

begin
wait for 100ns;

a1<='0';
b1<='0';
wait for 100ns;
```

```
a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;

a1<='1';
b1<='1';

wait;
end process;
end Dataflow;
```

# TBW Waveform

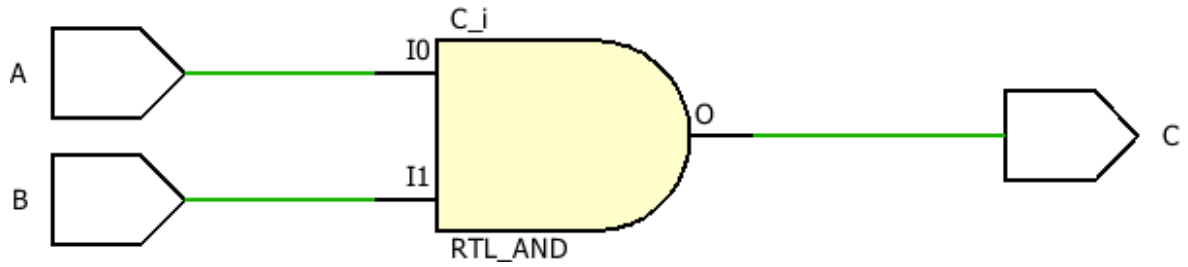


# AND Gate Behavioral Model

## VHD Code:

```
entity AND_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end AND_GATE_BV;

architecture Behavioral of AND_GATE_BV is

begin
process(A,B)
begin
    if(A='1' and B='1') then
        c<='1';
    else
        c<='0';
    end if;
end process;

end Behavioral;
```

# RTL Diagram

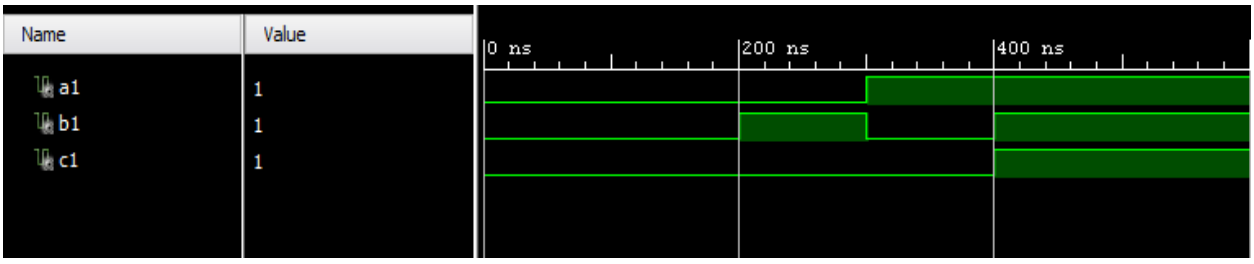



# TBW Code:

```
entity AND_GATE_TBW is
-- Port ( );
end AND_GATE_TBW;

architecture Behavioral of AND_GATE_TBW is
component AND_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin
```

```
UUT: AND_GATE_BV Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

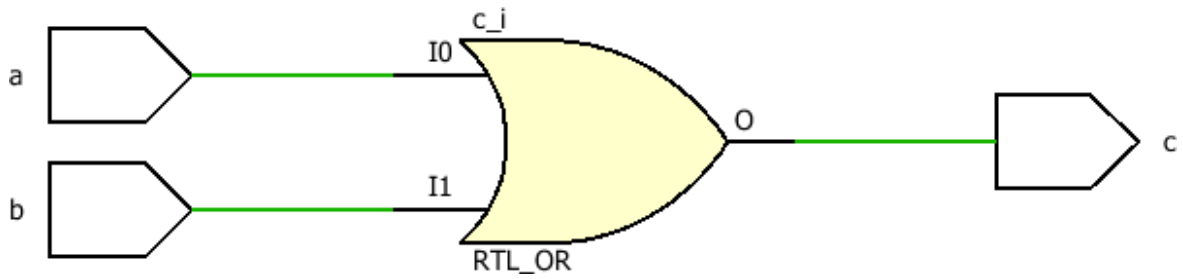


# OR Gate Dataflow Model

## VHD Code:

```
entity OR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end OR_DF;
architecture Dataflow of OR_DF is
begin
c <= a OR b;
end Dataflow;
```

## RTL Diagram

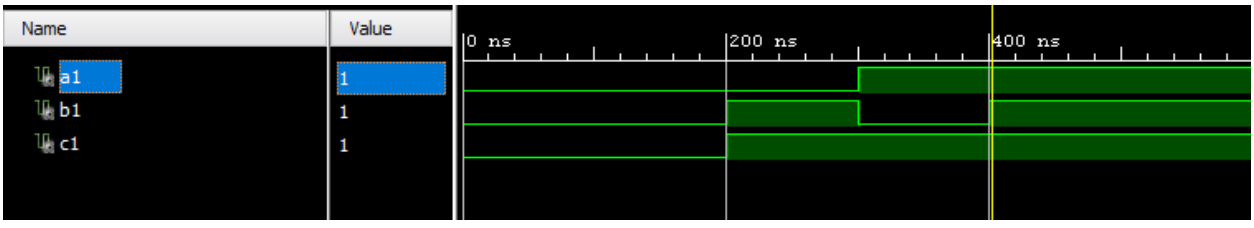


# TBW Code:

```
entity OR_DF_TBW is
-- Port ( );
end OR_DF_TBW;

architecture Dataflow of OR_DF_TBW is
component OR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin
UUT: OR_DF Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process

begin
wait for 100ns;

a1<='0';
b1<='0';
wait for 100ns;
```

```
a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;

a1<='1';
b1<='1';

wait;
end process;
end Dataflow;
```

# TBW Waveform

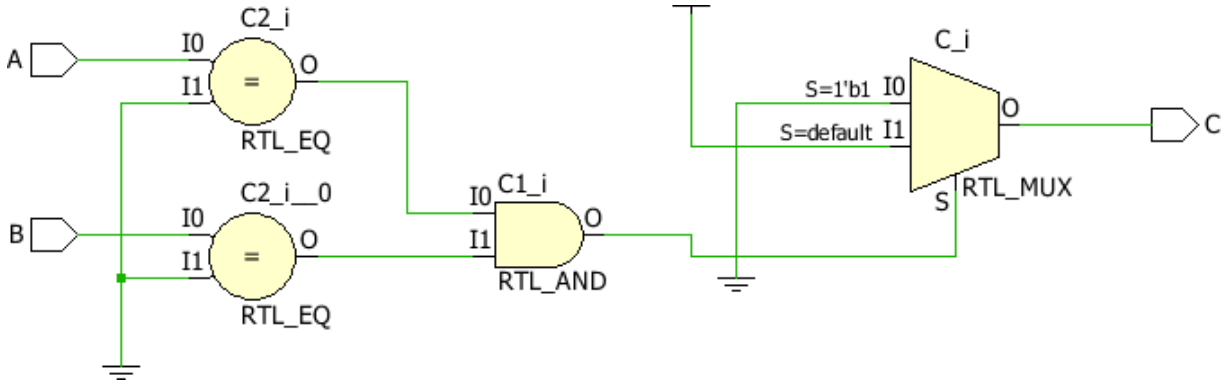


# OR Gate Behavioral Model

## VHD Code:

```
entity OR_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end OR_GATE_BV;
architecture Behavioral of OR_GATE_BV is
begin
process(A,B)
begin
    if(A='0' and B='0') then
        c<='0';
    else
        c<='1';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

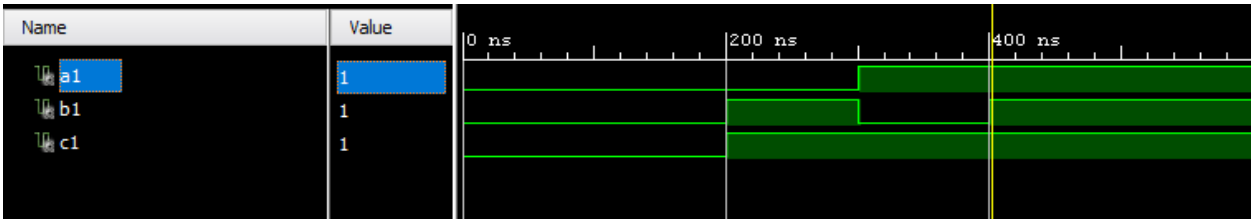



# TBW Code:

```
entity OR_GATE_TBW is
-- Port ( );
end OR_GATE_TBW;
architecture Behavioral of OR_GATE_TBW is
component OR_GATE_BV is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
```

```
UUT: OR_GATE_BV Port map(a=>a1, b=>b1, c=>c1);

stim_proc: process

begin

wait for 100ns;

a1<='0';

b1<='0';

wait for 100ns;

a1<='0';

b1<='1';

wait for 100ns;

a1<='1';

b1<='0';

wait for 100ns;

a1<='1';

b1<='1';

wait;

end process;

end Behavioral;
```

# TBW Waveform

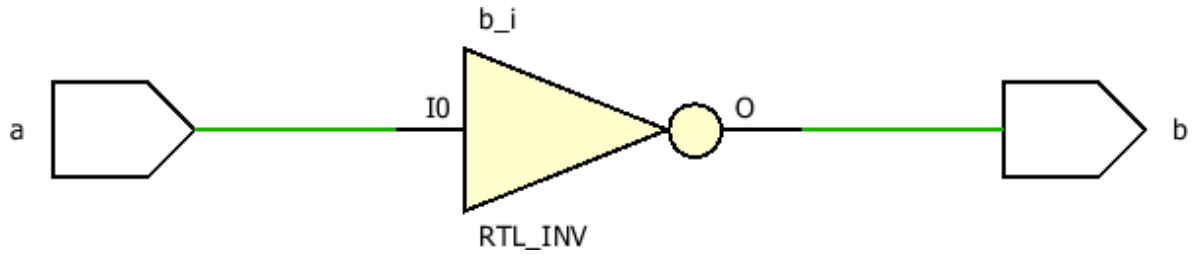


# NOT Gate Dataflow Model

## VHD Code:

```
entity NOT_DF is
    Port ( a : in STD_LOGIC;
           b : out STD_LOGIC);
end NOT_DF;
architecture Dataflow of NOT_DF is
begin
b <= NOT a;
end Dataflow;
```

## RTL Diagram

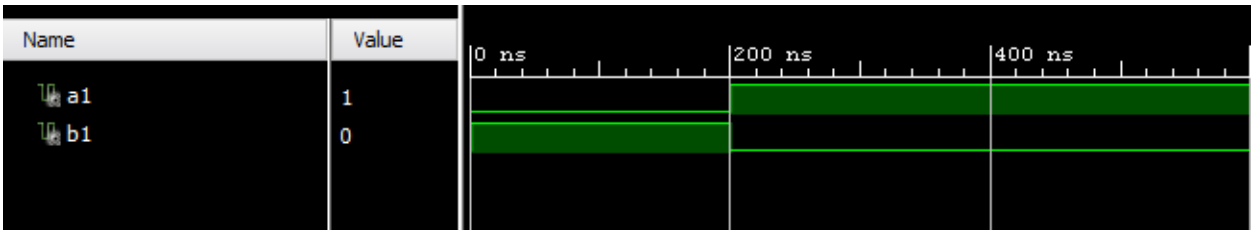



## TBW Code:

```
entity NOT_DF_TBW is
-- Port ( );
```

```
end NOT_DF_TBW;
architecture Dataflow of NOT_DF_TBW is
component NOT_DF is
    Port ( a : in STD_LOGIC;
           b : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC;
begin
UUT: NOT_DF Port map(a=>a1, b=>b1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
wait for 100ns;
a1<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform

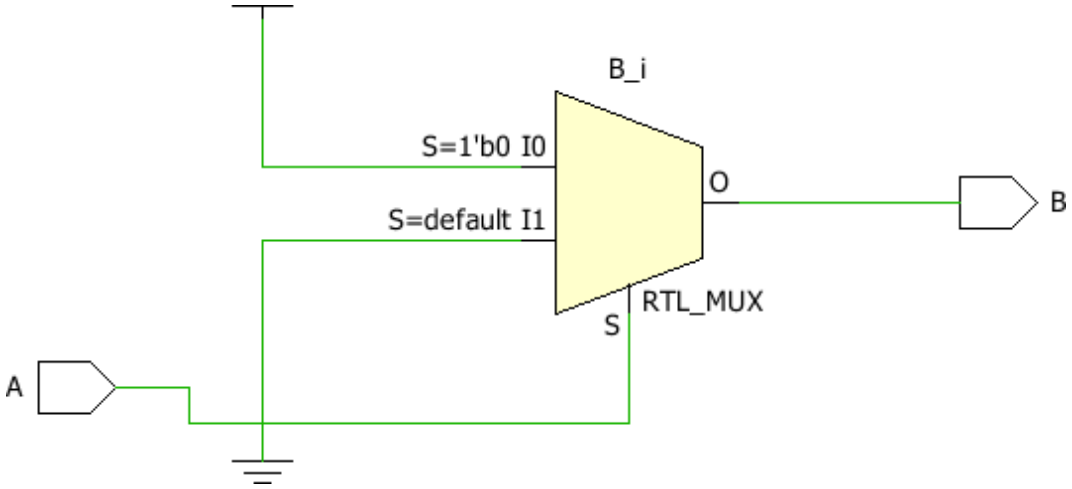


# NOT Gate Behavioral Model

## VHD Code:

```
entity NOT_GATE_BV is
    Port ( A : in
    STD_LOGIC;
          B : out STD_LOGIC);
end NOT_GATE_BV;
architecture Behavioral of NOT_GATE_BV is
begin
process(A)
begin
    if(A='0') then
        B<='1';
    else
        B<='0';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

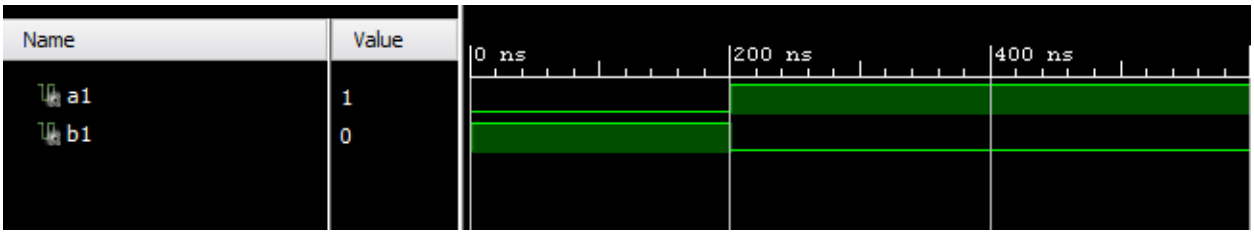



# TBW Code:

```
entity NOT_GATE_TBW is
-- Port ( );
end NOT_GATE_TBW;
architecture Behavioral of NOT_GATE_TBW is
component NOT_GATE_BV is
    Port ( a : in STD_LOGIC;
           b : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC;
```

```
begin
UUT: NOT_GATE_BV Port map(a=>a1, b=>b1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
wait for 100ns;
a1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

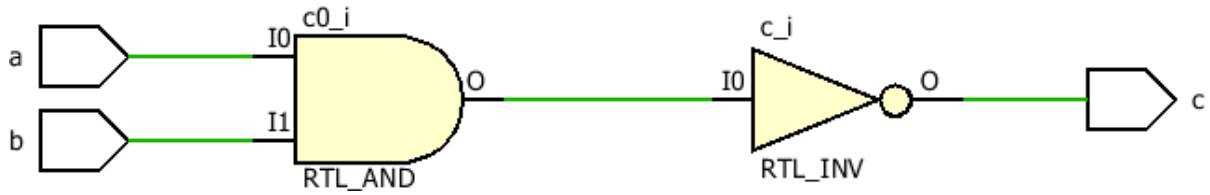


# NAND Gate Dataflow Model

## VHD Code:

```
entity NAND_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end NAND_DF;
architecture Dataflow of NAND_DF is
begin
c<=a NAND b;
end Dataflow;
```

## RTL Diagram



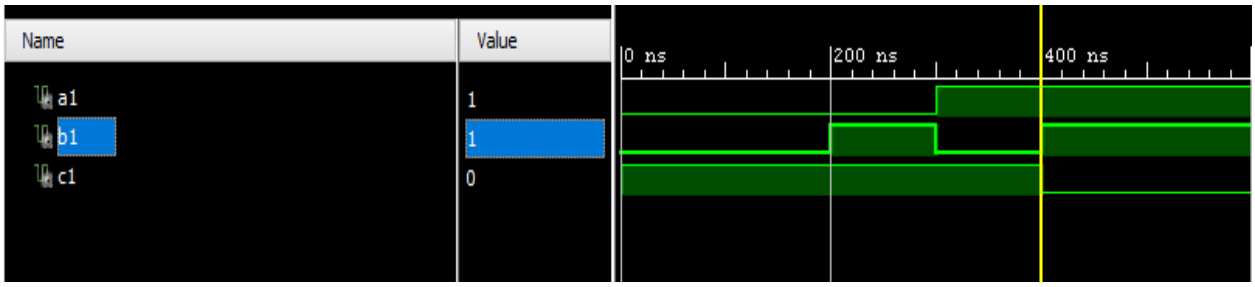

## TBW Code:

```
entity NAND_DF_TBW is
-- Port ( );
end NAND_DF_TBW;
```

```
architecture Dataflow of NAND_DF_TBW is
component NAND_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: NAND_DF Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
```

```
end process;

end Dataflow;
```

# TBW Waveform

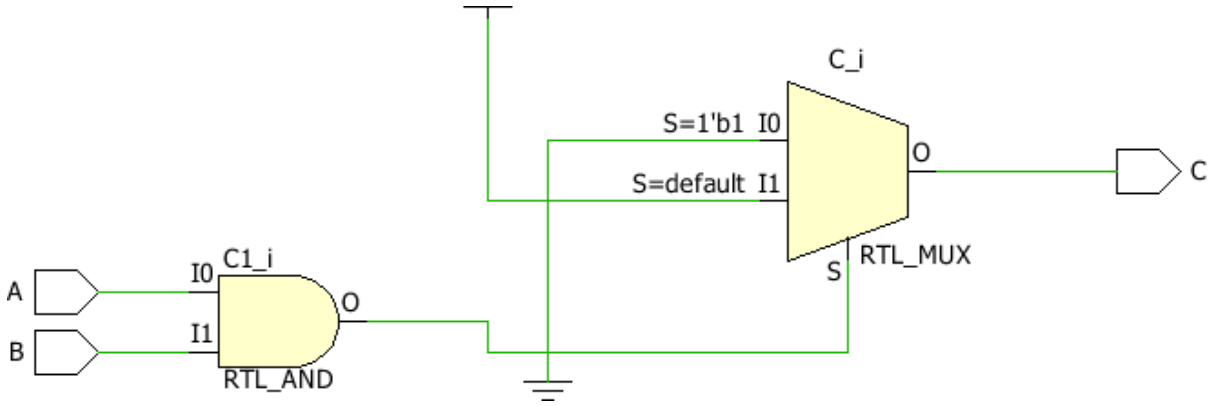


# NAND Gate Behavioral Model

## VHD Code:

```
entity NAND_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end NAND_GATE_BV;
architecture Behavioral of NAND_GATE_BV is
begin
process(A,B)
begin
    if(A='1' and B='1') then
       c<='0';
    else
       c<='1';
    end if;
end process;
end Behavioral;
```

## RTL Diagram

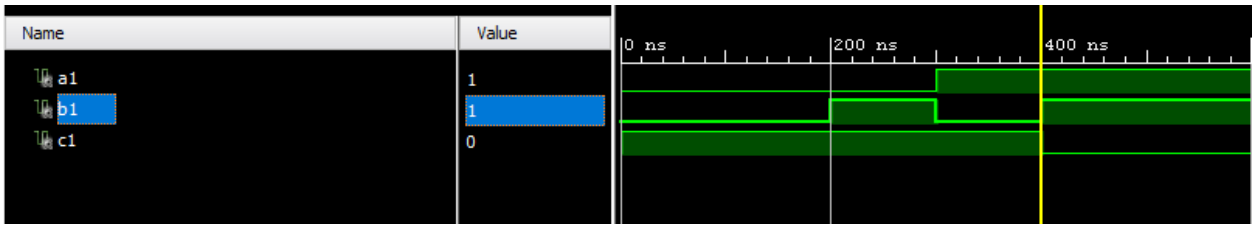



# TBW Code:

```
entity NAND_GATE_TBW is
-- Port ( );
end NAND_GATE_TBW;
architecture Dataflow of NAND_GATE_TBW is
component NAND_GATE_BV is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: NAND_GATE_BV Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process
begin
```

```
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

## TBW Waveform

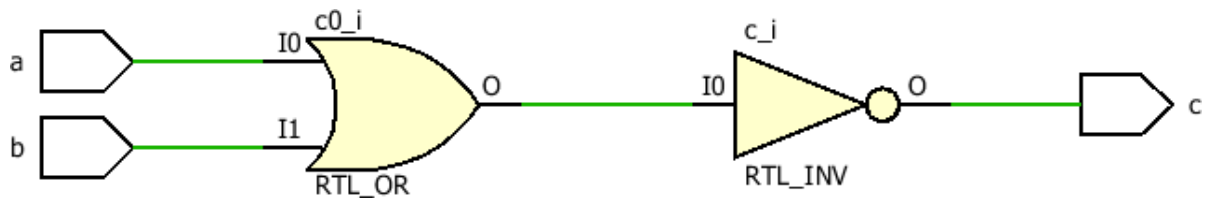



# NOR Gate Dataflow Model

## VHD Code:

```
entity NOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end NOR_DF;

architecture Dataflow of NOR_DF is

begin
c<=a NOR b;

end Dataflow;
```

# RTL Diagram



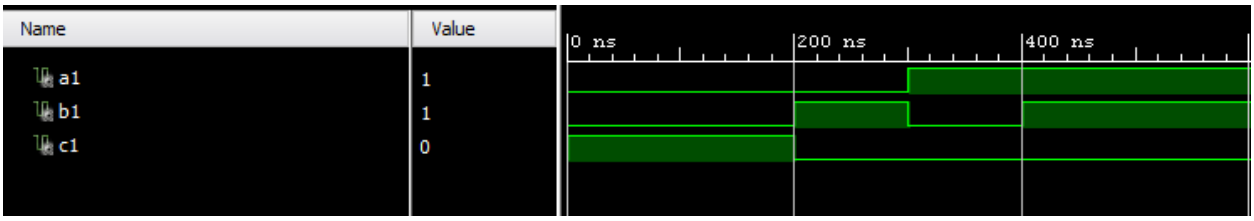

# TBW Code:

```
entity NOR_DF_TBW is
-- Port ( );
end NOR_DF_TBW;
```

```
architecture Dataflow of NOR_DF_TBW is
component NOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin

UUT: NOR_DF Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';
wait for 100ns;

a1<='1';
b1<='0';
```

```
wait for 100ns;


a1<='1';
b1<='1';


wait;
end process;
end Dataflow;
```

# TBW Waveform



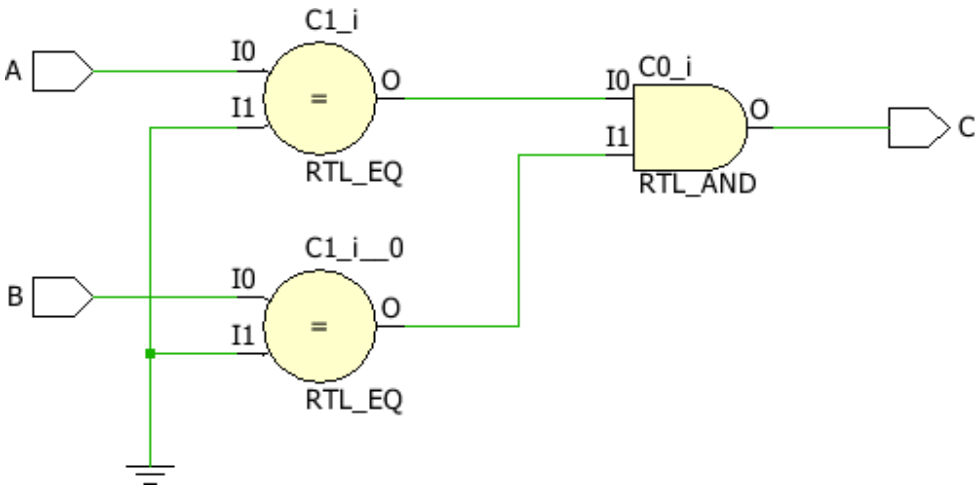

# NOR Gate Behavioral Model

# VHD Code:

```
entity NOR_GATE_BV is
    Port ( A : in
    STD_LOGIC;
```

```
        B : in STD_LOGIC;
        C : out STD_LOGIC);
end NOR_GATE_BV;
architecture Behavioral of NOR_GATE_BV is
begin
process(A,B)
begin
    if(A='0' and B='0') then
        C<='1';
    else
        C<='0';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

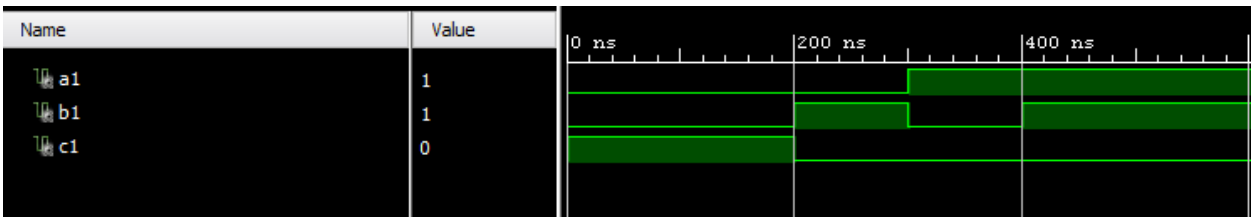



# TBW Code:

```
entity NOR_GATE_TBW is
-- Port ( );
end NOR_GATE_TBW;
architecture Behavioral of NOR_GATE_TBW is
component NOR_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: NOR_GATE_BV Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
```

```
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

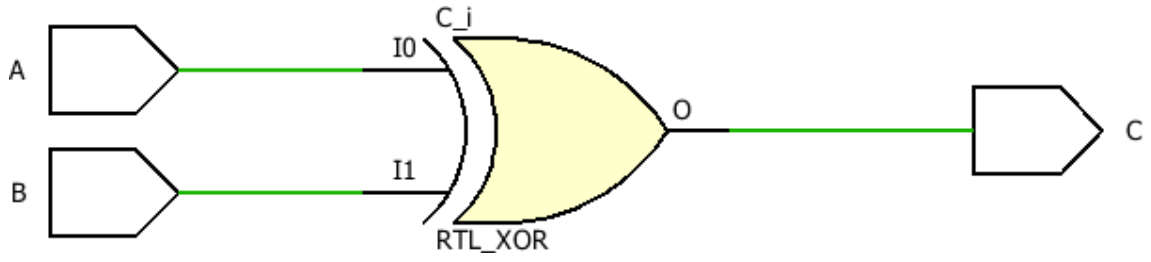



# XOR Gate Dataflow Model

# VHD Code:

```
entity XOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end XOR_DF;

architecture Dataflow of XOR_DF is

begin
c<=a XOR b;

end Dataflow;
```

# RTL Diagram

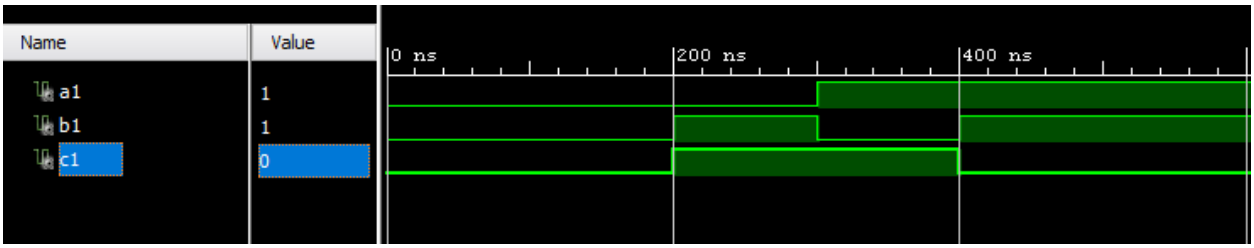



# TBW Code:

```
entity XOR_DF_TBW is
-- Port ( );
end XOR_DF_TBW;

architecture Dataflow of XOR_DF_TBW is
component XOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin

UUT: XOR_DF Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';
```

```
wait for 100ns;

a1<='1';
b1<='0';
wait for 100ns;

a1<='1';
b1<='1';

wait;
end process;
end Dataflow;
```

## TBW Waveform

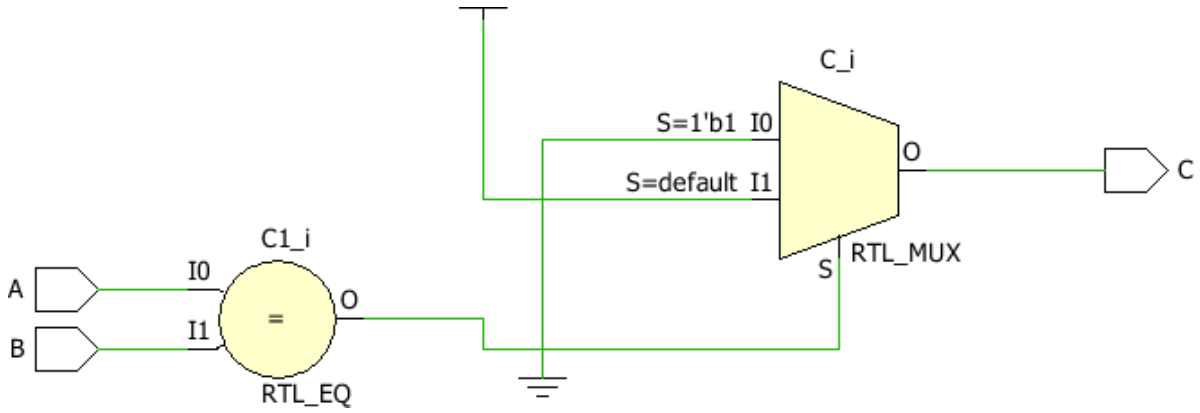



## XOR Gate Behavioral Model

# VHD Code:

```
entity XOR_GATE_BV is
    Port ( A : in
    STD_LOGIC;
          B : in STD_LOGIC;
          C : out STD_LOGIC);
end XOR_GATE_BV;
architecture Behavioral of XOR_GATE_BV is
begin
process(A,B)
begin
    if(A=B) then
        C<='0';
    else
        C<='1';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

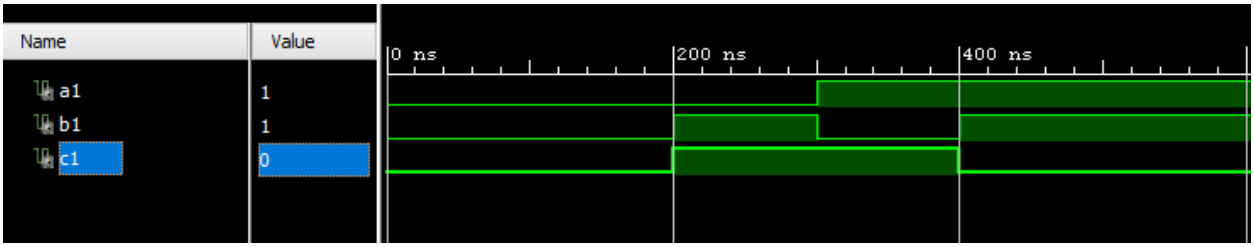



# TBW Code:

```
entity XOR_GATE_TBW is
-- Port ( );
end XOR_GATE_TBW;
architecture Behavioral of XOR_GATE_TBW is
component XOR_GATE_BV is
    Port ( A : in STD_LOGIC;
           B: in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;

begin
UUT: XOR_GATE_BV Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
```

```
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

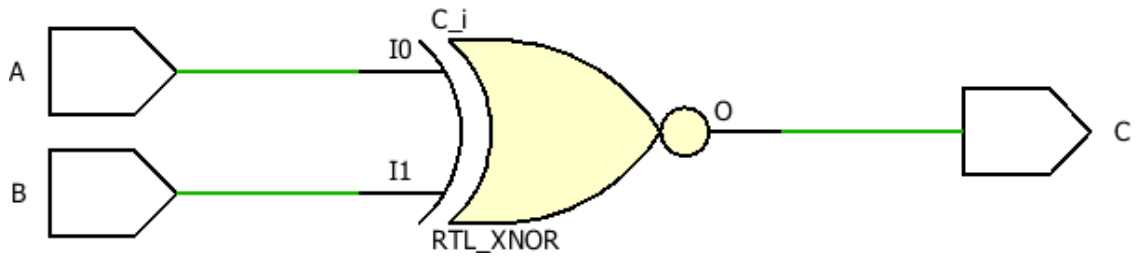



# XNOR Gate Dataflow Model

# VHD Code:

```
entity XNOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end XNOR_DF;

architecture Dataflow of XNOR_DF is

begin
c<=a XNOR b;

end Dataflow;
```

# RTL Diagram

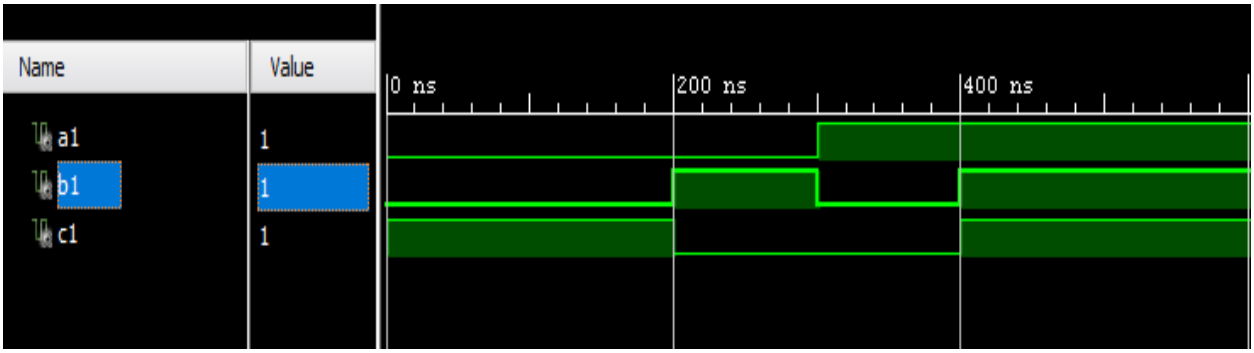



# TBW Code:

```
entity XNOR_DF_TBW is
-- Port ( );
end XNOR_DF_TBW;
architecture Dataflow of XNOR_DF_TBW is
component XNOR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: XNOR_DF Port map(a=>a1, b=>b1, c=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;

a1<='1';
b1<='0';
```

```
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

## TBW Waveform

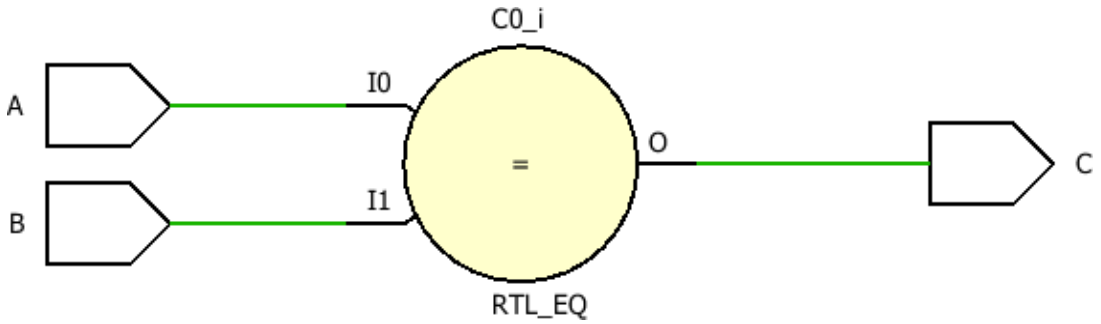



## XNOR Gate Behavioral Model

# VHD Code:

```
entity XNOR_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end XOR_GATE_BV;
architecture Behavioral of XNOR_GATE_BV is
begin
process(A,B)
begin
    if(A=B) then
        C<='1';
    else
        C<='0';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

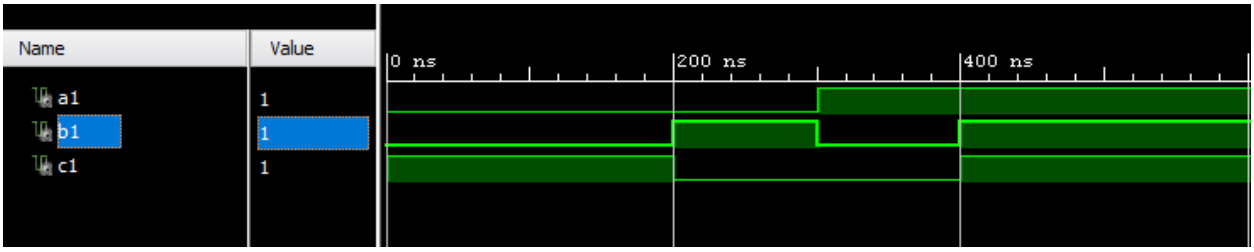



# TBW Code:

```
entity XNOR_GATE_TBW is
-- Port ( );
end XNOR_GATE_TBW;

architecture Behavioral of XNOR_GATE_TBW is
component XNOR_GATE_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: XNOR_GATE_BV Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
```

```
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

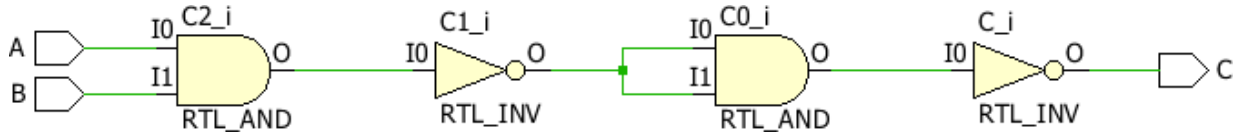



# AND_NAND Gate Dataflow Model

# VHD Code:

```
entity AND_NAND_DF is
    Port ( A : in
    STD_LOGIC;
          B : in STD_LOGIC;
          C : out STD_LOGIC);
end AND_NAND_DF;
architecture Dataflow of AND_NAND_DF is
begin
C<=(A NAND B) NAND (A NAND B);
end Dataflow;
```

# RTL Diagram

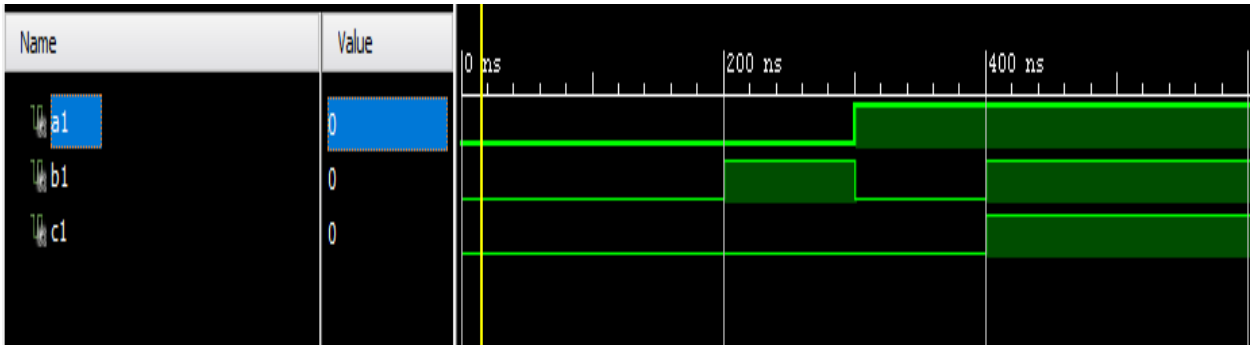



# TBW Code:

```
entity AND_NAND_TBW is
-- Port ( );
end AND_NAND_TBW;
architecture Dataflow of AND_NAND_TBW is
component NAND_DF is
```

```
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: AND_NAND_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

## TBW Waveform

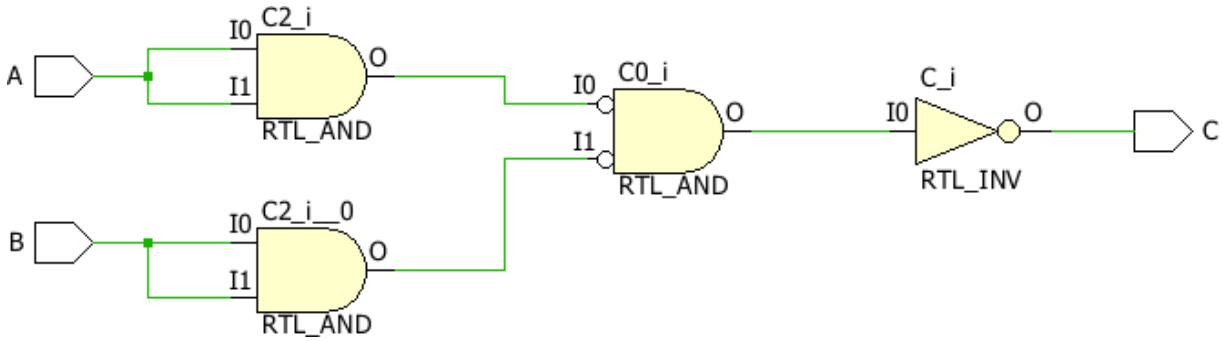



# OR_NAND Gate Dataflow Model

## VHD Code:

```
entity OR_NAND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end OR_NAND_DF;
architecture Dataflow of OR_NAND_DF is
begin
C<=((A NAND A) NAND (B NAND B));
end Dataflow;
```

# RTL Diagram



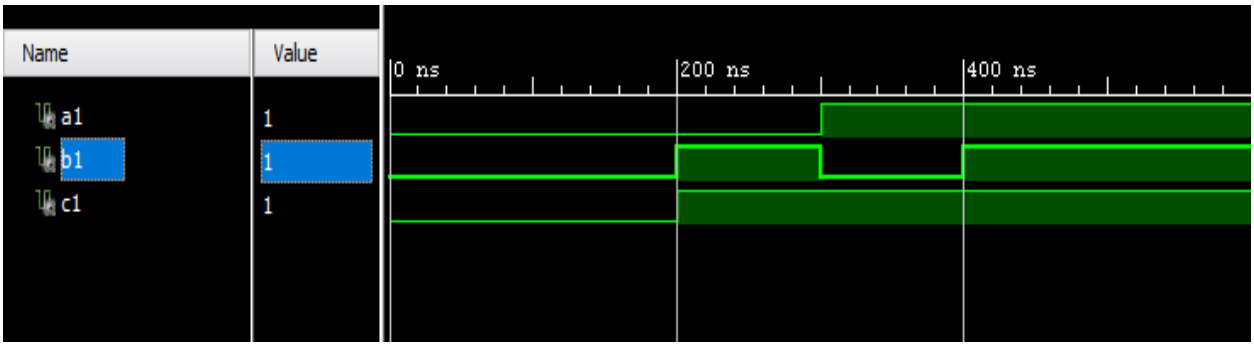

# TBW Code:

```
entity AND_NAND_TBW is
-- Port ( );
end AND_NAND_TBW;
architecture Dataflow of AND_NAND_TBW is
```

```
component NAND_DF is
    Port ( A : in
    STD_LOGIC;
          B : in STD_LOGIC;
          C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: AND_NAND_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
```

end Dataflow;

## TBW Waveform



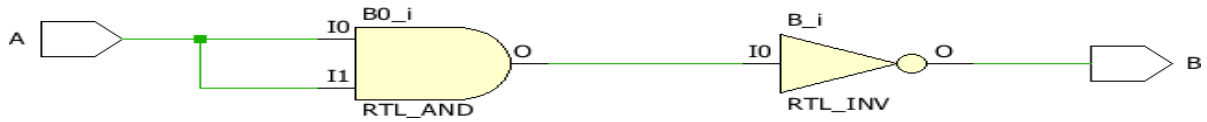

# NOT_NAND Gate Dataflow Model

## VHD Code:

```
entity NOT_NAND_DF is
    Port ( A : in
    STD_LOGIC;
          B : out STD_LOGIC);
```

```
end NOT_NAND_DF;
architecture Dataflow of NOT_NAND_DF is
begin
B<=(A NAND A);
end Dataflow;
```

# RTL Diagram

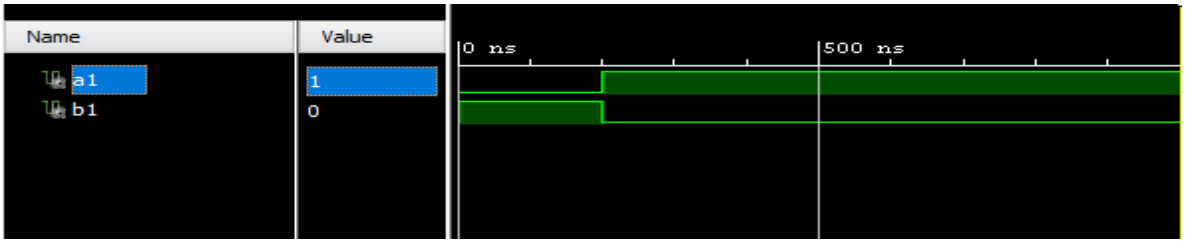



# TBW Code:

```
entity NOT_NAND_TBW is
-- Port ( );
end NOT_NAND_TBW;
architecture Dataflow of NOT_NAND_TBW is
component NOT_NAND_DF is
    Port ( A : in STD_LOGIC;
           B : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC;
begin
```

```
UUT: NOT_NAND_DF Port map(A=>a1, B=>b1);

stim_proc: process

begin

wait for 100ns;

a1<='0';

wait for 100ns;

a1<='1';

wait;

end process;

end Dataflow;
```

## TBW Waveform



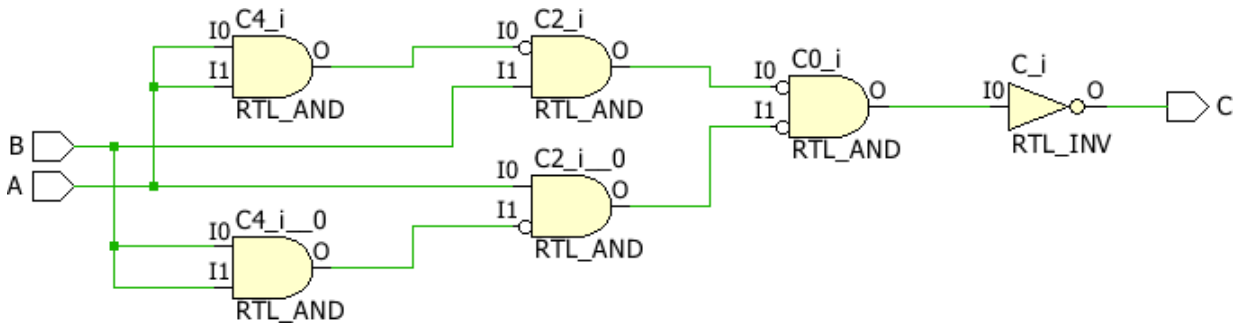

# XOR_NAND Gate Dataflow Model

## VHD Code:

```
entity XOR_NAND_DF is

    Port ( A : in

    STD_LOGIC;

          B : in STD_LOGIC;
```

```
        C : out STD_LOGIC);
end XOR_NAND_DF;

architecture Dataflow of XOR_NAND_DF is

begin
C<=((A NAND A) NAND B) NAND (A NAND (B NAND B));
end Dataflow;
```

# RTL Diagram




# TBW Code:

```
entity XOR_NAND_TBW is
-- Port ( );
end XOR_NAND_TBW;
architecture Dataflow of XOR_NAND_TBW is
component XOR_NAND_DF is
    Port ( A : in STD_LOGIC;
```

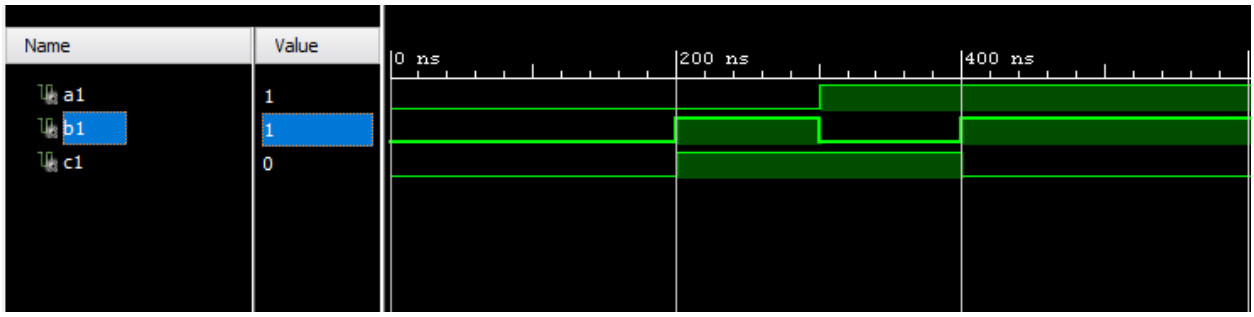
```
        B : in STD_LOGIC;
        C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: XOR_NAND_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;
a1<='1';
b1<='1';
```

wait;

end process;

end Dataflow;

## TBW Waveform



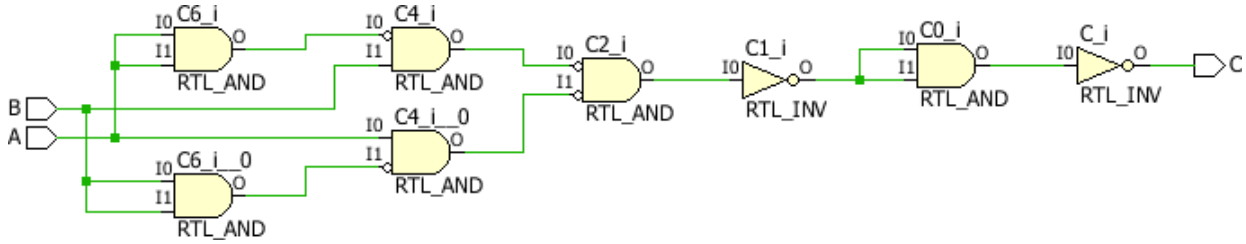

# XNOR_NAND Gate Dataflow Model

## VHD Code:

```
entity XNOR_NAND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
```

```
        C : out STD_LOGIC);
end XNOR_NAND_DF;

architecture Dataflow of XNOR_NAND_DF is

begin
C<=((A NAND A) NAND B) NAND (A NAND (B NAND B)) NAND ((A NAND A) NAND B) NAND (A NAND (B NAND B));
end Dataflow;
```

# RTL Diagram




# TBW Code:

```
entity XNOR_NAND_TBW is
-- Port ( );
end XNOR_NAND_TBW;
architecture Dataflow of XNOR_NAND_TBW is
component XNOR_NAND_DF is
    Port ( A : in
        STD_LOGIC; B : in
        STD_LOGIC;
```

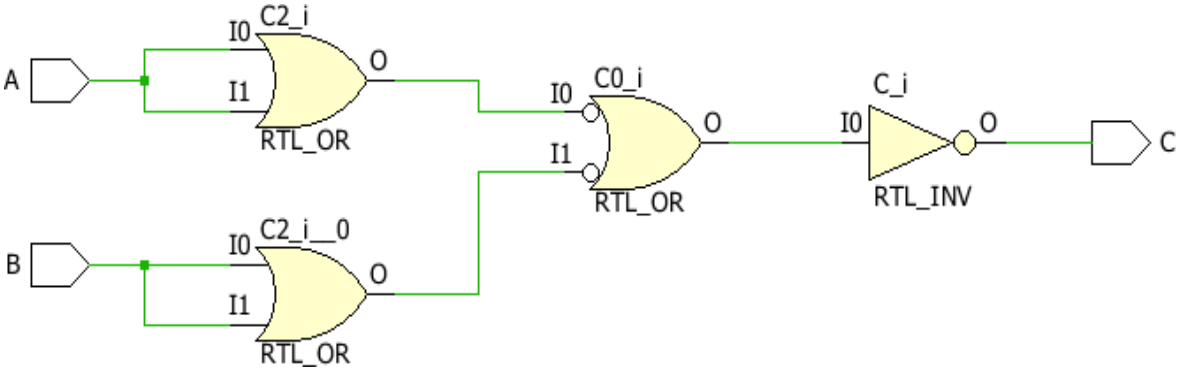
```
        C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: XNOR_NAND_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;
a1<='1';
b1<='1';
wai
```

# AND_NOR Gate Dataflow Model

## VHD Code:

```
entity AND_NOR_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end AND_NOR_DF;
architecture Dataflow of AND_NOR_DF is


begin
C<=((A NOR A) NOR (B NOR B));
end Dataflow;
```

## RTL Diagram

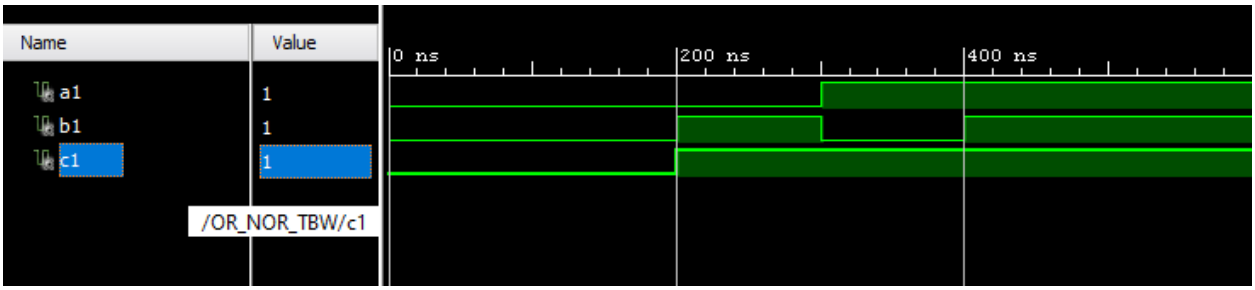



## TBW Code:

```
entity AND_NOR_TBW is
-- Port ( );
end AND_NOR_TBW;
architecture Dataflow of AND_NOR_TBW is
component AND_NOR_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
```

```
        C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0'
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: AND_NOR_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform



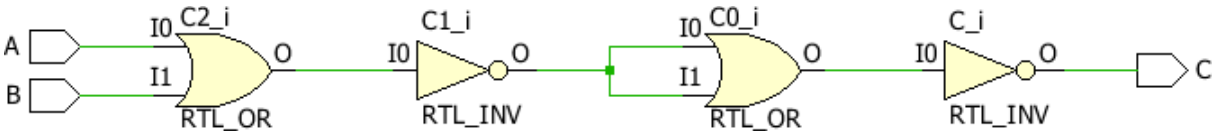

# OR_NOR Gate Dataflow Model

## VHD Code:

```
entity OR_NOR_GATE_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end OR_NOR_GATE_DF;

architecture Dataflow of OR_NOR_GATE_DF is
```

```
begin
C<=(A NOR B) NOR (A NOR B);
end Dataflow;
```

# RTL Diagram

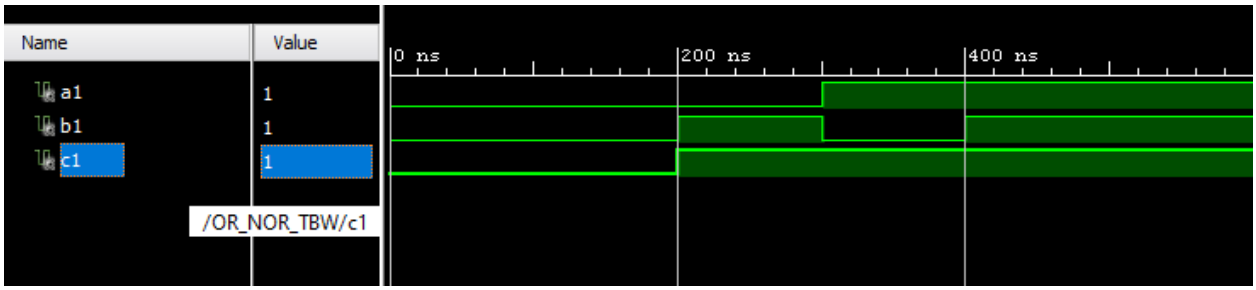



# TBW Code:

```
entity OR_NOR_TBW is
-- Port ( );
end OR_NOR_TBW;
architecture Dataflow of OR_NOR_TBW is
component OR_NOR_GATE_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
UUT: OR_NOR_GATE_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
```

```
wait for 100ns;

a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

## TBW Waveform



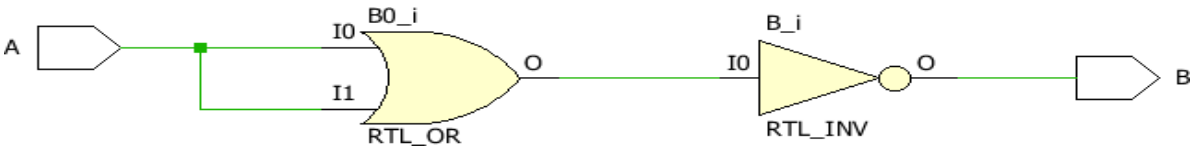

# NOT_NOR_Gate Dataflow Model

## VHD Code:

```
entity NOT_NOR _DF is
```

```
    Port ( A : in STD_LOGIC;
           B : out STD_LOGIC);
end NOT_NOR _DF;
architecture Dataflow of NOT_NOR_DF is
begin
B<=(A NOR A);
end Dataflow;
```

# RTL Diagram




# TBW Code:

```
entity NOT_NOR_TBW is
-- Port ( );
end NOT_NOR_TBW;
architecture Dataflow of NOT_NOR_TBW is
component NOT_NOR_DF is
    Port ( A : in STD_LOGIC;
           B : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC;
begin
UUT: NOT_NOR_DF Port map(A=>a1, B=>b1);
stim_proc: process
begin
wait for 100ns;
```

```
a1<='0';
wait for 100ns;
a1<='1';
wait;
end process;
end Dataflow;
```

## TBW Waveform


# XOR_NOR Gate Dataflow Model

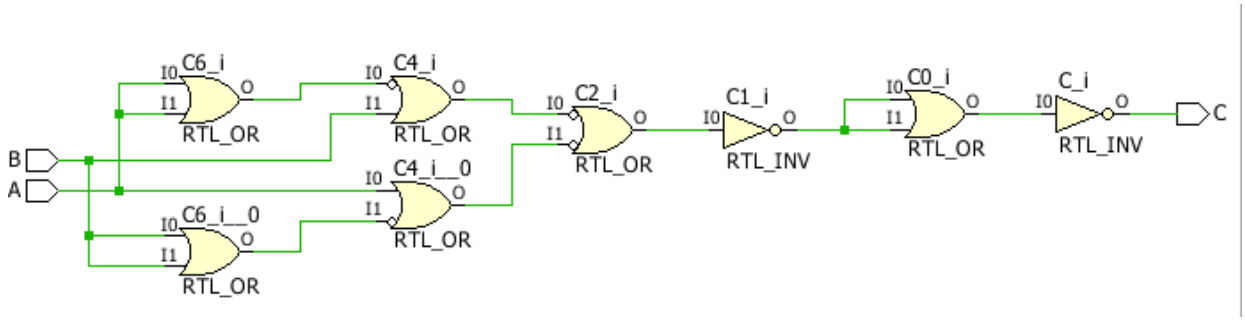
## VHD Code:

```
entity XOR_NOR_GATE_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end XOR_NOR_GATE _DF;

architecture Dataflow of XOR_NOR _GATE_DF is

begin
```

C<=((A NOR A) NOR B) NOR (A NOR (B NOR B)) NOR ((A NOR A) NOR B) NOR (A NOR (B NOR B));

end Dataflow;

# RTL Diagram

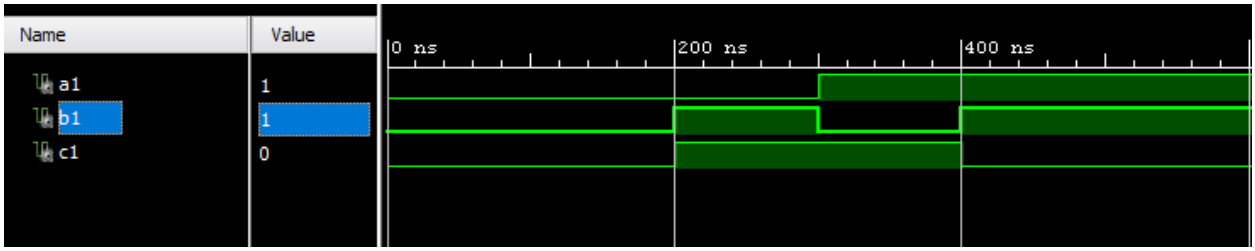



# TBW Code:

```
entity XOR_NOR_TBW is
-- Port ( );
end XOR_NOR _TBW;
architecture Dataflow of XOR_NOR _TBW is
component XOR_NOR_GATE _DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
```

```
Signal c1:STD_LOGIC;
begin
UUT: XOR_NOR_GATE _DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;

a1<='0';
b1<='1';

wait for 100ns;

a1<='1';
b1<='0';

wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform

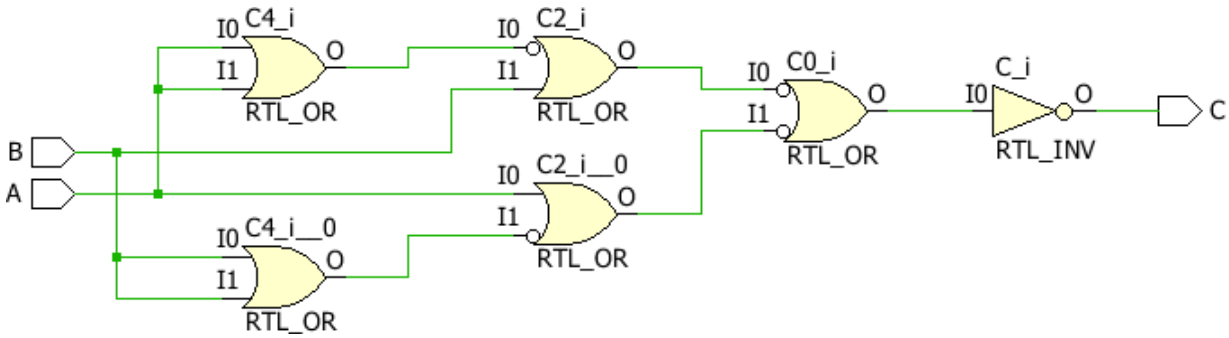



# XNOR_NOR Gate Dataflow Model

## VHD Code:

```
entity XNOR_NOR_DF is

    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end XNOR_NOR_DF;
architecture Dataflow of XNOR_NOR_DF is
begin
C<=((A NOR A) NOR B) NOR (A NOR (B NOR B));
end Dataflow;
```

## RTL Diagram

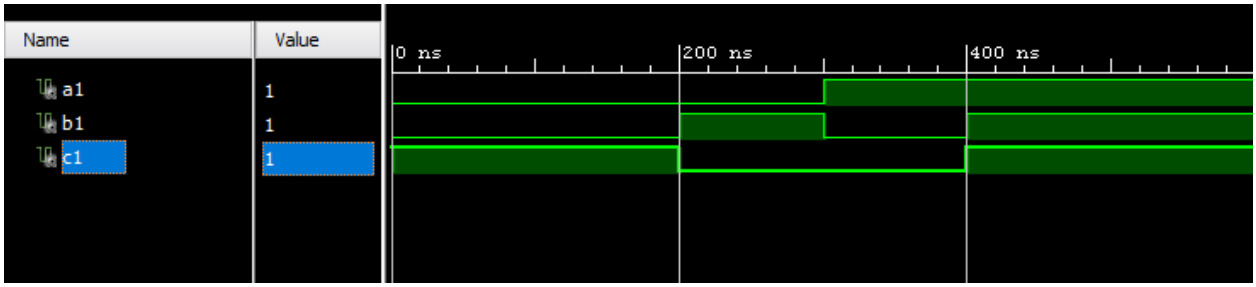



# TBW Code:

```
entity XNOR_NOR_TBW is
-- Port ( );
end XNOR_NOR_TBW;
architecture Dataflow of XNOR_NOR_TBW is
component XNOR_NOR_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
Signal c1:STD_LOGIC;
begin
```

```
UUT: XNOR_NOR_DF Port map(A=>a1, B=>b1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform

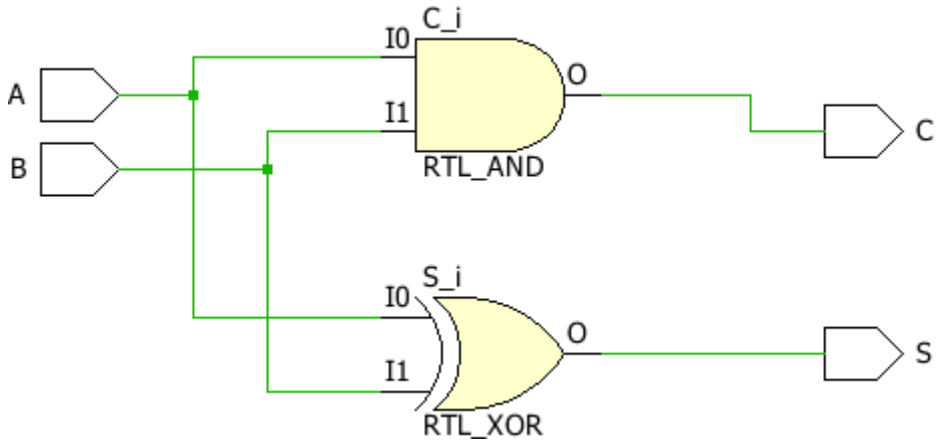


# HALF ADDER Dataflow Model

## VHD Code:

```
entity HALF_ADDER_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           S : out STD_LOGIC;
           C : out STD_LOGIC);
end HALF_ADDER_DF;
architecture Dataflow of HALF_ADDER_DF is
begin
S<=A xor B;
C<=A and B;

end Dataflow
```

# RTL Diagram

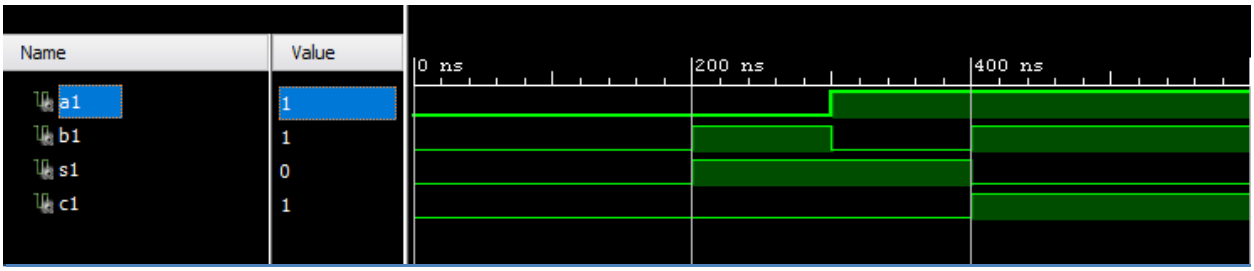



# TBW Code:

```
entity HALF_ADDER_TBW is
-- Port ( );
end HALF_ADDER_TBW;
architecture Behavioral of HALF_ADDER_TBW is
component HALF_ADDER_BV is
    Port ( A : in
           STD_LOGIC; B: in
           STD_LOGIC; S: out
           STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal b1:STD_LOGIC:='0';
```

```
Signal s1:STD_LOGIC;
Signal c1:STD_LOGIC;
begin
UUT: HALF_ADDER_BV Port map(A=>a1, B=>b1, S=>s1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform



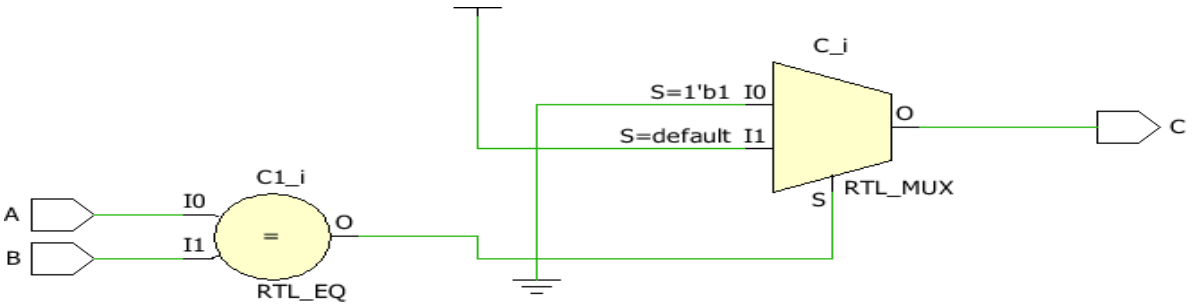

# HALF ADDER Behavioral Model

## VHD Code:

```
entity HALF_ADDER_BV is
Port ( A : in STD_LOGIC;
       B : in STD_LOGIC;
       S : out STD_LOGIC;
       C : out STD_LOGIC);
end HALF_ADDER_BV;
architecture Behavioral of HALF_ADDER_BV is
begin
process(A,B)
begin
    if(A=B) then
```

```
        S<='0';
    else
        S<='1';
    end if;
    if(A='1' and B='1') then
        C<='1';
    else
        C<='0';
    end if;
end process; end Behavioral;
```

# RTL Diagram:



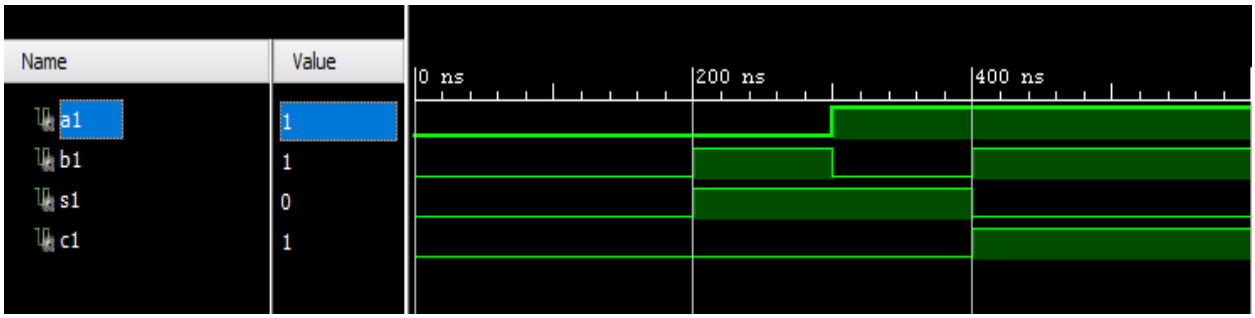

# TBW Code:

```
entity HALF_ADDER_TBW is
-- Port ( );
end HALF_ADDER_TBW;
architecture Behavioral of HALF_ADDER_TBW is
component HALF_ADDER_BV is
    Port ( A : in STD_LOGIC;
```

```
        B: in STD_LOGIC;
        S: out STD_LOGIC;
        C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal
b1:STD_LOGIC:='0';
Signal s1:STD_LOGIC;
Signal c1:STD_LOGIC;
begin
UUT: HALF_ADDER_BV Port map(A=>a1, B=>b1, S=>s1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
```

end Behavioral;

## TBW Waveform:



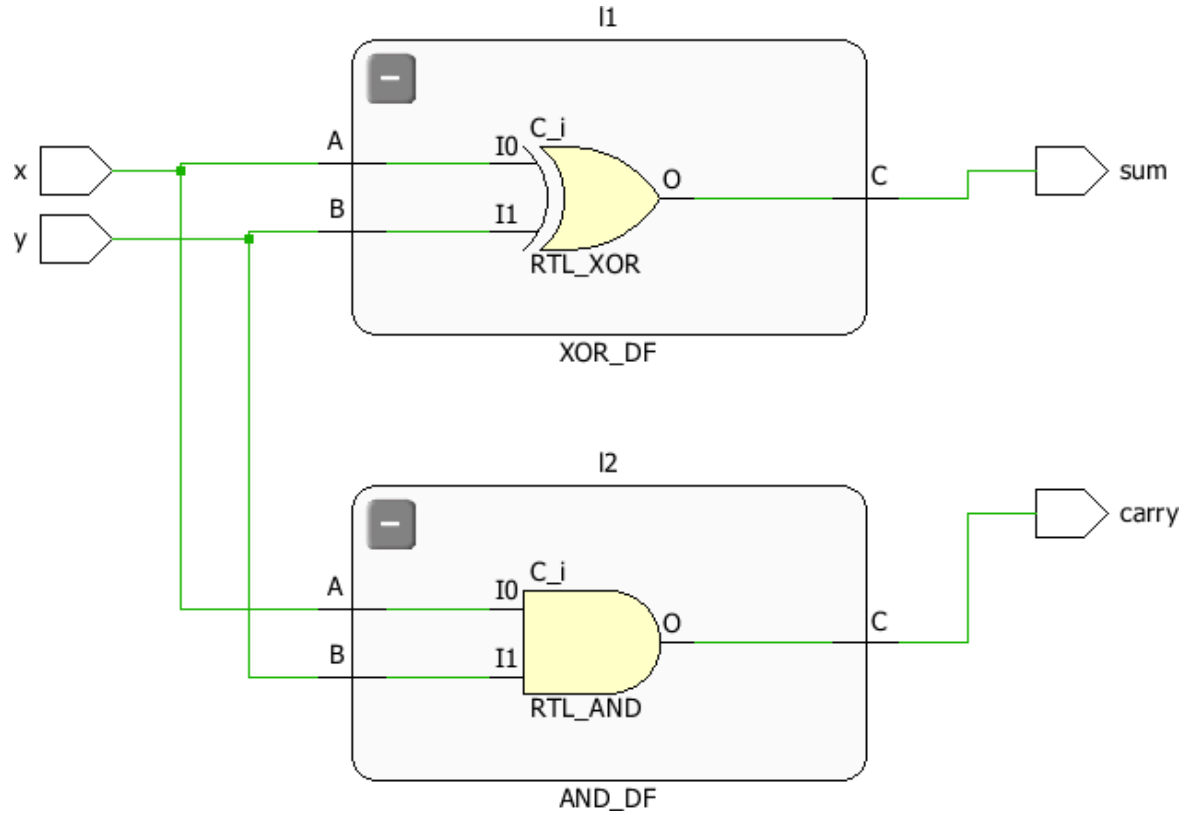

# HALF ADDER Structural Model

## VHD Code:

```
Entity HALF_ADDER_structural is
Port ( x : in STD_LOGIC;
       y : in STD_LOGIC;
       sum : out STD_LOGIC;
       carry : out STD_LOGIC);
end HALF_ADDER_structural;
architecture Structural of HALF_ADDER_structural is
component XOR_DF is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          C : out STD_LOGIC);
end component;
```

```
component AND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
begin
l1:XOR_DF port map(x,y,sum);
l2:AND_DF port map(x,y,carry);
end Structural;
```

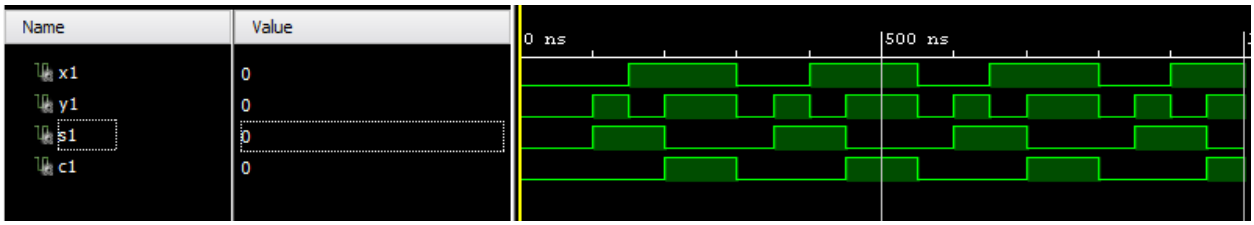
# RTL Diagram



# TBW Code:

```
entity HALF_ADDER_structural is
    Port ( x : in STD_LOGIC;
           y : in STD_LOGIC;
           sum : out STD_LOGIC;
           carry : out STD_LOGIC);
end HALF_ADDER_structural;
architecture Structural of HALF_ADDER_structural is

component XOR_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
component AND_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : out STD_LOGIC);
end component;
begin
l1:XOR_DF port map(x,y,sum);
l2:AND_DF port map(x,y,carry);
end Structural;
```

# TBW Waveform

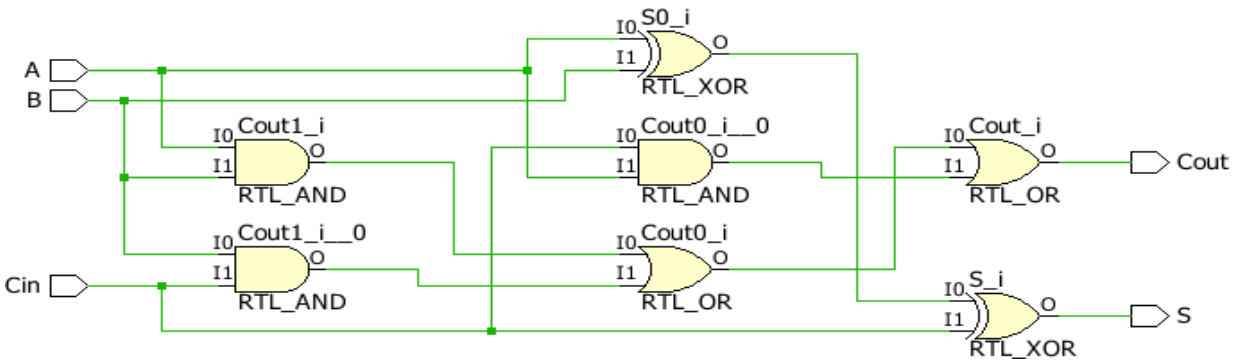



# FULL ADDER Dataflow Model

## VHD Code:

```
entity FULL_ADDER_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Cin : in STD_LOGIC;
           S : out STD_LOGIC;
           Cout : out STD_LOGIC);
end FULL_ADDER_DF;
architecture Behavioral of FULL_ADDER_DF is
begin
S<=(A xor B) xor Cin;
Cout<=(A and B) or (B and Cin) or (Cin and A);
end Behavioral;
```

# RTL Diagram




# TBW Code:

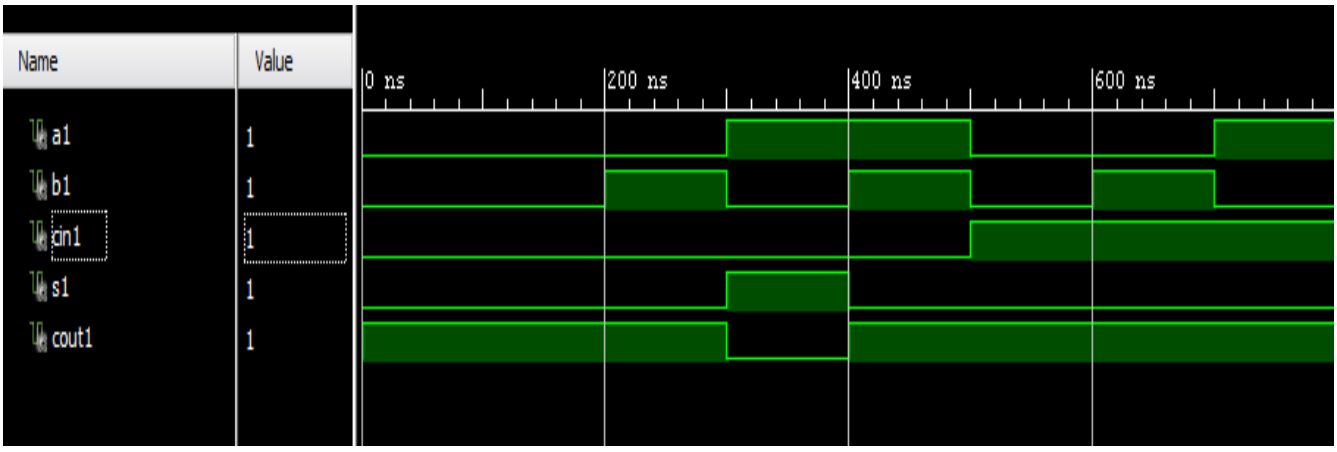
```
entity FULL_ADDER_TBW is
-- Port ( );
end FULL_ADDER_TBW;
architecture Behavioral of FULL_ADDER_TBW is
component FULL_ADDER_BV is
    Port ( A : in
```



```
        to set it up.
    STD_LOGIC; B: in
        STD_LOGIC; S: out
        STD_LOGIC;
        C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
Signal
b1:STD_LOGIC:='0';
Signal s1:STD_LOGIC;
Signal c1:STD_LOGIC;
begin
UUT: FULL_ADDER_BV Port map(A=>a1, B=>b1, S=>s1, C=>c1);
```

```
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform



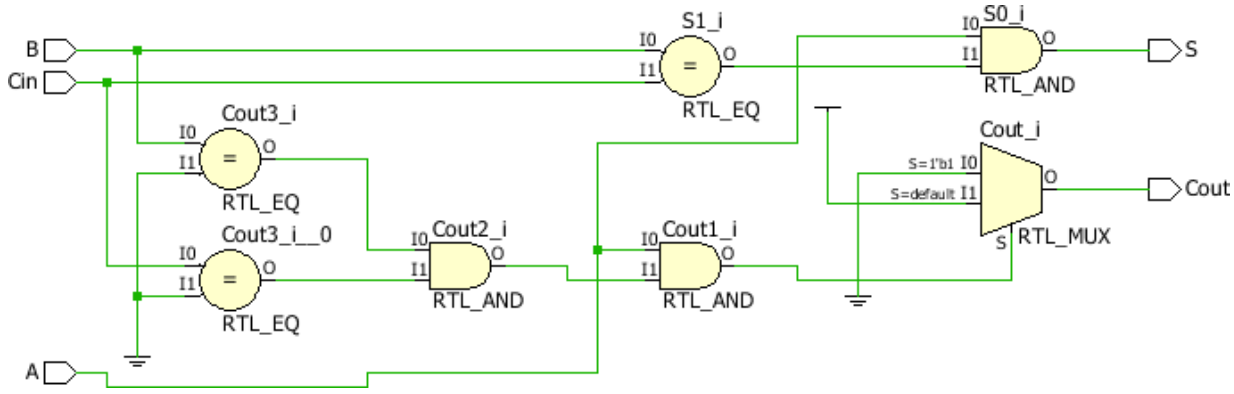
# FULL ADDER Behavioral Model

## VHD Code:

```
entity FULL_ADDER_BV is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Cin: in STD_LOGIC;
           S : out STD_LOGIC;
           Cout : out STD_LOGIC);
end FULL_ADDER_BV;
```

```
architecture Behavioral of FULL_ADDER_BV is
begin
process(A,B,Cin)
begin
    if((A='0') and (B=Cin)) then
        S<='0';
    else
        S<='1';
    end if;
    if((A='1') and (B=Cin)) then
        S<='1';
    else
        S<='0';
    end if;
    if((A='0') and (B='1' and Cin='1')) then
        Cout<='1';
    else
        Cout<='0';
    end if;
    if((A='1') and (B='0' and Cin='0')) then
        Cout<='0';
    else
        Cout<='1';
    end if;
end process;
end Behavioral;
```

# RTL Diagram

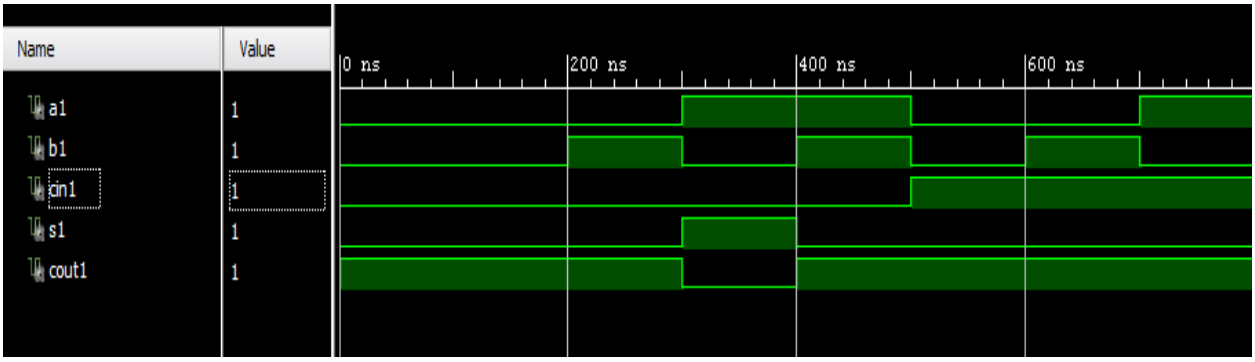



# TBW Code:

```
entity FULL_ADDER_TBW is
-- Port ( );
end FULL_ADDER_TBW;
architecture Behavioral of FULL_ADDER_TBW is
component FULL_ADDER_BV is
    Port ( A : in
           STD_LOGIC; B: in
           STD_LOGIC; S: out
           STD_LOGIC;
           C : out STD_LOGIC);
end component;
Signal a1:STD_LOGIC:='0';
```

```
Signal b1:STD_LOGIC:='0';
Signal s1:STD_LOGIC;
Signal c1:STD_LOGIC;
begin
UUT: FULL_ADDER_BV Port map(A=>a1, B=>b1, S=>s1, C=>c1);
stim_proc: process
begin
wait for 100ns;
a1<='0';
b1<='0';
wait for 100ns;
a1<='0';
b1<='1';
wait for 100ns;
a1<='1';
b1<='0';
wait for 100ns;
a1<='1';
b1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform



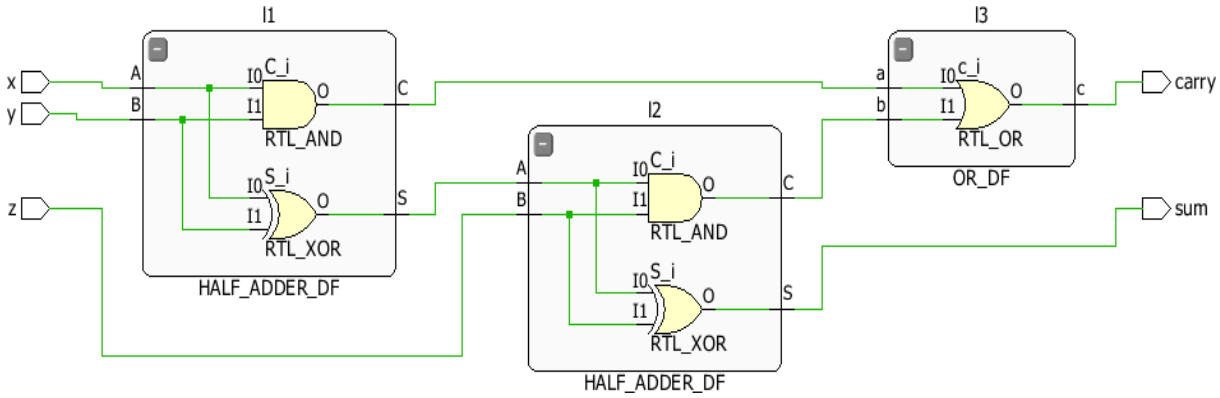

# FULL ADDER Structural Model

## VHD Code:

```
entity FULL_ADDER_structural is
    Port ( x : in STD_LOGIC;
           y : in STD_LOGIC;
           z : in STD_LOGIC;
           sum : out STD_LOGIC;
           carry : out STD_LOGIC);
end FULL_ADDER_structural;
```

```
architecture Structural of FULL_ADDER_structural is
component HALF_ADDER_DF is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           S : out STD_LOGIC;
           C : out STD_LOGIC);
end component;
component OR_DF is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c : out STD_LOGIC);
end component;
signal s1:std_logic;
signal c1:std_logic;
signal c2:std_logic;
begin
l1:HALF_ADDER_DF port map(x,y,s1,c1);
l2:HALF_ADDER_DF port map(s1,z,sum,c2);
l3:OR_DF port map(c1,c2,carry);
end Structural;
```

# RTL Diagram



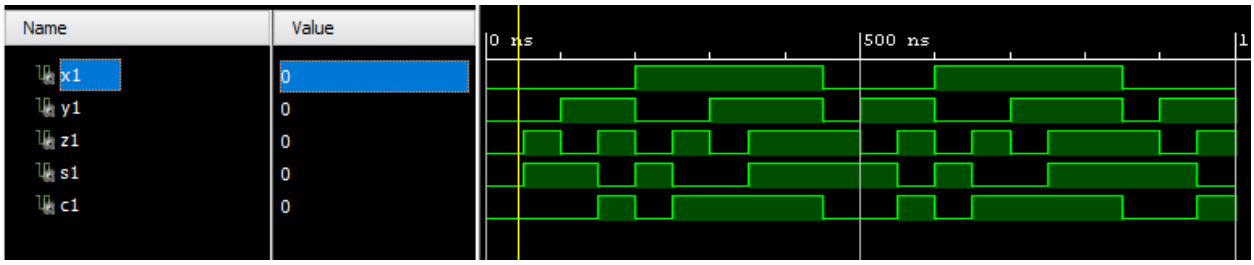

# TBW Code:

```
entity FULL_ADDER_TBW is
-- Port ( );
end FULL_ADDER_TBW;
architecture Structural of FULL_ADDER_TBW is
component FULL_ADDER_structural is
    Port ( x : in STD_LOGIC;
           y: in STD_LOGIC;
               z: in STD_LOGIC;
               sum: out STD_LOGIC;
           carry : out STD_LOGIC);
end component;
Signal x1:STD_LOGIC:='0';
Signal y1:STD_LOGIC:='0';
Signal z1:STD_LOGIC:='0';
Signal s1:STD_LOGIC;
Signal c1:STD_LOGIC;
begin
```

```
UUT: FULL_ADDER_structural Port map(x=>x1, y=>y1, z=>z1, sum=>s1,
carry=>c1);

stim_proc: process

begin

wait for 50ns;

x1<='0';

y1<='0';

z1<='1';

wait for 50ns;

x1<='0';

y1<='1';

z1<='0';

wait for 50ns;

x1<='0';

y1<='1';

z1<='1';

wait for 50ns;

x1<='1';

y1<='0';

z1<='0';

wait for 50ns;

x1<='1';

y1<='0';

z1<='1';

wait for 50ns;

x1<='1';

y1<='1';

z1<='0';

wait for 50ns;
```

```
    x1<='1';
    y1<='1';

    z1<='1';
    wait for 50ns;
    end process;
    end Structural;
```

# TBW WaveForm

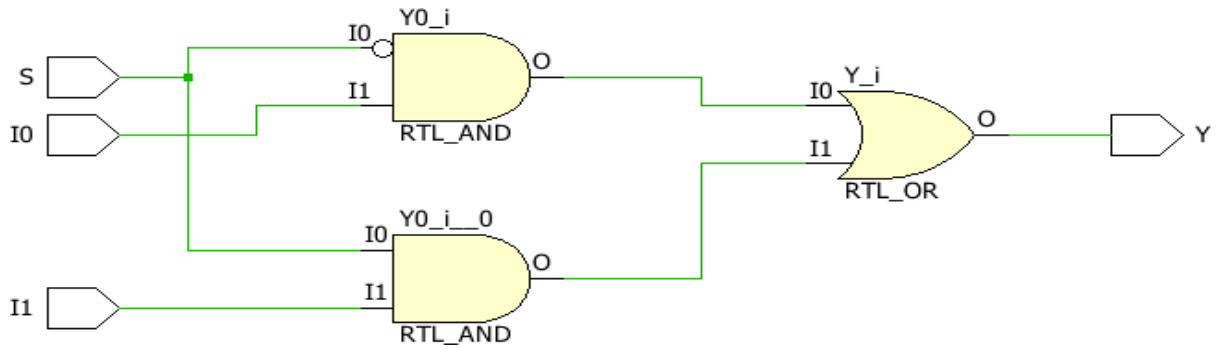



# 2:1 MUX Dataflow Model

## VHD Code:

```
entity MUX_2_1_DF is
    Port ( I0 : in STD_LOGIC;
           I1 : in STD_LOGIC;
           S : in STD_LOGIC;
           Y : out STD_LOGIC);
end MUX_2_1_DF;
architecture Dataflow of MUX_2_1_DF is
begin
Y<=((NOT S) AND I0) OR (S AND I1);
end Dataflow;
```

# RTL Diagram

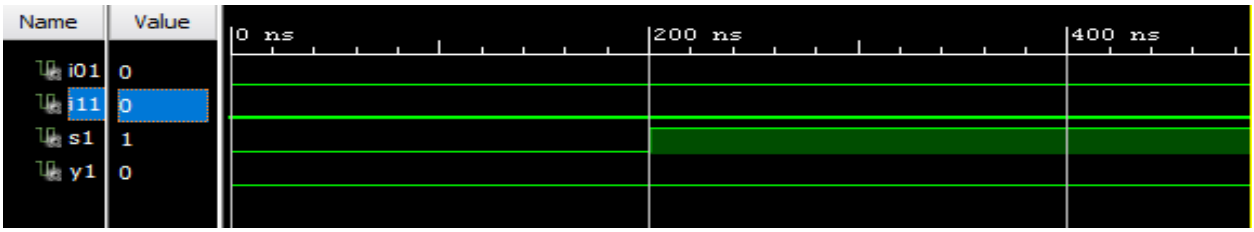



# TBW Code:

```
entity MUX_2_1_TBW is
-- Port ( );
end MUX_2_1_TBW;
architecture Dataflow of MUX_2_1_TBW is
component MUX_2_1_BV is
    Port ( I0 : in STD_LOGIC;
           I1 : in STD_LOGIC;
           S : in STD_LOGIC;
           Y : out STD_LOGIC);
end component;
signal i01: STD_LOGIC:='0';
signal i11: STD_LOGIC:='0';
signal s1: STD_LOGIC:='0';
signal y1: STD_LOGIC;
begin
UUT: MUX_2_1_DF Port map(I0=>i01, I1=>i11, S=>s1, Y=>y1);
```

```
stim_proc: process
begin
wait for 100ns;
s1<='0';
wait for 100ns;
s1<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform

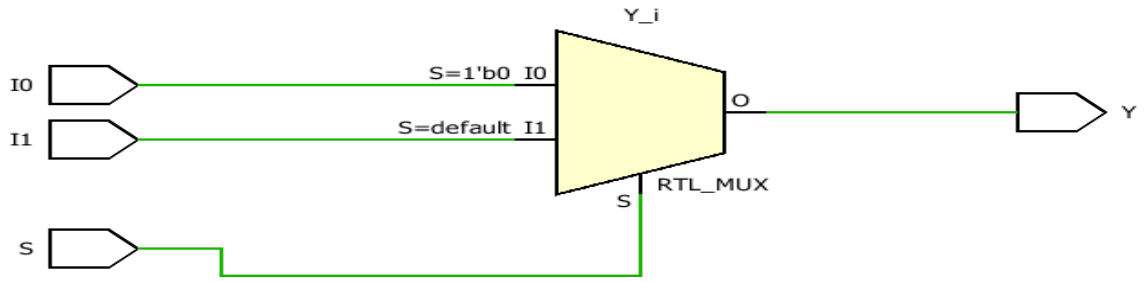



# 2:1 MUX Behavioral Model

## VHD Code:

```
entity MUX_2_1_DF is
    Port ( I0 : in STD_LOGIC;
           I1 : in STD_LOGIC;
           S : in STD_LOGIC;
           Y : out STD_LOGIC);
end MUX_2_1_DF;
architecture Behavioral of MUX_2_1_DF is
begin
process(I0,I1,S)
begin
    if(S='0') then
        Y <= I0;
    else
        Y<= I1;
    end if;
end process;
end Behavioral;
```

# RTL Diagram

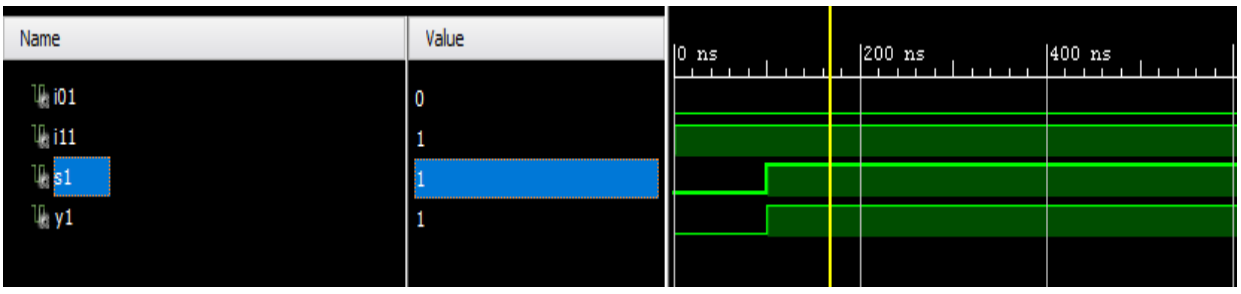



# TBW Code:

```
entity MUX_2_1_TBW is
-- Port ( );
end MUX_2_1_TBW;
```

```
architecture Behavioral of MUX_2_1_TBW is
component MUX_2_1_DF is
    Port ( I0 : in STD_LOGIC;
           I1: in STD_LOGIC;
               S: in STD_LOGIC;
           Y : out STD_LOGIC);
end component;
Signal i01:STD_LOGIC:='0';
Signal i11:STD_LOGIC:='0';
Signal s1:STD_LOGIC:='0';
Signal y1:STD_LOGIC;
begin
UUT: MUX_2_1_DF Port map(I0=>i01, I1=>i11, S=>s1, Y=>y1);
stim_proc: process
begin
wait for 100ns;
s1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

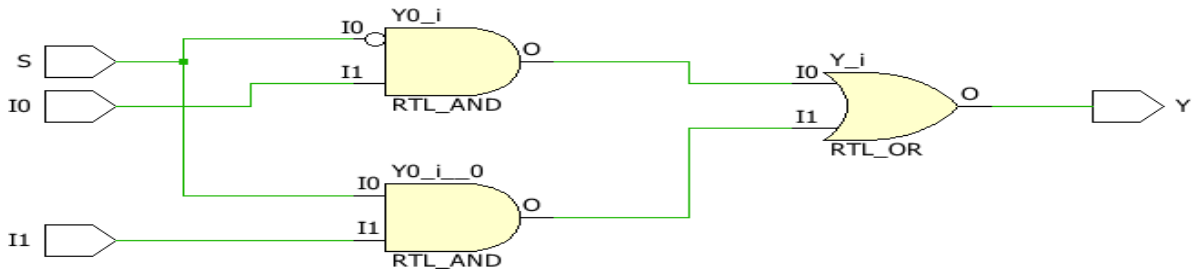


# 2:1 MUX Dataflow Model

## VHD Code:

```
entity MUX_2_1_dataflow is
    Port ( I0 : in STD_LOGIC;
           I1 : in STD_LOGIC;
           S : in STD_LOGIC;
           Y : out STD_LOGIC);
end MUX_2_1_dataflow;
architecture Dataflow of MUX_2_1_dataflow is
begin
Y <= (((not S) and I0) or (S and I1));
end Dataflow;
```

## RTL Diagram

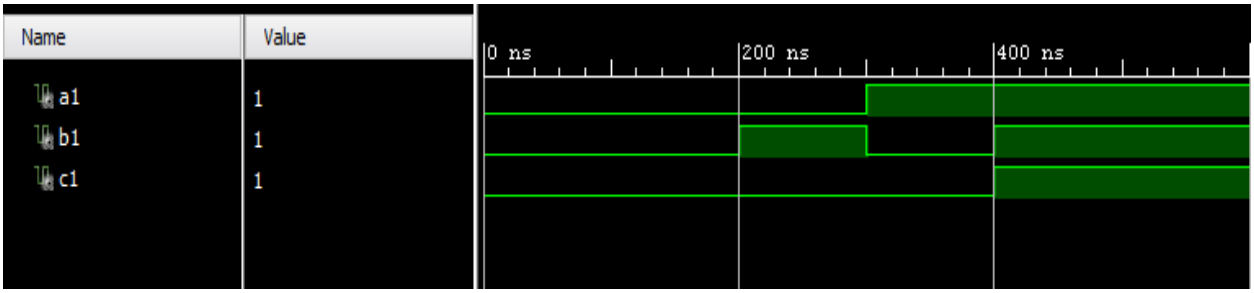



## TBW Code:

```
entity MUX_2_1_TBW is
-- Port ( );
```

```
end MUX_2_1_TBW;
architecture Behavioral of MUX_2_1_TBW is
component MUX_2_1_DF is
    Port ( I0 : in STD_LOGIC;
           I1: in STD_LOGIC;
               S: in STD_LOGIC;
           Y : out STD_LOGIC);
end component;
Signal i01:STD_LOGIC:='0';
Signal i11:STD_LOGIC:='0';
Signal s1:STD_LOGIC:='0';
Signal y1:STD_LOGIC;
begin
UUT: MUX_2_1_DF Port map(I0=>i01, I1=>i11, S=>s1, Y=>y1);
stim_proc: process
begin
wait for 100ns;
s1<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform

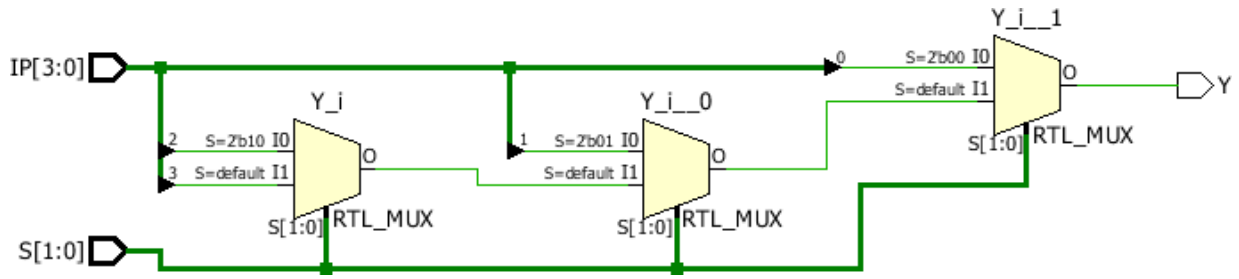


# 4:1 MUX Dataflow Model

## VHD Code:

```
entity MUX_4_1_DF is
    Port ( IP : in STD_LOGIC_VECTOR (3 downto 0);
           S : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC);
end MUX_4_1_DF;
architecture Dataflow of MUX_4_1_DF is
begin
Y <= IP(0) when S="00" else
     IP(1) when S="01" else
     IP(2) when S="10" else
     IP(3);

end Dataflow;
```

## RTL Diagram

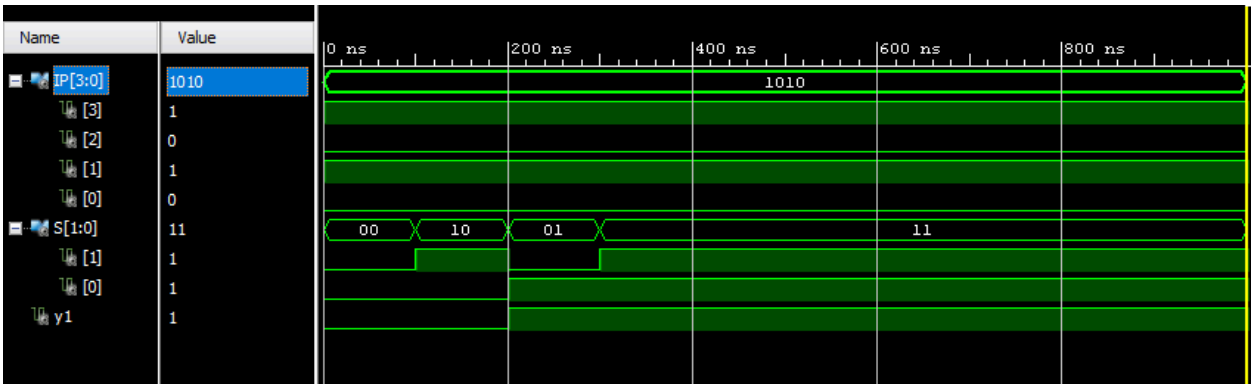



# TBW Code:

```
entity MUX_4_1_TBW is
-- Port ( );
end MUX_4_1_TBW;
architecture Behavioral of MUX_4_1_TBW is
component MUX_4_1_BV is
    Port ( IP : in STD_LOGIC_VECTOR (3 downto 0);
           S : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC);
end component;
Signal IP:STD_LOGIC_VECTOR(3 downto 0):="1010";
Signal S:STD_LOGIC_VECTOR(1 downto 0):="00";
Signal y1:STD_LOGIC;
begin
UUT: MUX_4_1_BV Port map(IP=>IP, S=>S, Y=>y1);
stim_proc: process
begin
```

```
wait for 100ns;
S(0)<='0';
S(1)<='1';
wait for 100ns;
S(0)<='1';
S(1)<='0';
wait for 100ns;
S(0)<='1';
S(1)<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform



# 4:1 MUX Behavioral Model

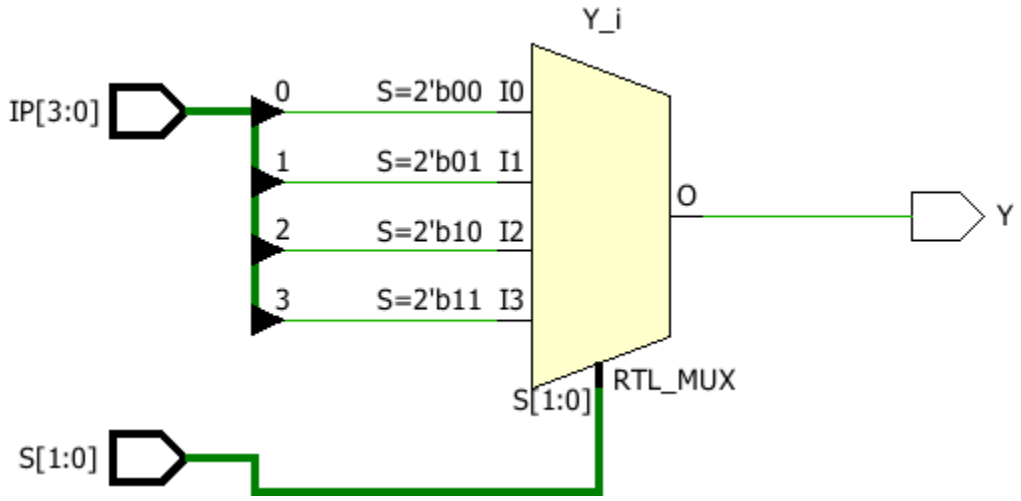
## VHD Code:

```
entity MUX_4_1_BV is
    Port ( IP : in STD_LOGIC_VECTOR (3 downto 0);
           S : in STD_LOGIC_VECTOR (1 downto 0);
           Y : out STD_LOGIC);
end MUX_4_1_BV;
architecture Behavioral of MUX_4_1_BV is
begin
process(IP,S)
begin
    case S is
        when "00" => Y <= IP(0);
        when "01" => Y <= IP(1);
        when "10" => Y <= IP(2);
        when "11" => Y <= IP(3);
```

```
        when others => NULL;
    end case;
end process;
end Behavioral;
```

---

# RTL Diagram

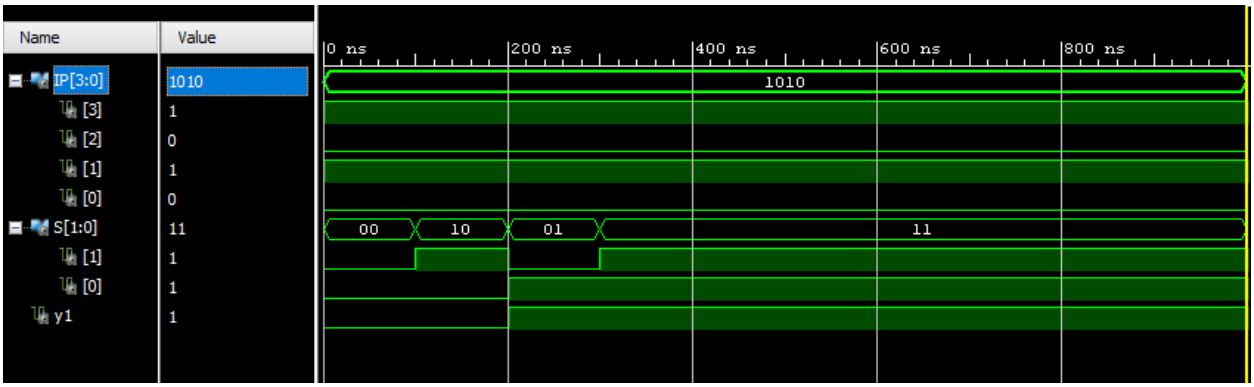



# TBW Code:

```
entity MUX_4_1_TBW is
-- Port ( );
end MUX_4_1_TBW;
architecture Behavioral of MUX_4_1_TBW is
component MUX_4_1_BV is
    Port ( IP : in STD_LOGIC_VECTOR (3 downto 0);
           S : in STD_LOGIC_VECTOR (1 downto 0);
```

```
        Y : out STD_LOGIC);
end component;
Signal IP:STD_LOGIC_VECTOR(3 downto 0):="1010";
Signal S:STD_LOGIC_VECTOR(1 downto 0):="00";
Signal y1:STD_LOGIC;
begin
UUT: MUX_4_1_BV Port map(IP=>IP, S=>S, Y=>y1);
stim_proc: process
begin
wait for 100ns;
S(0)<='0';
S(1)<='1';
wait for 100ns;
S(0)<='1';
S(1)<='0';
wait for 100ns;
S(0)<='1';
S(1)<='1';
wait;
end process;
end Behavioral;
```

# TBW Waveform



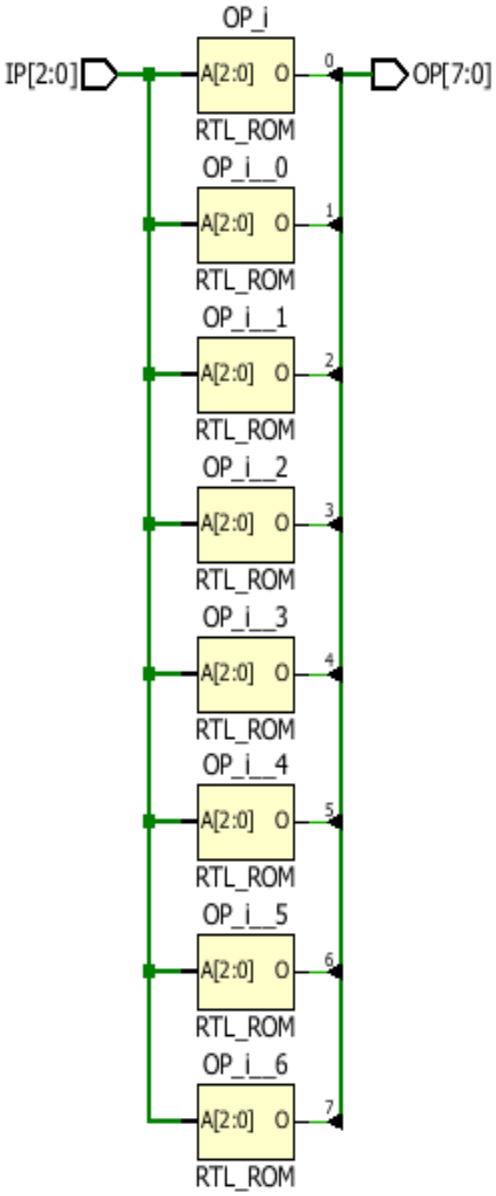

# 3:8 Decoder Dataflow Model

## VHD Code:

```
entity DECODER_3_8_DF is
    Port ( IP : in STD_LOGIC_VECTOR (2 downto 0);
           OP : out STD_LOGIC_VECTOR (7 downto 0));
end DECODER_3_8_DF;

architecture Dataflow of DECODER_3_8_DF is

begin
OP(0) <='1' when IP = "000" else '0';
OP(1) <='1' when IP = "001" else '0';
OP(2) <='1' when IP = "010" else '0';
OP(3) <='1' when IP = "011" else '0';
OP(4) <='1' when IP = "100" else '0';
```

```
OP(5) <='1' when IP = "101" else '0';

OP(6) <='1' when IP = "110" else '0';

OP(7) <='1' when IP = "111" else '0';


end Dataflow;
```

# RTL Diagram

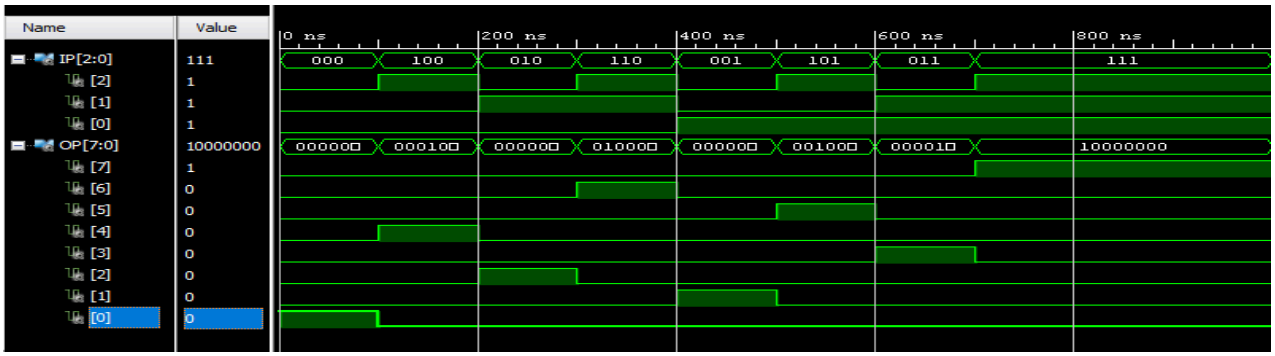


# TBW Code:

```
entity DECODER_3_8_TBW is
--  Port ( );
end DECODER_3_8_TBW;
architecture Dataflow of DECODER_3_8_TBW is
component DECODER_3_8_DF is
    Port ( IP : in STD_LOGIC_VECTOR (2 downto 0);
           OP : out STD_LOGIC_VECTOR (7 downto 0));
end component;
Signal IP:STD_LOGIC_VECTOR(2 downto 0):="000";
Signal OP:STD_LOGIC_VECTOR(7 downto 0);
begin
UUT: DECODER_3_8_DF Port map(IP=>IP, OP=>OP);
stim_proc: process
begin
wait for 100ns;
IP(0)<='0';
IP(1)<='0';
IP(2)<='1';
wait for 100ns;
IP(0)<='0';
IP(1)<='1';
IP(2)<='0';
wait for 100ns;
IP(0)<='0';
IP(1)<='1';
IP(2)<='1';
wait for 100ns;
IP(0)<='1';
```

```
IP(1)<='0';
IP(2)<='0';
wait for 100ns;
IP(0)<='1';
IP(1)<='0';
IP(2)<='1';
wait for 100ns;
IP(0)<='1';
IP(1)<='1';
IP(2)<='0';
wait for 100ns;
IP(0)<='1';
IP(1)<='1';
IP(2)<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform



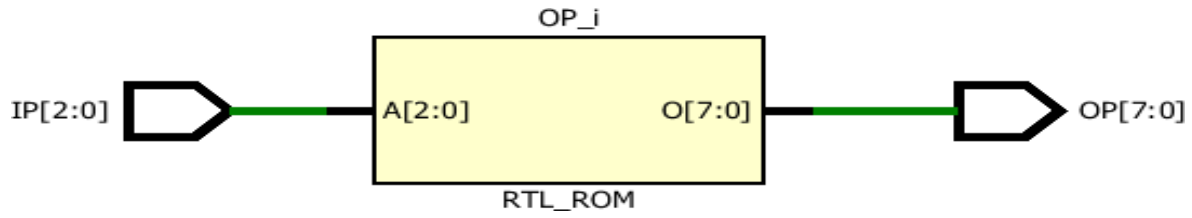

# 3:8 Decoder Behavioral Model

## VHD Code:

```
entity Decoder_3_8_BV is
```

```
    Port ( IP : in STD_LOGIC_VECTOR (2 downto 0);
           OP : out STD_LOGIC_VECTOR (7 downto 0));

    end Decoder_3_8_BV;

architecture Behavioral of Decoder_3_8_BV is
begin process(IP)
begin
    OP<="00000000";
    case IP is
        when "000" => OP(0) <= '1';
        when "001" => OP(1) <= '1';
        when "010" => OP(2) <= '1';
        when "011" => OP(3) <= '1';
        when "100" => OP(4) <= '1';
        when "101" => OP(5) <= '1';
        when "110" => OP(6) <= '1';
        when "111" => OP(7) <= '1';
        when others => NULL;
    end case;
end process;
end Behavioral;
```

# RTL Diagram

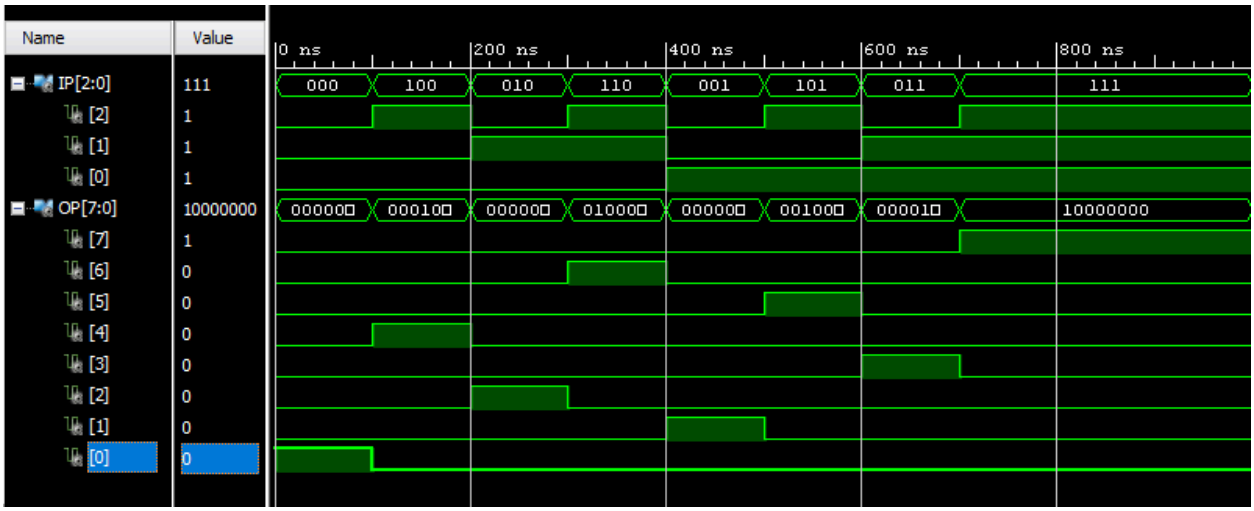


# TBW Code:

```
entity DECODER_3_8_TBW is
-- Port ( );
end DECODER_3_8_TBW;
architecture Dataflow of DECODER_3_8_TBW is
component DECODER_3_8_DF is
    Port ( IP : in STD_LOGIC_VECTOR (2 downto 0);
           OP : out STD_LOGIC_VECTOR (7 downto 0));
end component;
Signal IP:STD_LOGIC_VECTOR(2 downto 0):="000";
Signal OP:STD_LOGIC_VECTOR(7 downto 0);
begin
UUT: DECODER_3_8_DF Port map(IP=>IP, OP=>OP);
stim_proc: process
begin
wait for 100ns;
IP(0)<='0';
IP(1)<='0';
IP(2)<='1';
wait for 100ns;
IP(0)<='0';
IP(1)<='1';
IP(2)<='0';
wait for 100ns;
IP(0)<='0';
IP(1)<='1';
IP(2)<='1';
```

```
wait for 100ns;
IP(0)<='1';
IP(1)<='0';
IP(2)<='0';
wait for 100ns;
IP(0)<='1';
IP(1)<='0';
IP(2)<='1';
wait for 100ns;
IP(0)<='1';
IP(1)<='1';
IP(2)<='0';
wait for 100ns;
IP(0)<='1';
IP(1)<='1';
IP(2)<='1';
wait;
end process;
end Dataflow;
```

# TBW Waveform


Name
Value
0 ns
200 ns
400 ns
600 ns
800 ns
IP[2:0]
111
000
100
010
110
001
101
011
111
[2]
1
[1]
1
[0]
1
OP[7:0]
10000000
000000
000100
000000
010000
000000
001000
000010
10000000
[7]
1
[6]
0
[5]
0
[4]
0
[3]
0
[2]
0
[1]
0
[0]
0