

**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 & Behavioral Model) 2

OR GATE (Dataflow & Behavioral Model)8

NOT GATE (Dataflow & Behavioral Model) 14

NAND GATE (Dataflow & Behavioral Model) 19

NOR GATE (Dataflow & Behavioral Model) 25

XOR GATE (Dataflow & Behavioral Model) 31

XNOR GATE (Dataflow & Behavioral Model)37

AND GATE using NAND GATE 43

OR GATE using NAND GATE 46

NOT GATE using NAND GATE 49

XOR GATE using NAND GATE 51

XNOR GATE using NAND GATE 54

AND GATE using NOR GATE 57

OR GATE using NOR GATE 60

NOT GATE using NOR GATE 63

XOR GATE using NOR GATE 65

XNOR GATE using NOR GATE 68

HALF ADDER (Dataflow, Behavioral & Structural Model) 71

FULL ADDER (Dataflow, Behavioral & Structural Model) 80

2:1 MUX (Dataflow & Behavioral Model) 91

4:1 MUX (Dataflow & Behavioral Model) 97

3:8 DECODER (Dataflow & Behavioral Model) 103

# AND Gate Dataflow Model

**VHD Code:**

entity AND\_DF is

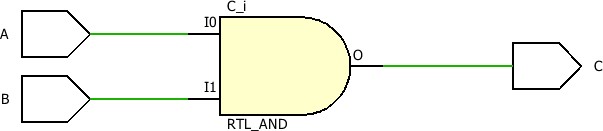
Port ( A : in STD\_LOGIC;

1. : in STD\_LOGIC;
2. : 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';

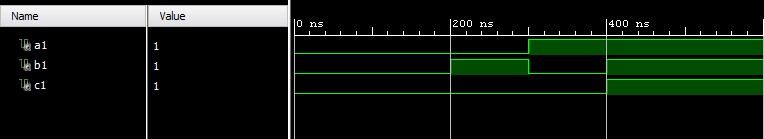
wait for 100ns;

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;

1. : in STD\_LOGIC;
2. : out STD\_LOGIC);

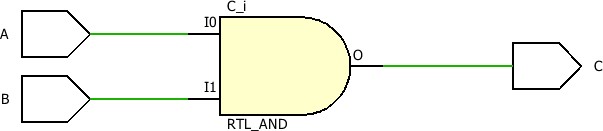
end AND\_GATE\_BV;

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;

1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end component;

Signal a1:STD\_LOGIC:='0';

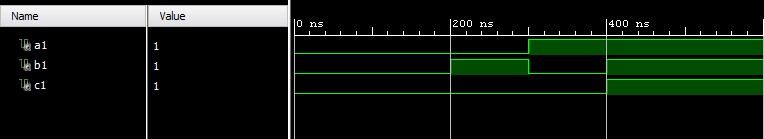
Signal b1:STD\_LOGIC:='0';

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



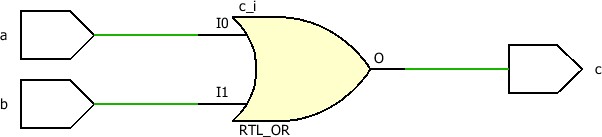
# OR Gate Dataflow Model

**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';

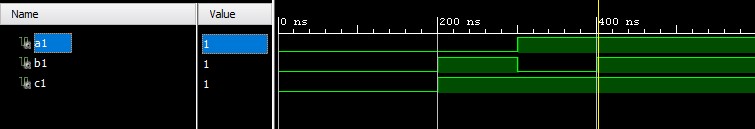
wait for 100ns;

a1<='1';

b1<='1';

wait; end process; end Dataflow;

## TBW Waveform



# OR Gate Behavioral Model

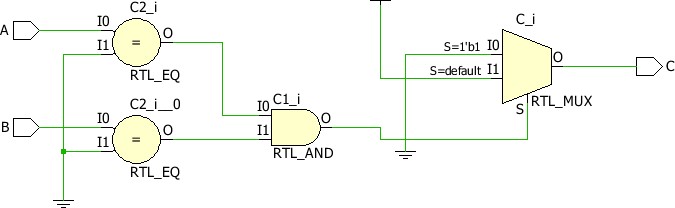
**VHD Code:**

entity OR\_GATE\_BV is

Port ( A : in STD\_LOGIC;

1. : in STD\_LOGIC;
2. : 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

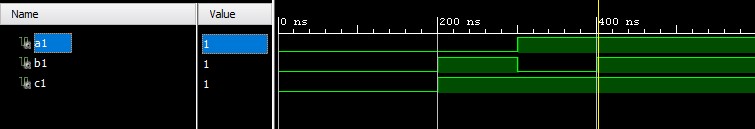
Port ( a : in STD\_LOGIC; b : in STD\_LOGIC; c : out STD\_LOGIC); end component;

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:**



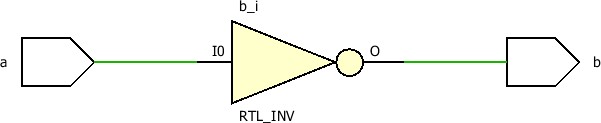
entity NOT\_DF is

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

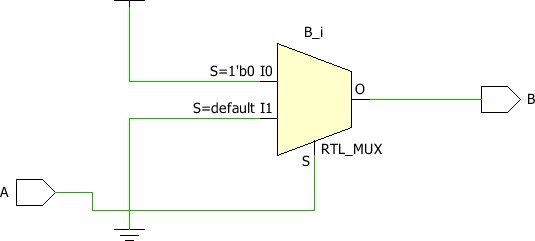
STD\_LOGIC;

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;

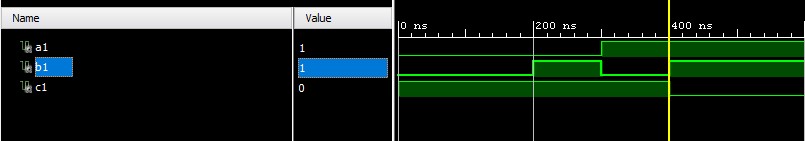
Signal a1:STD\_LOGIC:='0';

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:**

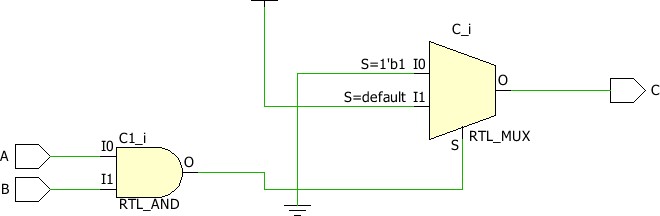
entity NAND\_GATE\_BV is

Port ( A : in STD\_LOGIC;

1. : in STD\_LOGIC;
2. : 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;

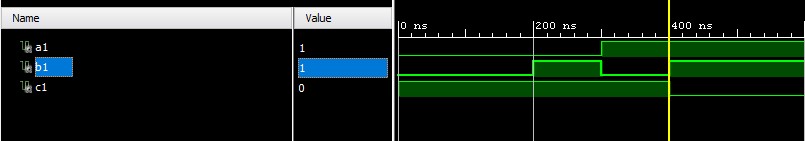
Signal a1:STD\_LOGIC:='0';

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;

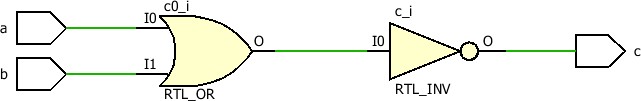
architecture Dataflow of NOR\_DF is

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;

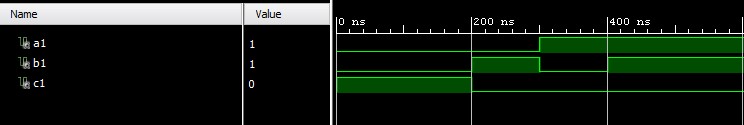
a1<='1'; b1<='0'; wait for 100ns;

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;

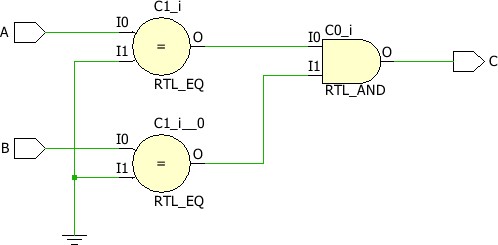
1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end NOR\_GATE\_BV; architecture Behavioral of NOR\_GATE\_BV is begin process(A,B) begin if(A='0' and B='0') then

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;

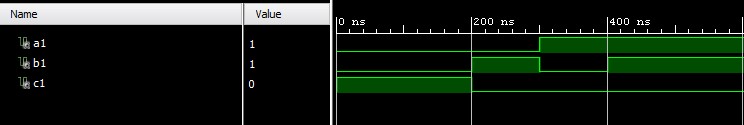
C : out STD\_LOGIC); end component;

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;

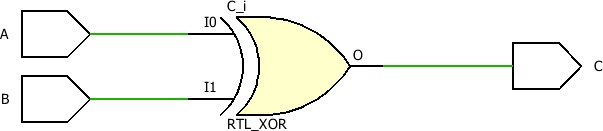
architecture Dataflow of XOR\_DF is

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;

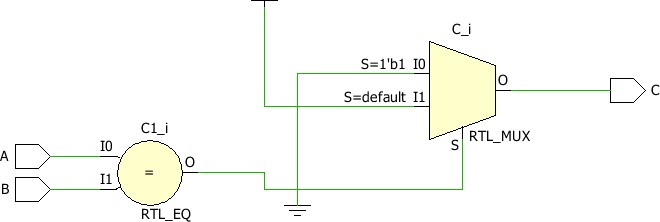
1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end XOR\_GATE\_BV; architecture Behavioral of XOR\_GATE\_BV is begin process(A,B) begin if(A=B) then

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;

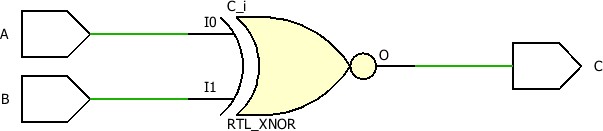
architecture Dataflow of XNOR\_DF is

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';

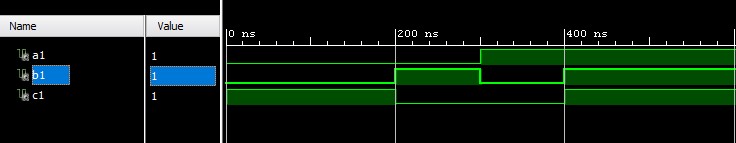
Signal b1:STD\_LOGIC:='0'; Signal c1:STD\_LOGIC; begin

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;

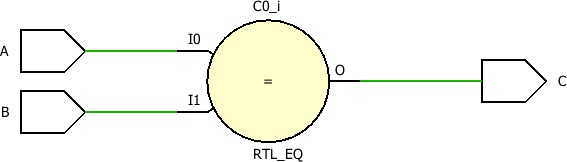
1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end XOR\_GATE\_BV; architecture Behavioral of XNOR\_GATE\_BV is begin process(A,B) begin if(A=B) then

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;

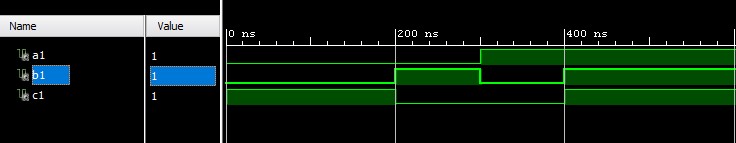
C : out STD\_LOGIC); end component;

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

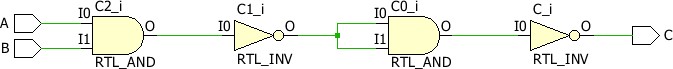
STD\_LOGIC;

1. : in STD\_LOGIC;
2. : 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;

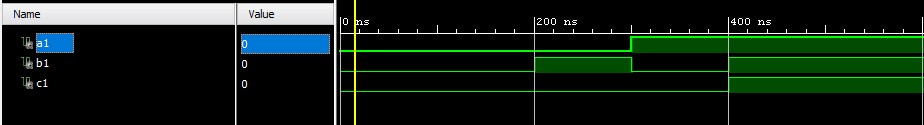
1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end component;

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

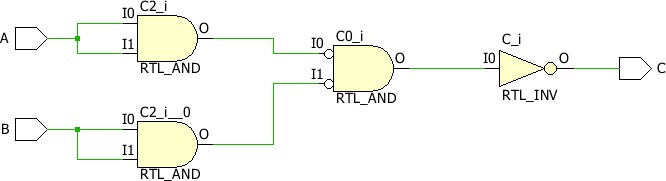
Port ( A : in STD\_LOGIC; B : in STD\_LOGIC;

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;

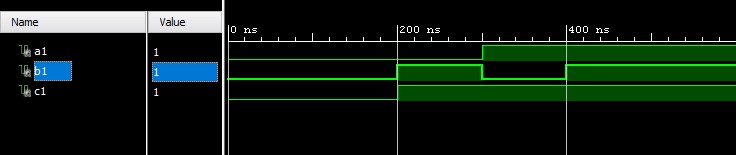
1. : in STD\_LOGIC;
2. : out STD\_LOGIC); end component;

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;

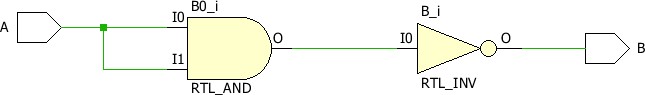
B : out STD\_LOGIC);

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

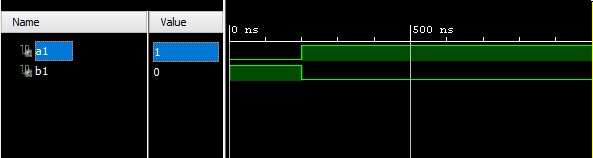
Port ( A : in STD\_LOGIC;

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;

1. : in STD\_LOGIC;
2. : out STD\_LOGIC);

end XOR\_NAND\_DF;

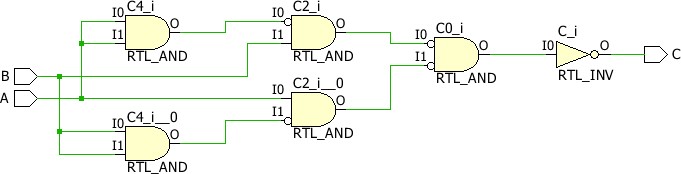
architecture Dataflow of XOR\_NAND\_DF is

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;

1. : in STD\_LOGIC;
2. : 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;

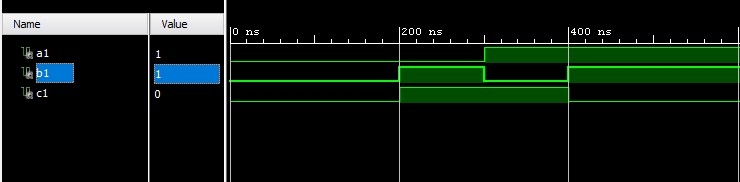
a1<='1';

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;

1. : in STD\_LOGIC;
2. : out STD\_LOGIC);

end XNOR\_NAND\_DF;

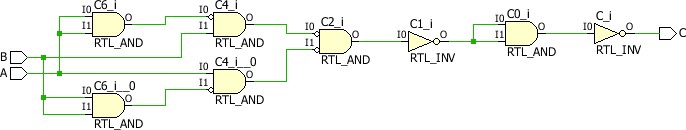
architecture Dataflow of XNOR\_NAND\_DF is

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;

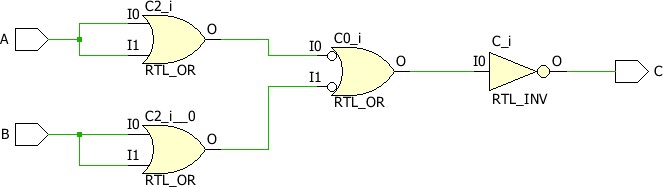
1. : in STD\_LOGIC;
2. : out STD\_LOGIC);

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;

1. : in STD\_LOGIC;
2. : 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;

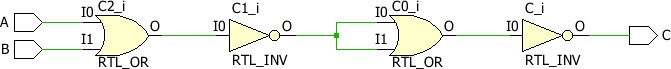
architecture Dataflow of OR\_NOR\_GATE\_DF is

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:**

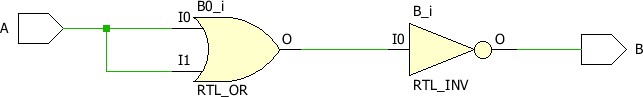
entity NOT\_NOR \_DF is

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;

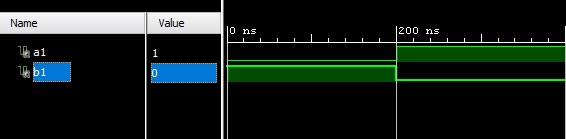
B : out STD\_LOGIC); end component;

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;

1. : in STD\_LOGIC;
2. : out STD\_LOGIC);

end XOR\_NOR\_GATE \_DF;

architecture Dataflow of XOR\_NOR \_GATE\_DF is

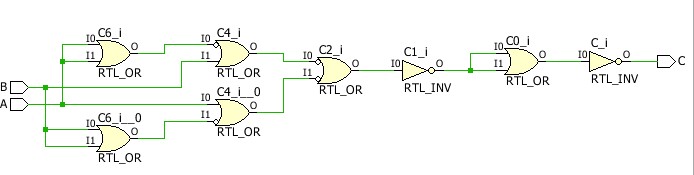
begin

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';

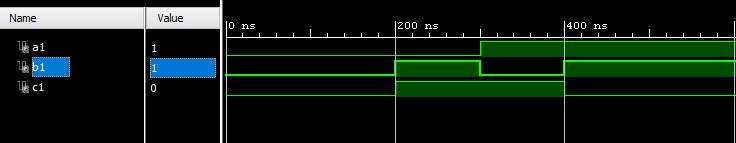
wait for 100ns;

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

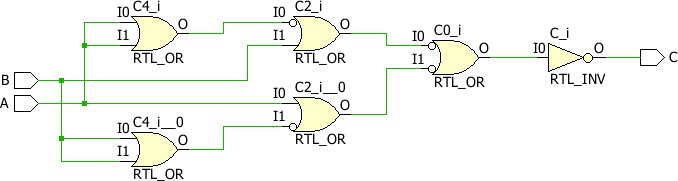
Port ( A : in STD\_LOGIC;

1. : in STD\_LOGIC;
2. : 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;

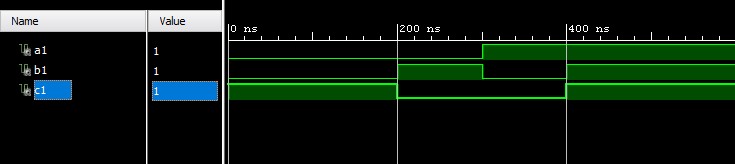
C : out STD\_LOGIC); end component;

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;

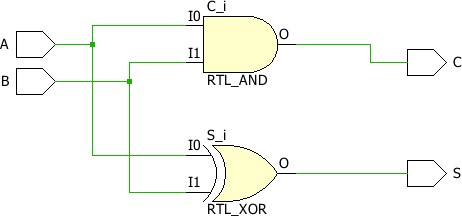
C : out STD\_LOGIC); end HALF\_ADDER\_DF; architecture Dataflow of HALF\_ADDER\_DF is begin

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;

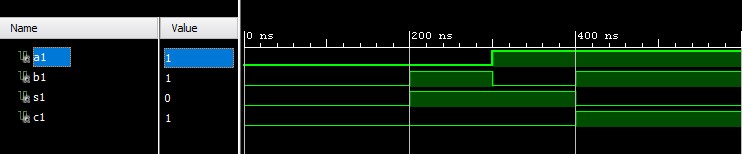
Signal a1:STD\_LOGIC:='0';

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

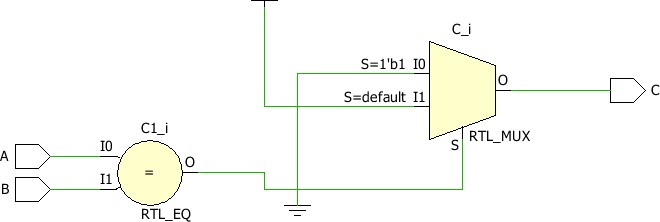
S<='0'; else

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;

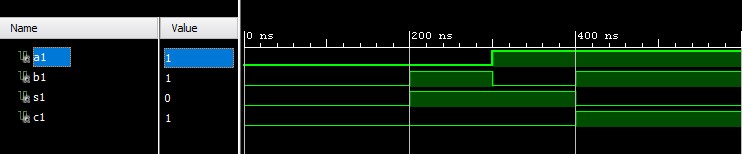
Signal a1:STD\_LOGIC:='0'; Signal b1:STD\_LOGIC:='0';

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;

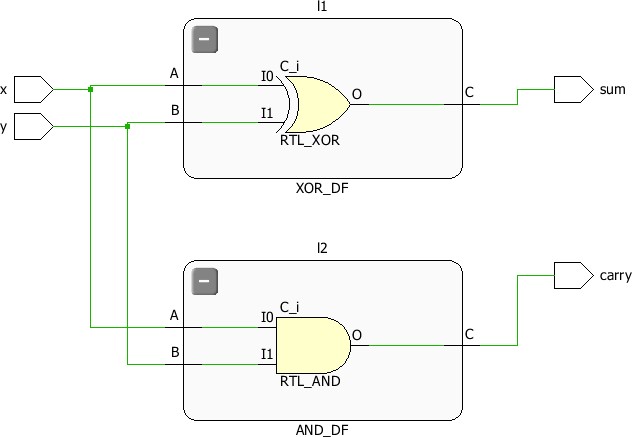
C : out STD\_LOGIC); end component;

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

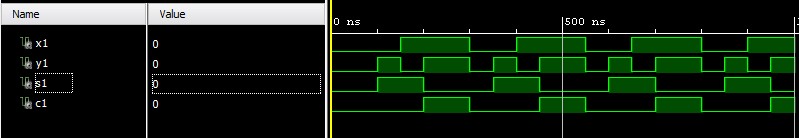
Port ( A : in STD\_LOGIC;

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;

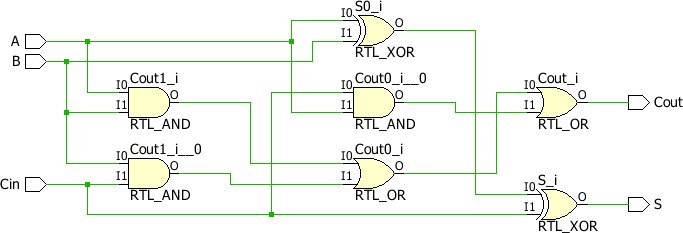
S : out STD\_LOGIC;

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

AI Page Translation is now live! Click here

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

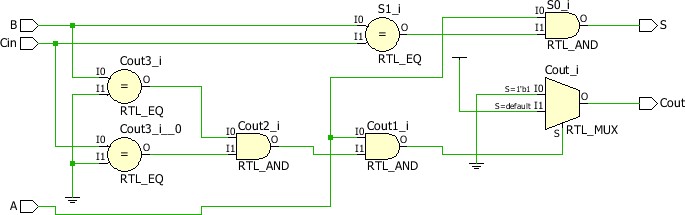
Cout<='1'; else

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;

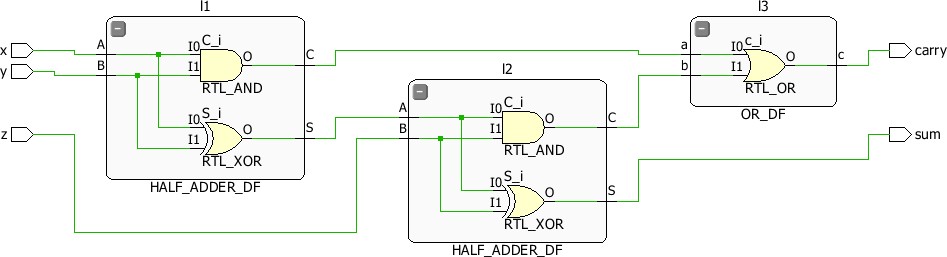
S : out STD\_LOGIC;

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';

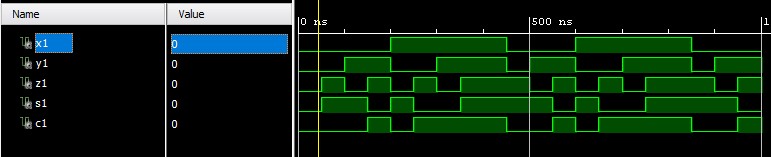
Signal s1:STD\_LOGIC; Signal c1:STD\_LOGIC; begin

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

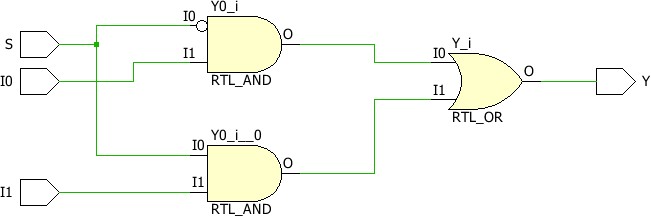
Port ( I0 : in STD\_LOGIC;

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;

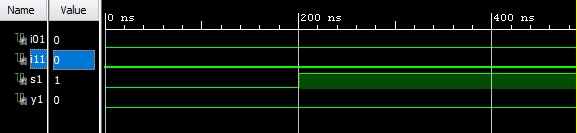
S : in STD\_LOGIC;

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;

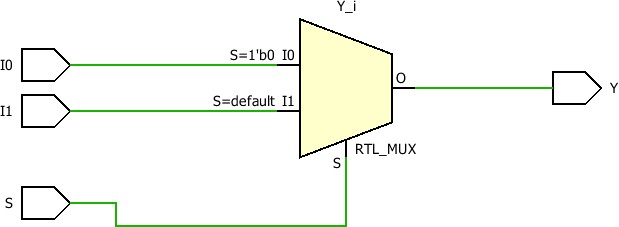
S : in STD\_LOGIC; Y : out STD\_LOGIC); end MUX\_2\_1\_DF; architecture Behavioral of MUX\_2\_1\_DF is begin process(I0,I1,S) begin if(S='0') then Y <= I0; else Y<= I1; end if;

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;

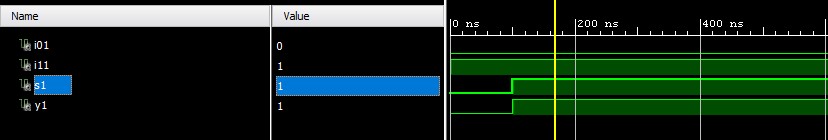
Signal i01:STD\_LOGIC:='0';

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;

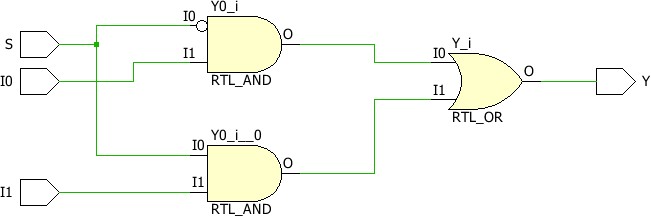
S : in STD\_LOGIC;

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;

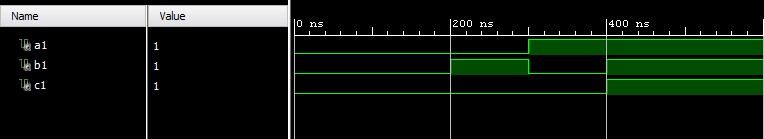
Signal i01:STD\_LOGIC:='0';

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

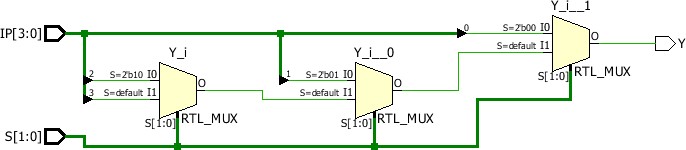
IP(2) when S="10" else

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;

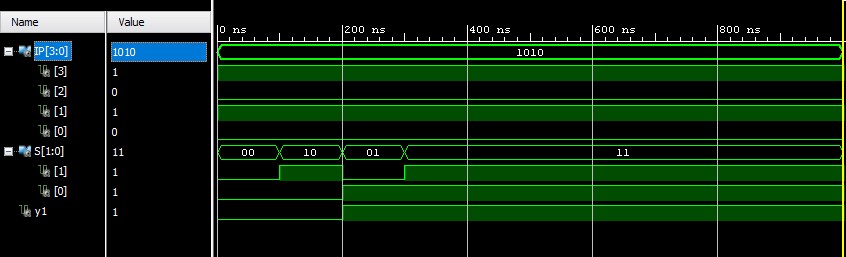
Signal IP:STD\_LOGIC\_VECTOR(3 downto 0):="1010";

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);

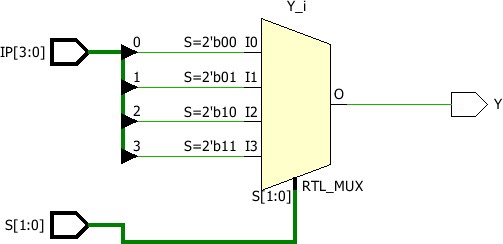
S : in STD\_LOGIC\_VECTOR (1 downto 0);

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);

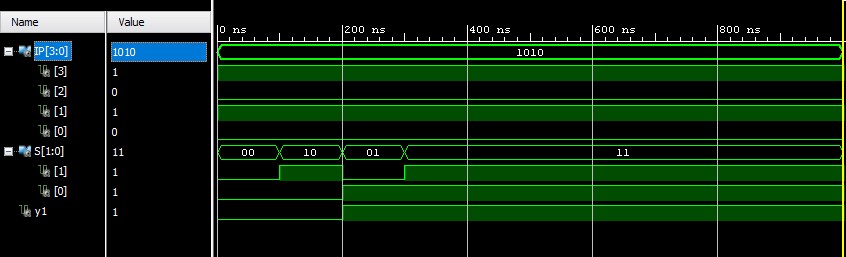
Y : out STD\_LOGIC); end component;

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';

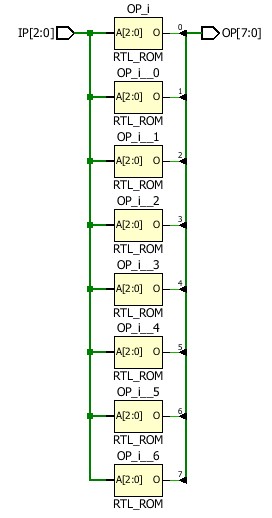
OP(4) <='1' when IP = "100" else '0';

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';

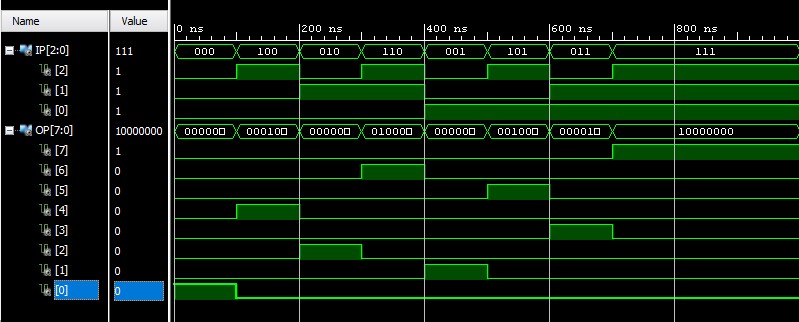
IP(1)<='1'; IP(2)<='1'; wait for 100ns; IP(0)<='1'; IP(1)<='0'; IP(2)<='0'; wait for 100ns; IP(0)<='1';

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));

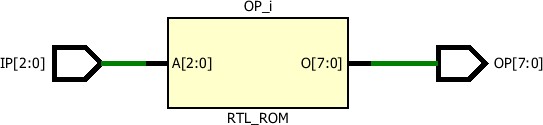
end Decoder\_3\_8\_BV;

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

