**D FLIP FLOP**

Library ieee;

Use ieee.std\_logic\_1164.all;

entity dffb is

port( d: in std\_logic;

clk: in std\_logic;

rst: in std\_logic;

q: out std\_logic);

end dffb;

architecture Behavioral of dffb is

begin

process(rst,clk,d)

begin

if (rst='1') then

q<='0';

elsif(rising\_edge(clk)) then

q<= d;

end if;

end process;

end Behavioral;

**TESTBENCH**

LIBRARY ieee;

USEieee.std\_logic\_1164.ALL;

ENTITY dffbt IS

END dffbt;

ARCHITECTURE behavior OF dffbt IS

COMPONENT dffb

PORT(

d : IN std\_logic;

clk : IN std\_logic;

rst : IN std\_logic;

q : OUT std\_logic

);

END COMPONENT;

--Inputs

signal d : std\_logic := '0';

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

--Outputs

signal q : std\_logic;

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: dffb PORT MAP (

d => d,

clk => clk,

rst => rst,

q => q

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

wait for 100 ns;

wait for clk\_period\*10;

-- insert stimulus here

rst <= '1';

wait for 50 ns;

rst <= '0';

d <= '0';

wait for 50 ns;

rst <= '0';

d <= '1';

wait;

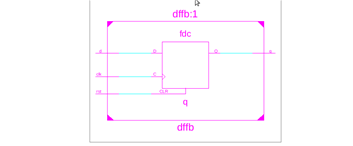
end process;

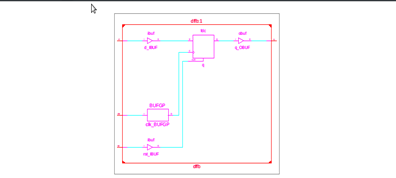
END;

**SIMULATION**

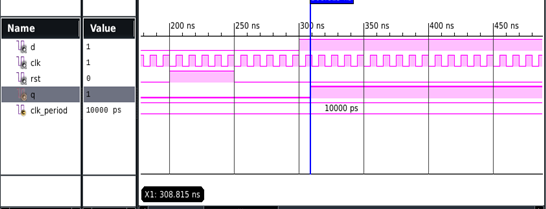


**RTL SCHEMATIC**



**TECHNOLOGY SCHEMATIC** 

**POST SYNTHESIS**



AREA

Number of bonded IOBs: 4 out of 190

IOB Flip Flops/Latches: 1

Number of BUFG/BUFGCTRLs: 1 out of 16

FREQUENCY = 1/ 3.597 GHZ

POWER = 0.029W

**ALU STRUCTURE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity HA is

Port ( HAA, HAB : in STD\_LOGIC;

SUM, CARRY : out STD\_LOGIC);

end HA;

architecture dataflow of HA is

begin

SUM <= HAA XOR HAB;

CARRY <= HAA AND HAB;

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity HS is

Port ( HSA, HSB : in STD\_LOGIC;

DIFFERENCE, BORROW : out STD\_LOGIC);

end HS;

architecture dataflow of HS is

begin

DIFFERENCE <= HSA XOR HSB;

BORROW <= (not HSA) AND HSB;

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity multiplier is

Port ( MA, MB : in STD\_LOGIC;

PRODUCT : out STD\_LOGIC);

end multiplier;

architecture dataflow of multiplier is

begin

PRODUCT <= MA AND MB;

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity MUX is

Port ( A1,A2,A3,A4 : in STD\_LOGIC;

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

X : out STD\_LOGIC);

end mux;

architecture dataflow of MUX is

begin

with S select

X <= A1 when "00",

A2 when "01",

A3 when "10",

A4 when others;

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ground is

Port ( N : inout STD\_LOGIC);

end ground;

architecture dataflow of ground is

begin

N <= 'U';

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ALU is

Port ( A,B,SEL1,SEL2 : in STD\_LOGIC;

ALU1,ALU2 : out STD\_LOGIC);

end ALU;

architecture Structural of ALU is

component HA is

Port ( HAA, HAB : in STD\_LOGIC;

SUM, CARRY : out STD\_LOGIC);

end component;

component HS is

Port ( HSA, HSB : in STD\_LOGIC;

DIFFERENCE, BORROW : out STD\_LOGIC);

end component;

component multiplier is

Port ( MA, MB : in STD\_LOGIC;

PRODUCT : out STD\_LOGIC);

end component;

component MUX is

Port ( A1,A2,A3,A4 : in STD\_LOGIC;

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

X : out STD\_LOGIC);

end component;

component ground is

Port ( N : inout STD\_LOGIC);

end component;

signal S0,S1,S2,S3,S4,S5: STD\_LOGIC;

begin

U0: ground PORT MAP(N=>S5);

U1: HA PORT MAP(HAA=>A,HAB=>B,SUM=>S0,CARRY=>S3);

U2: HS PORT MAP(HSA=>A,HSB=>B,DIFFERENCE=>S1,BORROW=>S4);

U3: multiplier PORT MAP(MA=>A,MB=>B,PRODUCT=>S2);

U4: MUX PORT MAP(A1=>S0,A2=>S1,A3=>S2,A4=>S5,X=>ALU1,S(0)=>SEL1,S(1)=>SEL2);

U5: MUX PORT MAP(A1=>S3,A2=>S4,A3=>S5,A4=>S5,X=>ALU2,S(0)=>SEL1,S(1)=>SEL2);

end Structural;

**TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY alust IS

END alust;

ARCHITECTURE behavior OF alust IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT ALU

PORT(

A : IN std\_logic;

B : IN std\_logic;

SEL1 : IN std\_logic;

SEL2 : IN std\_logic;

ALU1 : OUT std\_logic;

ALU2 : OUT std\_logic

);

END COMPONENT;

--Inputs

signal A : std\_logic := '0';

signal B : std\_logic := '0';

signal SEL1 : std\_logic := '0';

signal SEL2 : std\_logic := '0';

--Outputs

signal ALU1 : std\_logic;

signal ALU2 : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ALU PORT MAP (

A => A,

B => B,

SEL1 => SEL1,

SEL2 => SEL2,

ALU1 => ALU1,

ALU2 => ALU2

);

-- Stimulus process

stim\_proc: process

begin

-- insert stimulus here

A<='0';B<='1';

SEL1<='0';SEL2<='0';

wait for 100 ns;

SEL1<='0';SEL2<='1';

wait for 100 ns;

SEL1<='1';SEL2<='0';

wait for 100 ns;

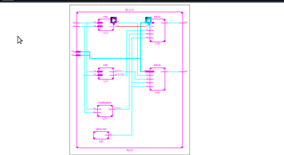
SEL1<='1';SEL2<='1';

wait;

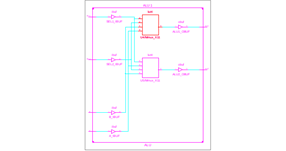
end process;

END;

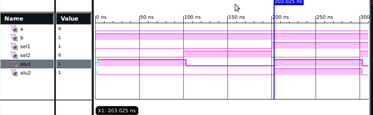
**RTL SCHEMATIC**

****

**TECHNOLOGY SCHEMATIC**

****

**POST SYNTHESIS**

****

AREA

Number of Slice LUTs: 2 out of 15032

Number of bonded IOBs: 6 out of 190

FREQUENCY 1/ 5.439 GHZ

POWER = 0.029W

**ALU BEHAVIOURAL**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity alub1 is

Port ( a : in signed(3 downto 0);

b : in signed(3 downto 0);

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

outp : out signed(3 downto 0));

end alub1;

architecture Behavioral of alub1 is

begin

process(a,b,sel)

begin

case sel is

when "000" =>

outp<= a + b; --addition

when "001" =>

outp<= a - b; --subtraction

when "010" =>

outp<= a - 1; --sub 1

when "011" =>

outp<= a + 1; --add 1

when "100" =>

outp<= a and b; --AND gate

when "101" =>

outp<= a or b; --OR gate

when "110" =>

outp<= not a ; --NOT gate

when "111" =>

outp<= a xor b; --XOR gate

when others =>

NULL;

end case;

end process;

end Behavioral;

**TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY alubt IS

END alubt;

ARCHITECTURE behavior OF alubt IS

-- Component Declaration for the Unit Under Test (UUT) COMPONENT alub1

PORT(

a : IN signed(3 downto 0);

b : IN signed(3 downto 0);

sel : IN std\_logic\_vector(2 downto 0);

outp : OUT signed(3 downto 0)

);

END COMPONENT;

--Inputs

signal a : signed(3 downto 0) := (others => '0');

signal b : signed(3 downto 0) := (others => '0');

signal sel : std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal outp : signed(3 downto 0);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: alub1 PORT MAP (

a => a,

b => b,

sel => sel,

outp => outp

);

-- Stimulus process

stim\_proc: process

begin

-- insert stimulus here

a <= "1001";

b <= "1111";

sel <= "000";

wait for 100 ns;

sel <= "001";

wait for 100 ns;

sel <= "010";

wait for 100 ns;

sel <= "011";

wait for 100 ns;

sel <= "100";

wait for 100 ns;

sel <= "101";

wait for 100 ns;

sel <= "110";

wait for 100 ns;

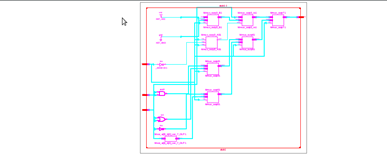
sel <= "111";

wait;

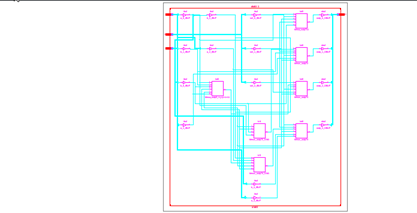
end process;

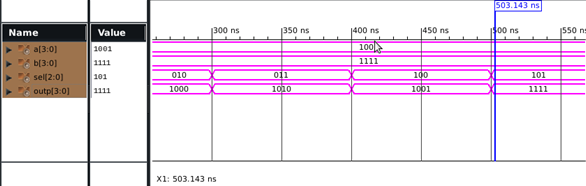
END;

**RTL SCHEMATIC**

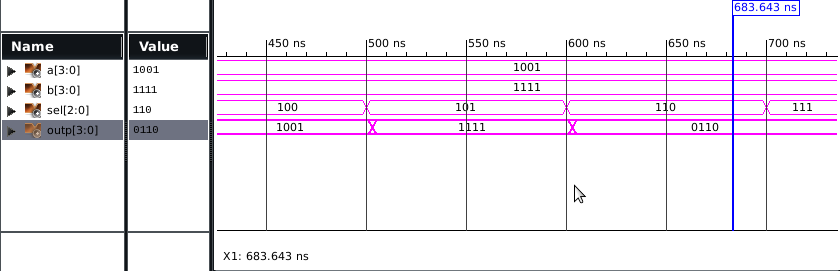
****

**TECHNOLOGY SCHEMATIC**

****

**TESTBENCH WAVEFORM **

**POST SYNTHESIS**

****

AREA

Number of Slice LUTs: 7 out of 15032

Number of bonded IOBs: 15 out of 190

FREQUENCY = 1/ 7.756 GHZ

POWER = 0.029W

**ALU DATAFLOW**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity alud1 is

Port ( a : in signed(3 downto 0);

b : in signed(3 downto 0);

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

outp : out signed(3 downto 0));

end alud1;

architecture Behavioral of alud1 is

begin

with sel select

outp<= a + b when "000", --addition

a - b when "001", --subtraction

a - 1 when "010", --sub 1

a + 1 when "011", --add 1

a and b when "100" , --AND gate

a or b when "101", --OR gate

not a when "110", --NOT gate

a xor b when "111"; --XOR gate

end Behavioral;

**TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY aludt IS

END aludt;

ARCHITECTURE behavior OF aludt IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT alud1

PORT(

a : IN signed(3 downto 0);

b : IN signed(3 downto 0);

sel : IN std\_logic\_vector(2 downto 0);

outp : OUT signed(3 downto 0)

);

END COMPONENT;

--Inputs

signal a : signed(3 downto 0) := (others => '0');

signal b : signed(3 downto 0) := (others => '0');

signal sel : std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal outp : signed(3 downto 0);

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: alud1 PORT MAP (

a => a,

b => b,

sel => sel,

outp => outp

);

-- Stimulus process stim\_proc: process

begin

-- insert stimulus here a <= "1001";

b <= "1111";

sel <= "000";

wait for 100 ns;

sel <= "001";

wait for 100 ns;

sel <= "010";

wait for 100 ns;

sel <= "011";

wait for 100 ns;

sel <= "100";

wait for 100 ns;

sel <= "101";

wait for 100 ns;

sel <= "110";

wait for 100 ns;

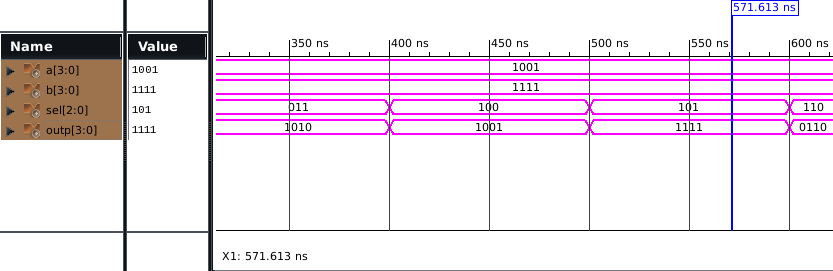
sel <= "111";

wait;

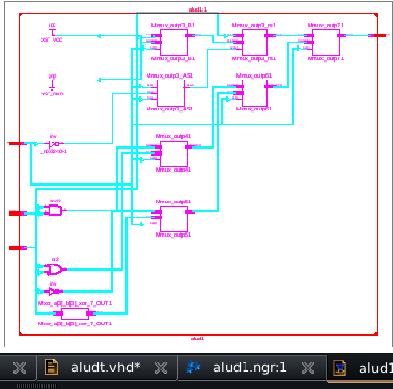
end process;

END;

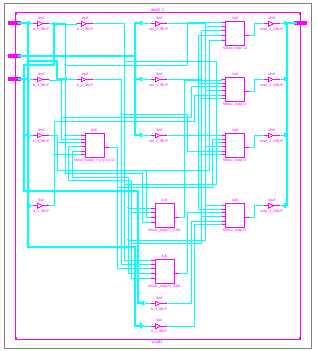
**TESTBENCH WAVEFORM**

****

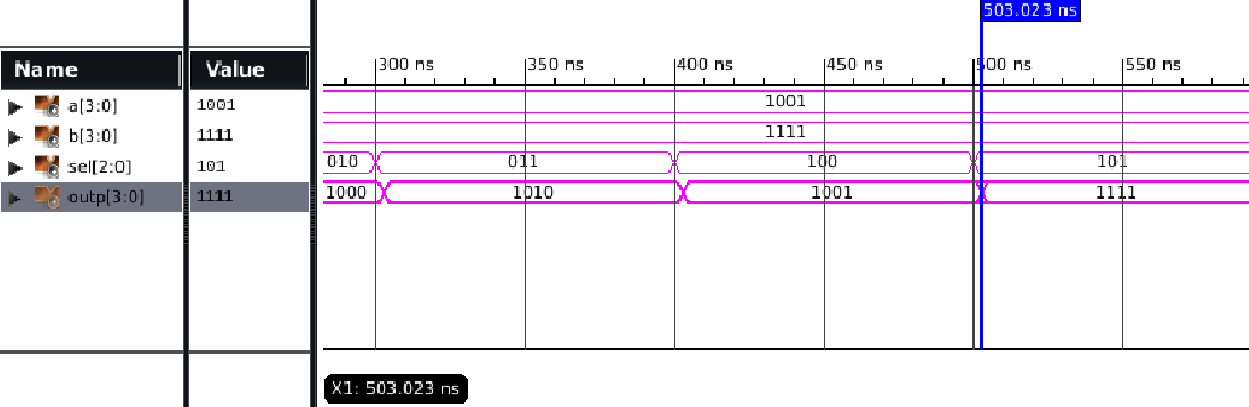
**RTL LOGIC**

****

**TECHNOLOGY SCHEMATIC**

****

**POST SYNTHESIS**

****

AREA

Number of Slice LUTs: 7 out of 15032

Number of bonded IOBs: 15 out of 190

FREQUENCY = 1/ 7.756 GHZ

POWER = 0.029W

**SHIFT REGISTER BEHAVIOURAL**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity shiftb1 is

port( res: in std\_logic;

sin: in std\_logic;

clk: in std\_logic;

pout: out std\_logic\_vector(3 downto 0));

end shiftb1;

architecture Behavioral of shiftb1 is

signal temp: std\_logic\_vector( 3 downto 0);

begin

process( clk, res)

begin

if(res='1') then

temp<="0000";

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

temp(3)<=sin;

temp(2)<=temp(3);

temp(1)<=temp(2);

temp(0)<=temp(1);

end if;

end process;

pout<=temp;

end Behavioral;

**TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY shiftbt IS

END shiftbt;

ARCHITECTURE behavior OF shiftbt IS

-- Component

Declaration for the Unit Under Test (UUT)

COMPONENT shiftb1

PORT(

res : IN std\_logic;

sin : IN std\_logic;

clk : IN std\_logic;

pout : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal res : std\_logic := '0';

signal sin : std\_logic := '0';

signal clk : std\_logic := '0';

--Outputs

signal pout : std\_logic\_vector(3 downto 0);

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: shiftb1 PORT MAP (

res => res,

sin => sin,

clk => clk,

pout => pout

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

wait for 100 ns;

wait for clk\_period\*10;

-- insert stimulus here

sin<='0';

wait for 50 ns;

sin<='0';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='0';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='0';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='0';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='1';

wait for 50 ns;

sin<='0';

wait for 50 ns;

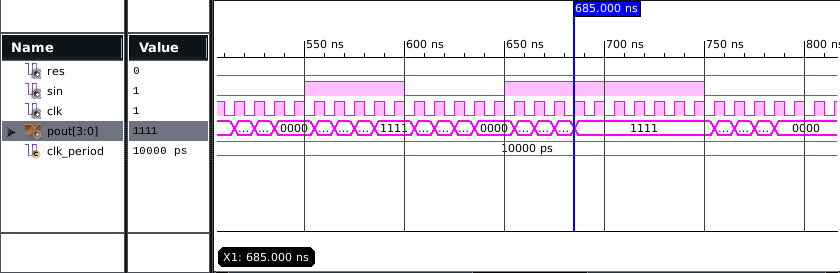
sin<='0';

wait;

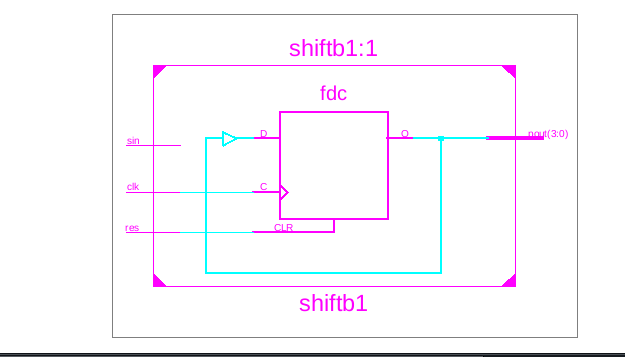
end process;

END;

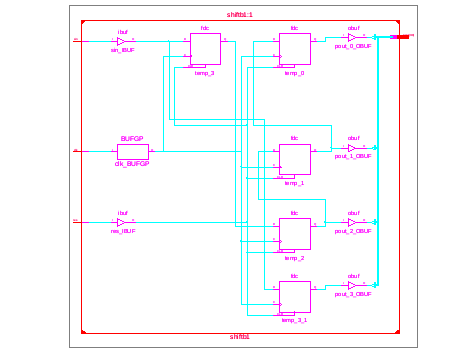
**TESTBENCH WAVEFORM**

****

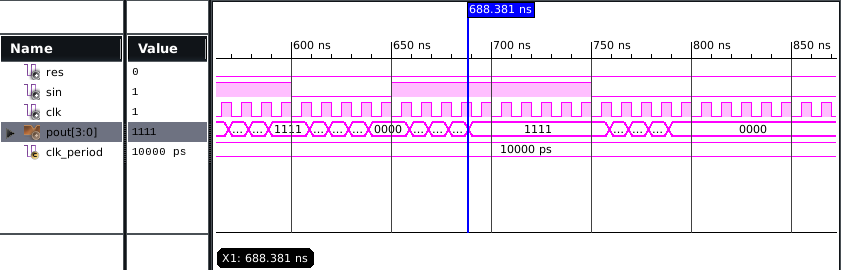
**RTL SCHEMATIC**

****

**TECHNOLOGY SCHEMATIC**

****

**POST SYNTHESIS**

****

AREA

Number of Slice Registers: 4 out of 30064

Number of LUT Flip Flop pairs used: 4

Number of bonded IOBs: 7 out of 190

Number of BUFG/BUFGCTRLs: 1 out of 16

FREQUENCY = 1/ 3.634 GHZ

POWER = 0.029W

**2 BIT COMPAROTOR(DATAFLOW):**

**CODE:**

library IEEE;

use IEEE.std\_logic\_1164.all;

entity comparator\_VHDL is

port (

A,B: in std\_logic\_vector(1 downto 0);

A\_less\_B: out std\_logic;

A\_equal\_B: out std\_logic;

A\_greater\_B: out std\_logic

);

end comparator\_VHDL;

architecture comparator\_structural of comparator\_VHDL is

signal tmp1,tmp2,tmp3,tmp4,tmp5,tmp6,tmp7,tmp8:std\_logic;

begin

tmp1 <= A(1) xnor B(1);

tmp2 <= A(0) xnor B(0);

A\_equal\_B <= tmp1 and tmp2;

tmp3 <= (not A(0)) and (not A(1)) and B(0);

tmp4 <= (not A(1)) and B(1);

tmp5 <= (not A(0)) and B(1) and B(0);

A\_less\_B <= tmp3 or tmp4 or tmp5;

tmp6 <= (not B(0)) and (not B(1)) and A(0);

tmp7 <= (not B(1)) and A(1);

tmp8 <= (not B(0)) and A(1) and A(0);

A\_greater\_B <= tmp6 or tmp7 or tmp8;

end comparator\_structural

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.std\_logic\_unsigned.all;

ENTITY ttbit IS

END ttbit;

ARCHITECTURE behavior OF ttbit IS

COMPONENT comparator\_VHDL

PORT(A : IN std\_logic\_vector(1 downto 0);

B : IN std\_logic\_vector(1 downto 0);

A\_less\_B : OUT std\_logic; A\_equal\_B : OUT std\_logic;

A\_greater\_B : OUT std\_logic);

END COMPONENT;

signal A : std\_logic\_vector(1 downto 0) := (others => '0');

signal B : std\_logic\_vector(1 downto 0) := (others => '0');

signal A\_less\_B : std\_logic; signal A\_equal\_B : std\_logic;

signal A\_greater\_B : std\_logic;

BEGIN

uut: comparator\_VHDL PORT MAP (

A => A, B => B,

A\_less\_B => A\_less\_B,

A\_equal\_B => A\_equal\_B, A\_greater\_B => A\_greater\_B);

stim\_proc: process

begin

for i in 0 to 3 loop

A <= std\_logic\_vector(to\_unsigned(i,2));

B <= std\_logic\_vector(to\_unsigned(i+1,2));

wait for 20 ns;

end loop;

for i in 0 to 3 loop

A <= std\_logic\_vector(to\_unsigned(i+1,2));

B <= std\_logic\_vector(to\_unsigned(i,2));

wait for 20 ns;

end loop;

for i in 0 to 3 loop

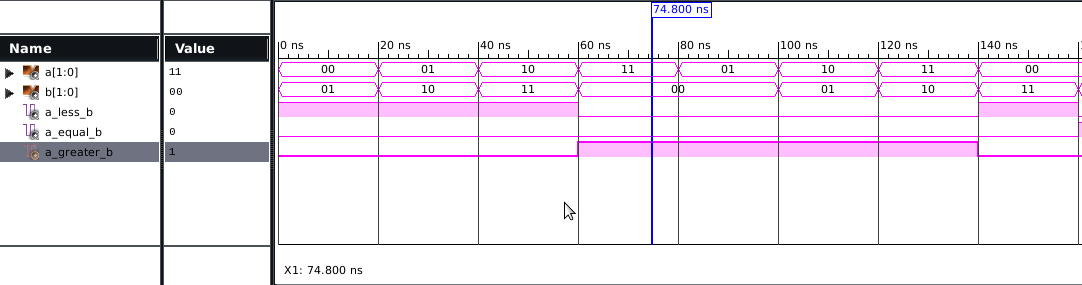
A <= std\_logic\_vector(to\_unsigned(i,2));

B <= std\_logic\_vector(to\_unsigned(i,2));

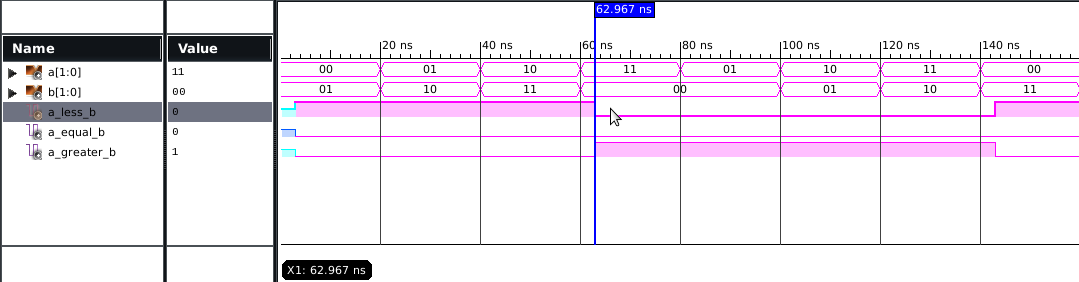
wait for 20 ns;

end loop; end process; END;

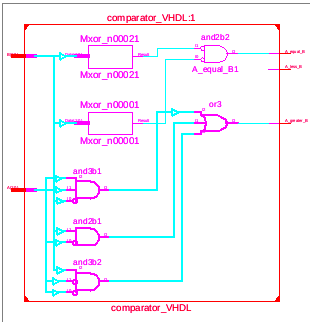
**SIMULATION:**



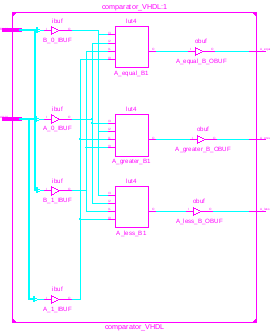
**POST MAP:**



RTL VIEW:



TECHNICAL VIEW:



|  |  |  |
| --- | --- | --- |
| AREA  NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY= 1/5.473 GHZ

POWER = 0.029W

**2 BIT COMPARATOR(BEHAVIOR):**

**CODE:**

library IEEE;

use IEEE.std\_logic\_1164.all;

entity comparator is

port (

A,B: in std\_logic\_vector(1 downto 0);

A\_less\_B: out std\_logic;

A\_equal\_B: out std\_logic;

A\_greater\_B: out std\_logic

);

end comparator;

architecture comparator\_structural of comparator is

begin

process(A,B)

begin

if A="00" and B="00" then

A\_greater\_B<='0';A\_equal\_B<='1';A\_less\_B<='0';

elsif A="00" and B="01" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="00" and B="10" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="00" and B="11" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="01" and B="00" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="01" and B="01" then

A\_greater\_B<='0';A\_equal\_B<='1';A\_less\_B<='0';

elsif A="01" and B="10" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="01" and B="11" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="10" and B="00" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="10" and B="01" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="10" and B="10" then

A\_greater\_B<='0';A\_equal\_B<='1';A\_less\_B<='0';

elsif A="10" and B="11" then

A\_greater\_B<='0';A\_equal\_B<='0';A\_less\_B<='1';

elsif A="11" and B="00" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="11" and B="01" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="11" and B="10" then

A\_greater\_B<='1';A\_equal\_B<='0';A\_less\_B<='0';

elsif A="11" and B="11" then

A\_greater\_B<='0';A\_equal\_B<='1';A\_less\_B<='0';

end if;

end process;

end comparator\_structural;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

ENTITY testbeh IS

END testbeh;

ARCHITECTURE behavior OF testbeh IS

COMPONENT comparator

PORT(

A : IN std\_logic\_vector(1 downto 0);

B : IN std\_logic\_vector(1 downto 0);

A\_less\_B : OUT std\_logic;

A\_equal\_B : OUT std\_logic;

A\_greater\_B : OUT std\_logic);

END COMPONENT;

signal A : std\_logic\_vector(1 downto 0) := (others => '0');

signal B : std\_logic\_vector(1 downto 0) := (others => '0');

signal A\_less\_B : std\_logic;

signal A\_equal\_B : std\_logic;

signal A\_greater\_B : std\_logic;

BEGIN

uut: comparator PORT MAP (

A => A,

B => B,

A\_less\_B => A\_less\_B,

A\_equal\_B => A\_equal\_B,

A\_greater\_B => A\_greater\_B);

stim\_proc: process

begin

for i in 0 to 3 loop

A <= std\_logic\_vector(to\_unsigned(i,2));

B <= std\_logic\_vector(to\_unsigned(i+1,2));

wait for 20 ns;

end loop;

for i in 0 to 3 loop

A <= std\_logic\_vector(to\_unsigned(i+1,2));

B <= std\_logic\_vector(to\_unsigned(i,2));

wait for 20 ns;

end loop;

for i in 0 to 3 loop

A <= std\_logic\_vector(to\_unsigned(i,2));

B <= std\_logic\_vector(to\_unsigned(i,2));

wait for 20 ns;

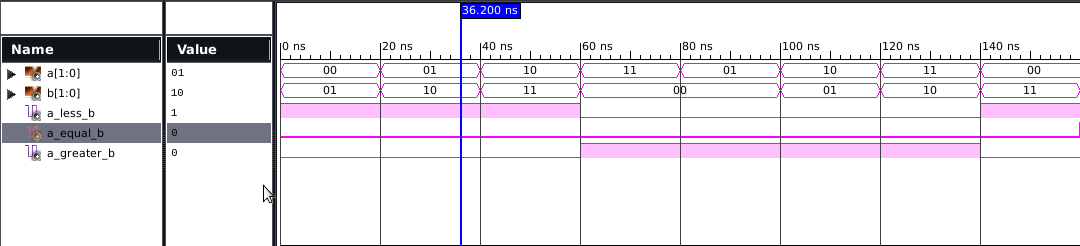
end loop;

wait;

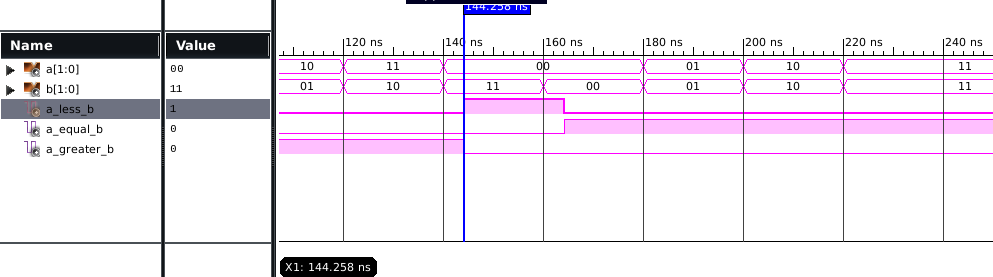
end process;

END;

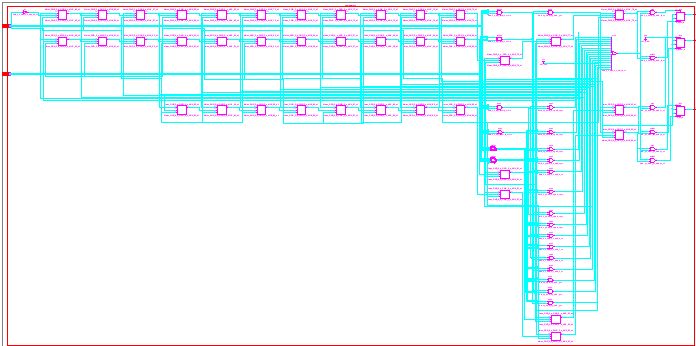
SIMULATION:



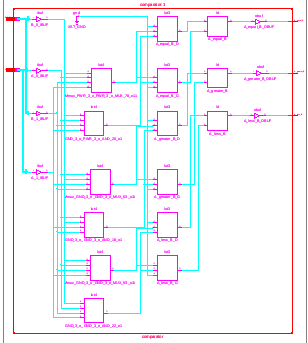
POST MAP SIMULATION:



RTL VIEW:



TECHNICAL VIEW:



AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 12 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY = 1/3.504 GHZ

POWER = 0.029W

**MUX 4:1 (DATAFLOW)**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity mux4 is

port(d0,d1,d2,d3,s0,s1 :in std\_logic;

y :out std\_logic);

end mux4;

architecture dataflow of mux4 is

begin

y <= ((d0 and (not s0) and (not s1)) or (d1 and s1 and (not s0)) or (d2 and (not s1) and s0) or (d3 and s0 and s1));

end dataflow;

**TESTBENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_arith.all;

ENTITY tmuxdata IS

END tmuxdata;

ARCHITECTURE behavior OF tmuxdata IS

COMPONENT mux4

PORT(

d0 : IN std\_logic;

d1 : IN std\_logic;

d2 : IN std\_logic;

d3 : IN std\_logic;

s0 : IN std\_logic;

s1 : IN std\_logic;

y : OUT std\_logic

);

END COMPONENT;

signal d0 : std\_logic := '0';

signal d1 : std\_logic := '0';

signal d2 : std\_logic := '0';

signal d3 : std\_logic := '0';

signal s0 : std\_logic := '0';

signal s1 : std\_logic := '0';

signal y : std\_logic;

BEGIN

uut: mux4 PORT MAP (

d0 => d0,

d1 => d1,

d2 => d2,

d3 => d3,

s0 => s0,

s1 => s1,

y => y);

stim\_proc: process

begin

d0<='0';d1<='1';d2<='0';d3<='1';

s0<='0';s1<='0';

wait for 100 ns;

s0<='0';s1<='1';

wait for 100 ns;

s0<='1';s1<='0';

wait for 100 ns;

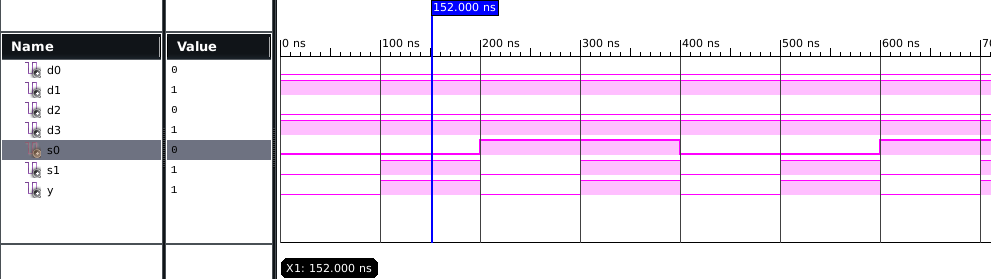
s0<='1';s1<='1';

wait for 100 ns;

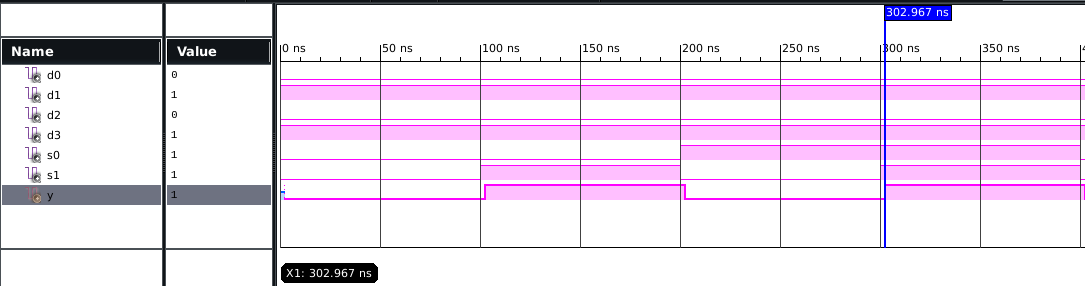
end process;

END;

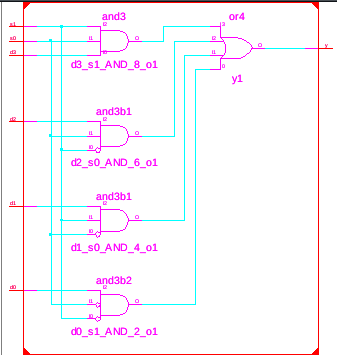
**SIMULATION:**



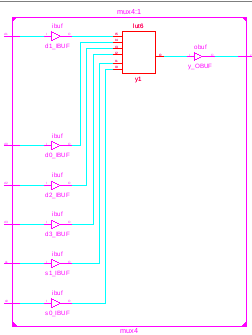
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 1 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY= 1/5.519 GHZ

POWER= 0.029W

**MUX 4:1(BEHAVIOUR)**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux is

port(a,b,c,d:in std\_logic;

s0,s1:in std\_logic;

z:out std\_logic);

end mux;

architecture Behavioral of mux is

begin

process(a,b,c,d,s0,s1)

begin

if (s0='0' and s1='0')then

z<=a;

elsif (s0='0' and s1='1')then

z<=b;

elsif (s0='1' and s1='0')then

z<=c;

else

z<=d;

end if;

end process;

end Behavioral;

**TESTBENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tmux IS

END tmux;

ARCHITECTURE behavior OF tmux IS

COMPONENT mux

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : IN std\_logic;

d : IN std\_logic;

s0 : IN std\_logic;

s1 : IN std\_logic;

z : OUT std\_logic);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic := '0';

signal d : std\_logic := '0';

signal s0 : std\_logic := '0';

signal s1 : std\_logic := '0';

signal z : std\_logic;

BEGIN

uut: mux PORT MAP (

a => a,

b => b,

c => c,

d => d,

s0 => s0,

s1 => s1,

z => z

);

proc: process

begin

a<='0';

b<='1';

c<='0';

d<='1';

s0<='0';

s1<='0';

wait for 100 ns;

s0<='0';

s1<='1';

wait for 100 ns;

s0<='1';

s1<='0';

wait for 100 ns;

s0<='1';

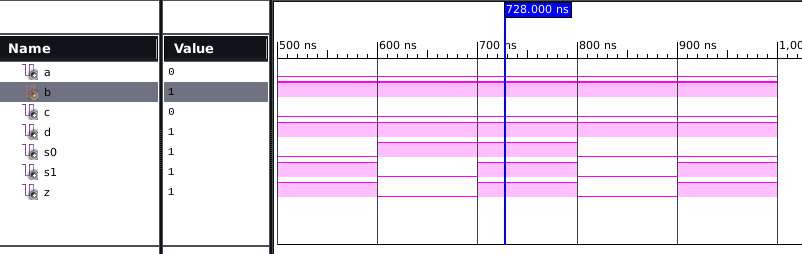
s1<='1';

wait for 100 ns;

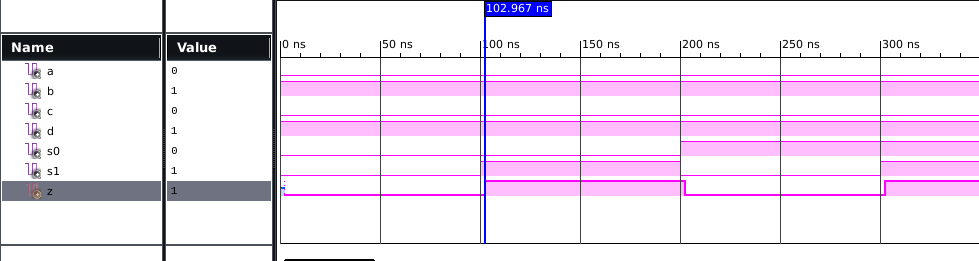
end process;

END;

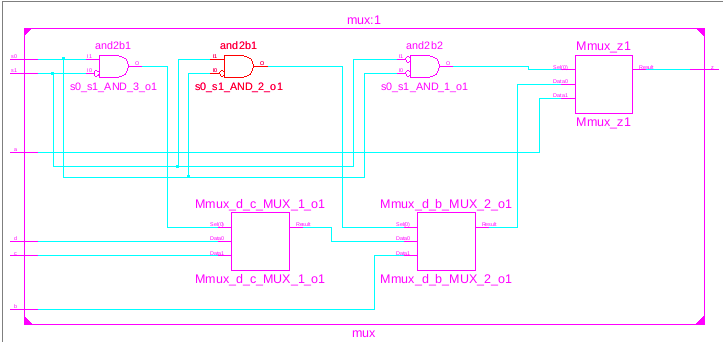
**SIMULATION:**

****

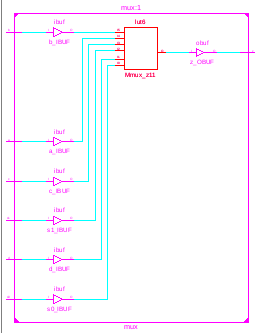
**POST MAP SIMULATION:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

|  |  |  |
| --- | --- | --- |
| AREA  NO. OF SLICE LUTs | 1 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY= 1/5.519 GHZ

POWER=0,029W

**MUX 4:1 (STRUCTURAL)**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity invv is

port (pin : in std\_logic;

pout :out std\_logic);

end invv;

architecture str of invv is

begin

pout<=not pin;

end str;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity andd is

port (a0,a1,a2: in std\_logic;

aout:out std\_logic);

end andd;

architecture str1 of andd is

begin

aout<=a0 and a1 and a2;

end str1;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity orr is

port(r0,r1,r2,r3:in std\_logic;

rout:out std\_logic);

end orr;

architecture str2 of orr is

begin

rout<=r0 or r1 or r2 or r3;

end str2;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity MUX4\_1 is

port ( Sel0,Sel1 : in std\_logic;

A, B, C, D : in std\_logic;

Y : out std\_logic );

end MUX4\_1;

architecture structural of MUX4\_1 is

component invv is

port (pin : in std\_logic;

pout :out std\_logic);

end component;

component andd is

port (a0,a1,a2: in std\_logic;

aout:out std\_logic);

end component;

component orr is

port (r0,r1,r2,r3:in std\_logic;

rout:out std\_logic);

end component;

signal selbar0,selbar1,t1,t2,t3,t4:std\_logic;

begin

INV0: invv port map (Sel0, selbar0);

INV1: invv port map (Sel1, selbar1);

A1: andd port map (A, selbar0, selbar1, t1);

A2: andd port map (B, Sel0, selbar1, t2);

A3: andd port map (C, selbar0, Sel1, t3);

A4: andd port map (D, Sel0, Sel1, t4);

O1: orr port map (t1, t2, t3, t4, Y);

end structural;

**TEST BENCH:**

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

ENTITY tmuxst IS

END tmuxst;

ARCHITECTURE behavior OF tmuxst IS

COMPONENT MUX4\_1

PORT(

Sel0 : IN std\_logic;

Sel1 : IN std\_logic;

A : IN std\_logic;

B : IN std\_logic;

C : IN std\_logic;

D : IN std\_logic;

Y : OUT std\_logic);

END COMPONENT;

--Inputs

signal Sel0 : std\_logic := '0';

signal Sel1 : std\_logic:= '0';

signal A : std\_logic:= '0';

signal B : std\_logic:= '0';

signal C : std\_logic:= '0';

signal D : std\_logic:= '0';

signal Y : std\_logic;

BEGIN

uut: MUX4\_1 PORT MAP (

Sel0 => Sel0,

Sel1 => Sel1,

A => A,

B => B,

C => C,

D => D,

Y => Y

);

stim\_proc: process

begin

A<='0';

B<='1';

C<='0';

D<='1';

sel0<='0';

sel1<='0';

wait for 100 ns;

sel0<='0';

sel1<='1';

wait for 100 ns;

sel0<='1';

sel1<='0';

wait for 100 ns;

sel0<='1';

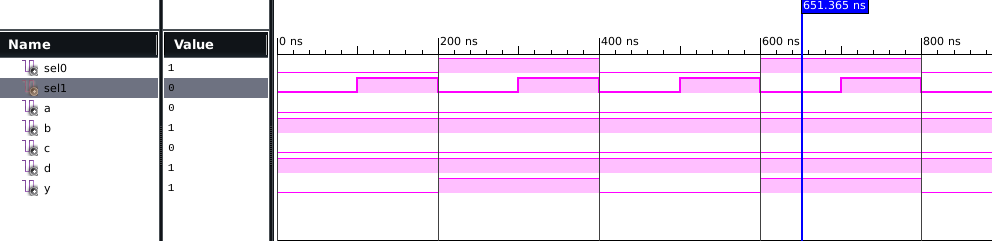
sel1<='1';

wait for 100 ns;

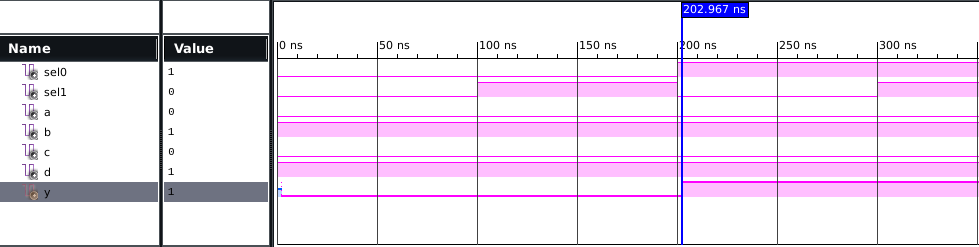
end process;

END;

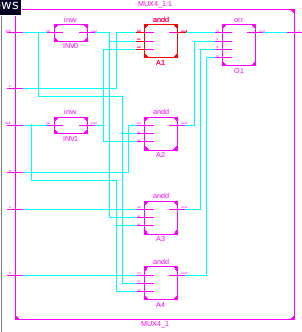
**SIMULATION:**

****

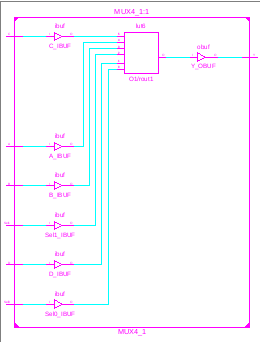
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |
| --- |
| NO. OF SLICE LUTs 1 OUT OF 15032 |
| NO. OF BONDED IOB 7 OUT OF 190 |

POWER = 0.029W

FREQUENCY = 1/5.519 GHZ

**DEMUX 1:4 (DATAFLOW)**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity demux4 is

port ( Y : in std\_logic;

SEL : in std\_logic\_vector (1 downto 0);

D : out std\_logic\_vector (3 downto 0) );

end demux4;

architecture dataflow of demux4 is

begin

D(0) <= (not SEL(0)) and (not SEL(1));

D(1) <= SEL(0) and (not SEL(1));

D(2) <= (not SEL(0)) and SEL(1);

D(3) <= SEL(0) and SEL(1);

end dataflow;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY demux4\_tb IS

END demux4\_tb;

ARCHITECTURE demux4\_tb OF demux4\_tb IS

COMPONENT demux4

PORT(

Y : IN std\_logic;

SEL : IN std\_logic\_vector(1 downto 0);

D : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

signal Y : std\_logic := '1';

signal SEL : std\_logic\_vector(1 downto 0) := (others => '0');

signal D : std\_logic\_vector(3 downto 0);

BEGIN

uut: demux4 PORT MAP (

Y => Y,

SEL => SEL,

D => D

);

stim\_proc: process

begin

SEL(0) <= '0';

SEL(1) <= '0';

wait for 50 ns;

SEL(0) <= '1';

SEL(1) <= '0';

wait for 50 ns;

SEL(0) <= '0';

SEL(1) <= '1';

wait for 50 ns;

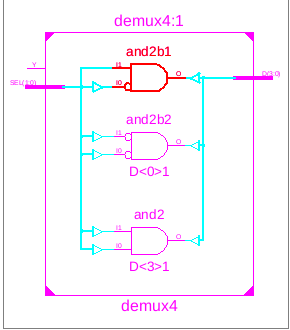
SEL(0) <= '1';

SEL(1) <= '1';

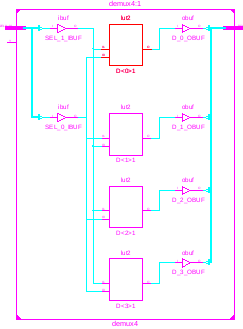
wait; end process;

END;

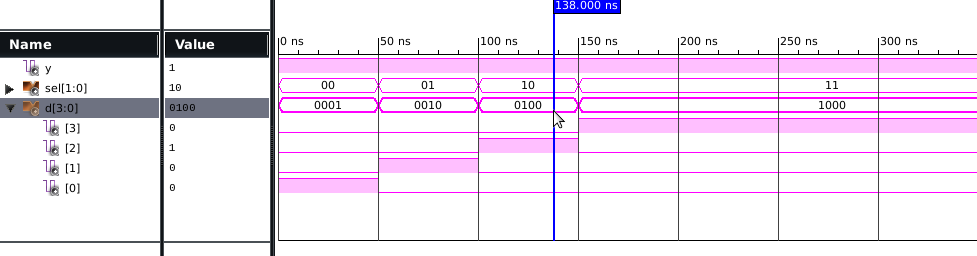
**RTL VIEW:**

****

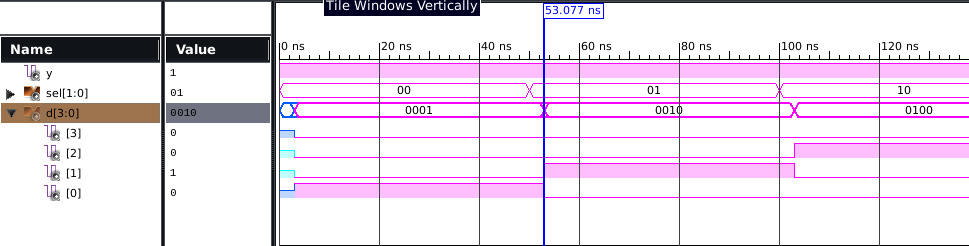
**TECHNICAL VIEW:**

****

**SIMULATION:**

****

**POST MAP:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF BONDED IOB | 7 | 190 |
| NO. OF SLICE LUTs | 4 | 15032 |
|  |  |  |

FREQUENCY = 1/5.362 GHZ

POWER = 0.029W

**DEMUX 1:4 (BEHAVIOUR)**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity demux is

port(f:in std\_logic;

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

a,b,c,d:out std\_logic);

end demux;

architecture Behavioral of demux is

begin

process(f,s)

begin

case(s) is

when "00"=>a<=f;

when "01"=>b<=f;

when "10"=>c<=f;

when others=>d<=f;

end case;

end process;

end Behavioral;

**TEST BENCH:**

LIBRARY ieee;

ENTITY tdemux IS

END tdemux;

ARCHITECTURE behavior OF tdemux IS

COMPONENT demux

PORT(

f : IN std\_logic;

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

a : OUT std\_logic;

b : OUT std\_logic;

c : OUT std\_logic;

d : OUT std\_logic

);

END COMPONENT;

signal f : std\_logic := '0';

signal s : std\_logic\_vector(1 downto 0) := (others => '0');

signal a : std\_logic;

signal b : std\_logic;

signal c : std\_logic;

signal d : std\_logic;

BEGIN

f => f,

s => s,

a => a,

b => b,

c => c,

d => d

);

proc:process

begin

f<='1';

s<="00";

wait for 100ns;

s<="01";

wait for 100ns;

s<="10";

wait for 100ns;

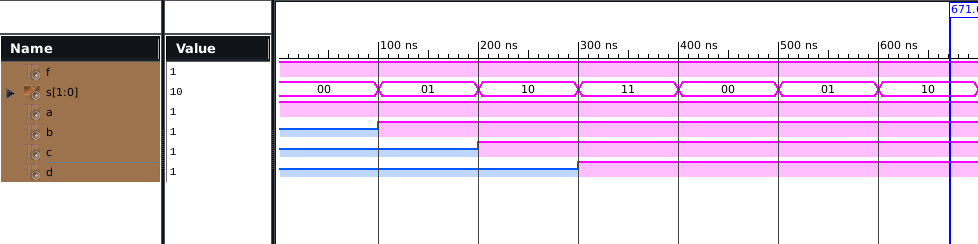
s<="11";

wait for 100ns;

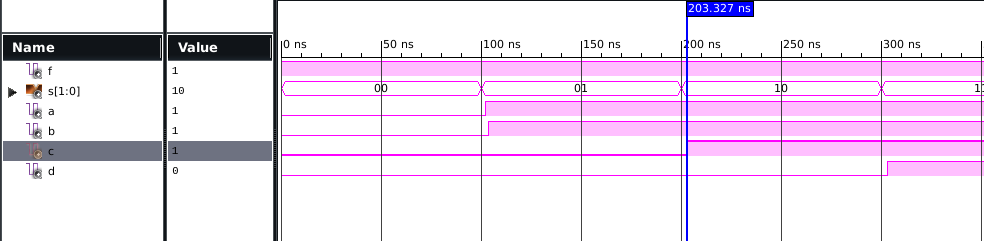
end process proc;

END;

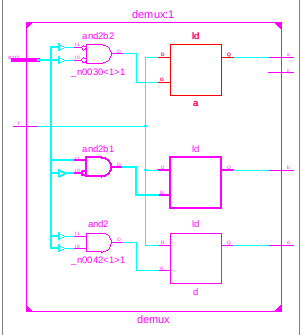
**SIMULATION:**

****

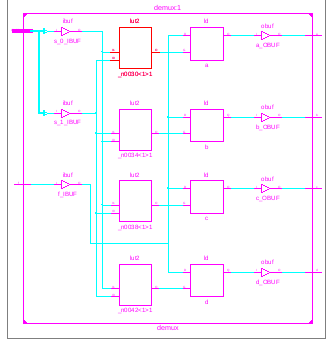
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 4 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY = 1/1.942 GHZPOWER = 0.029W

**DEMUX 1:4 (STRUCTURAL)**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity inv2 is

port(pin:in std\_logic;

pout:out std\_logic);

end inv2;

architecture str of inv2 is

begin

pout<=not pin;

end str;

library ieee;

use ieee.std\_logic\_1164.all;

entity anddd is

port(a0,a1,a2:in std\_logic;

aout:out std\_logic);

end anddd;

architecture strl of anddd is

begin

aout<=a0 and a1 and a2;

end strl;

library ieee;

use ieee.std\_logic\_1164.all;

entity demux4\_1 is

port ( Sel0,Sel1,inp: in std\_logic;

A, B, C, D : out std\_logic);

end demux4\_1;

architecture structural of demux4\_1 is

component inv2

port (pin : in std\_logic;

pout :out std\_logic);

end component;

component anddd

port (a0,a1,a2: in std\_logic;

aout:out std\_logic);

end component;

signal selbar0,selbar1: std\_logic;

begin

INV0: inv2 port map (Sel0, selbar0);

INV1: inv2 port map (Sel1, selbar1);

A1: anddd port map (inp,selbar1,selbar0,A);

A2: anddd port map (inp,selbar1,sel0,B);

A3: anddd port map (inp,sel1,selbar0,C);

A4: anddd port map (inp,sel1,sel0,D);

end structural;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tstrde IS

END tstrde;

ARCHITECTURE behavior OF tstrde IS

COMPONENT demux4\_1

PORT(

Sel0 : IN std\_logic;

Sel1 : IN std\_logic;

A : OUT std\_logic;

B : OUT std\_logic;

C : OUT std\_logic;

D : OUT std\_logic;

inp : IN std\_logic

);

END COMPONENT;

--Inputs

signal Sel0 : std\_logic := '0';

signal Sel1 : std\_logic := '0';

signal inp : std\_logic:= '0';

signal A : std\_logic;

signal B : std\_logic;

signal C : std\_logic;

signal D : std\_logic;

BEGIN

uut: demux4\_1 PORT MAP (

Sel0 => Sel0,

Sel1 => Sel1,

A => A,

B => B,

C => C,

D => D,

inp => inp

);

stim\_proc: process

begin

inp<='1';

SEL0 <= '0';

SEL1 <= '0';

wait for 50 ns;

SEL0 <= '1';

SEL1 <= '0';

wait for 50 ns;

SEL0 <= '0';

SEL1 <= '1';

wait for 50 ns;

SEL0 <= '1';

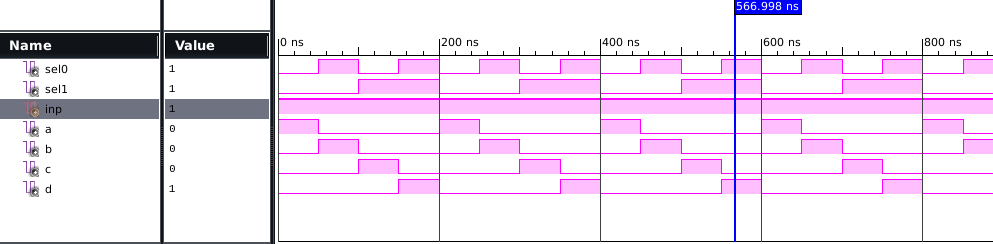
SEL1 <= '1';

wait for 50 ns;

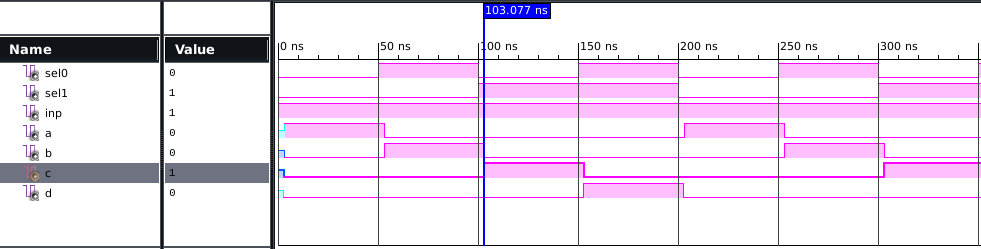
end process;

END;

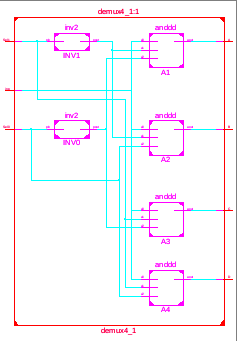
**SIMULATION:**

****

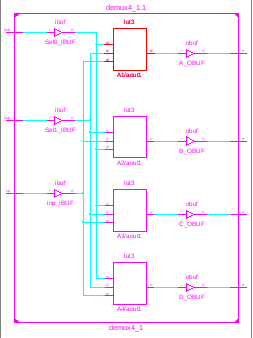
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 4 | 15032 |
| NO. OF BONDED IOB | 7 | 190 |

FREQUENCY = 1/5.488 GHZ POWER=0.029W

**BINARY TO GREY(DATAFLOW):**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity btg is

port(a:in std\_logic\_vector(3 downto 0);

b:out std\_logic\_vector(3 downto 0));

end btg;

architecture Behavioral of btg is

begin

b(3)<=a(3);

b(2)<=a(3) xor a(2);

b(1)<=a(2) xor a(1);

b(0)<=a(1) xor a(0);

end Behavioral;

**TESTBENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tbtg IS

END tbtg;

ARCHITECTURE behavior OF tbtg IS

COMPONENT btg

PORT(

a : IN std\_logic\_vector(3 downto 0);

b : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

signal a : std\_logic\_vector(3 downto 0) := (others => '0');

signal b : std\_logic\_vector(3 downto 0);

BEGIN

uut: btg PORT MAP (

a => a,

b => b

);

proc: process

begin

a<="0000"; wait for 10ns;

a<="0001"; wait for 10ns;

a<="0010"; wait for 10ns;

a<="0011"; wait for 10ns;

a<="0100"; wait for 10ns;

a<="0101"; wait for 10ns;

a<="0110"; wait for 10ns;

a<="0111"; wait for 10ns;

a<="1000"; wait for 10ns;

a<="1001"; wait for 10ns;

a<="1010"; wait for 10ns;

a<="1011"; wait for 10ns;

a<="1100"; wait for 10ns;

a<="1101"; wait for 10ns;

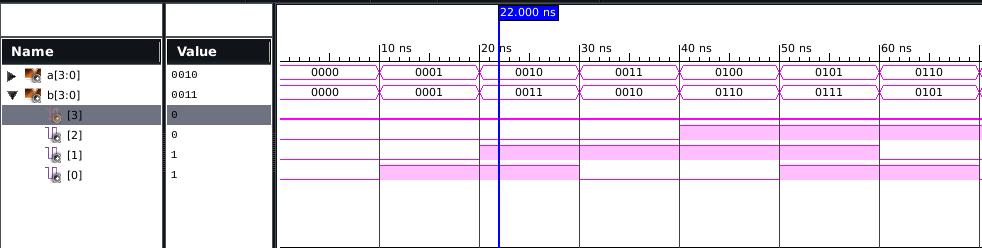
a<="1110"; wait for 10ns;

a<="1111"; wait for 10ns;

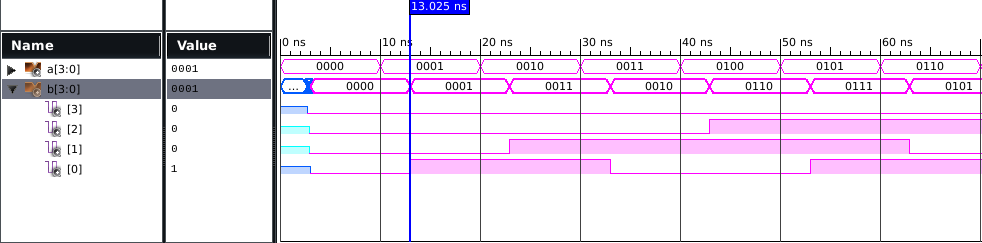
wait;

end process; END;

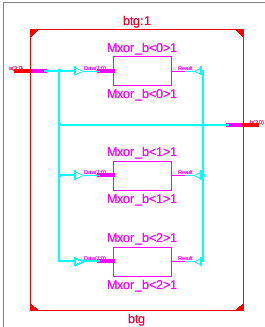
**SIMULATION:**

****

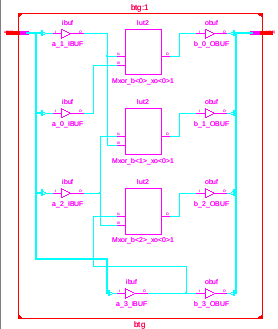
**POST-MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FRQUENCY = 1/5.296 GHZ POWER=0.029W

**BINARY TO GREY (BEHAVIOUR):**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity bbinary\_to\_gray is

port(

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

dout : out STD\_LOGIC\_VECTOR(3 downto 0)

);

end bbinary\_to\_gray;

architecture beh of bbinary\_to\_gray is

begin

process (din) is

begin

if (din="0000") then

dout <= "0000";

elsif (din="0001") then

dout <= "0001";

elsif (din="0010") then

dout <= "0011";

elsif (din="0011") then

dout <= "0010";

elsif (din="0100") then

dout <= "0110";

elsif (din="0101") then

dout <= "0111";

elsif (din="0110") then

dout <= "0101";

elsif (din="0111") then

dout <= "0100";

elsif (din="1000") then

dout <= "1100";

elsif (din="1001") then

dout <= "1101";

elsif (din="1010") then

dout <= "1111";

elsif (din="1011") then

dout <= "1110";

elsif (din="1100") then

dout <= "1010";

elsif (din="1101") then

dout <= "1011";

elsif (din="1110") then

dout <= "1001";

else

dout <= "1000";

end if;

end process;

end beh;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tbbtg IS

END tbbtg;

ARCHITECTURE behavior OF tbbtg IS

COMPONENT bbinary\_to\_gray

PORT(

din : IN std\_logic\_vector(3 downto 0);

dout : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

signal din : std\_logic\_vector(3 downto 0) := (others => '0');

signal dout : std\_logic\_vector(3 downto 0);

BEGIN

uut: bbinary\_to\_gray PORT MAP (

din => din,

dout => dout

);

proc: process

begin

din<="0000"; wait for 10ns;

din<="0001"; wait for 10ns;

din<="0010"; wait for 10ns;

din<="0011"; wait for 10ns;

din<="0100"; wait for 10ns;

din<="0101"; wait for 10ns;

din<="0110"; wait for 10ns;

din<="0111"; wait for 10ns;

din<="1000"; wait for 10ns;

din<="1001"; wait for 10ns;

din<="1010"; wait for 10ns;

din<="1011"; wait for 10ns;

din<="1100"; wait for 10ns;

din<="1101"; wait for 10ns;

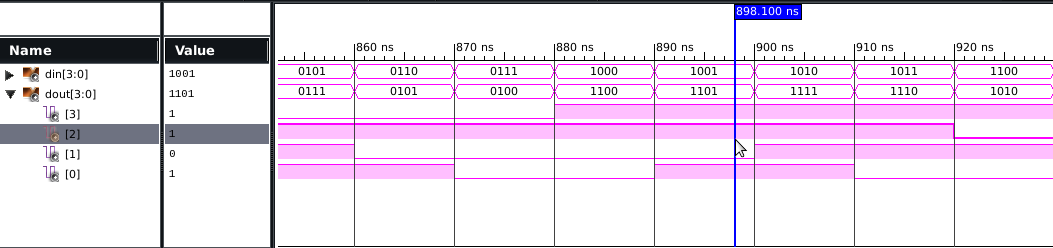
din<="1110"; wait for 10ns;

din<="1111"; wait for 10ns;

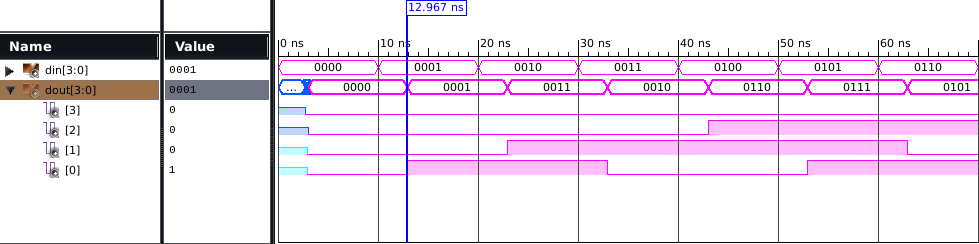
end process;

END;

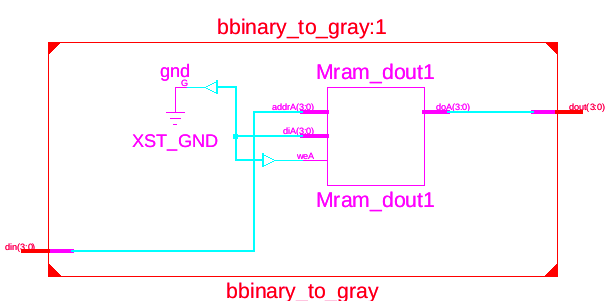
**SIMULATION:**

****

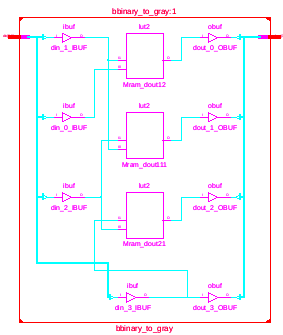
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FRQUENCY = 1/5.296 GHZ

POWER=0.029W

**BINARY TO GREY(STRUCTURAL):**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity xor\_gate is

port(l,m:in std\_logic;

n:out std\_logic);

end xor\_gate;

architecture behv of xor\_gate is

begin

n<=l xor m;

end behv;

library ieee;

use ieee.std\_logic\_1164.all;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity sbtog is

port(a: in std\_logic\_vector(3 downto 0);

b: out std\_logic\_vector(3 downto 0));

end sbtog;

architecture structure of sbtog is

component xor\_gate is

port(l,m:in std\_logic;n: out std\_logic);

end component;

begin

b(3)<=a(3);

x2: xor\_gate port map(a(3),a(2),b(2));

x3: xor\_gate port map(a(2),a(1),b(1));

x4: xor\_gate port map(a(1),a(0),b(0));

end structure

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tstrbtg IS

END tstrbtg;

ARCHITECTURE behavior OF tstrbtg IS

COMPONENT sbtog

PORT(

a : IN std\_logic\_vector(3 downto 0);

b : OUT std\_logic\_vector(3 downto 0));

END COMPONENT;

signal a : std\_logic\_vector(3 downto 0) := (others => '0');

signal b : std\_logic\_vector(3 downto 0);

BEGIN

uut: sbtog PORT MAP (

a => a,

b => b

);

-- Stimulus process

stim\_proc: process

begin

a<="0000"; wait for 10ns;

a<="0001"; wait for 10ns;

a<="0010"; wait for 10ns;

a<="0011"; wait for 10ns;

a<="0100"; wait for 10ns;

a<="0101"; wait for 10ns;

a<="0110"; wait for 10ns;

a<="0111"; wait for 10ns;

a<="1000"; wait for 10ns;

a<="1001"; wait for 10ns;

a<="1010"; wait for 10ns;

a<="1011"; wait for 10ns;

a<="1100"; wait for 10ns;

a<="1101"; wait for 10ns;

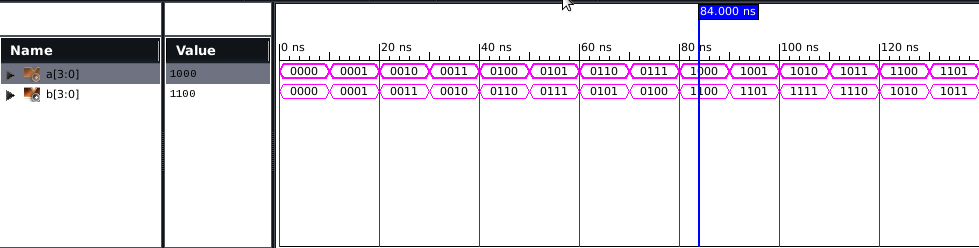
a<="1110"; wait for 10ns;

a<="1111"; wait for 10ns;

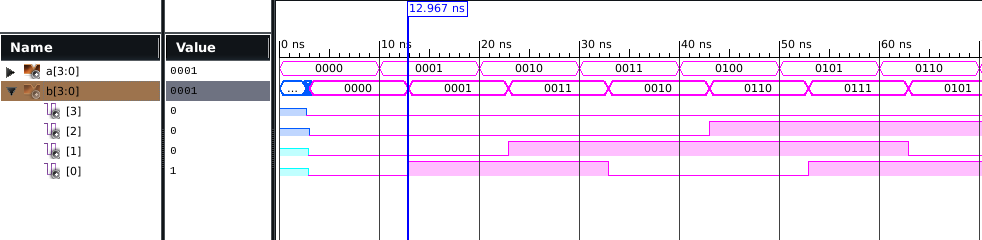
end process;

END;

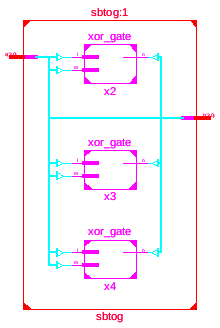
**SIMULATION:**

****

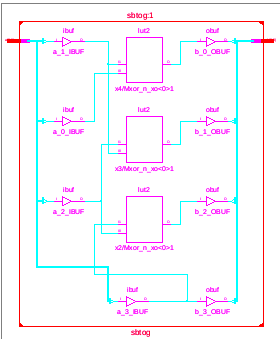
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FRQUENCY = 1/5.296 GHZ POWER=0.029W

**GREY TO BINARY(DATAFLOW):**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity gtb is

port(a:in std\_logic\_vector(3 downto 0);

b:out std\_logic\_vector(3 downto 0));

end gtb;

architecture Behavioral of gtb is

begin

b(3)<=a(3);

b(2)<=a(3) xor a(2);

b(1)<=a(3) xor a(2) xor a(1);

b(0)<=a(3) xor a(2) xor a(1) xor a(0);

end Behavioral;

**TESTBENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tgtb IS

END tgtb;

ARCHITECTURE behavior OF tgtb IS

COMPONENT gtb

PORT(

a : IN std\_logic\_vector(3 downto 0);

b : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

signal a : std\_logic\_vector(3 downto 0) := (others => '0');

signal b : std\_logic\_vector(3 downto 0);

BEGIN

uut: gtb PORT MAP (

a => a,

b => b

);

proc: process

begin

a<="0000"; wait for 10ns;

a<="0001"; wait for 10ns;

a<="0010"; wait for 10ns;

a<="0011"; wait for 10ns;

a<="0100"; wait for 10ns;

a<="0101"; wait for 10ns;

a<="0110"; wait for 10ns;a<="0111"; wait for 10ns;

a<="1000"; wait for 10ns;

a<="1001"; wait for 10ns;

a<="1010"; wait for 10ns;

a<="1011"; wait for 10ns;

a<="1100"; wait for 10ns;

a<="1101"; wait for 10ns;

a<="1110"; wait for 10ns;

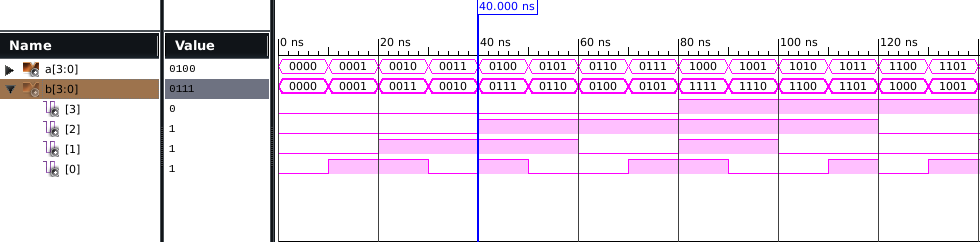
a<="1111"; wait for 10ns;

wait;

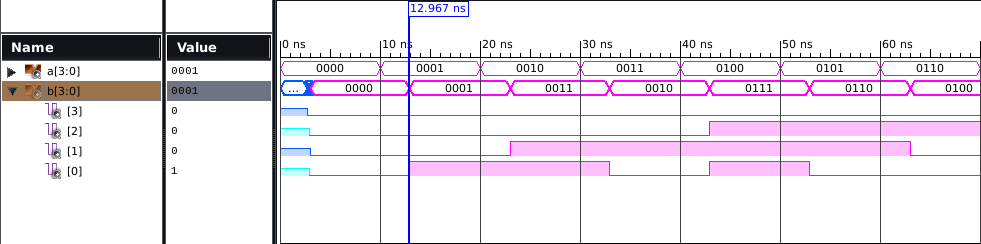
end process;

END;

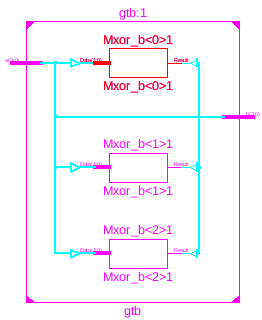
**SIMULATION:**

****

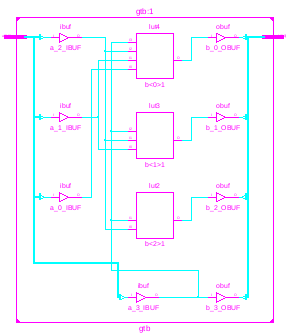
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FREQUENCY= 1/5.422GHZ POWER = 0.029W

**GREY TO BINARY(BEHAVIOUR):**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity bhvgtb is

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

dout : out STD\_LOGIC\_VECTOR(3 downto 0)

);

end bhvgtb;

architecture Behavioral of bhvgtb is

begin

process (din) is

begin

if (din="0000") then

dout <= "0000";

elsif (din="0001") then

dout <= "0001";

elsif (din="0010") then

dout <= "0011";

elsif (din="0011") then

dout <= "0010";

elsif (din="0100") then

dout <= "0111";

elsif (din="0101") then

dout <= "0110";

elsif (din="0110") then

dout <= "0100";

elsif (din="0111") then

dout <= "0101";

elsif (din="1000") then

dout <= "1111";

elsif (din="1001") then

dout <= "1110";

elsif (din="1010") then

dout <= "1100";

elsif (din="1011") then

dout <= "1101";

elsif (din="1100") then

dout <= "1000";

elsif (din="1101") then

dout <= "1001";

elsif (din="1110") then

dout <= "1011";

else

dout <="1010";

end if;

end process;

end Behavioral;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tbhvgtb IS

END tbhvgtb;

ARCHITECTURE behavior OF tbhvgtb IS

COMPONENT bhvgtb

PORT(

din : IN std\_logic\_vector(3 downto 0);

dout : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal din : std\_logic\_vector(3 downto 0) := (others => '0');

--Outputs

signal dout : std\_logic\_vector(3 downto 0);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: bhvgtb PORT MAP (

din => din,

dout => dout

);

-- Stimulus process

stim\_proc: process

begin

din<="0000"; wait for 10ns;

din<="0001"; wait for 10ns;

din<="0010"; wait for 10ns;

din<="0011"; wait for 10ns;

din<="0100"; wait for 10ns;

din<="0101"; wait for 10ns;

din<="0110"; wait for 10ns;

din<="0111"; wait for 10ns;

din<="1000"; wait for 10ns;

din<="1001"; wait for 10ns;

din<="1010"; wait for 10ns;

din<="1011"; wait for 10ns;

din<="1100"; wait for 10ns;

din<="1101"; wait for 10ns;

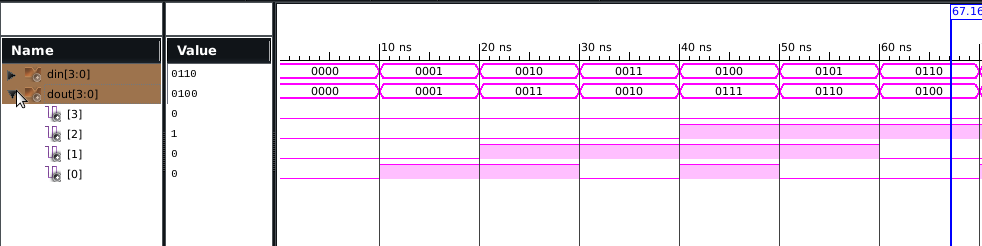
din<="1110"; wait for 10ns;

din<="1111"; wait for 10ns;

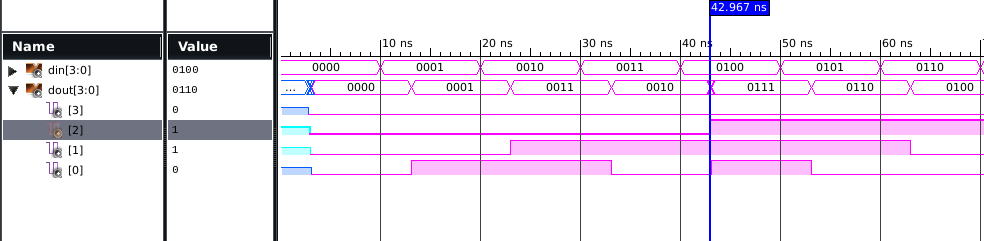
end process;

END;

**SIMULATION:**

****

**POST MAP:**

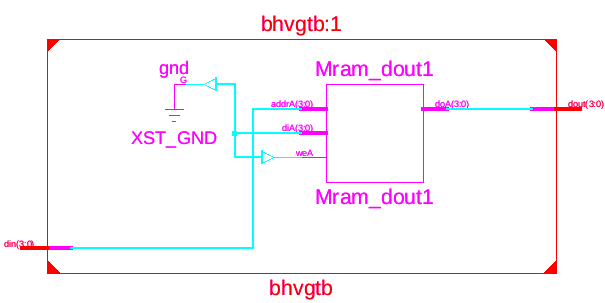
****

AREA

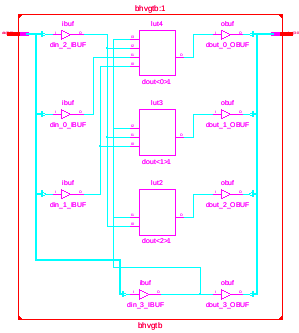
|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FREQUENCY= 1/5.439GHZ POWER = 0.029W

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

**GREY TO BINARY(STRUCTURAL):**

**CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity xor\_gate is

port(l,m:in std\_logic;

n:out std\_logic);

end xor\_gate;

architecture behv of xor\_gate is

begin

n<=l xor m;

end behv;

library ieee;

use ieee.std\_logic\_1164.all;

entity str\_gtb is

port(a: in std\_logic\_vector(3 downto 0);

b: inout std\_logic\_vector(3 downto 0));

end str\_gtb;

architecture structure of str\_gtb is

component xor\_gate is

port(l,m:in std\_logic;n: out std\_logic);

end component;

begin

b(3)<=a(3);

x2: xor\_gate port map(a(3),a(2),b(2));

x3: xor\_gate port map(b(2),a(1),b(1));

x4: xor\_gate port map(b(1),a(0),b(0));

end structure;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tstr\_gtb IS

END tstr\_gtb;

ARCHITECTURE behavior OF tstr\_gtb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT str\_gtb

PORT(

a : IN std\_logic\_vector(3 downto 0);

b : INOUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal a : std\_logic\_vector(3 downto 0) := (others => '0');

--BiDirs

signal b : std\_logic\_vector(3 downto 0);

BEGIN

uut: str\_gtb PORT MAP (

a => a,

b => b

);

stim\_proc: process

begin

a<="0000"; wait for 10ns;

a<="0001"; wait for 10ns;

a<="0010"; wait for 10ns;

a<="0011"; wait for 10ns;

a<="0100"; wait for 10ns;

a<="0101"; wait for 10ns;

a<="0110"; wait for 10ns;

a<="0111"; wait for 10ns;

a<="1000"; wait for 10ns;

a<="1001"; wait for 10ns;

a<="1010"; wait for 10ns;

a<="1011"; wait for 10ns;

a<="1100"; wait for 10ns;

a<="1101"; wait for 10ns;

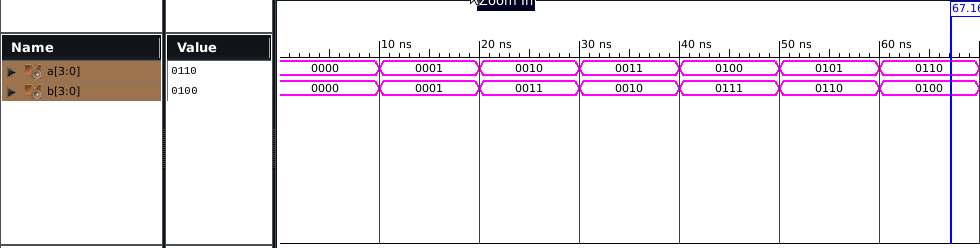
a<="1110"; wait for 10ns;

a<="1111"; wait for 10ns;

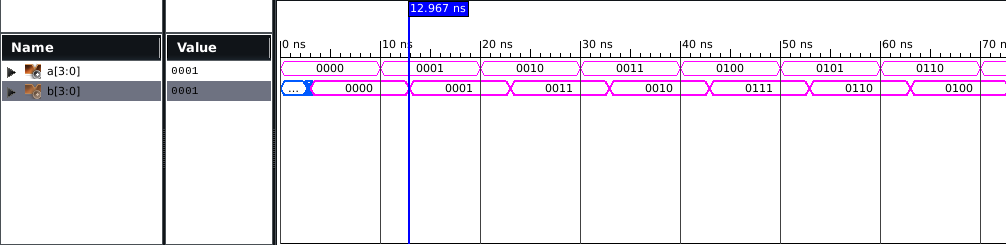
end process;

END;

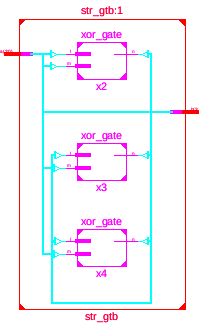
**SIMULATION:**

****

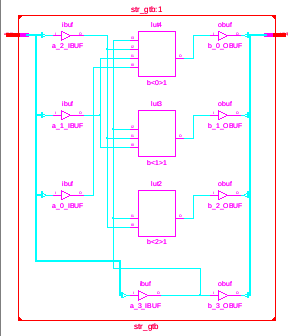
**POST MAP:**

****

**RTL VIEW:**

****

**TECHNICAL VIEW:**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 3 | 15032 |
| NO. OF BONDED IOB | 8 | 190 |

FREQUENCY= 1/5.422GHZ POWER = 0.029W

**DESIGN ALL GATES USING VHDL**

**VHDL CODE FOR AND GATE**

**DATA FLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and1 is

port(a,b : in std\_logic ;

c : out std\_logic );

end and1;

architecture Behavioral of and1 is

begin

c <= a and b;

end Behavioral;

**BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and1beh is

port(a,b: in std\_logic;

c : out std\_logic);

end and1beh;

architecture Behavioral of and1beh is

begin

process(a,b)

begin

if(a='1' and b='1')then

c<='1';

else

c<='0';

end if;

end process;

end Behavioral;

**VHDL TESTBENCH CODE FOR AND GATE**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testand IS

END testand;

ARCHITECTURE behavior OF testand IS

COMPONENT and1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

uut: and1 PORT MAP (

a => a,

b => b,

c => c

);

-- Stimulus process

proc: process

begin

a<='0';

b<='0';

wait for 50 ns;

a<='0';

b<='1';

wait for 50 ns;

a<='1';

b<='0';

wait for 50 ns;

a<='1';

b<='1';

wait for 50 ns;

end process;

END;

**BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testand1beh IS

END testand1beh;

ARCHITECTURE behavior OF testand1beh IS

COMPONENT and1beh

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: and1beh PORT MAP (

a => a,

b => b,

c => c);

-- Stimulus process

stim\_proc: process

begin

a<='0';

b<='0';

wait for 50 ns;

a<='0';

b<='1';

wait for 50 ns;

a<='1';

b<='0';

wait for 50 ns;

a<='1';

b<='1';

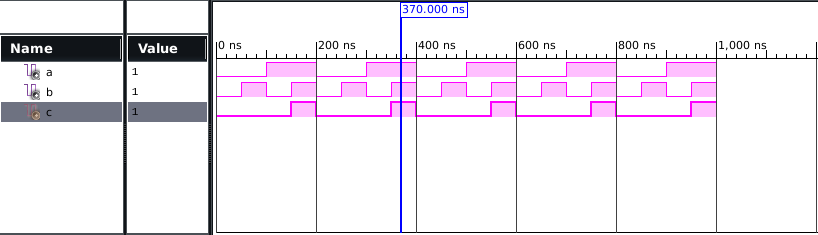
wait for 50 ns;

end process;

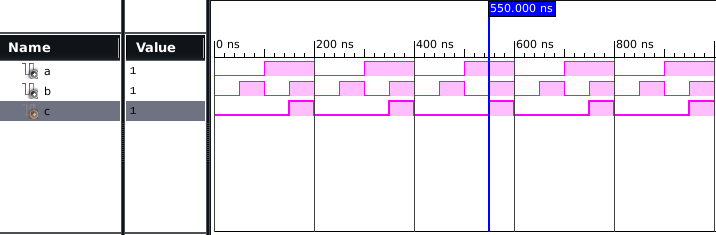
END;

**WAVEFORM**

**DATAFLOW MODELLING**

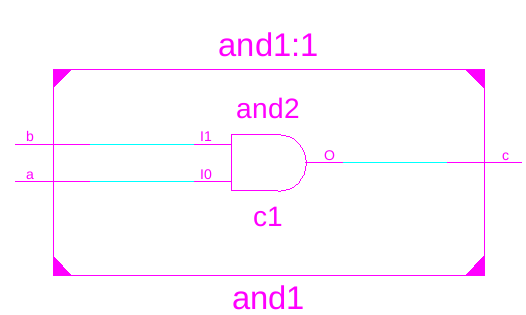


**BEHAVIORAL MODELLING**

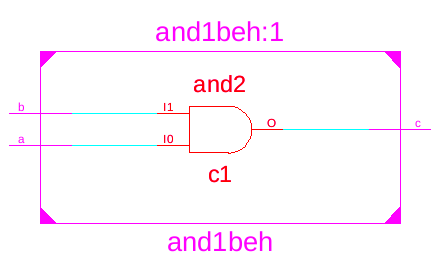
****

**RTL SCHEMATIC**

**DATAFLOW MODELLING**

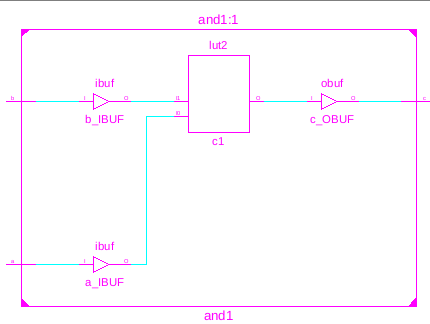
****

**BEHAVIORAL MODELLING**

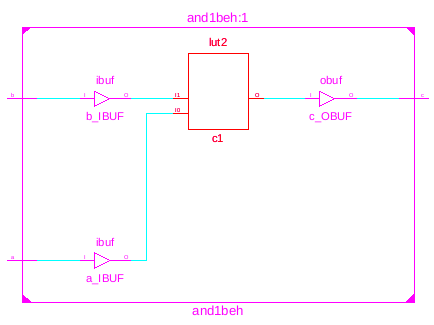
****

**TECHNOLOGY SCHEMATIC**

**DATAFLOW MODELLING**

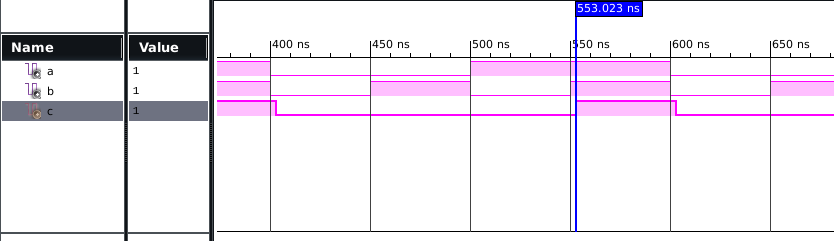
****

**BEHAVIORAL MODELLING**

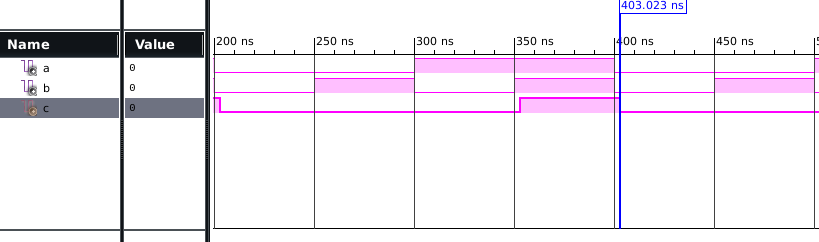
****

**POST-MAP SYNTHESIS**

**DATAFLOW MODELLING**

****

**BEHAVIORAL MODELLING**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 1 | 15032 |
| NO. USING O6 OUTPUT ONLY | 1 |  |
| NO. OF BONDED IOB | 3 | 190 |
| AVG. FANOUT OF NON CLK NETS | 1 |  |

FREQUENCY = 1/5.299GHZ POWER=0.029W

**VHDL CODE FOR OR GATE**

1. **DATA FLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or1 is

port(a,b : in std\_logic;

c : out std\_logic);

end or1;

architecture Behavioral of or1 is

begin

c<=a or b;

end Behavioral;

1. **BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity or1beh is

port(a,b : in std\_logic;

c: out std\_logic);

end or1beh;

architecture Behavioral of or1beh is

begin

process(a,b)

begin

if(a='0' and b='0')then

c<='0';

else

c<='1';

end if;

end process;

end Behavioral;

**VHDL TESTBENCH CODE FOR OR GATE**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testor1 IS

END testor1;

ARCHITECTURE behavior OF testor1 IS

COMPONENT or1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

uut: or1 PORT MAP (

a => a,

b => b,

c => c

);

-- Stimulus process

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

wait for 100 ns;

end process;

END;

1. **BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testor1beh IS

END testor1beh;

ARCHITECTURE behavior OF testor1beh IS

COMPONENT or1beh

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic;

BEGIN

uut: or1beh PORT MAP (

a => a,

b => b,

c => c

);

-- Stimulus process

stim\_proc: process

begin

a<='0';

b<='0';

wait for 50 ns;

a<='0';

b<='1';

wait for 50 ns;

a<='1';

b<='0';

wait for 50 ns;

a<='1';

b<='1';

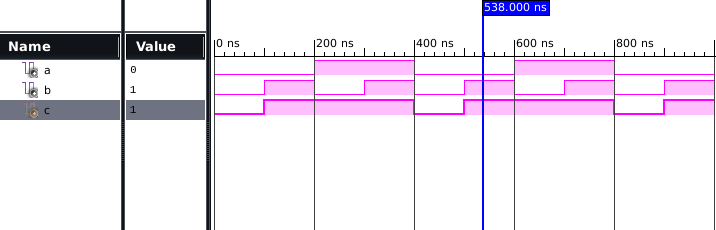
wait for 50 ns;

end process;

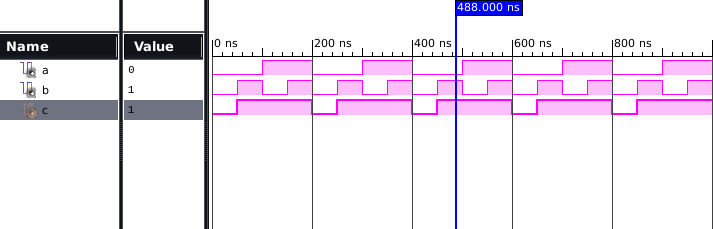
END;

**WAVEFORM**

1. **DATAFLOW MODELLING**

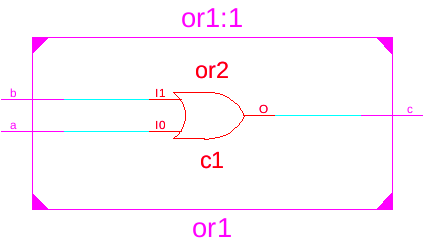
****

**BEHAVIORAL MODELLING**

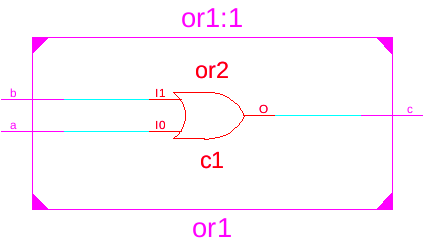
****

**RTL SCHEMATIC**

**DATAFLOW MODELLING**

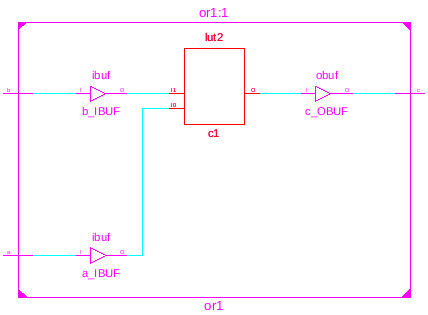
****

**BEHAVIORAL MODELLING**

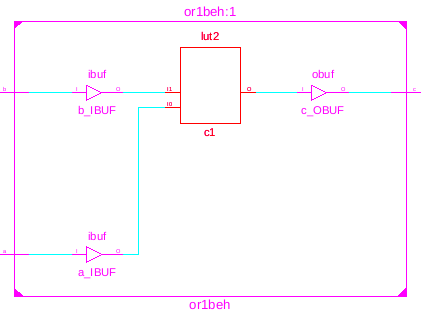
****

**TECHNOLOGY SCHEMATIC**

**DATAFLOW MODELLING**

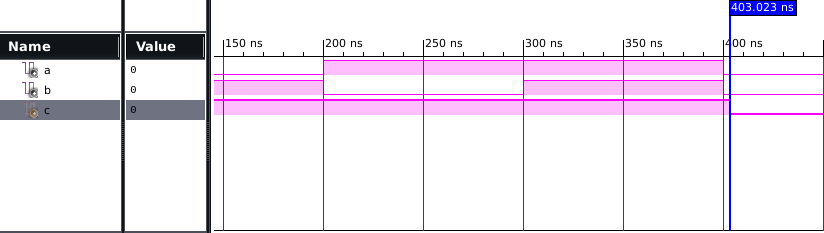
****

**BEHAVIORAL MODELLING**

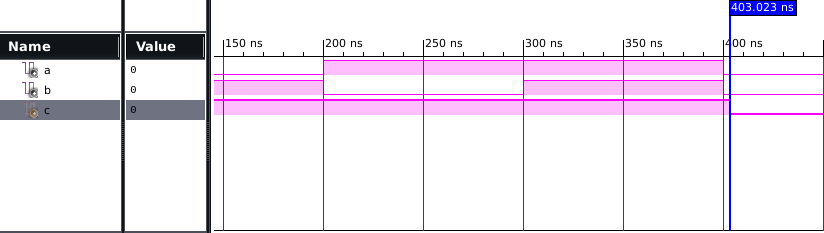
****

**POST – MAP SYNTHESIS**

**DATAFLOW MODELLING**

****

**BEHAVIORAL MODELLING**

****

|  |  |  |
| --- | --- | --- |
| AREA  NO. OF SLICE LUTs | 1 | 15032 |
| NO. USING O6 OUTPUT ONLY | 1 |  |
| NO. OF BONDED IOB | 3 | 190 |
| AVG. FANOUT OF NON CLK NETS | 1 |  |

FREQUENCY= 1/5.299GHZ

POWER=0.029W

**VHDL CODE FOR NOT GATE**

1. **DATA FLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not1 is

port(a : in std\_logic;

c : out std\_logic);

end not1;

architecture Behavioral of not1 is

begin

c<= not a;

end Behavioral;

1. **BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity not1beh is

port(a:in std\_logic;

c: out std\_logic);

end not1beh;

architecture Behavioral of not1beh is

begin

process(a)

begin

if (a='0')then

c<='1';

else

c<='0';

end if;

end process;

end Behavioral;

**VHDL TESTBENCH CODE FOR NOT GATE**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testnot1 IS

END testnot1;

ARCHITECTURE behavior OF testnot1 IS

COMPONENT not1

PORT(

a : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: not1 PORT MAP (

a => a,

c => c

);

stim\_proc: process

begin

a<='0';

wait for 100 ns;

a<='1';

wait for 100 ns;

end process;

END;

1. **BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testnot1beh IS

END testnot1beh;

ARCHITECTURE behavior OF testnot1beh IS

COMPONENT not1beh

PORT(

a : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

uut: not1beh PORT MAP (

a => a,

c => c

);

stim\_proc: process

begin

a<='0';

wait for 100 ns;

a<='1';

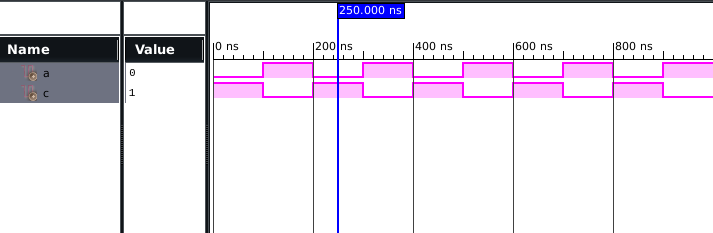
wait for 100 ns;

end process;

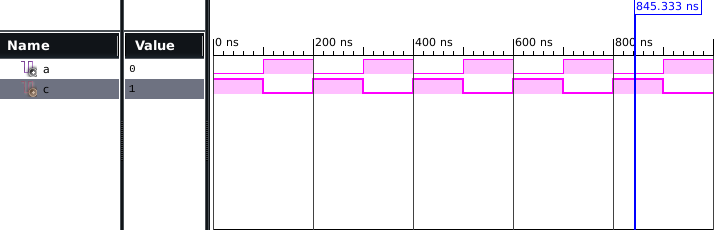
END;

**WAVEFORM**

**DATAFLOW MODELLING**

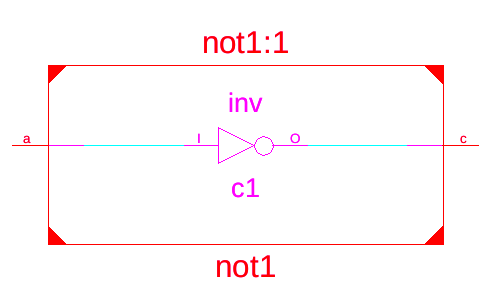
****

**BEHAVIORAL MODELLING**

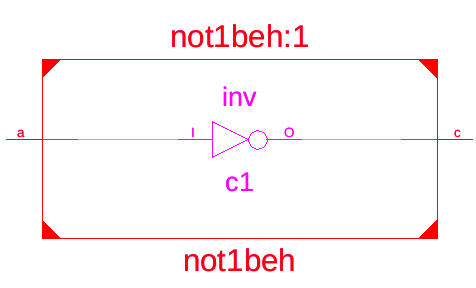
****

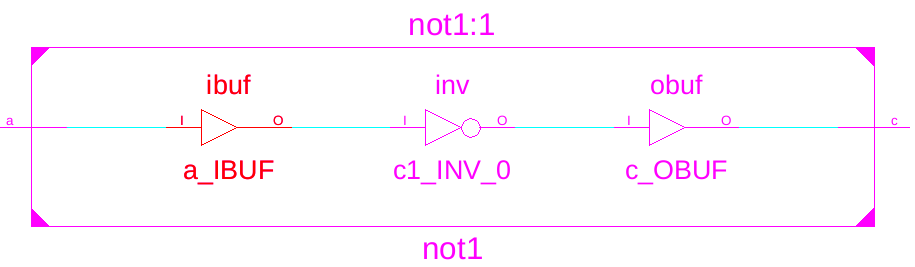
**RTL SCHEMATIC**

**DATAFLOW MODELLING**

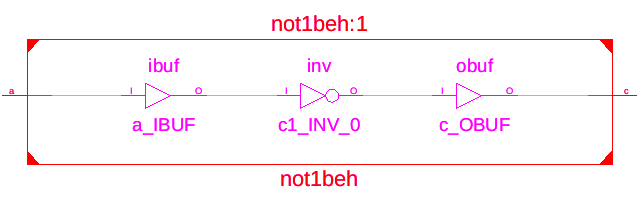


**BEHAVIORAL MODELLING**

****

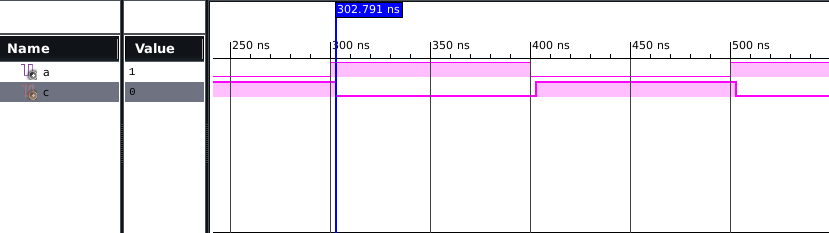
**TECHNOLOGY SCHEMATIC DATAFLOW MODELLING**

**BEHAVIORAL MODELLING**

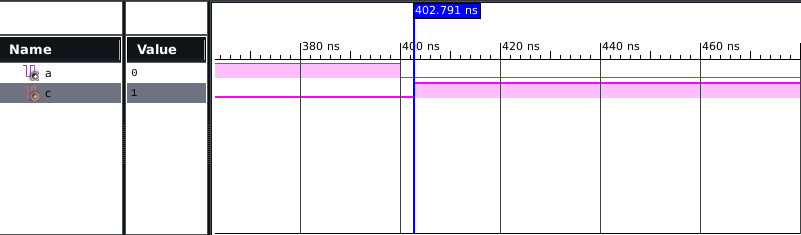
****

**POST-MAP SYNTHESIS**

**DATAFLOW MODELLING**

****

**BEHAVIORAL MODELLING**

****

**VHDL CODE FOR NAND GATE**

1. **DATA FLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity nand1 is

port(a,b: in std\_logic;

c: out std\_logic);

end nand1;

architecture Behavioral of nand1 is

begin

c<= a nand b;

end Behavioral;

1. **BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity nand1beh is

port(a,b: in std\_logic; c : out std\_logic);

end nand1beh;

architecture Behavioral of nand1beh is

begin

process(a,b)

begin

if(a='1' and b='1')then c<='0';

else c<='1';

end if;

end process;

end Behavioral

**VHDL TESTBENCH CODE FOR NAND GATE**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testnand1 IS

END testnand1;

ARCHITECTURE behavior OF testnand1 IS

COMPONENT nand1

PORT(a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

uut: nand1 PORT MAP (

a => a,

b => b,

c => c

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

wait for 100 ns;

end process;

END;

1. **BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testnand1beh IS

END testnand1beh;

ARCHITECTURE behavior OF testnand1beh IS

COMPONENT nand1beh

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: nand1beh PORT MAP (

a => a,

b => b,

c => c

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

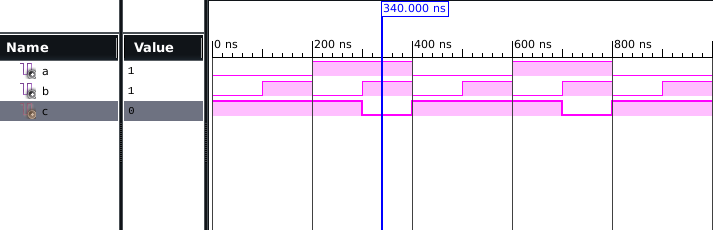
wait for 100 ns;

end process;

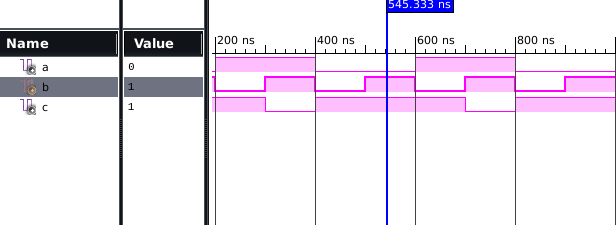
END;

**WAVEFORM**

**DATAFLOW MODELLING**

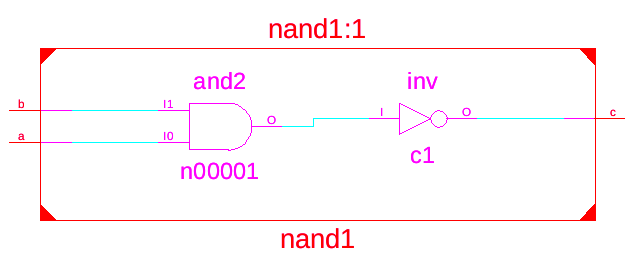
****

**BEHAVIORAL MODELLING**

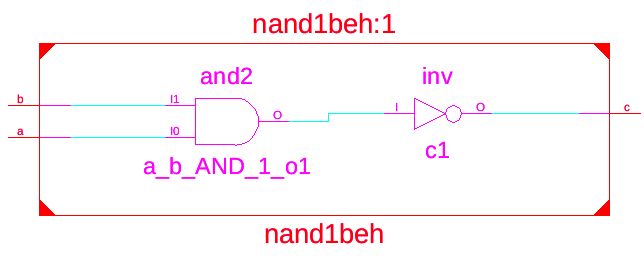
****

**RTL SCHEMATIC**

**DATAFLOW MODELLING**

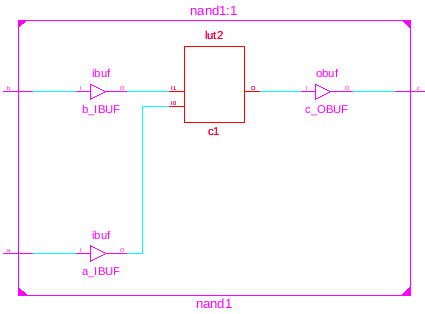
****

**BEHAVIORAL MODELLING**

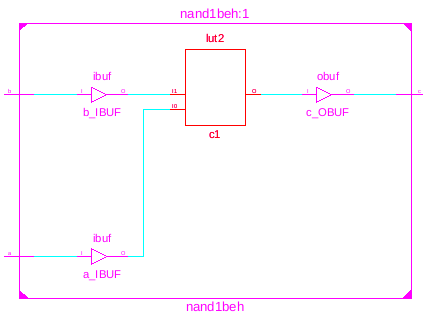
****

**TECHNOLOGY SCHEMATIC**

**DATAFLOW MODELLING**

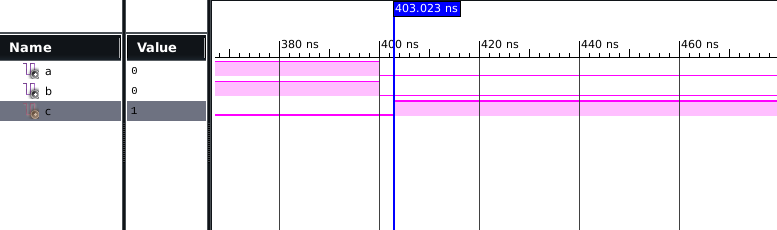
****

**BEHAVIORAL MODELLING**

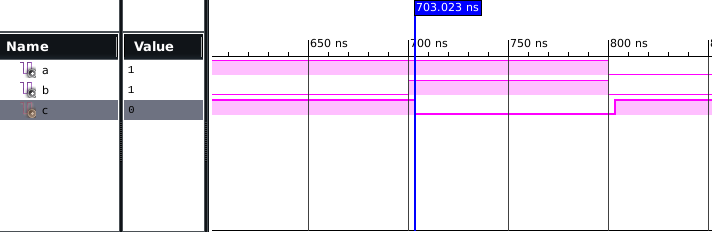
****

**POST-MAP SYNTHESIS**

**DATAFLOW MODELLING**

****

**BEHAVIORAL MODELLING**

****

|  |  |  |
| --- | --- | --- |
| AREA  NO. OF SLICE LUTs | 1 | 15032 |
| NO. USING O6 OUTPUT ONLY | 1 |  |
| NO. OF BONDED IOB | 3 | 190 |

**DESIGN HALF ADDER AND FULL ADDER USING VHDL**

**VHDL CODE FOR HALF ADDER**

1. **DATAFLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ha1 is

port(a,b:in std\_logic;

c,s:out std\_logic);

end ha1;

architecture Behavioral of ha1 is

begin

s<=a xor b;

c<=a and b;

end Behavioral;

**BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ha1beh is

port(a,b:in std\_logic;

c,s: out std\_logic);

end ha1beh;

architecture Behavioral of ha1beh is

begin

process(a,b)

begin

if (a='0' and b='0')then

s<='0';

c<='0';

elsif(a='0' and b='1')then

s<='1';

c<='0';

elsif(a='1' and b='0')then

s<='1';

c<='0';

else

s<='0';

c<='1';

end if;

end process;

end Behavioral;

**STRUCTURAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity andgate is

port(a,b:in std\_logic;

z:out std\_logic);

end andgate;

architecture e1 of andgate is

begin

z<=a and b;

end e1;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity xorgate is

port(a,b:in std\_logic;

z:out std\_logic);

end xorgate;

architecture e2 of xorgate is

begin

z<=a xor b;

end e2;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ha1str is

port(a,b:in std\_logic;

s,c:out std\_logic);

end ha1str;

architecture Behavioral of ha1str is

component andgate

port(a,b:in std\_logic;

z:out std\_logic);

end component;

component xorgate

port(a,b:in std\_logic;

z:out std\_logic);

end component;

begin

u1:andgate port map(a,b,c);

u2:xorgate port map(a,b,s);

end Behavioral;

**VHDL TESTBENCH CODE FOR HALF ADDER**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testha1 IS

END testha1;

ARCHITECTURE behavior OF testha1 IS

COMPONENT ha1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic;

s : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

signal s : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ha1 PORT MAP (

a => a,

b => b,

c => c,

s => s

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

wait for 100 ns;

end process;

END;

**BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testha1beh IS

END testha1beh;

ARCHITECTURE behavior OF testha1beh IS

COMPONENT ha1beh

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic;

s : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

signal s : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ha1beh PORT MAP (

a => a,

b => b,

c => c,

s => s

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

wait for 100 ns;

end process;

END;

**STRUCTURAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testha1str IS

END testha1str;

ARCHITECTURE behavior OF testha1str IS

COMPONENT ha1str

PORT(

a : IN std\_logic;

b : IN std\_logic;

s : OUT std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal s : std\_logic;

signal c : std\_logic;

BEGIN

uut: ha1str PORT MAP (

a => a,

b => b,

s => s,

c => c

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

b<='1';

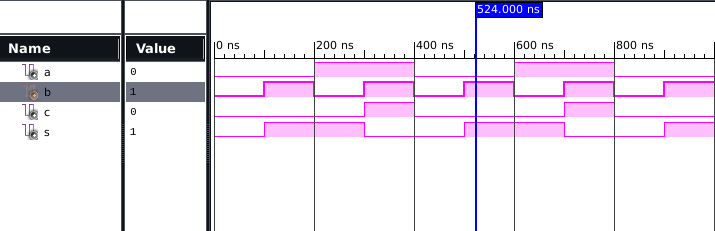
wait for 100 ns;

end process;

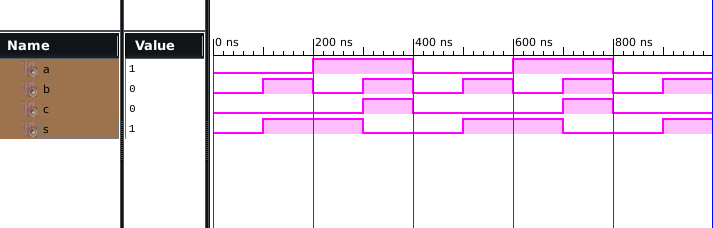
END;

**WAVEFORM**

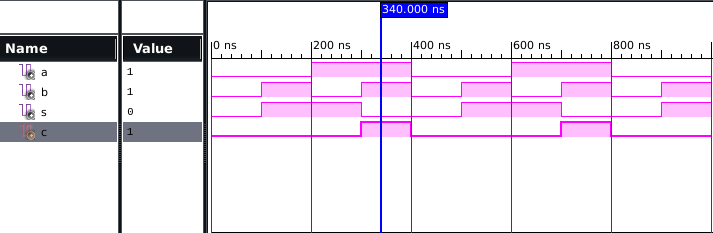
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

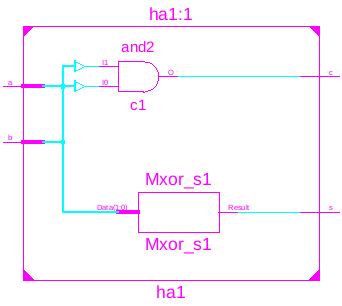
****

1. **STRUCTURAL MODELLING**

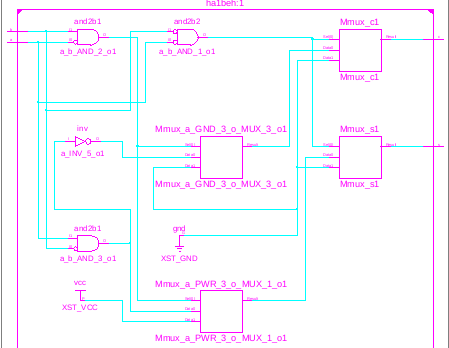
****

**RTL SCHEMATIC**

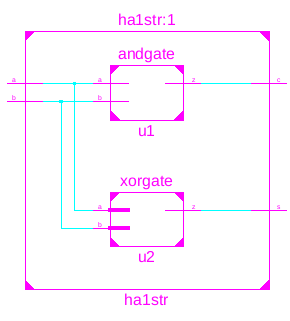
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

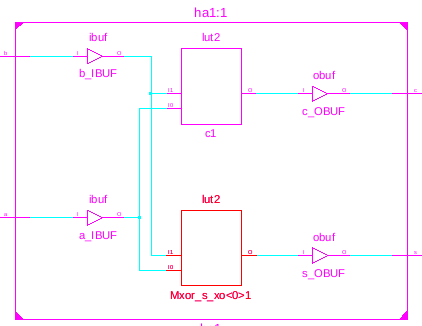
****

1. **STRUCTURAL MODELLING**

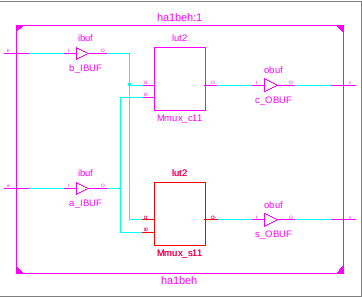
****

**TECHNOLOGY SCHEMATIC**

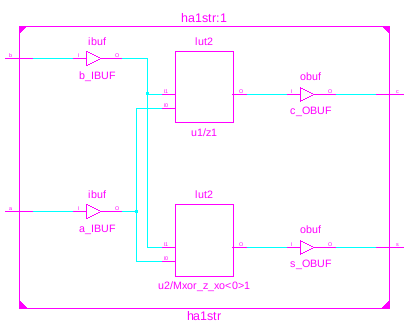
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

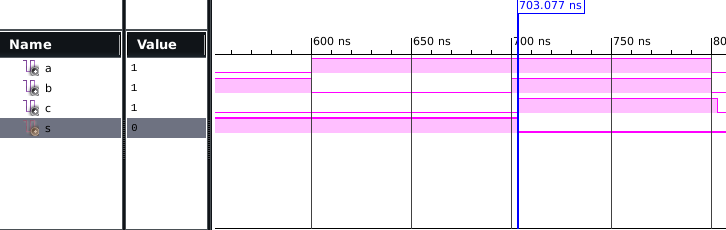
****

1. **STRUCTURAL MODELLING**

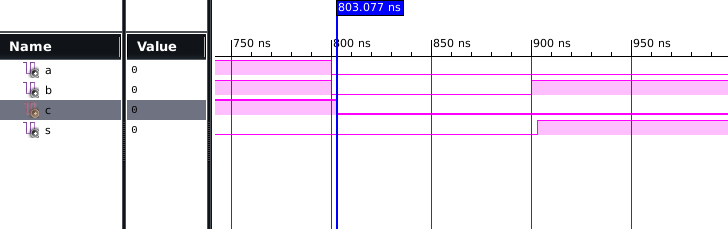
****

**POST-MAP SYNTHESIS**

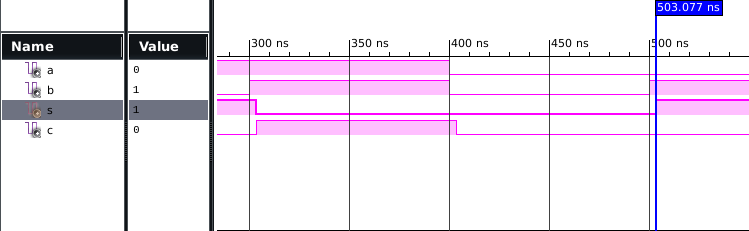
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**



1. **STRUCTURAL MODELLING**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 1 | 15032 |
| NO. OF BONDED IOB | 4 | 190 |
| AVG. FANOUT OF NON CLK NETS | 1 |  |
| NO USING O5 &O6 | 1 |  |

FREQUENCY= 1/5.266 GHZ POWER=0.029W

**VHDL CODE FOR FULL ADDER**

1. **DATAFLOW MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fa1 is

port(a,b,c:in std\_logic;

s,o: out std\_logic);

end fa1;

architecture Behavioral of fa1 is

begin

s<=(a xor b) xor c ;

o<=(a and b) or(c and(a xor b)) ;

end Behavioral;

1. **BEHAVIORAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fa1beh is

port(a,b,c:in std\_logic;

s,o: out std\_logic);

end fa1beh;

architecture Behavioral of fa1beh is

begin

process(a,b,c)

begin

if(a='0' and b='0' and c='0')then

s<='0';

o<='0';

elsif( a='0' and b='0' and c='1')then

s<='1';

o<='0';

elsif( a='0' and b='1' and c='0')then

s<='1';

o<='0';

elsif( a='0' and b='1' and c='1')then

s<='0';

o<='1';

elsif( a='1' and b='0' and c='0')then

s<='1';

o<='0';

elsif( a='1' and b='0' and c='1')then

s<='0';

o<='1';

elsif( a='1' and b='1' and c='0')then

s<='0';

o<='1';

else

s<='1';

o<='1';

end if;

end process;

end Behavioral;

1. **STRUCTURAL MODELLING**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ha is

port(a,b : in std\_logic;

s,c : out std\_logic);

end ha;

architecture e1 of ha is

begin

s<=a xor b;

c<=a and b;

end e1;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity orr is

port(x,y:in std\_logic;

z:out std\_logic);

end orr;

architecture e2 of orr is

begin

z<=x or y;

end e2;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fa1str is

port(Fa,Fb,Fc:in std\_logic;

gs,go : out std\_logic);

end fa1str;

architecture Behavioral of fa1str is

component ha is

port(a,b:in std\_logic;

s,c:out std\_logic);

end component;

component orr is

port(x,y:in std\_logic;

z:out std\_logic);

end component;

signal s0,s1,s2:std\_logic;

begin

u1: ha port map(Fa,Fb,s0,s1);

u2: ha port map(s0,Fc,gs,s2);

u3: orr port map(s2,s1,go);

end Behavioral;

**VHDL TESTBENCH CODE FOR FULL ADDER**

1. **DATAFLOW MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testfa1 IS

END testfa1;

ARCHITECTURE behavior OF testfa1 IS

COMPONENT fa1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : IN std\_logic;

s : OUT std\_logic;

o : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic := '0';

--Outputs

signal s : std\_logic;

signal o : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: fa1 PORT MAP (

a => a,

b => b,

c => c,

s => s,

o => o

);

stim\_proc: process

begin

a<='0';

b<='0';

c<='0';

wait for 100 ns;

a<='0';

b<='0';

c<='1';

wait for 100 ns;

a<='0';

b<='1';

c<='0';

wait for 100 ns;

a<='0';

b<='1';

c<='1';

wait for 100 ns;

a<='1';

b<='0';

c<='0';

wait for 100 ns;

a<='1';

b<='0';

c<='1';

wait for 100 ns;

a<='1';

b<='1';

c<='0';

wait for 100 ns;

a<='1';

b<='1';

c<='1';

wait for 100 ns;

end process;

END;

1. **BEHAVIORAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testfa1beh IS

END testfa1beh;

ARCHITECTURE behavior OF testfa1beh IS

COMPONENT fa1beh

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : IN std\_logic;

s : OUT std\_logic;

o : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal c : std\_logic := '0';

--Outputs

signal s : std\_logic;

signal o : std\_logic;

BEGIN

uut: fa1beh PORT MAP (

a => a,

b => b,

c => c,

s => s,

o => o

);

stim\_proc: process

begin

a<='0';

b<='0';

c<='0';

wait for 100 ns;

a<='0';

b<='0';

c<='1';

wait for 100 ns;

a<='0';

b<='1';

c<='0';

wait for 100 ns;

a<='0';

b<='1';

c<='1';

wait for 100 ns;

a<='1';

b<='0';

c<='0';

wait for 100 ns;

a<='1';

b<='0';

c<='1';

wait for 100 ns;

a<='1';

b<='1';

c<='0';

wait for 100 ns;

a<='1';

b<='1';

c<='1';

wait for 100 ns;

end process;

END;

**STRUCTURAL MODELLING**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testfa1str IS

END testfa1str;

ARCHITECTURE behavior OF testfa1str IS

COMPONENT fa1str

PORT(

Fa : IN std\_logic;

Fb : IN std\_logic;

Fc : IN std\_logic;

gs : OUT std\_logic;

go : OUT std\_logic

);

END COMPONENT;

signal Fa : std\_logic := '0';

signal Fb : std\_logic := '0';

signal Fc : std\_logic := '0';

--Outputs

signal gs : std\_logic;

signal go : std\_logic;

BEGIN

uut: fa1str PORT MAP (

Fa => Fa,

Fb => Fb,

Fc => Fc,

gs => gs,

go => go

);

stim\_proc: process

begin

Fa<='0';

Fb<='0';

Fc<='0';

wait for 100 ns;

Fa<='0';

Fb<='0';

Fc<='1';

wait for 100 ns;

Fa<='0';

Fb<='1';

Fc<='0';

wait for 100 ns;

Fa<='0';

Fb<='1';

Fc<='1';

wait for 100 ns;

Fa<='1';

Fb<='0';

Fc<='0';

wait for 100 ns;

Fa<='1';

Fb<='0';

Fc<='1';

wait for 100 ns;

Fa<='1';

Fb<='1';

Fc<='0';

wait for 100 ns;

Fa<='1';

Fb<='1';

Fc<='1';

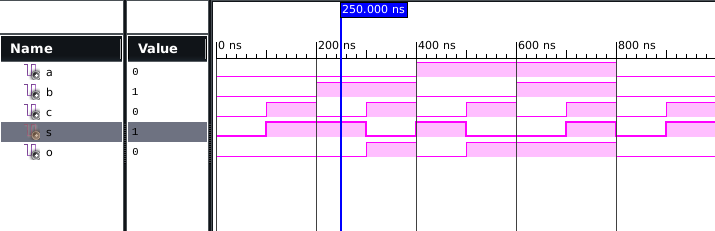
wait for 100 ns;

end process;

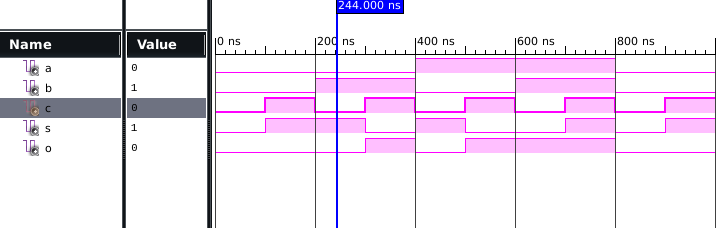
END;

**WAVEFORM**

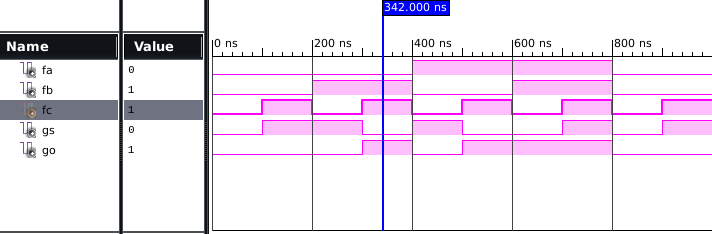
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

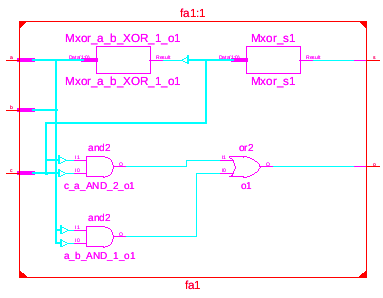
****

1. **STRUCTURAL MODELLING**

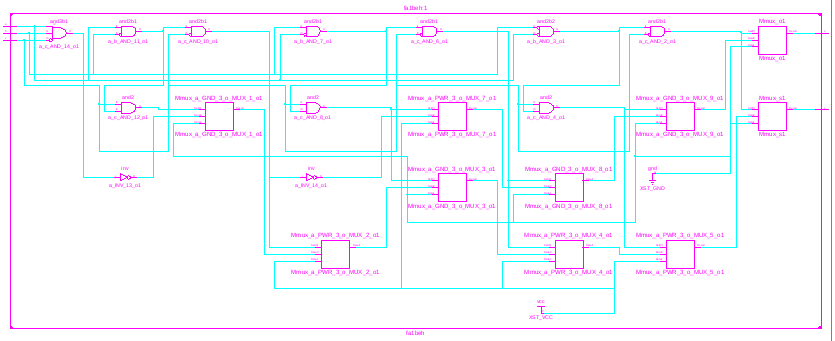
****

**RTL SCHEMATIC**

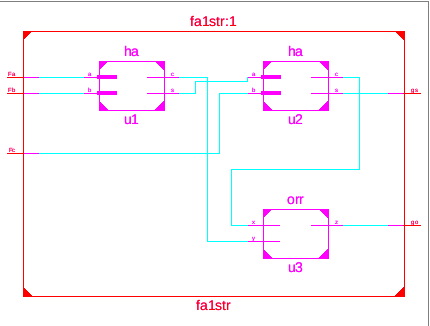
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

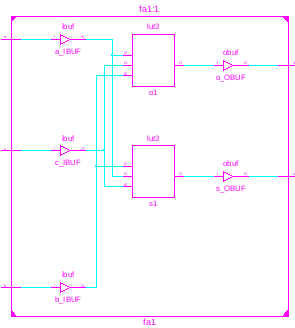
****

1. **STRUCTURAL MODELLING**

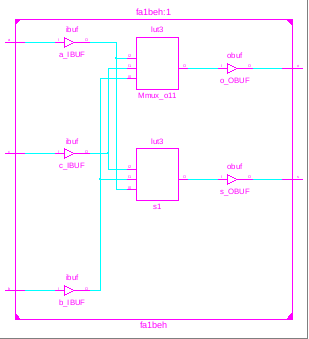
****

**TECHNOLOGY SCHEMATIC**

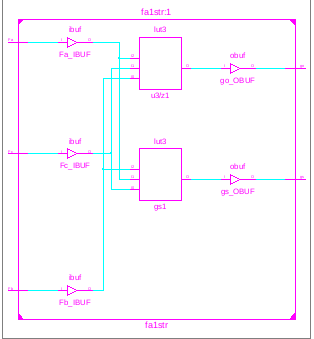
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**

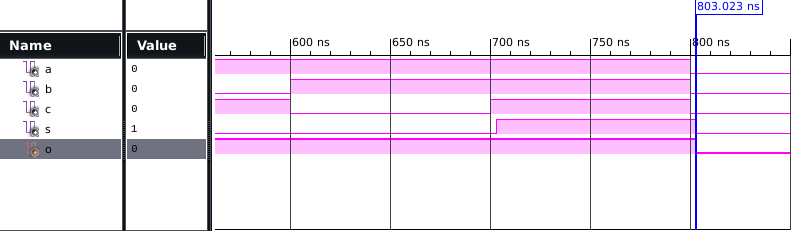
****

1. **STRUCTURAL MODELLING**

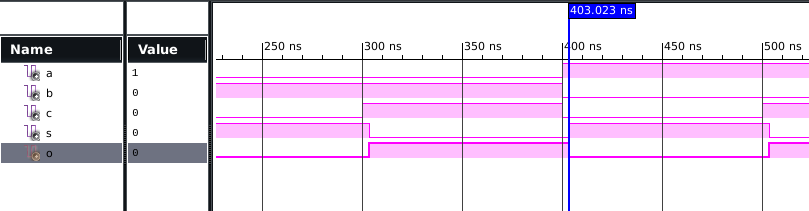
****

**POST-MAP SYNTHESIS**

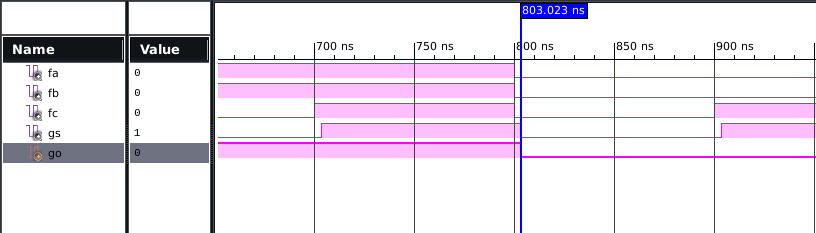
1. **DATAFLOW MODELLING**

****

1. **BEHAVIORAL MODELLING**



1. **STRUCTURAL MODELLING**

****

|  |  |  |
| --- | --- | --- |
| AREA  NO. OF SLICE LUTs | 1 | 15032 |
| NO. OF BONDED IOB | 5 | 190 |
| AVG. FANOUT OF NON CLK NETS | 1 |  |
| NO USING O5 &O6 | 1 |  |

FREQUENCY= 1/5.412 GHZ POWER=0.029W

**Decoder (behaviour):**

**code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity decoder is

port(

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

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

end decoder;

architecture bhv of decoder is

begin

process(a)

begin

case a is

when "00" => b <= "0001";

when "01" => b <= "0010";

when "10" => b <= "0100";

when others => b <= "1000";

end case;

end process;

end bhv;

**testbench:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testdec IS

END testdec;

ARCHITECTURE behavior OF testdec IS

COMPONENT decoder

PORT(

a : IN std\_logic\_vector(1 downto 0);

b : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal a : std\_logic\_vector(1 downto 0) := (others => '0');

--Outputs

signal b : std\_logic\_vector(3 downto 0);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: decoder PORT MAP (

a => a,

b => b

);

-- Stimulus process

stim\_proc: process

begin

a<="00";

wait for 50ns;

a<="01";

wait for 50ns;

a<="10";

wait for 50ns;

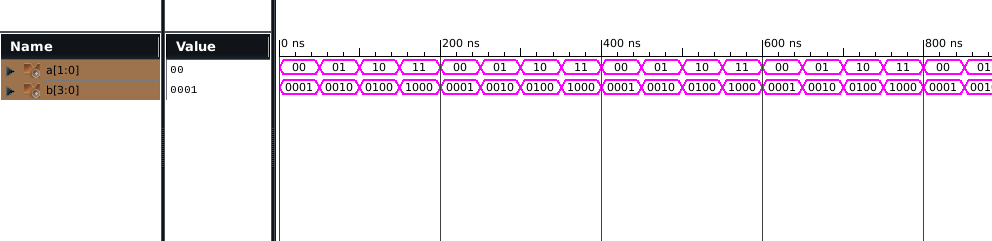
a<="11";

wait for 50ns;

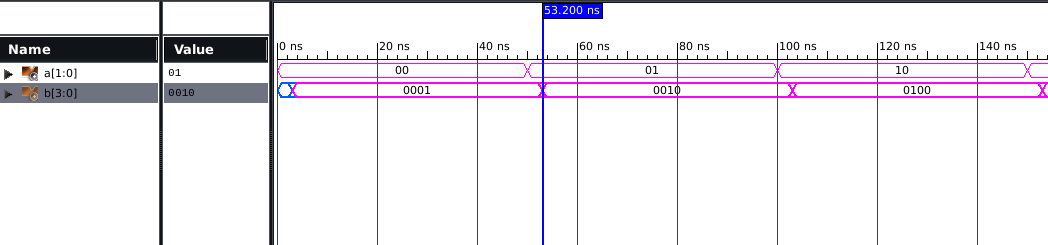
end process;

END;

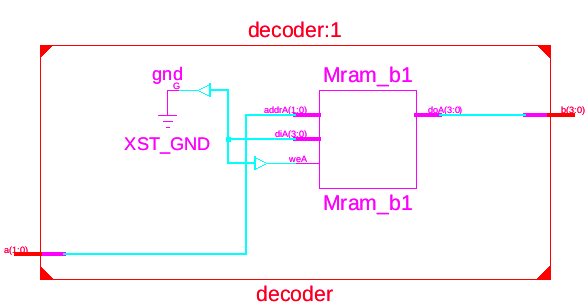
**simulation:**

****

**post map:**

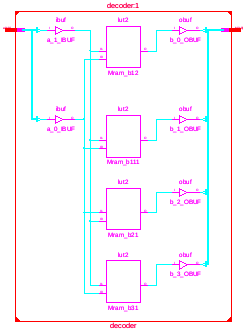
****

**RTL view:**

****

AREA

**technical view:**

****

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 4 | 15032 |
| NO. OF BONDED IOB | 6 | 190 |

FREQUENCY = 1/5.332 GHZ POWER=0.029W

**DECODER(DATAFLOW)**

**VHDL CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity dec1 is

port(a,b:in std\_logic;

c,d,e,f : out std\_logic);

end dec1;

architecture Behavioral of dec1 is

begin

c<=((not a) and (not b));

d<=((not a) and b);

e<=(a and (not b));

f<=(a and b);

end Behavioral;

**VHDL TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testdec1 IS

END testdec1;

ARCHITECTURE behavior OF testdec1 IS

COMPONENT dec1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic;

d : OUT std\_logic;

e : OUT std\_logic;

f : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

signal d : std\_logic;

signal e : std\_logic;

signal f : std\_logic;

BEGIN

uut: dec1 PORT MAP (

a => a,

b => b,

c => c,

d => d,

e => e,

f => f

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

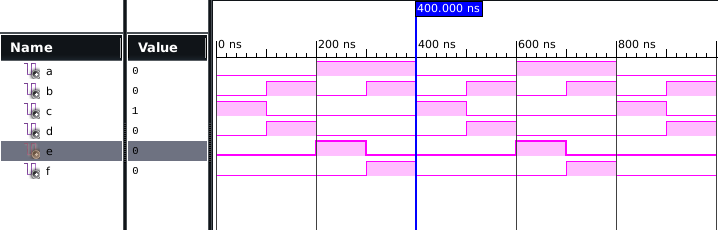
b<='1';

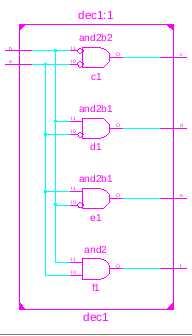
wait for 100 ns;

end process;

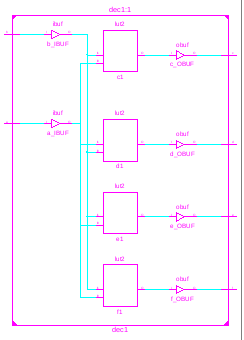
END;

**WAVEFORM**

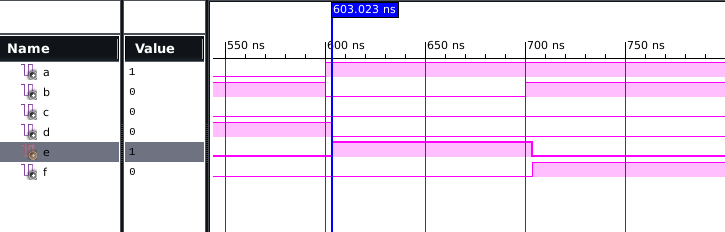
****

**RTL SCHEMATIC**

**TECHNOLOGY SCHEMATIC**

****

**POST MAP SYNTHESIS**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 2 | 15032 |
| NO. OF BONDED IOB | 6 | 190 |
| AVG. FANOUT OF NON CLK NET | 1.33 |  |

FREQUENCY = 1/5.322 GHZ POWER=0.029W

**ENCODER DATAFLOW:**

**CODE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity ENCODER\_SOURCE is

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

Y : out STD\_LOGIC\_VECTOR (1 downto 0));

end ENCODER\_SOURCE;

architecture dataflow of ENCODER\_SOURCE is

begin

Y <= "00" when I <= "0001"

else "01" when I <= "0010"

else "10" when I <= "0100"

else "11" when I <= "1000";

end dataflow;

**TEST BENCH:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testencdat IS

END testencdat;

ARCHITECTURE behavior OF testencdat IS

COMPONENT ENCODER\_SOURCE

PORT(

I : IN std\_logic\_vector(3 downto 0);

Y : OUT std\_logic\_vector(1 downto 0)

);

END COMPONENT;

--Inputs

signal I : std\_logic\_vector(3 downto 0) := (others => '0');

--Outputs

signal Y : std\_logic\_vector(1 downto 0);

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: ENCODER\_SOURCE PORT MAP (

I => I,

Y => Y );

-- Stimulus process

stim\_proc: process

begin

I<="0001";

wait for 50ns;

I<="0010";

wait for 50ns;

I<="0100";

wait for 50ns;

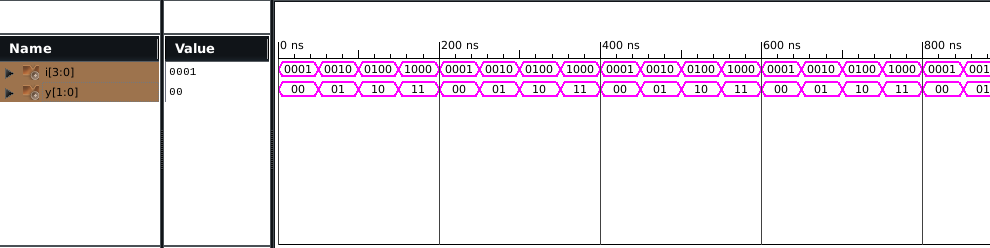
I<="1000";

wait for 50ns;

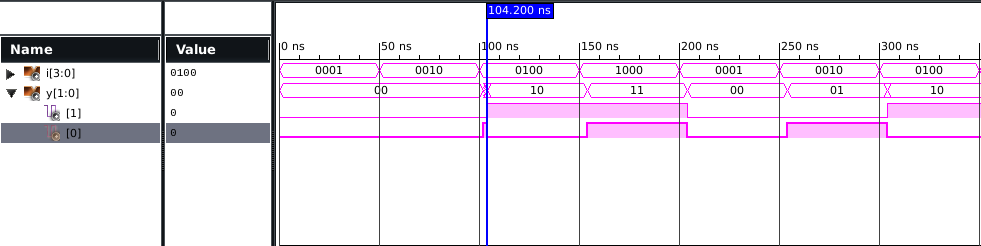
end process;

END;

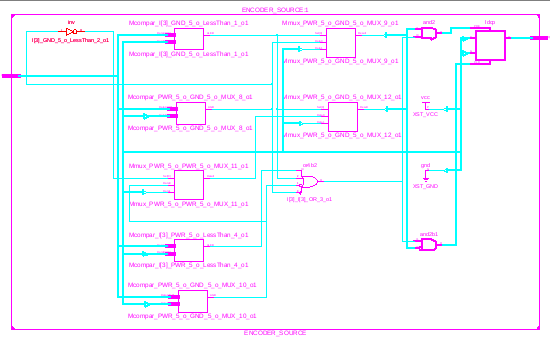
**SIMULATION:**



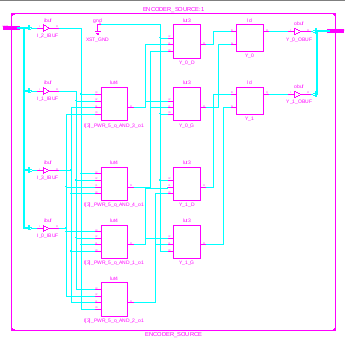
**POST MAP:**



**RTL VIEW:**



**TECHNICAL VIEW:**



AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 4 | 15032 |
| NO. OF BONDED IOB | 6 | 190 |

FREQUENCY = 1/5.332 GHZ POWER=0.029W

# **ENCODER BEHAVIOURAL**

# **CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity behave is

Port (

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

Y : out STD\_LOGIC\_VECTOR (1 downto 0));

end behave;

architecture Behavioral of behave is

begin

process (I)

begin

case I is

when "0001" => Y <= "00" ;

when "0010" => Y <= "01" ;

when "0100" => Y <= "10" ;

when others => Y <= "11" ;

end case;

end process;

end Behavioral;

**TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY test IS

END test;

ARCHITECTURE behavior OF test IS

COMPONENT behave

PORT(

I : IN std\_logic\_vector(3 downto 0);

Y : OUT std\_logic\_vector(1 downto 0)

);

END COMPONENT;

signal I : std\_logic\_vector(3 downto 0) := (others => '0');

signal Y : std\_logic\_vector(1 downto 0);

constant a\_period : time := 100 ns;

BEGIN

uut: behave PORT MAP (

I => I,

Y => Y

);

-- Stimulus process

stim\_proc: process

begin

I<="0001";

wait for a\_period;

I<="0010";

wait for a\_period;

I<="0100";

wait for a\_period;

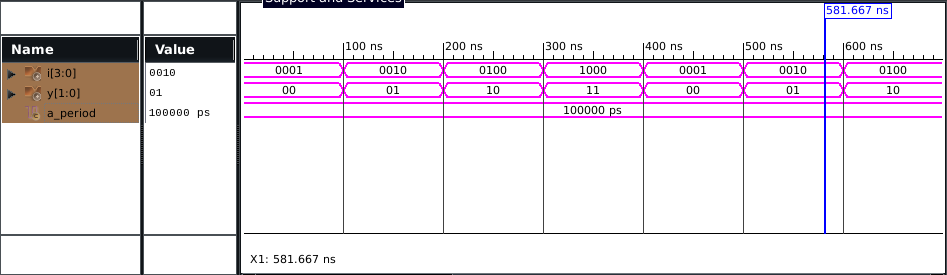
I<="1000";

wait for a\_period;

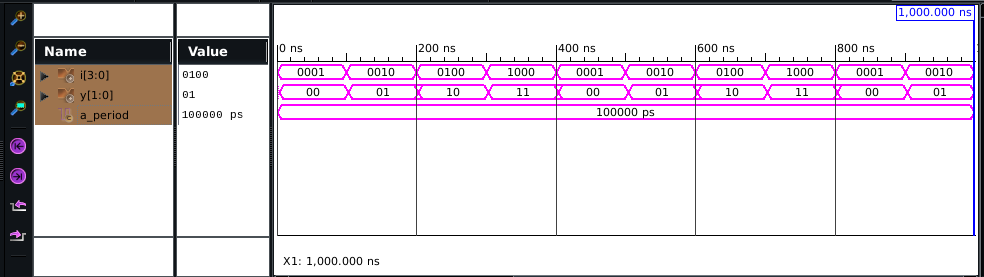
end process;

END;

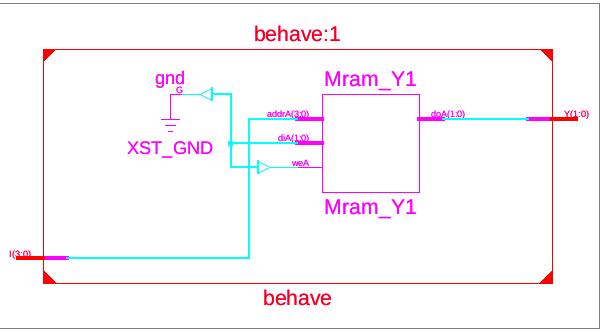
# **SIMULATION**



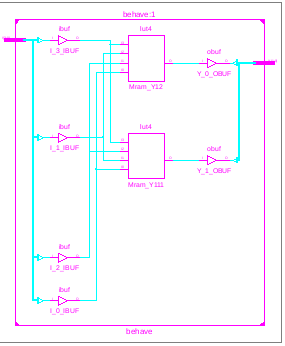
# **POST SIMULATION**



# **RTL**



**TECH VIEW**



AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 2 | 15032 |
| NO. OF BONDED IOB | 6 | 190 |

FREQUENCY = 1/5.439 GHZ POWER=0.029W

**COUNTER (BEHAVIOURAL)**

# **CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity up\_counter is

Port ( clk: in std\_logic; -- clock input

reset: in std\_logic; -- reset input

counter: out std\_logic\_vector(3 downto 0) -- output 4-bit counter

);

end up\_counter;

architecture Behavioral of up\_counter is

signal counter\_up: std\_logic\_vector(3 downto 0);

begin

process(clk,reset)

begin

if(rising\_edge(clk)) then

if(reset='1') then

counter\_up <= "0000";

else

counter\_up <= counter\_up +"1";

end if;

end if;

end process;

counter <= counter\_up;

end Behavioral;

# 

# **TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY test IS

END test;

ARCHITECTURE behavior OF test IS

COMPONENT up\_counter

PORT(

clk : IN std\_logic;

reset : IN std\_logic;

counter : OUT std\_logic\_vector(3 downto 0));

END COMPONENT;

--Inputs

signal clk : std\_logic := '0';

signal reset : std\_logic := '0';

--Outputs

signal counter : std\_logic\_vector(3 downto 0);

-- Clock period definitions

constant clk\_period : time := 20 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: up\_counter PORT MAP (

clk => clk,

reset => reset,

counter => counter

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

reset <= '1';

wait for 20 ns;

reset <= '0';

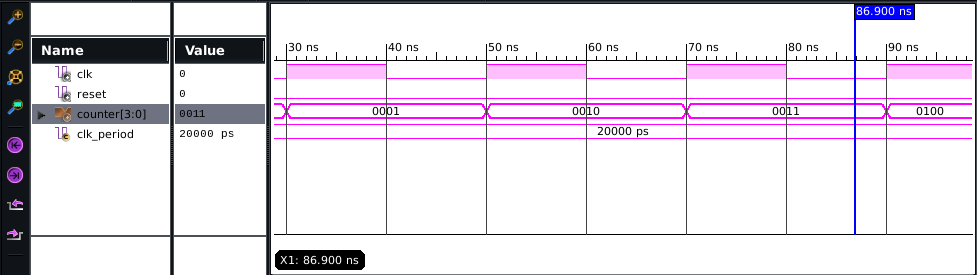
wait;

end process;

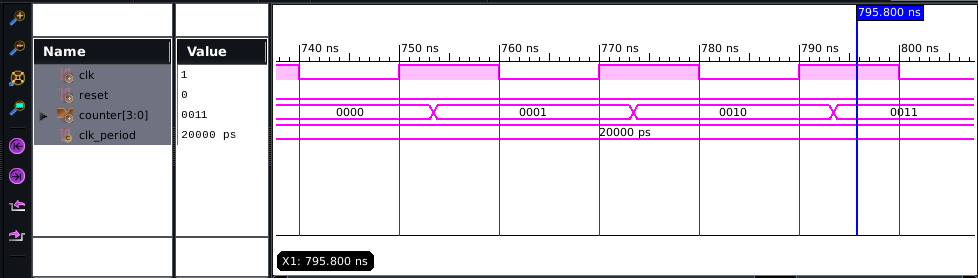
END;

# 

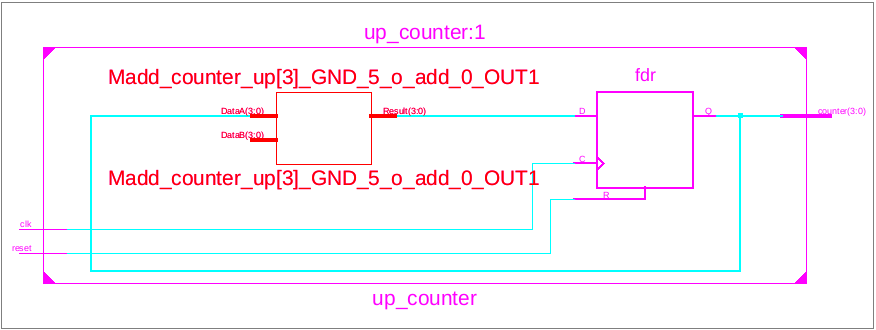
# **SIMULATION**



# **POST SIMULATION**

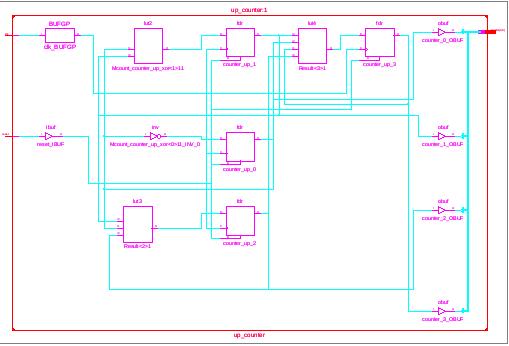


# **RTL VIEW**



# 

# **TECHNICAL VIEW**



AREA

|  |  |  |
| --- | --- | --- |
| NO.OF SLICE REGISTOR | 4 | 30064 |
| NO. OF SLICE LUTs | 4 | 15032 |
| NO. OF BONDED IOB | 6 | 190 |
| NO. OF LUT FF PAIR USED | 8 |  |

FREQUENCY = 1/3.732 GHZ POWER=0.029W

**NAND Structural**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity andgate is

port( w,q: in std\_logic;

d: out std\_logic);

end andgate;

architecture e1 of andgate is

begin

d<=w and q;

end e1;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity notgate is

port( f: in std\_logic;

y: out std\_logic);

end notgate;

architecture e2 of notgate is

begin

y<=not f;

end e2;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity nand1str is

port(a,b : in std\_logic;

c : out std\_logic);

end nand1str;

architecture Behavioral of nand1str is

component andgate is

port(w,q:in std\_logic;

d:out std\_logic);

end component;

component notgate is

port(f:in std\_logic;

y:out std\_logic);

end component;

signal s: std\_logic;

begin

a1 : andgate port map(a,b,s);

n1 : notgate port map(s,c);

end Behavioral;

**TESTBENCH CODE OF NAND GATE**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testnand1str IS

END testnand1str;

ARCHITECTURE behavior OF testnand1str IS

COMPONENT nand1str

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: nand1str PORT MAP (

a => a,

b => b,

c => c

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

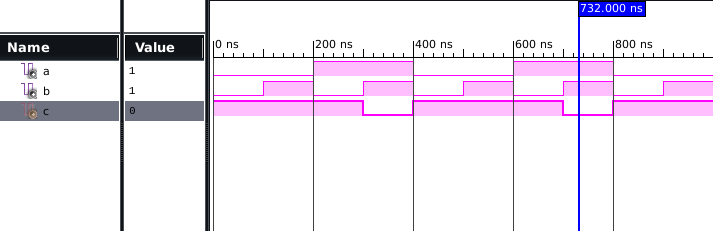
a<='1';

b<='1';

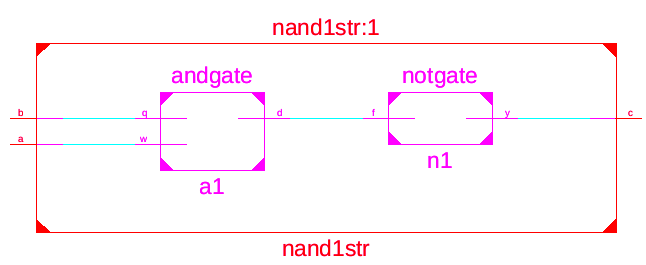
wait for 100 ns;

end process;

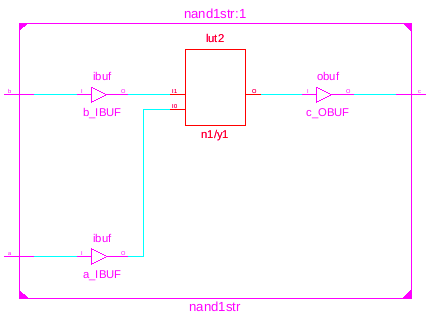
END;

**WAVEFORM FOR NAND  
**

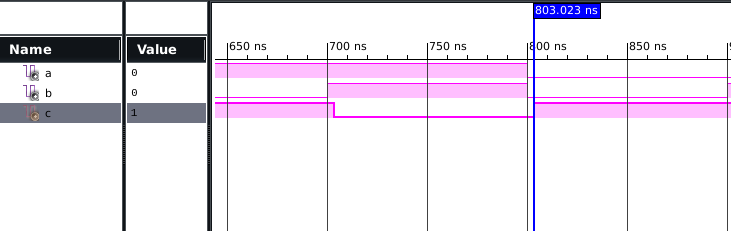
**RTL SCHEMATIC OF NAND GATE**

****

**TECHNOLOGY SCHEMATIC**

****

**POST MAP SYNTHESIS**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 1 | 15032 |
| NO. OF BONDED IOB | 3 | 190 |
| NO. USING O6 OUTPUT ONLY | 1 |  |

FREQUENCY = 1/5.229 POWER=0.029W

**COMPARATOR STRUCTURAL**

**VHDL CODE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity andgate is

port(q,w :in std\_logic;

j: out std\_logic);

end andgate;

architecture e1 of andgate is

begin

j<=q and w;

end e1;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity notgate is

port(r:in std\_logic;

t: out std\_logic );

end notgate;

architecture e2 of notgate is

begin

t<=not r;

end e2;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity xorgate is

port(y,u: in std\_logic;

i: out std\_logic );

end xorgate;

architecture e3 of xorgate is

begin

i<=y xor u;

end e3;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity comp1 is

port(a,b: in std\_logic;

c,d,e: out std\_logic );

end comp1;

architecture Behavioral of comp1 is

component andgate is

port(q,w : in std\_logic;

j : out std\_logic);

end component;

component notgate is

port(r : in std\_logic;

t : out std\_logic);

end component;

component xorgate is

port(y,u : in std\_logic;

i : out std\_logic);

end component;

signal z,m,v:std\_logic;

begin

u1:notgate port map(b,z);

u2:andgate port map(a,z,c);

u3:notgate port map(a,m);

u4:andgate port map(b,m,d);

u5:xorgate port map(a,b,v);

u6:notgate port map(v,e);

end Behavioral;

**VHDL TESTBENCH**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY testcomp1str IS

END testcomp1str;

ARCHITECTURE behavior OF testcomp1str IS

COMPONENT comp1

PORT(

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic;

d : OUT std\_logic;

e : OUT std\_logic

);

END COMPONENT;

--Inputs

signal a : std\_logic := '0';

signal b : std\_logic := '0';

--Outputs

signal c : std\_logic;

signal d : std\_logic;

signal e : std\_logic;

BEGIN

uut: comp1 PORT MAP (

a => a,

b => b,

c => c,

d => d,

e => e

);

stim\_proc: process

begin

a<='0';

b<='0';

wait for 100 ns;

a<='0';

b<='1';

wait for 100 ns;

a<='1';

b<='0';

wait for 100 ns;

a<='1';

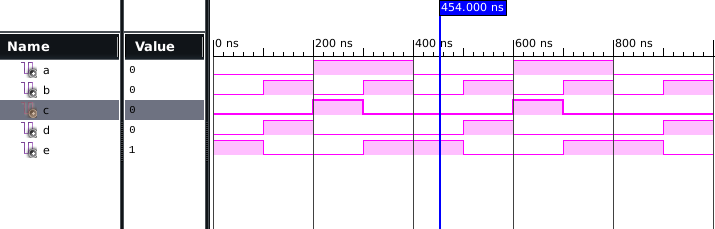
b<='1';

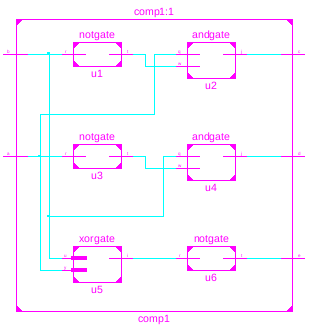
wait for 100 ns;

end process;

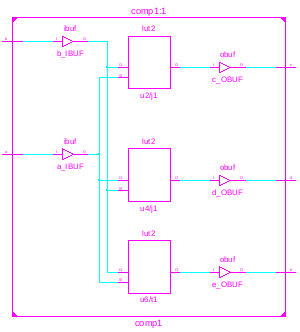
END;

**WAVEFORM**

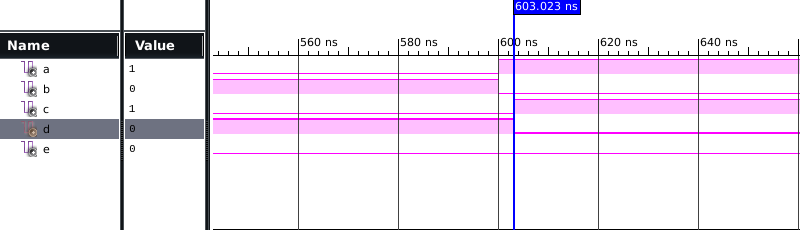
****

**RTL SCHEMATIC**

**TECHNOLOGY SCHEMATIC**

****

**POST MAP SYNTHESIS**

****

AREA

|  |  |  |
| --- | --- | --- |
| NO. OF SLICE LUTs | 2 | 150323 |
| NO. OF BONDED IOB | 5 | 190 |

FREQUENCY = 1/5.3 GHZ POWER=0.029W