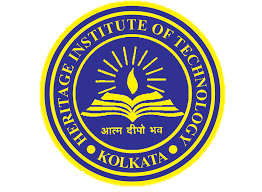
****

**HERITAGE INSTITUTE OF TECHNOLOGY**

**COMPUTER ARCHITECTURE LAB**

CSE2252

|  |  |  |
| --- | --- | --- |
| **NAME** | **ROLL NO.** | **AUTONOMY ROLL NO.** |
| SAUMYA TIWARI | 2362057 |  |
| ROHIT DUTTA | 2362059 |  |
| SUKRITI KUMARI | 2362061 |  |
| DEBAYUSH DEY | 2362062 |  |
| ABHINAV MISHRA | 2362063 |  |
| VEDANT MEHTA | 2362064 |  |
| NAVNEET VERMA | 2362065 |  |

**TABLE OF CONTENT**

**Basic Gates**

**Data Flow Model for AND Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**VHD Source Code:**

**AND\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity AND\_DF is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC);

end AND\_DF;

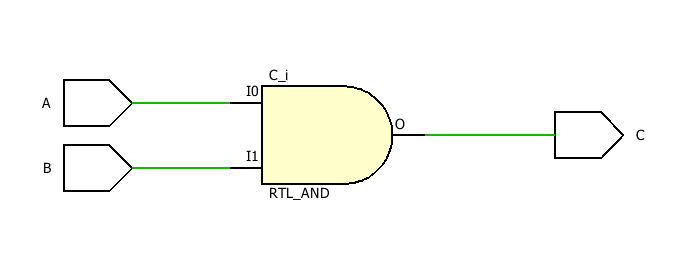
architecture Dataflow of AND\_DF is

begin

C <= A AND B;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**AND\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity AND\_DF\_TBW is

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<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

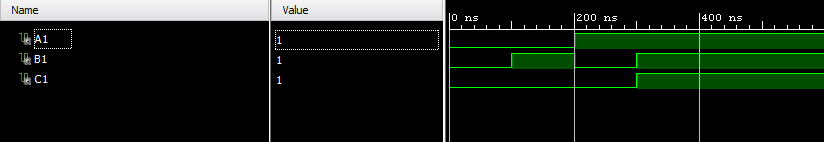
B1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for OR Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**VHD Source Code:**

**OR\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity OR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end OR\_DF;

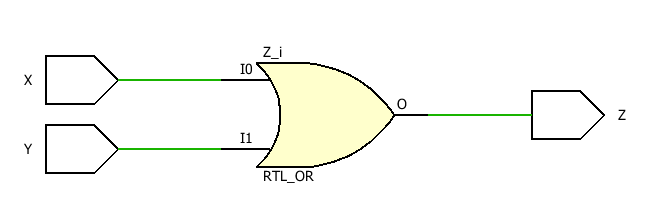
architecture Dataflow of OR\_DF is

begin

Z <= X OR Y;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**OR\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity OR\_DF\_TBW is

end OR\_DF\_TBW;

architecture Dataflow of OR\_DF\_TBW is

component OR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

Signal X1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC := '0';

Signal Z1 : STD\_LOGIC ;

begin

uut: OR\_DF port map(X=>X1,Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1<='0';

Y1<='1';

wait for 100ns;

X1<='1';

Y1<='0';

wait for 100ns;

X1<='1';

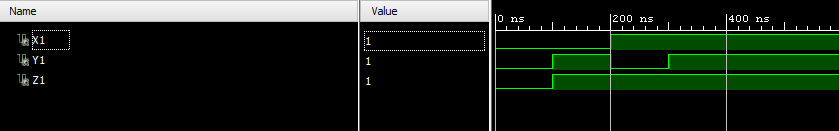
Y1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for NOT Gate**

**Truth Table:**

|  |  |
| --- | --- |
| X | Y |
| 0 | 1 |
| 1 | 0 |

**VHD Source Code:**

**NOT\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity NOT\_DF is

Port ( X : in STD\_LOGIC;

Y : out STD\_LOGIC);

end NOT\_DF;

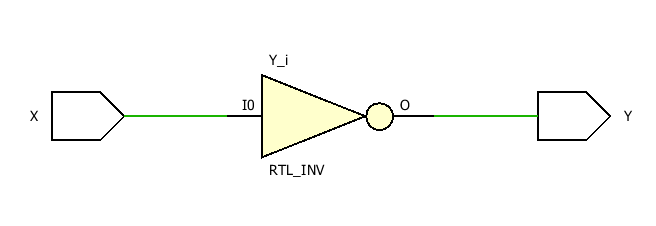
architecture Dataflow of NOT\_DF is

begin

Y<= NOT X;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**NOT\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity NOT\_DF\_TBW is

-- Port ( );

end NOT\_DF\_TBW;

architecture Dataflow of NOT\_DF\_TBW is

component NOT\_DF is

Port ( X : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

Signal X1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC;

begin

uut: NOT\_DF port map(X=>X1,Y=>Y1);

stim\_proc: process

begin

wait for 100ns;

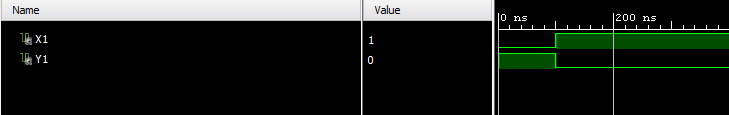
X1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for XOR Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**VHD Source Code:**

**XOR\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity XOR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end XOR\_DF;

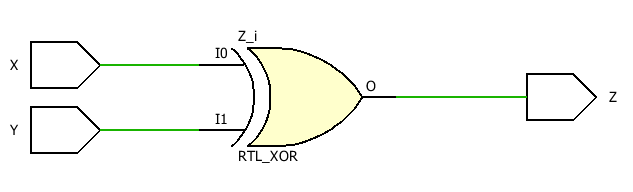
architecture Dataflow of OR\_DF is

begin

Z <= X XOR Y;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**XOR\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity XOR\_DF\_TBW is

-- Port ( );

end XOR\_DF\_TBW;

architecture Dataflow of XOR\_DF\_TBW is

component XOR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

Signal X1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC := '0';

Signal Z1 : STD\_LOGIC ;

begin

uut: XOR\_DF port map(X=>X1,Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1<='0';

Y1<='1';

wait for 100ns;

X1<='1';

Y1<='0';

wait for 100ns;

X1<='1';

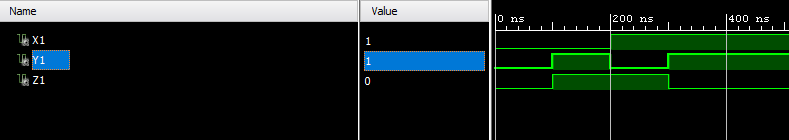
Y1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for NAND Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**VHD Source Code:**

**NAND\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity NAND\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end NAND\_DF;

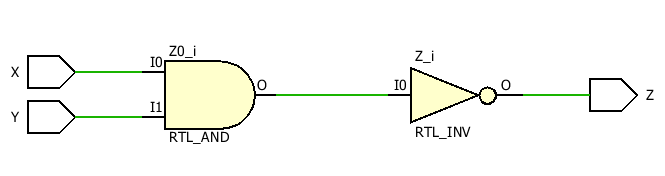
architecture Dataflow of NAND\_DF is

begin

C <= A NAND B;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**NAND\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity NAND\_DF\_TBW is

-- Port ( );

end NAND\_DF\_TBW;

architecture Dataflow of NAND\_DF\_TBW is

component NAND\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

Signal X1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC := '0';

Signal Z1 : STD\_LOGIC ;

begin

uut: NAND\_DF port map(X=>X1,Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1<='0';

Y1<='1';

wait for 100ns;

X1<='1';

Y1<='0';

wait for 100ns;

X1<='1';

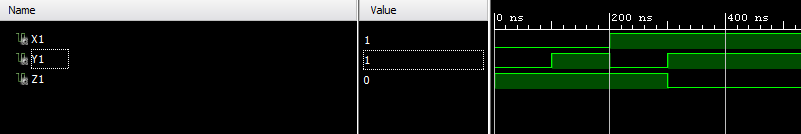
Y1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for XNOR Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**VHD Source Code:**

**XNOR\_DF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity XNOR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end XNOR\_DF;

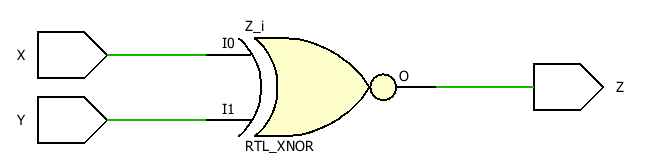
architecture Dataflow of XNOR\_DF is

begin

Z <= X XNOR Y;

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**XNOR\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity XNOR\_DF\_TBW is

end XNOR\_DF\_TBW;

architecture Dataflow of XNOR\_DF\_TBW is

component XNOR\_DF is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

Signal X1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC := '0';

Signal Z1 : STD\_LOGIC ;

begin

uut: XNOR\_DF port map(X=>X1,Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1<='0';

Y1<='1';

wait for 100ns;

X1<='1';

Y1<='0';

wait for 100ns;

X1<='1';

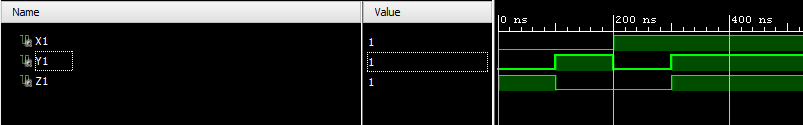
Y1<='1';

wait;

end process;

end Dataflow;

**Waveform:**



**Data Flow Model for NOT Gate Using NAND Gate**

**Truth Table:**

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

**VHD Source Code:**

**NOT\_NAND.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not\_nand is

Port ( A : in STD\_LOGIC;

B : out STD\_LOGIC);

end not\_nand;

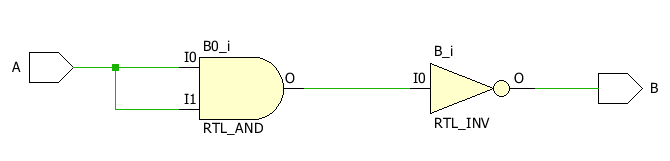
architecture dataflow of not\_nand is

begin

B<=A nand A;

end dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**Not\_nand\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not\_nand\_tbw is

-- Port ( );

end not\_nand\_tbw;

architecture dataflow of not\_nand\_tbw is

component not\_nand 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 port map(A=>A1,B=>B1);

stim\_proc: process

begin

wait for 100ns;

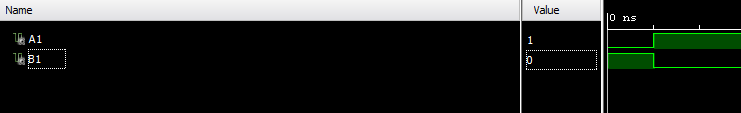
A1<='1';

wait;

end process;

end dataflow;

**Waveform:**

****

**Data Flow Model for OR Gate Using NAND Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**VHD Source Code:**

**or\_nand.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or\_nand is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC);

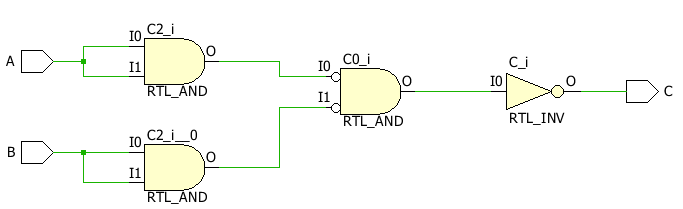
end or\_nand;

architecture dataflow of or\_nand is

begin

C<=(A nand A) nand (B nand B);

end dataflow;

**Schematic Diagram:**

**Test Bench Code:**

**or\_nand\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or\_nand\_tbw is

-- Port ( );

end or\_nand\_tbw;

architecture dataflow of or\_nand\_tbw is

component or\_nand 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\_nand port map(A=>A1,B=>B1, C=>C1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

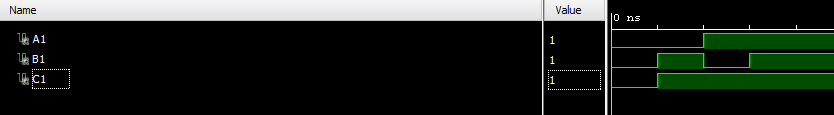
B1<='1';

wait;

end process;

end dataflow;

**Waveform:**

****

**Data Flow Model for AND Gate Using NAND Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**VHD Source Code:**

**And\_nand.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_nand is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC);

end and\_nand;

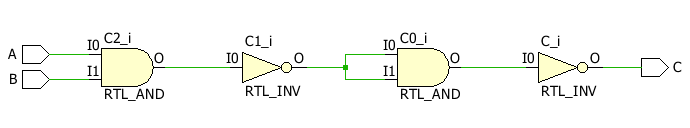
architecture dataflow of and\_nand is

begin

C<=(A nand B) nand (A nand B);

end dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**And\_nand\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_nand\_tbw is

-- Port ( );

end and\_nand\_tbw;

architecture dataflow of and\_nand\_tbw is

component and\_nand 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 port map(A=>A1, B=>B1, C=>C1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

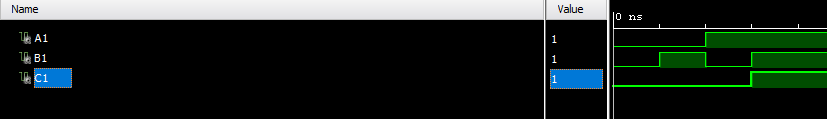
B1<='1';

wait;

end process;

end dataflow;

**Waveform:**



**Data Flow Model for NOT Gate Using NOR Gate**

**Truth Table:**

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

**VHD Source Code:**

**not\_nor.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not\_nor is

Port ( A : in STD\_LOGIC;

B : out STD\_LOGIC);

end not\_nor;

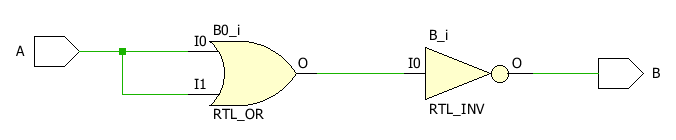
architecture dataflow of not\_nor is

begin

B<=A nor A;

end dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**Not\_nor\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not\_nor\_tbw is

-- Port ( );

end not\_nor\_tbw;

architecture dataflow of not\_nor\_tbw is

component not\_nor 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 port map(A=>A1,B=>B1);

stim\_proc: process

begin

wait for 100ns;

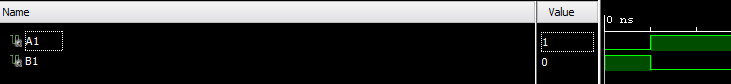
A1<='1';

wait;

end process;

end dataflow;

**Waveform:**

****

**Data Flow Model for OR Gate Using NOR Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**VHD Source Code:**

**or\_nor.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or\_nor is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC);

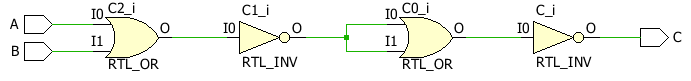
end or\_nor;

architecture dataflow of or\_nor is

begin

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

end dataflow;

**Schematic Diagram:** 

**Test Bench Code:**

**or\_nor\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or\_nor\_tbw is

-- Port ( );

end or\_nor\_tbw;

architecture dataflow of or\_nor\_tbw is

component or\_nor 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 port map(A=>A1,B=>B1, C=>C1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

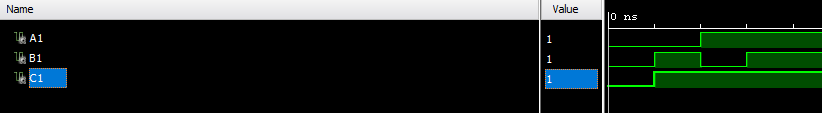
B1<='1';

wait;

end process;

end dataflow;

**Waveform:**

****

**Data Flow Model for AND Gate Using NOR Gate**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**VHD Source Code:**

**And\_nor.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_nor is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : out STD\_LOGIC);

end and\_nor;

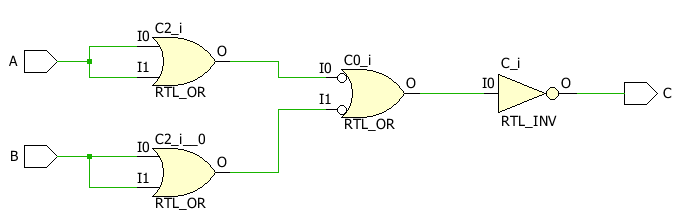
architecture dataflow of and\_nor is

begin

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

end dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**And\_nor\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_nor\_tbw is

-- Port ( );

end and\_nor\_tbw;

architecture dataflow of and\_nor\_tbw is

component and\_nor 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 port map(A=>A1, B=>B1, C=>C1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

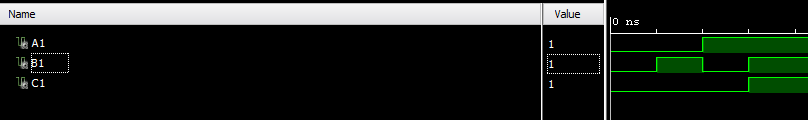
B1<='1';

wait;

end process;

end dataflow;

**Waveform:**



**Behavioral Model for Half Adder**

**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**VHD Source Code:**

**halfadder.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity halfadder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

S : out STD\_LOGIC;

C : out STD\_LOGIC);

end halfadder;

architecture Behavioral of halfadder is

begin

process(A, B)

begin

if(A = '0')then

S <= B;

C <= '0';

else

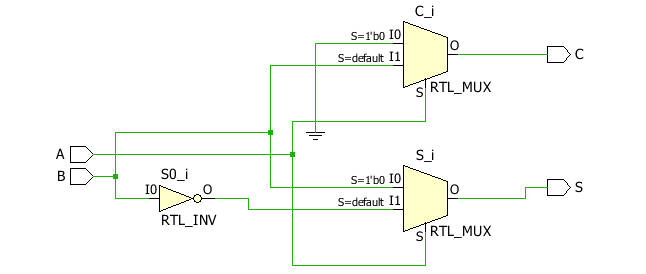
S <= not B;

C <= B;

end if;

end process;

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**Halfadder\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity halfadder\_tbw is

end halfadder\_tbw;

architecture Behavioral of halfadder\_tbw is

component halfadder 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: halfadder port map(A=>A1,B=>B1,S=>S1,C=>C1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='1';

wait for 100ns;

A1<='1';

B1<='0';

wait for 100ns;

A1<='1';

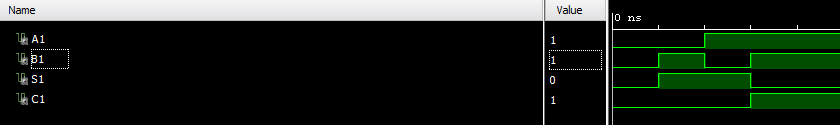
B1<='1';

wait;

end process;

end Behavioral;

**Waveform:**

****

**Behavioral Model for Full Adder**

**Truth Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | S | Ca |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**VHD Source Code:**

**fulladder.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fulladder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Ca : out STD\_LOGIC);

end fulladder;

architecture Behavioral of fulladder is

begin

process(A, B, C)

begin

if(A = '0') then

if(B = C) then

S <= '0';

Ca <= B;

else

S <= '1';

Ca <= '0';

end if;

else

if(B = C) then

S <= '1';

Ca <= B;

else

S <= '0';

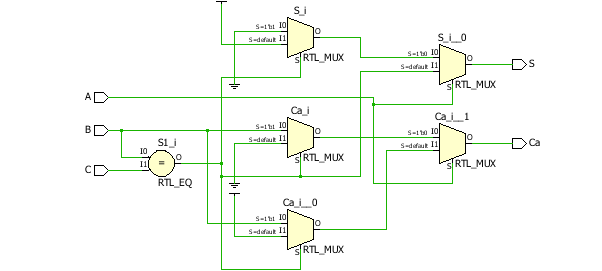
Ca <= '1';

end if;

end if;

end process;

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**fulladder\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fulladder\_tbw is

-- Port ( );

end fulladder\_tbw;

architecture Behavioral of fulladder\_tbw is

component fulladder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Ca : out STD\_LOGIC);

end component;

Signal A1 : STD\_LOGIC := '0';

Signal B1 : STD\_LOGIC := '0';

Signal C1 : STD\_LOGIC := '0';

Signal S1 : STD\_LOGIC ;

Signal Ca1 : STD\_LOGIC ;

begin

uut: fulladder port map(A=>A1,B=>B1,C=>C1,S=>S1,Ca=>Ca1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='0';

C1<='1';

wait for 100ns;

A1<='0';

B1<='1';

C1<='0';

wait for 100ns;

A1<='0';

B1<='1';

C1<='1';

wait for 100ns;

A1<='1';

B1<='0';

C1<='0';

wait for 100ns;

A1<='1';

B1<='0';

C1<='1';

wait for 100ns;

A1<='1';

B1<='1';

C1<='0';

wait for 100ns;

A1<='1';

B1<='1';

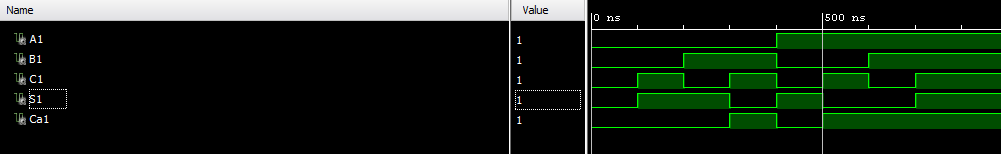
C1<='1';

wait;

end process;

end Behavioral;

**Waveform:**

**Behavioral Model for MUX2:1**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| I | S | Y |
| I(0)=0 | 0 | 0 |
| I(0)=1 | 0 | 1 |
| I(1)=0 | 1 | 0 |
| I(1)=1 | 1 | 0 |

**VHD Source Code:**

**MUX21.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MUX21 is

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

S : in STD\_LOGIC;

Y : out STD\_LOGIC);

end MUX21;

architecture Behavioral of MUX21 is

begin

process(I,S)

begin

if (S='0')then

y<=I(0);

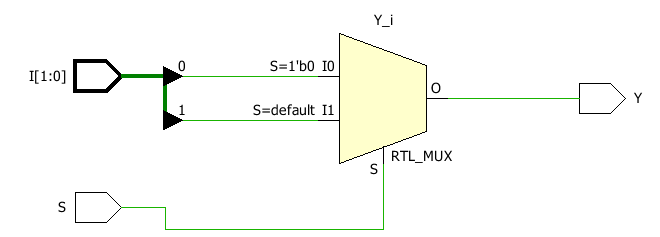
else

y<=I(1);

end if;

end process;

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**MUX21\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MUX21\_tbw is

-- Port ( );

end MUX21\_tbw;

architecture Behavioral of MUX21\_tbw is

component MUX21 is

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

S : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

Signal I0 : STD\_LOGIC := '0';

Signal I1 : STD\_LOGIC := '0';

Signal S1 : STD\_LOGIC := '0';

Signal Y1 : STD\_LOGIC ;

begin

uut: MUX21 port map(I(0)=>I0,I(1)=>I1,S=>S1,Y=>Y1);

stim\_proc: process

begin

wait for 100ns;

I0<='1';

wait for 100ns;

S1<='1';

I1<='0';

wait for 100ns;

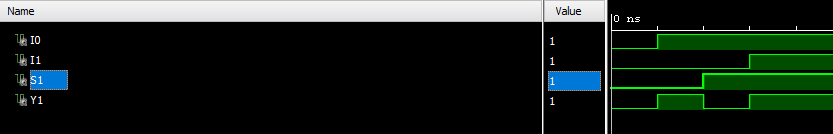
I1<='1';

wait;

end process;

end Behavioral;

**Waveform:**



**Dataflow Model for MUX4:1**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| S | | op |
| 0 | 0 | Ip(0) |
| 0 | 1 | Ip(1) |
| 1 | 0 | Ip(2) |
| 1 | 1 | Ip(3) |

**VHD Source Code:**

**MUX41.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux41 is

Port ( ip : in STD\_LOGIC\_VECTOR (3 downto 0);

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

op : out STD\_LOGIC);

end mux41;

architecture Dataflow of mux41 is

begin

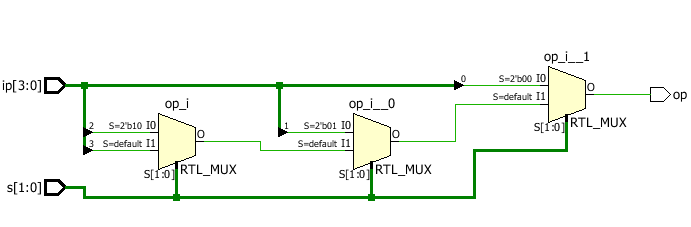
op <= ip(0) when s="00" else

ip(1) when s="01" else

ip(2) when s="10" else

ip(3);

end Dataflow;

**Schematic Diagram: **

**Test Bench Code:**

**MUX41\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux41\_tbw is

-- Port ( );

end mux41\_tbw;

architecture Dataflow of mux41\_tbw is

component mux41 is

Port ( ip : in STD\_LOGIC\_VECTOR (3 downto 0);

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

op : out STD\_LOGIC);

end component;

Signal ip1: STD\_LOGIC\_VECTOR (3 downto 0):="1010";

Signal s1: STD\_LOGIC\_VECTOR (1 downto 0):="00";

Signal op1: STD\_LOGIC;

begin

uut: mux41 port map(ip=>ip1,s=>s1,op=>op1);

stim\_proc: process

begin

wait for 100ns;

s1<="01";

wait for 100ns;

s1<="10";

wait for 100ns;

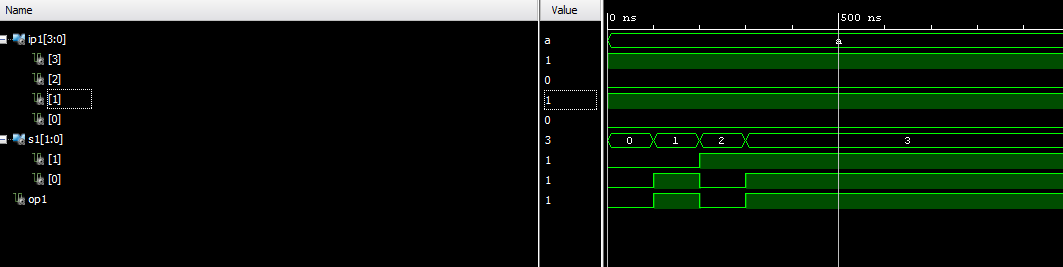
s1<="11";

wait;

end process;

end Dataflow;

**Waveform:**

****

**Behavioral Model for MUX4:1**

**Truth Table:**

|  |  |  |
| --- | --- | --- |
| S | | op |
| 0 | 0 | Ip(0) |
| 0 | 1 | Ip(1) |
| 1 | 0 | Ip(2) |
| 1 | 1 | Ip(3) |

**VHD Source Code:**

**MUX41\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux41\_b is

Port ( ip : in STD\_LOGIC\_VECTOR (3 downto 0);

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

op : out STD\_LOGIC);

end mux41\_b;

architecture Behavioral of mux41\_b is

begin

process(ip,s)

begin

case s is

when "00" => op<=ip(0);

when "01" => op<=ip(1);

when "10" => op<=ip(2);

when "11" => op<=ip(3);

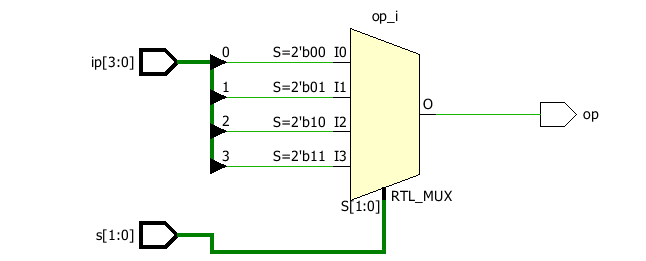
when others => NULL;

end case;

end process;

end Behavioral;

**Schematic Diagram:**



**Test Bench Code:**

**MUX41\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux41\_tbw is

-- Port ( );

end mux41\_tbw;

architecture Dataflow of mux41\_tbw is

component mux41\_b is

Port ( ip : in STD\_LOGIC\_VECTOR (3 downto 0);

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

op : out STD\_LOGIC);

end component;

Signal ip1: STD\_LOGIC\_VECTOR (3 downto 0):="1010";

Signal s1: STD\_LOGIC\_VECTOR (1 downto 0):="00";

Signal op1: STD\_LOGIC;

begin

uut: mux41\_b port map(ip=>ip1,s=>s1,op=>op1);

stim\_proc: process

begin

wait for 100ns;

s1<="01";

wait for 100ns;

s1<="10";

wait for 100ns;

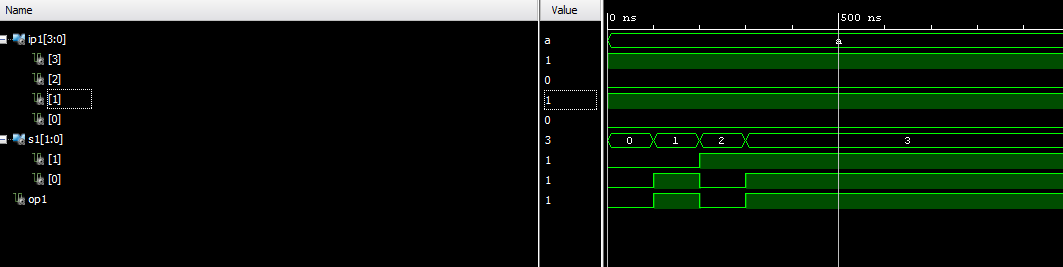
s1<="11";

wait;

end process;

end Dataflow;

**Waveform:**

****

**Dataflow Model for 3:8 Decoder**

**Truth Table:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | | | D(0) | D(1) | D(2) | D(3) | D(4) | D(5) | D(6) | D(7) |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**VHD Source Code:**

**Decoder38.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity decoder38 is

Port ( ip : in STD\_LOGIC\_VECTOR (2 downto 0);

op : out STD\_LOGIC\_VECTOR (7 downto 0));

end decoder38;

architecture Decoder of decoder38 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';

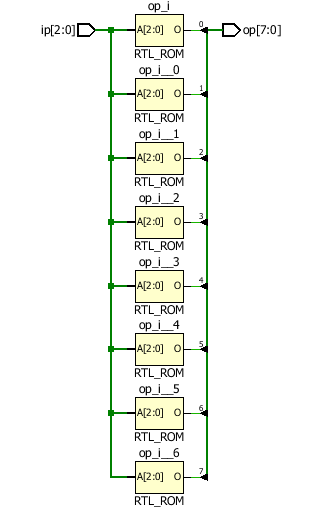
op(5)<='1' when ip="101" else '0';

op(6)<='1' when ip="110" else '0';

op(7)<='1' when ip="111" else '0';

end Decoder;

**Schematic Diagram:**



**Test Bench Code:**

**Decoder38\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity decoder38\_tbw is

-- Port ( );

end decoder38\_tbw;

architecture Dataflow of decoder38\_tbw is

component decoder38 is

Port ( ip : in STD\_LOGIC\_VECTOR (2 downto 0);

op : out STD\_LOGIC\_VECTOR (7 downto 0));

end component;

Signal ip1: STD\_LOGIC\_VECTOR (2 downto 0):="000";

Signal op1: STD\_LOGIC\_VECTOR (7 downto 0);

begin

uut: decoder38 port map(ip=>ip1,op=>op1);

stim\_proc: process

begin

wait for 100ns;

ip1<="001";

wait for 100ns;

ip1<="010";

wait for 100ns;

ip1<="011";

wait for 100ns;

ip1<="100";

wait for 100ns;

ip1<="101";

wait for 100ns;

ip1<="110";

wait for 100ns;

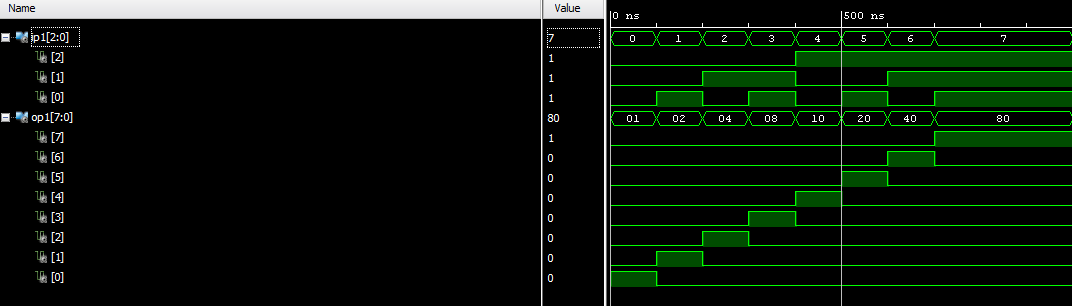
ip1<="111";

wait;

end process;

end Dataflow;

**Waveform:**

****

**Behavioral Model for 3:8 Decoder**

**Truth Table:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | | | D(0) | D(1) | D(2) | D(3) | D(4) | D(5) | D(6) | D(7) |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**VHD Source Code:**

**Decoder38\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity decoder38\_b is

Port ( ip : in STD\_LOGIC\_VECTOR (2 downto 0);

op : out STD\_LOGIC\_VECTOR (7 downto 0));

end decoder38\_b;

architecture Behavioral of decoder38\_b 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';

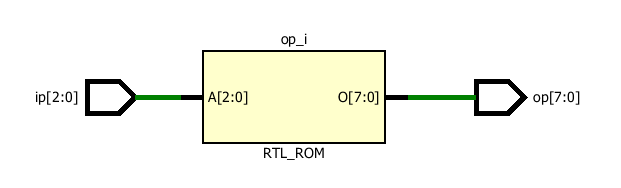
when "111"=>op(7)<='1';

when others=>NULL;

end case;

end process;

end Behavioral;

**Schematic Diagram:** **Test Bench Code:**

**Decoder38\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity decoder38\_tbw is

-- Port ( );

end decoder38\_tbw;

architecture Dataflow of decoder38\_tbw is

component decoder38 is

Port ( ip : in STD\_LOGIC\_VECTOR (2 downto 0);

op : out STD\_LOGIC\_VECTOR (7 downto 0));

end component;

Signal ip1: STD\_LOGIC\_VECTOR (2 downto 0):="000";

Signal op1: STD\_LOGIC\_VECTOR (7 downto 0);

begin

uut: decoder38\_b port map(ip=>ip1,op=>op1);

stim\_proc: process

begin

wait for 100ns;

ip1<="001";

wait for 100ns;

ip1<="010";

wait for 100ns;

ip1<="011";

wait for 100ns;

ip1<="100";

wait for 100ns;

ip1<="101";

wait for 100ns;

ip1<="110";

wait for 100ns;

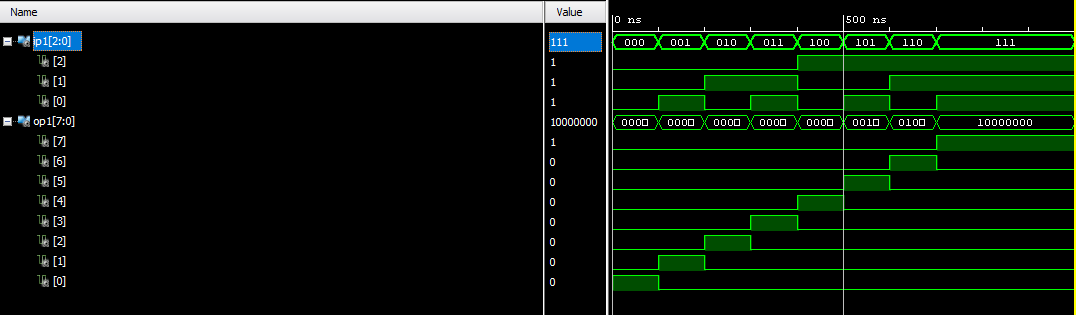
ip1<="111";

wait;

end process;

end Dataflow;

**Waveform:**

****

**Dataflow Model for Comparator**

**VHD Source Code:**

**Comparator.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comparator is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

eq : out STD\_LOGIC;

gt : out STD\_LOGIC;

lt : out STD\_LOGIC);

end comparator;

architecture Dataflow of comparator is

begin

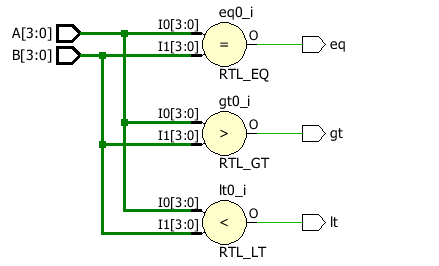
eq<='1' when A=B else '0';

gt<='1' when A>B else '0';

lt<='1' when A<B else '0';

end Dataflow;

**Schematic Diagram:**



**Test Bench Code:**

**Decoder38\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comp\_df\_tbw is

-- Port ( );

end comp\_df\_tbw;

architecture Behavioral of comp\_df\_tbw is

component comp\_df is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

eq : out STD\_LOGIC;

gt : out STD\_LOGIC;

lt : out STD\_LOGIC);

end component;

signal A1:std\_logic\_vector (3 downto 0):="0000";

signal B1:std\_logic\_vector (3 downto 0):="0000";

signal eq1:std\_logic;

signal gt1:std\_logic;

signal lt1:std\_logic;

begin

uut:comp\_df port map(A=>A1, B=>B1, eq=>eq1, gt=>gt1, lt=>lt1);

stim\_proc:process

begin

wait for 100ns;

A1<="0001";

B1<="0001";

wait for 100ns;

A1<="1000";

B1<="0010";

wait for 100ns;

A1<="0010";

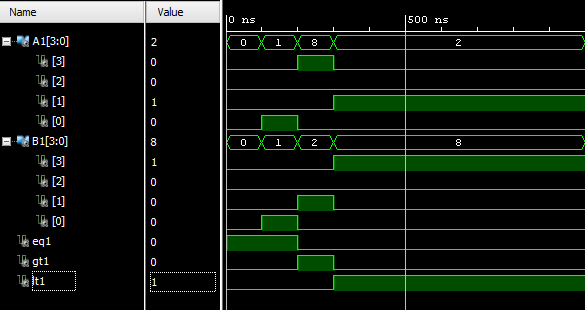
B1<="1000";

wait;

end process;

end Dataflow;

**Waveform:**

**Behavioral Model for ALU**

**Truth Table:**

|  |  |
| --- | --- |
| Ch | Output |
| 000 | A + B |
| 001 | A - B |
| 110 | A + 1 |
| 011 | B - 1 |
| 100 | A AND B |
| 101 | A OR B |
| 110 | A XOR B |
| 111 | NOT A |

**VHD Source Code:**

**ALU\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity ALU\_b is

Port ( A : in unsigned (3 downto 0);

B : in unsigned (3 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

C : out unsigned (3 downto 0));

end ALU\_b;

architecture Behavioral of ALU\_b is

begin

process(A,B,S)

begin

case S is

when "000"=>C<=A + B;

when "001"=>C<=A - B;

when "010"=>C<=A + 1;

when "011"=>C<=A - 1;

when "100"=>C<=A AND B;

when "101"=>C<=A OR B;

when "110"=>C<=A XOR B;

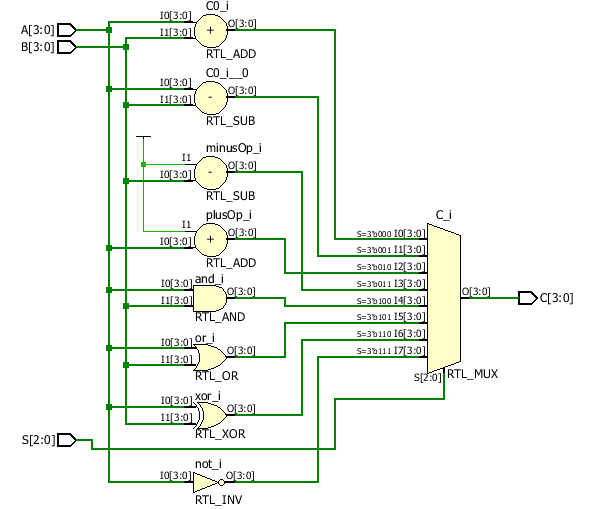
when "111"=>C<=NOT A;

when others=>NULL;

end case;

end process;

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**ALU\_B\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity ALU\_B\_TBW is

-- Port ( );

end ALU\_B\_TBW;

architecture Behavioral of ALU\_B\_TBW is

component ALU\_b is

Port ( A : in unsigned (3 downto 0);

B : in unsigned (3 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

C : out unsigned (3 downto 0));

end component;

Signal A1: unsigned(3 downto 0):="1010";

Signal B1: unsigned(3 downto 0):="0101";

Signal S1: STD\_LOGIC\_VECTOR(2 downto 0):="000";

Signal C1: unsigned(3 downto 0);

begin

uut:ALU\_b port map(A=>A1,B=>B1,S=>S1,C=>C1);

stim\_proc:process

begin

wait for 100ns;

S1<="001";

wait for 100ns;

S1<="010";

wait for 100ns;

S1<="011";

wait for 100ns;

S1<="100";

wait for 100ns;

S1<="101";

wait for 100ns;

S1<="110";

wait for 100ns;

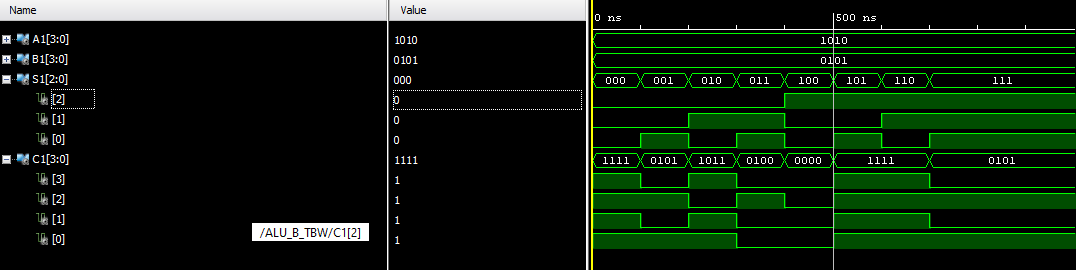
S1<="111";

wait;

end process;

end Behavioral;

**Waveform:**



**Behavioral Model for ALU**

**Truth Table:**

|  |  |
| --- | --- |
| Ch | Output |
| 000 | A + B |
| 001 | A - B |
| 110 | A + 1 |
| 011 | B - 1 |
| 100 | A AND B |
| 101 | A OR B |
| 110 | A XOR B |
| 111 | NOT A |

**VHD Source Code:**

**ALU\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity MUX81\_st is

Port ( I : in STD\_LOGIC\_VECTOR (7 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

Y : out STD\_LOGIC);

end MUX81\_st;

architecture Behavioral of MUX81\_st is

component MUX21 is

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

S : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

component mux41\_b is

Port ( ip : in STD\_LOGIC\_VECTOR (3 downto 0);

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

op : out STD\_LOGIC);

end component;

Signal Y1 : STD\_LOGIC;

Signal Y2 : STD\_LOGIC;

Signal temp : STD\_LOGIC\_VECTOR(1 downto 0);

begin

l1: mux41\_b port map(I(3 downto 0),S(1 downto 0),Y1);

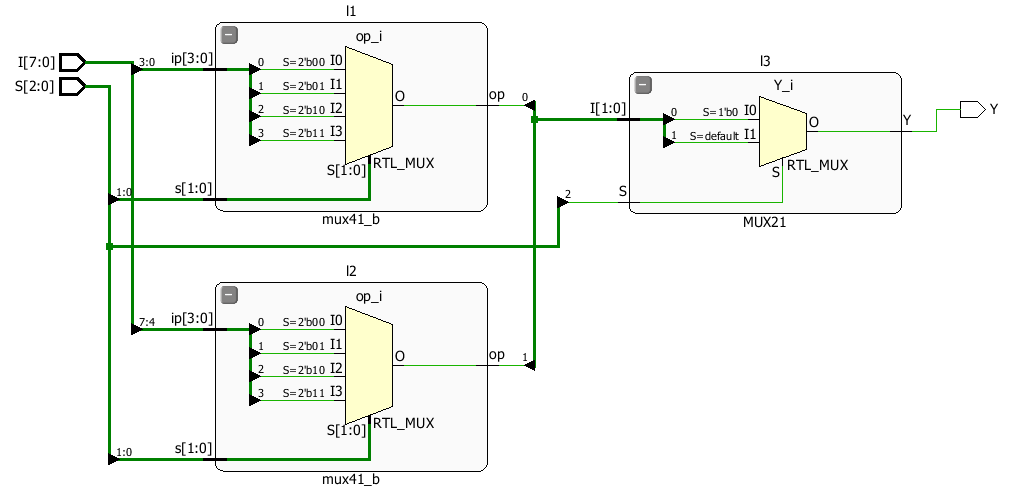
l2: mux41\_b port map(I(7 downto 4),S(1 downto 0),Y2);

temp(0)<=Y1;

temp(1)<=Y2;

l3: MUX21 port map(temp,S(2),Y);

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**ALU\_B\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity MUX81\_st\_tbw is

-- Port ( );

end MUX81\_st\_tbw;

architecture Behavioral of MUX81\_st\_tbw is

component MUX81\_st is

Port ( I : in STD\_LOGIC\_VECTOR (7 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

Y : out STD\_LOGIC);

end component;

Signal I1:STD\_LOGIC\_VECTOR (7 downto 0):="01010101";

Signal S1:STD\_LOGIC\_VECTOR (2 downto 0):="000";

Signal Y1:STD\_LOGIC;

begin

uut:MUX81\_st port map(I=>I1,S=>S1,Y=>Y1);

stim\_proc:process

begin

wait for 100ns;

S1<="001";

wait for 100ns;

S1<="010";

wait for 100ns;

S1<="011";

wait for 100ns;

S1<="100";

wait for 100ns;

S1<="101";

wait for 100ns;

S1<="110";

wait for 100ns;

S1<="111";

wait;

end process;

end Behavioral;

**Waveform:**



**Dataflow Model for Halfadder**

**Truth Table:**

|  |  |
| --- | --- |
| Ch | Output |
| 000 | A + B |
| 001 | A - B |
| 110 | A + 1 |
| 011 | B - 1 |
| 100 | A AND B |
| 101 | A OR B |
| 110 | A XOR B |
| 111 | NOT A |

**VHD Source Code:**

**ALU\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity HA\_df is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

S : out STD\_LOGIC;

C : out STD\_LOGIC);

end HA\_df;

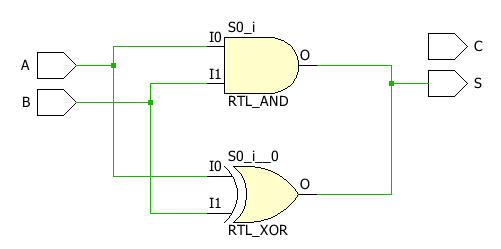
architecture Dataflow of HA\_df is

begin

S<=A Xor B;

S<=A AND B;

end Dataflow;

**Schematic Diagram: **

**Behavioral Model for ALU**

**Truth Table:**

|  |  |
| --- | --- |
| Ch | Output |
| 000 | A + B |
| 001 | A - B |
| 110 | A + 1 |
| 011 | B - 1 |
| 100 | A AND B |
| 101 | A OR B |
| 110 | A XOR B |
| 111 | NOT A |

**VHD Source Code:**

**ALU\_b.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FA\_st is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : in STD\_LOGIC;

Sm : out STD\_LOGIC;

C : out STD\_LOGIC);

end FA\_st;

architecture Behavioral of FA\_st is

component HA\_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 ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

Signal S1:STD\_LOGIC;

Signal C1:STD\_LOGIC;

Signal C2:STD\_LOGIC;

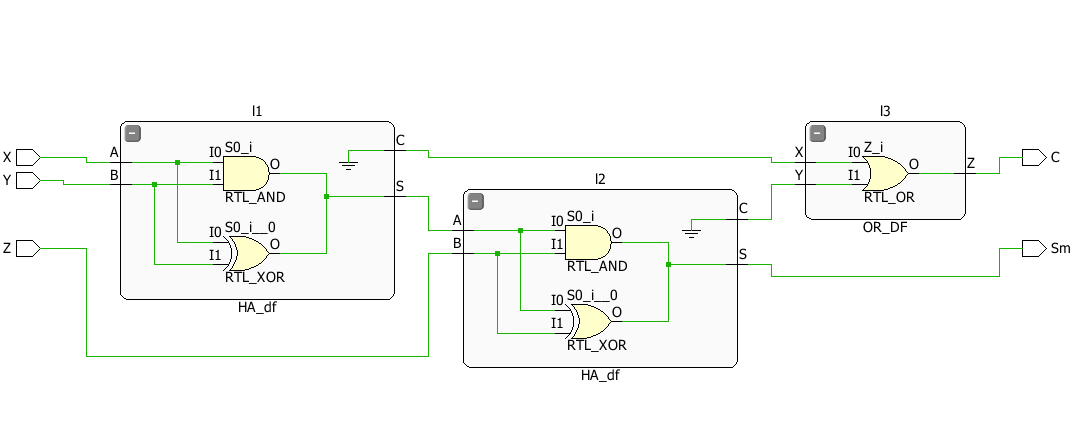
begin

l1:HA\_df port map(X,Y,S1,C1);

l2:HA\_df port map(S1,Z,Sm,C2);

l3:OR\_DF port map(C1,C2,C);

end Behavioral;

**Schematic Diagram: **

**Test Bench Code:**

**ALU\_B\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FA\_st\_tbw is

-- Port ( );

end FA\_st\_tbw;

architecture Behavioral of FA\_st\_tbw is

component FA\_st is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : in STD\_LOGIC;

Sm : out STD\_LOGIC;

C : out STD\_LOGIC);

end component;

Signal A1 : STD\_LOGIC := '0';

Signal B1 : STD\_LOGIC := '0';

Signal C1 : STD\_LOGIC := '0';

Signal S1 : STD\_LOGIC ;

Signal Ca1 : STD\_LOGIC ;

begin

uut: FA\_st port map(X=>A1,Y=>B1,Z=>C1,Sm=>S1,C=>Ca1);

stim\_proc: process

begin

wait for 100ns;

A1<='0';

B1<='0';

C1<='1';

wait for 100ns;

A1<='0';

B1<='1';

C1<='0';

wait for 100ns;

A1<='0';

B1<='1';

C1<='1';

wait for 100ns;

A1<='1';

B1<='0';

C1<='0';

wait for 100ns;

A1<='1';

B1<='0';

C1<='1';

wait for 100ns;

A1<='1';

B1<='1';

C1<='0';

wait for 100ns;

A1<='1';

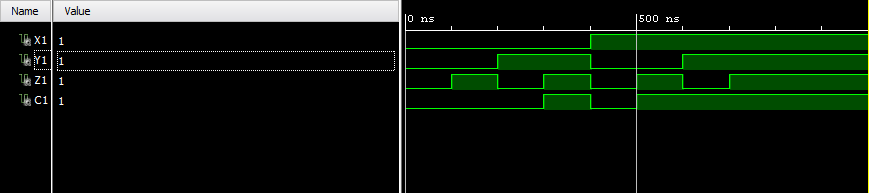
B1<='1';

C1<='1';

wait;

end process;

**Waveform:**



**Dataflow Model for Full Adder**

**Truth Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | S | Ca |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**VHD Source Code:**

**fulladder.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fa\_df is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Ca : out STD\_LOGIC);

end fa\_df;

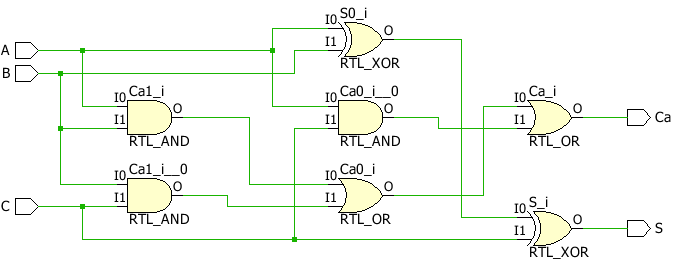
architecture Dataflow of fa\_df is

begin

S<=A xor B xor C;

Ca<= (A and B)or(B and C)or(A and C);

end Dataflow;

**Schematic Diagram:****Ripple Carry Full Adder**

**VHD Source Code:**

**rc\_st.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity rc\_st is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

C : in STD\_LOGIC;

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Ca : out STD\_LOGIC);

end rc\_st;

architecture Behavioral of rc\_st is

component fa\_df is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Ca : out STD\_LOGIC);

end component;

Signal C1 : STD\_LOGIC;

Signal C2 : STD\_LOGIC;

Signal C3 : STD\_LOGIC;

begin

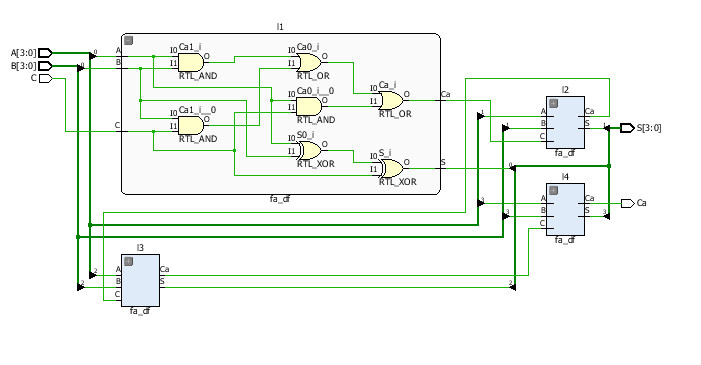
l1: fa\_df port map(A(0),B(0),C,S(0),C1);

l2: fa\_df port map(A(1),B(1),C1,S(1),C2);

l3: fa\_df port map(A(2),B(2),C2,S(2),C3);

l4: fa\_df port map(A(3),B(3),C3,S(3),Ca);

end Behavioral;

**Schematic Diagram:** **Test Bench Code:**

**ALU\_B\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity rc\_st\_tbw is

-- Port ( );

end rc\_st\_tbw;

architecture Behavioral of rc\_st\_tbw is

component rc\_st is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

C : in STD\_LOGIC;

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Ca : out STD\_LOGIC);

end component;

Signal A1:STD\_LOGIC\_VECTOR (3 downto 0):="0000";

Signal B1:STD\_LOGIC\_VECTOR (3 downto 0):="0001";

Signal C1:STD\_LOGIC:='0';

Signal S1:STD\_LOGIC\_VECTOR (3 downto 0);

Signal Ca1:STD\_LOGIC;

begin

uut:rc\_st port map(A=>A1,B=>B1,C=>C1,Ca=>Ca1,S=>S1);

stim\_proc:process

begin

wait for 100ns;

A1<="1001";

B1<="1100";

wait for 100ns;

A1<="0111";

B1<="0100";

wait for 100ns;

A1<="1000";

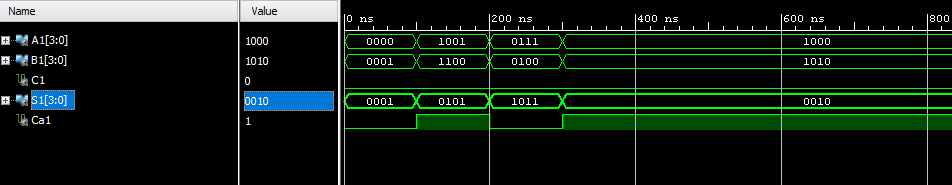
B1<="1010";

wait;

end process;

end Behavioral;

**Waveform:**



**Ripple Carry Full Adder**

**VHD Source Code:**

**rc\_st.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity ASC is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

C : in STD\_LOGIC;

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Ca : out STD\_LOGIC);

end ASC;

architecture Behavioral of ASC is

component fa\_df is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

S : out STD\_LOGIC;

Ca : out STD\_LOGIC);

end component;

Signal C1 : STD\_LOGIC;

Signal C2 : STD\_LOGIC;

Signal C3 : STD\_LOGIC;

Signal temp: STD\_LOGIC\_VECTOR (3 downto 0);

begin

gk:for i in 0 to 3 generate

temp(i)<=B(i) xor C;

end generate gk;

l1: fa\_df port map(A(0),temp(0),C,S(0),C1);

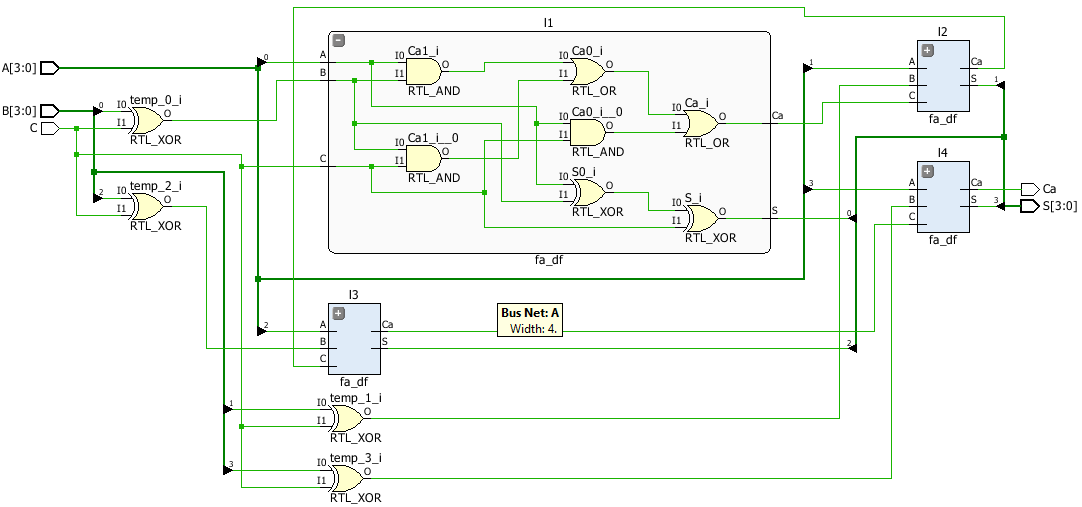
l2: fa\_df port map(A(1),temp(1),C1,S(1),C2);

l3: fa\_df port map(A(2),temp(2),C2,S(2),C3);

l4: fa\_df port map(A(3),temp(3),C3,S(3),Ca);

end Behavioral;

**Schematic Diagram:**



**Test Bench Code:**

**ALU\_B\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity asc\_tbw is

-- Port ( );

end asc\_tbw;

architecture Behavioral of asc\_tbw is

component ASC is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

C : in STD\_LOGIC;

S : out STD\_LOGIC\_VECTOR (3 downto 0);

Ca : out STD\_LOGIC);

end component;

Signal A1:STD\_LOGIC\_VECTOR (3 downto 0):="0000";

Signal B1:STD\_LOGIC\_VECTOR (3 downto 0):="0001";

Signal C1:STD\_LOGIC:='1';

Signal S1:STD\_LOGIC\_VECTOR (3 downto 0);

Signal Ca1:STD\_LOGIC;

begin

uut:ASC port map(A=>A1,B=>B1,C=>C1,Ca=>Ca1,S=>S1);

stim\_proc:process

begin

wait for 100ns;

A1<="1001";

B1<="1100";

wait for 100ns;

A1<="0111";

B1<="0100";

C1<='0';

wait for 100ns;

A1<="1000";

B1<="1010";

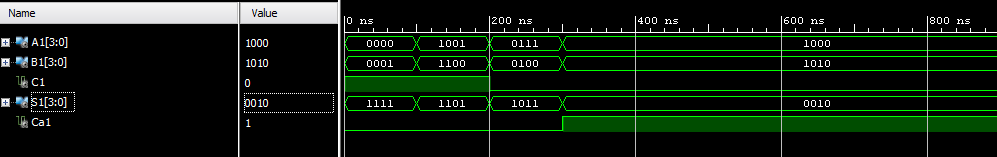
C1<='0';

wait;

end process;

end Behavioral;

**Waveform:**



**Behavioral Flow Model for SR Flip Flop**

**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| S | R | Qn+1 | Q’n+1 |
| 0 | 0 | Qn | Qn’ |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | X | X |

**VHD Source Code:**

**SR\_FF.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity sr\_ff is

Port ( S : in STD\_LOGIC;

R : in STD\_LOGIC;

Q : out STD\_LOGIC;

Qn : out STD\_LOGIC;

clk : in STD\_LOGIC);

end sr\_ff;

architecture Behavioral of sr\_ff is

begin

process(S,R,clk)

begin

if(clk' event and clk='1')then

if(S='0' and R='1')then

Q<='0';

Qn<='1';

elsif(S='1' and R='0')then

Q<='1';

Qn<='0';

elsif(S='1' and R='1')then

Q<='X';

Qn<='X';

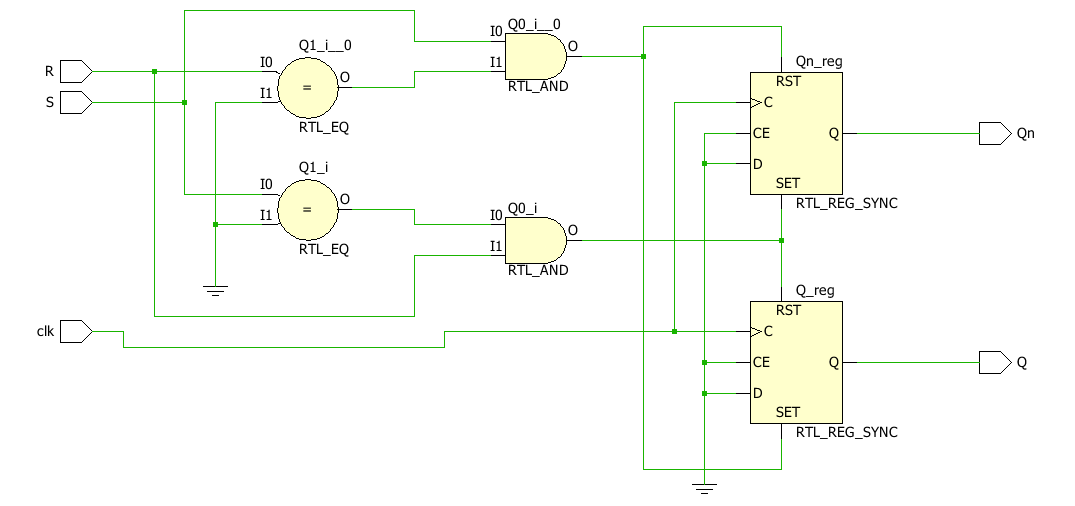
end if;

end if;

end process;

end Behavioral;

**Schematic Diagram:**



**Test Bench Code:**

**AND\_DF\_TBW.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity sr\_ff\_tbw is

-- Port ( );

end sr\_ff\_tbw;

architecture Behavioral of sr\_ff\_tbw is

component sr\_ff is

Port ( S : in STD\_LOGIC;

R : in STD\_LOGIC;

Q : out STD\_LOGIC;

Qn : out STD\_LOGIC;

clk : in STD\_LOGIC);

end component;

constant clock\_period:time:=60ns;

Signal S1 : STD\_LOGIC := '0';

Signal R1 : STD\_LOGIC := '0';

Signal clk1 : STD\_LOGIC := '0';

Signal Q1 : STD\_LOGIC ;

Signal Qn1 : STD\_LOGIC ;

begin

uut:sr\_ff port map(S=>S1,R=>R1,clk=>clk1,Q=>Q1,Qn=>Qn1);

clk1 <= not clk1 after clock\_period/2;

stim\_proc: process

begin

wait for 100ns;

S1<='0';

R1<='1';

wait for 100ns;

S1<='1';

R1<='0';

wait for 100ns;

S1<='1';

R1<='1';

wait for 100ns;

S1<='1';

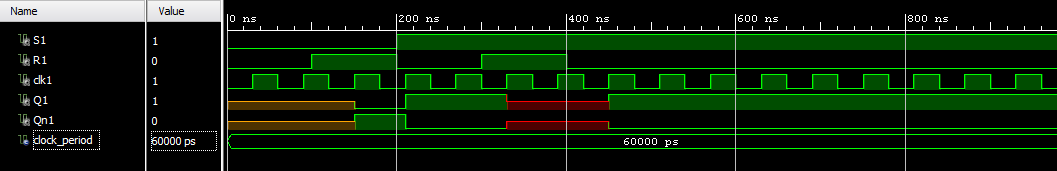
R1<='0';

wait;

end process;

end Behavioral;

**Waveform:**



**Behavioral Flow Model for JK Flip Flop**

**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| J | K | Qn+1 | Q’n+1 |
| 0 | 0 | Qn | Qn’ |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Qn’ | Qn |

**VHD Source Code:**

**jk\_ff.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity jk\_ff is

Port ( J : in STD\_LOGIC;

K : in STD\_LOGIC;

clk : in STD\_LOGIC;

Q : inout STD\_LOGIC;

Qn : inout STD\_LOGIC);

end jk\_ff;

architecture Behavioral of jk\_ff is

begin

process(J,K,clk)

begin

if(clk' event and clk='1')then

if(J='0' and K='1')then

Q<='0';

Qn<='1';

elsif(J='1' and K='0')then

Q<='1';

Qn<='0';

elsif(J='1' and K='1')then

Q<=Qn;

Qn<=Q;

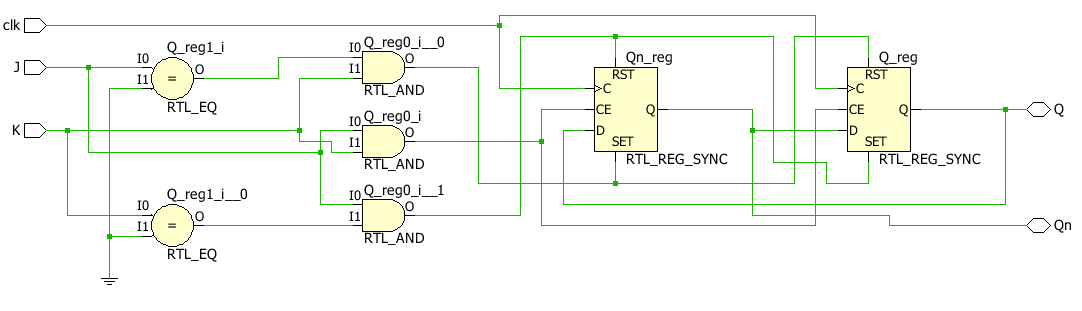
end if;

end if;

end process;

end Behavioral;

**Schematic Diagram:**



**Test Bench Code:**

**Jk\_ff\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity jk\_tbw is

-- Port ( );

end jk\_tbw;

architecture Behavioral of jk\_tbw is

component jk\_ff is

Port ( J : in STD\_LOGIC;

K : in STD\_LOGIC;

clk : in STD\_LOGIC;

Q : inout STD\_LOGIC;

Qn : inout STD\_LOGIC);

end component;

constant clock\_period:time:=60ns;

Signal J1 : STD\_LOGIC := '0';

Signal K1 : STD\_LOGIC := '0';

Signal clk1 : STD\_LOGIC := '0';

Signal Q1 : STD\_LOGIC ;

Signal Qn1 : STD\_LOGIC ;

begin

uut:jk\_ff port map(J=>J1,K=>K1,clk=>clk1,Q=>Q1,Qn=>Qn1);

clk1 <= not clk1 after clock\_period/2;

stim\_proc: process

begin

J1<='1';

K1<='0';

wait for 100ns;

J1<='1';

K1<='1';

wait for 100ns;

J1<='0';

K1<='1';

wait for 100ns;

J1<='1';

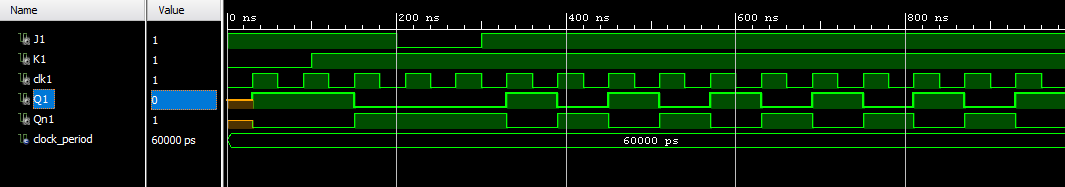
K1<='1';

wait;

end process;

end Behavioral;

**Waveform:**



**Behavioral Flow Model for D Flip Flop**

**Truth Table:**

|  |  |
| --- | --- |
| D | Q |
| 0 | 0 |
| 1 | 1 |

**VHD Source Code:**

**D\_ff.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity D\_ff is

Port ( D : in STD\_LOGIC;

clk : in STD\_LOGIC;

Q : out STD\_LOGIC);

end D\_ff;

architecture Behavioral of D\_ff is

begin

process(D,clk)

begin

if(clk' event and clk='1')then

if(D='0')then

Q<='0';

elsif(D='1')then

Q<='1';

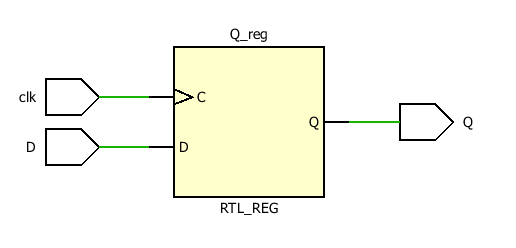
end if;

end if;

end process;

end Behavioral;

**Schematic Diagram:**



**Test Bench Code:**

**D\_ff\_tbw.vhd**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity d\_ff\_tbw is

-- Port ( );

end d\_ff\_tbw;

architecture Behavioral of d\_ff\_tbw is

component D\_ff is

Port ( D : in STD\_LOGIC;

clk : in STD\_LOGIC;

Q : out STD\_LOGIC);

end component;

constant clock\_period:time:=60ns;

Signal D1 : STD\_LOGIC := '0';

Signal clk1 : STD\_LOGIC := '0';

Signal Q1 : STD\_LOGIC ;

begin

uut:D\_ff port map(D=>D1,clk=>clk1,Q=>Q1);

clk1 <= not clk1 after clock\_period/2;

stim\_proc: process

begin

wait for 100ns;

D1<='1';

wait;

end process;

end Behavioral;

**Waveform:**

