**ASSIGNMENT- 3 VLSI DESIGN ROLLNO. 23M1064**

Question- Describe a 32 bit Brent Kung adder in VHDL and simulate it using a test bench. Your description should use std\_logic types for various signals. You can use the public domain tool “ghdl” for VHDL simulation.

Q–1 Describe a 32 bit Brent Kung adder in VHDL and simulate it using a test bench. Your description should use std\_logic types for various signals. You can use the public domain tool “ghdl” for VHDL simulation.

Solution- VHDL code for Brentkung 32bit adder is

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all ;

entity brentkung32bit123 is

port(a,b : in std\_logic\_vector(31 downto 0);

cin : in std\_logic;

sum: out std\_logic\_vector(31 downto 0);

cout : out std\_logic);

end brentkung32bit123;

architecture brent\_kung\_arch of brentkung32bit123 is

signal C : std\_logic\_vector(32 downto 0);

signal temp1, temp2 : std\_logic;

signal P1,G1 : std\_logic\_vector(31 downto 0);

signal P2,g2 : std\_logic\_vector(15 downto 0);

signal p3,G3 : std\_logic\_vector(7 downto 0);

signal p4,g4 : std\_logic\_vector(3 downto 0);

signal p5,g5 : std\_logic\_vector(1 downto 0);

signal P6,G6 : std\_logic;

-- Component institation

component and\_gate

port(a,b: in std\_logic;

y: out std\_logic );

end component;

component xor\_gate

port(a,b: in std\_logic;

y: out std\_logic );

end component;

component or\_gate

port(a,b: in std\_logic;

y: out std\_logic );

end component;

component logic

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

y: out std\_logic );

end component;

component Cin\_map\_G

port(A, B, Cin: in std\_ulogic;

Bit0\_G: out std\_ulogic);

end component;

begin

-- first order generate and propogate

C(0) <= cin;

order1\_propogate : for i in 0 to 31 generate

propogate1 : xor\_gate portmap(a(i),b(i),P1(i));

end generate;

order1\_generate : for i in 1 to 31 generate

generate1 : and\_gate port map(a(i),b(i),G1(i));

end generate;

-- first level generation

x1 : Cin\_map\_G port map(a(0), b(0), c(0),c(1));

G1(0) <= C(1);

order2\_propogate : for i in 0 to 15 generate

propogate : and\_gate port map(P1((2\*i)+1), P1(2\*i),P2(i));

end generate;

order2\_generate : for i in 1 to 15 generate

generateo : logic port map(G1(2\*i+1),P1(2\*i+1),G1(2\*i),g2(i));

end generate;

x4 : logic port map(G1(1),P1(1),C(1),C(2));

g2(0) <= C(2);

O3\_propogate : for i in 0 to 7 generate

propogate3 : and\_gate port map(P2(2\*i+1), P2(2\*i),p3(i));

end generate;

O3\_generate : for i in 1 to 7 generate

generate3 : logic port map(g2(2\*i+1),P2(2\*i+1),g2(2\*i),G3(i));

end generate;

x5 : logic port map(g2(1),P2(1),C(2),C(4));

G3(0) <= C(4);

o4\_propogate : for i in 0 to 3 generate

propogate4 : and\_gate port map(p3(2\*i+1), p3(2\*i),p4(i));

end generate;

o4\_generate : for i in 1 to 3 generate

generate4 : logic port map(G3(2\*i+1),p3(2\*i+1),G3(2\*i),g4(i));

end generate;

x6 : logic port map(G3(1),p3(1),C(4),C(8));

g4(0) <= C(8);

O5\_propogate : for i in 0 to 1 generate

propogate5 : and\_gate port map(p4(2\*i+1), p4(2\*i),p5(i));

end generate;

x7: logic port map(g4(3),p4(3),g4(2),g5(1));

x8 :logic port map(g4(1),p4(1),C(8),C(16));

g5(0) <= C(16);

x9 : and\_gate port map(p5(1),p5(0),P6);

x10 : logic port map(g5(1),p5(1),C(16),C(32));

G6 <= C(32);

cout <= C(32);

--computation of second and other level carries

c3 : logic port map(G1(2),P1(2),C(2),C(3));

c5 : logic port map(G1(4),P1(4),C(4),C(5));

c6 : logic port map(g2(2),P2(2),C(4),C(6));

c7 : logic port map(G1(6),P1(6),C(6),C(7));

c9 : logic port map(G1(8),P1(8),C(8),C(9));

c10 : logic port map(g2(4),P2(4),C(8),C(10));

c11 : logic port map(G1(10),P1(10),C(10),C(11));

c12 : logic port map(G3(2),p3(2),C(8),C(12));

c13 : logic port map(G1(12),P1(12),C(12),C(13));

c14 : logic port map(g2(6),P2(6),C(12),C(14));

c15 : logic port map(G1(14),P1(14),C(14),C(15));

c17 : logic port map(G1(16),P1(16),C(16),C(17));

c18 : logic port map(g2(8),P2(8),C(16),C(18));

c19 : logic port map(G1(18),P1(18),C(18),C(19));

c20 : logic port map(G3(4),p3(4),C(16),C(20));

c21 : logic port map(G1(20),P1(20),C(20),C(21));

c22 : logic port map(g2(10),P2(10),C(20),C(22));

c23 : logic port map(G1(22),P1(22),C(22),C(23));

c24 : logic port map(g4(2),p4(2),C(16),C(24));

c25 : logic port map(G1(24),P1(24),C(24),C(25));

c26 : logic port map(g2(12),P2(12),C(24),C(26));

c27 : logic port map(G1(26),P1(26),C(26),C(27));

c28 : logic port map(G3(6),p3(6),C(24),C(28));

c29 : logic port map(G1(28),P1(28),C(28),C(29));

c30 : logic port map(g2(14),P2(14),C(28),C(30));

c31 : logic port map(G1(30),P1(30),C(30),C(31));

sum32bit : for i in 0 to 31 generate

x111 : xor\_gate port map(P1(i),C(i),Sum(i));

end generate;

end brent\_kung\_arch;

after writing code testing the final adder with a test bench which reads pairs of 16 bit words and a single bit input carry value from a file, applies these to the adder and compares the result with the expected 16 bit sum and 1 bit carry values stored in the same file.

Delay of 16 bits is 18960 ps which is showing in picture I m attaching with the file

