VHDL Source Codes and VHDL Testbench Codes

Based on the given information, design a 1-bit RCA using Behavioral Model.

VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;      -- Using STD_LOGIC_1164 package

entity onebitrca is
port (a : in STD_LOGIC;                  -- a is used as input port
        b : in STD_LOGIC;                -- b is used as input port
        Carryin : in STD_LOGIC;          -- Carryin is used as input port
        Sum : out STD_LOGIC;             -- Sum is used as output port
        Carryout : out STD_LOGIC);       -- Carryout is used as output port
end onebitrca;

architecture Behavioral of onebitrca is
begin
Sum <= a xor b xor Carryin;                                          -- Sum Equation
Carryout <= (a and b) or (Carryin and b) or (Carryin and a);     -- Carryout equation
end Behavioral;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                 -- Using STD_LOGIC_1164 package

ENTITY onebitrca_tb IS
END onebitrca_tb;

ARCHITECTURE behavior OF onebirrca_tb IS

 COMPONENT onebitrca                     -- Component declaration
 PORT(
       a : IN  std_logic;                -- a is declared as input port
       b : IN  std_logic;                -- b is declared as input port
       Carryin : IN  std_logic;          -- Carryin is declared as input port
       Sum : OUT  std_logic;             -- Sum is declared as output port
       Carryout : OUT  std_logic         -- Carryput is declared as output port
       );
   END COMPONENT

              -- Signal declaration

  signal a : std_logic := '0';
  signal b : std_logic := '0';
  signal Carryin : std_logic := '0';
  signal Sum : std_logic;
  signal Carryout : std_logic;

BEGIN
 -- Instantiate the Unit Under Test (UUT)
  uut: onebitrca PORT MAP (
        a => a,
        b => b,
        Carryin => Carryin,
        Sum => Sum,
        Carryout => Carryout
      );
   --Process name :   stim_proc
   -- This process tests the implementation by possible input values
  stim_proc: process
  begin
    wait for 100 ns;
     a <= '0';
     b <= '0';
     Carryin <= '0';
     wait for 10 ns;

     a <= '0';
     b <= '0';
    Carryin <= '1';
    wait for 10 ns;

    a <= '0';
    b <= '1';
    Carryin <= '0';
    wait for 10 ns;

    a <= '0';
    b <= '1';
    Carryin <= '1';
    wait for 10 ns;

    a <= '1';
    b <= '0';
    Carryin <= '0';
    wait for 10 ns;

    a <= '1';
    b <= '0';
    Carryin <= '1';
    wait for 10 ns;

    a <= '1';
    b <= '1';
```

```
     Carryin <= '0';
    wait for 10 ns;

    a <= '1';
    b <= '1';
    Carryin <= '1';
    wait for 10 ns;

    end process;
    END;
```

Design a 4-bit RCA using Structural Model based on your 1-bit RCA.

VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;                    -- Using STD_LOGIC_LIBRARY_1164 package
        entity four_bit_rca is
port( A:IN STD_LOGIC_VECTOR(3 downto 0);    -- Declaring A as four bit input port
      B:IN STD_LOGIC_VECTOR(3 downto 0);    -- Declaring B as four bit input port
      Cin:IN STD_LOGIC;                     -- Declaring Cin as input port
      S:out STD_LOGIC_VECTOR(3 downto 0);  -- Declaring S as four bit output port
      Carryout:out std_logic                -- Declaring Carryout as output port
      );
end fourbitrca;
        architecture Structural of four_bit_rca is        -- architecture declaration
 --Component declaration
component fa is
port(a,b,c:in std_logic;                    -- Declaring a, b and c as inputs
     Cout:out std_logic;                    --  Declaring Cout as output
     s:out std_logic                        --  Declaring s as output
);
end component;
 signal carry : STD_LOGIC_VECTOR (3 downto 0);
 begin
 --Port mapping
fa0:fa port map(a=>A(0), b=>B(0), c=>Cin, Cout=>carry(0), s=>S(0));
fa1:fa port map(a=>A(1), b=>B(1), c=>carry(0), Cout=>carry(1), s=>S(1));
fa2:fa port map(a=>A(2), b=>B(2), c=>carry(1), Cout=>carry(2), s=>S(2));
fa3:fa port map(a=>A(3), b=>B(3), c=>carry(2), Cout=>Carryout, s=>S(3));
end Structural;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;              -- Using STD_LOGIC_LIBRARY_1164 package
entity fa is
port( a,b,c:in std_logic;                 --  Declaring a, b and c as inputs
    Cout:out std_logic;                   --  Declaring Cout as output
            s:out std_logic               --  Declaring s as output
         );
```

```
end fa;
architecture fa_architecture of fa is          -- architecture declaration
begin
s<=a xor b xor c;                              -- Sum result expression
Cout<=(a and b) or (c and a) or (c and b);     -- Cout expression
end fa_architecture;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                    -- Using STD_LOGIC_LIBRARY_1164 package
ENTITY four_bit_rca_tb IS                       -- fourbit rca Entity declaration
END four_bit_rca_tb;
ARCHITECTURE behavior OF four_bit_rca_tb IS            -- architecture declaration
  -- Component Declaration for the Unit Under Test (UUT)
 COMPONENT fourbitrca
 PORT(A : IN  std_logic_vector(3 downto 0);            -- A is used as input port
         B : IN  std_logic_vector(3 downto 0);         -- B is used as input port
         Cin : IN  std_logic;                          -- Cin is used as input port
          S : OUT  std_logic_vector(3 downto 0);       -- S is used as output port
          Carryout : OUT  std_logic                    -- Carryout is used as output port
    );
  END COMPONENT;
 --Inputs
  signal A : std_logic_vector(3 downto 0) := (others => '0');
  signal B : std_logic_vector(3 downto 0) := (others => '0');
  signal Cin : std_logic := '0';
        --Outputs
  signal S : std_logic_vector(3 downto 0);
  signal Carryout : std_logic;
  BEGIN
  uut: fourbitrca PORT MAP (
      A => A,
      B => B,
      Cin => Cin,
      S => S,
      Carryout => Carryout
    );
  stim_proc_A: process
  begin
  A<="0000";
  wait for 100 ns;
  A<="0001";
  wait;
  end process;
  stim_proc_B: process
  begin
  B<="1110";
```

```
   wait for 100 ns;
   B<="0010";
    wait;
   end process;
   stim_proc_Cin: process
   begin
   Cin<='0';
    wait for 100 ns;
   Cin<='1';
   wait;
   end process;
   END;
```

Design a 32-bit RCA using Structural model based on your 4-bit RCA

VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;                        -- Using STD_LOGIC_LIBRARY_1164 package

USE ieee.std_logic_arith.ALL;                      -- Using STD_LOGIC_LIBRARY_arith package

entity thirtytwo_bit_rca is                         -- Entity declaration

port(A:IN STD_LOGIC_VECTOR(31 downto 0);        -- Declaring A as 32 bit input port
B:IN STD_LOGIC_VECTOR(31 downto 0);              -- Declaring B as 32 bit input port
Cin:IN STD_LOGIC;                                    -- Declaring Cin as input port
S:OUT STD_LOGIC_VECTOR(31 downto 0);            -- Declaring S as 32 bit output port
Carryout:OUT std_logic                              -- Declaring Carryout as output port
);
end thirtytwo_bit_rca;

architecture Structural of thirtytwo_bit_rca is
--Component Declaration
component fa is
port(a,b,c:in std_logic;                           -- a,b and c is used as input port
Cout:out std_logic;                                -- Cout is used as output port
s:out std_logic                                    -- s is used as output port
);
end component;
signal carry : STD_LOGIC_VECTOR (30 downto 0);
begin
--Port mapping
fa0:fa port map(a=>A(0), b=>B(0), c=>Cin, Cout=>carry(0), s=>S(0));
fa1:fa port map(a=>A(1), b=>B(1), c=>carry(0), Cout=>carry(1), s=>S(1));
fa2:fa port map(a=>A(2), b=>B(2), c=>carry(1), Cout=>carry(2), s=>S(2));
fa3:fa port map(a=>A(3), b=>B(3), c=>carry(2), Cout=>Carry(3), s=>S(3));
fa4:fa port map(a=>A(4), b=>B(4), c=>carry(3), Cout=>Carry(4), s=>S(4));
fa5:fa port map(a=>A(5), b=>B(5), c=>carry(4), Cout=>Carry(5), s=>S(5));
```

```
fa6:fa port map(a=>A(6), b=>B(6), c=>carry(5), Cout=>Carry(6), s=>S(6));
fa7:fa port map(a=>A(7), b=>B(7), c=>carry(6), Cout=>Carry(7), s=>S(7));
fa8:fa port map(a=>A(8), b=>B(8), c=>carry(7), Cout=>Carry(8), s=>S(8));
fa9:fa port map(a=>A(9), b=>B(9), c=>carry(8), Cout=>Carry(9), s=>S(9));
fa10:fa port map(a=>A(10), b=>B(10), c=>carry(9), Cout=>Carry(10), s=>S(10));
fa11:fa port map(a=>A(11), b=>B(11), c=>carry(10), Cout=>Carry(11), s=>S(11));
fa12:fa port map(a=>A(12), b=>B(12), c=>carry(11), Cout=>Carry(12), s=>S(12));
fa13:fa port map(a=>A(13), b=>B(13), c=>carry(12), Cout=>Carry(13), s=>S(13));
fa14:fa port map(a=>A(14), b=>B(14), c=>carry(13), Cout=>Carry(14), s=>S(14));
fa15:fa port map(a=>A(15), b=>B(15), c=>carry(14), Cout=>Carry(15), s=>S(15));
fa16:fa port map(a=>A(16), b=>B(16), c=>carry(15), Cout=>Carry(16), s=>S(16));
fa17:fa port map(a=>A(17), b=>B(17), c=>carry(16), Cout=>Carry(17), s=>S(17));
fa18:fa port map(a=>A(18), b=>B(18), c=>carry(17), Cout=>Carry(18), s=>S(18));
fa19:fa port map(a=>A(19), b=>B(19), c=>carry(18), Cout=>Carry(19), s=>S(19));
fa20:fa port map(a=>A(20), b=>B(20), c=>carry(19), Cout=>Carry(20), s=>S(20));
fa21:fa port map(a=>A(21), b=>B(21), c=>carry(20), Cout=>Carry(21), s=>S(21));
fa22:fa port map(a=>A(22), b=>B(22), c=>carry(21), Cout=>Carry(22), s=>S(22));
fa23:fa port map(a=>A(23), b=>B(23), c=>carry(22), Cout=>Carry(23), s=>S(23));
fa24:fa port map(a=>A(24), b=>B(24), c=>carry(23), Cout=>Carry(24), s=>S(24));
fa25:fa port map(a=>A(25), b=>B(25), c=>carry(24), Cout=>Carry(25), s=>S(25));
fa26:fa port map(a=>A(26), b=>B(26), c=>carry(25), Cout=>Carry(26), s=>S(26));
fa27:fa port map(a=>A(27), b=>B(27), c=>carry(26), Cout=>Carry(27), s=>S(27));
fa28:fa port map(a=>A(28), b=>B(28), c=>carry(27), Cout=>Carry(28), s=>S(28));
fa29:fa port map(a=>A(29), b=>B(29), c=>carry(28), Cout=>Carry(29), s=>S(29));
fa30:fa port map(a=>A(30), b=>B(30), c=>carry(29), Cout=>Carry(30), s=>S(30));
fa31:fa port map(a=>A(31), b=>B(31), c=>carry(30), Cout=>Carryout, s=>S(31));
end Structural;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;                      -- Using STD_LOGIC_LIBRARY_1164 package
--Entity Declaration
entity fa is
port(a,b,c:in std_logic;                          -- a,b and c are used as inputs
     Cout:out std_logic;                          --  Cout used as output
     s s:out std_logic                            -- s used as output
         );
end fa;
architecture fa_architecture of fa is             -- architecture declaration
begin
s<=a xor b xor c;                          -- Sum output equation
Cout<=(a and b) or (c and a) or (c and b);   -- Cout output equation
end fa_architecture;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                        -- Using STD_LOGIC_LIBRARY_1164 package

USE ieee.std_logic_arith.ALL;                         -- Using STD_LOGIC_LIBRARY_arith package
```

```
--Entity Declaration
ENTITY thirtytwo_bit_rca_tb IS
END thirtytwo_bit_rca_tb;

ARCHITECTURE behavior OF thirtytwo_bit_rca_tb IS                    -- architecture declaration
 -- Component Declaration for the Unit Under Test (UUT)
 COMPONENT thirtytwo_bit_rca
 PORT( A : IN  std_logic_vector(31 downto 0);                     -- A is used as input port
          B : IN  std_logic_vector(31 downto 0);                   -- B is used as input port
          Cin : IN  std_logic;                                      -- Cin is used as input port
          S : OUT  std_logic_vector(31 downto 0);                  -- S is used as output port
          Carryout : OUT  std_logic                                 -- Carryout is used as input port
          );
   END COMPONENT;
   --Inputs
  signal A : std_logic_vector(31 downto 0) := (others => '0');
  signal B : std_logic_vector(31 downto 0) := (others => '0');
  signal Cin : std_logic := '0';

 --Outputs
  signal S : std_logic_vector(31 downto 0);
  signal Carryout : std_logic;


BEGIN

-- Instantiate the Unit Under Test (UUT)
  uut: thirtytwo_bit_rca PORT MAP (
        A => A,
        B => B,
        Cin => Cin,
        S => S,
        Carryout => Carryout
      );
stim_proc: process
  begin
          Cin <= '0';
          wait for 10 ns;
          A <= "00000000000000000000000000000000";
          wait for 10 ns;
          B <= "00000000000000000000000000000000";
          wait for 10 ns;
          Cin <= '1';
          wait for 10 ns;
          A <= "00000000000000000000000000000010";
          wait for 10 ns;
          B <= "00000000000000000000000000001000";
          wait for 10 ns;
          A <= "11000000000000000000000000000010";
          wait for 10 ns;
          B <= "11000000000000000000000000001000";
          wait for 10 ns;
          A <= x"0AF70000";
          wait for 10 ns;
          B <= x"23640000";
          wait for 10 ns;
          end process;
          END;
```

Design a 4-bit CLA based on the information given.

<u>VHDL code:</u>

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                          -- Using STD_LOGIC_LIBRARY_1164 package

-- entity declaration
ENTITY four_bit_cla IS
  PORT
    (a     :  IN   STD_LOGIC_VECTOR(3 DOWNTO 0);          -- a is used as input port
     b     :  IN   STD_LOGIC_VECTOR(3 DOWNTO 0);          -- b is used as input port
     carry_in  :  IN   STD_LOGIC;                          -- carry_in is used as input port
     sum       :  OUT  STD_LOGIC_VECTOR(3 DOWNTO 0);      -- sum is used as input port
     carry_out :  OUT  STD_LOGIC                           -- carry_out is used as output port
    );
END four_bit_cla ;

ARCHITECTURE behavioral OF four_bit_cla  IS                  -- architecture declaration
--Signal declaration
SIGNAL   sum_new   :   STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL   g         :   STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL   p         :   STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL   carry     :   STD_LOGIC_VECTOR(3 DOWNTO 1);
BEGIN
  sum_new <= a XOR b;    -- sum_new signal equation
  g <= a AND b;          -- generate signal equation
  p <= a OR b;           -- propagate signal equation
  PROCESS (g,p,carry)
  BEGIN
  carry(1) <= g(0) OR (p(0) AND carry_in);
    inst: FOR i IN 1 TO 2 LOOP                          -- for loop
       carry(i+1) <= g(i) OR (p(i) AND carry(i));
       END LOOP;
  carry_out <= g(3) OR (p(3) AND carry(3));
  END PROCESS;
```

```
   sum(0) <= sum_new(0) XOR carry_in;
   sum(3 DOWNTO 1) <= sum_new(3 DOWNTO 1) XOR carry(3 DOWNTO 1);
END behavioral;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                      -- Using STD_LOGIC_LIBRARY_1164 package
--Entity Declaration
ENTITY four_bit_cla_tb IS
END four_bit_cla_tb;

ARCHITECTURE behavior OF four_bit_cla_tb IS
  -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT four_bit_cla
   PORT(
       a : IN  std_logic_vector(3 downto 0);              -- a is used as input
       b : IN  std_logic_vector(3 downto 0);              -- b is used as input
       carry_in : IN  std_logic;                          -- carry_in is used as input
       sum : OUT  std_logic_vector(3 downto 0);           -- sum is used as output
       carry_out : OUT  std_logic                         -- carry_out is used as output
      );
   END COMPONENT;


  --Input signals
  signal a : std_logic_vector(3 downto 0) := (others => '0');
  signal b : std_logic_vector(3 downto 0) := (others => '0');
  signal carry_in : std_logic := '0';
 --Output signals
  signal sum : std_logic_vector(3 downto 0);
  signal carry_out : std_logic;
  BEGIN
          -- Instantiate the Unit Under Test (UUT)
  uut: four_bit_cla PORT MAP (
        a => a,
        b => b,
        carry_in => carry_in,
        sum => sum,
        carry_out => carry_out
      );
  -- Stimulus process
  stim_proc: process
  begin
  carry_in <= '0';
  a <= "0001";
          wait for 10 ns;
          b <= "0001";
          wait for 10 ns;
          carry_in <= '1';
          wait for 10 ns;
          a <= "1000";
          wait for 10 ns;
          b <= "1000";
          wait for 10 ns;
          a <= "0101";
          wait for 10 ns;
          b <= "0101";
          wait for 10 ns;
    end process;
   END;
```

Design a 16-bit CLA based on the information.

VHDL code:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                                    -- Using STD_LOGIC_LIBRARY_1164 package
ENTITY sixteen_bit_cla IS                                        -- entity declaration
   PORT
      ( a      :  IN   STD_LOGIC_VECTOR(15 DOWNTO 0);            -- a is used as input port
       b      :  IN   STD_LOGIC_VECTOR(15 DOWNTO 0);             -- b is used as input port
       carry_in  :  IN   STD_LOGIC;                              -- carry_in is used as input port
       sum       :  OUT  STD_LOGIC_VECTOR(15 DOWNTO 0);          -- sum is used as output port
       carry_out :  OUT  STD_LOGIC                               -- carry_out is used as input port
      );
END sixteen_bit_cla ;
--Architecture declaration
ARCHITECTURE behavioral OF sixteen_bit_cla  IS
--Signal declaration
SIGNAL   sum_new   :    STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL   g              :    STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL   p              :    STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL   carry          :    STD_LOGIC_VECTOR(15 DOWNTO 1);

BEGIN
   sum_new <= a XOR b;        -- sum_new signal logic equation
   g <= a AND b;              -- generate signal logic equation
   p <= a OR b;               -- propagate signal logic equation
   PROCESS (g,p,carry)
   BEGIN
   carry(1) <= g(0) OR (p(0) AND carry_in);
      inst: FOR i IN 1 TO 14 LOOP         -- for loop
```

```
          carry(i+1) <= g(i) OR (p(i) AND carry(i));
          END LOOP;
   carry_out <= g(15) OR (p(15) AND carry(15));
   END PROCESS;

   sum(0) <= sum_new(0) XOR carry_in;
   sum(15 DOWNTO 1) <= sum_new(15 DOWNTO 1) XOR carry(15 DOWNTO 1);
   END behavioral;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                          -- Use Library IEEE_STD_LOGIC_1164 package
ENTITY sixteen_bit_cla_tb IS                          -- entity declaration
END sixteen_bit_cla_tb;

ARCHITECTURE behavior OF sixteen_bit_cla_tb IS        -- architecture declaration
-- Component Declaration for the Unit Under Test (UUT)
   COMPONENT sixteen_bit_cla
   PORT(
       a : IN  std_logic_vector(15 downto 0);              -- a is used as input port
       b : IN  std_logic_vector(15 downto 0);              -- b is used as input port
      carry_in : IN  std_logic;                            -- carry_in is used as input port
       sum : OUT  std_logic_vector(15 downto 0);           -- sum is used as output port
       carry_out : OUT  std_logic                          -- carry_out is used as input port
      );
   END COMPONENT;



  --Inputs
  signal a : std_logic_vector(15 downto 0) := (others => '0');
  signal b : std_logic_vector(15 downto 0) := (others => '0');
  signal carry_in : std_logic := '0';
--Outputs
  signal sum : std_logic_vector(15 downto 0);
  signal carry_out : std_logic;
  BEGIN
 -- Instantiate the Unit Under Test (UUT)
 uut: sixteen_bit_cla  PORT MAP (
        a => a,
        b => b,
        carry_in => carry_in,
        sum => sum,
        carry_out => carry_out
      );
 -- Stimulus process
  stim_proc: process
  begin
  carry_in <= '0';
  a  <= "0000000000000000";
  wait for 10 ns;
  b  <= "0000000000000000";
  wait for 10 ns;
  a  <= "0000000000000001";
  wait for 10 ns;
  b  <= "0000000000000111";
  wait for 10 ns;
  a  <= "1100000000000110";
  wait for 10 ns;
  b  <= "1100000000001011";
  wait for 10 ns;
  end process;
  END;
```

Design a 32-bit CLA based on the information given.

VHDL code:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                                 -- Use Library IEEE_STD_LOGIC_1164 package

ENTITY thirty_two_bit_cla IS                                 -- entity declaration
   PORT
      (a      :  IN   STD_LOGIC_VECTOR(31 DOWNTO 0);              -- a is used as input port
       b      :  IN   STD_LOGIC_VECTOR(31 DOWNTO 0);              -- b is used as input port
       carry_in  :  IN   STD_LOGIC;                               -- carry_in is used as input port
       sum       :  OUT  STD_LOGIC_VECTOR(31 DOWNTO 0);           -- sum is used as output port
       carry_out :  OUT  STD_LOGIC                                -- carry_out is used as input port
      );
END thirty_two_bit_cla ;

--architecture declaration

ARCHITECTURE behavioral OF thirty_two_bit_cla  IS

--signal declaration

SIGNAL   sum_new   :   STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL   g         :   STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL   p         :   STD_LOGIC_VECTOR(31 DOWNTO 0);
SIGNAL   carry     :   STD_LOGIC_VECTOR(31 DOWNTO 1);

BEGIN
   sum_new <= a XOR b;                    -- sum_new signal logic equation
   g <= a AND b;                          -- generate signal logic equation
   p <= a OR b;                           -- propagate signal logic equation
   PROCESS (g,p,carry)
   BEGIN
   carry(1) <= g(0) OR (p(0) AND carry_in);
      inst: FOR i IN 1 TO 30 LOOP                              for loop
           carry(i+1) <= g(i) OR (p(i) AND carry(i));
           END LOOP;
   carry_out <= g(31) OR (p(31) AND carry(31));
   END PROCESS;

   sum(0) <= sum_new(0) XOR carry_in;
sum(31 DOWNTO 1) <= sum_new(31 DOWNTO 1) XOR carry(31 DOWNTO 1);
END behavioral;
```

Test Bench:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;                              -- Use Library IEEE_STD_LOGIC_1164 package
ENTITY thirthy_two_bit_cla_tb IS                                     -- entity declaration
END thirtytwo _bit_cla_tb;

--architecture declaration

ARCHITECTURE behavior OF thirty_two_bit_cla_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT thirty_two_bit_cla
   PORT(
       a : IN  std_logic_vector(31 downto 0);                     -- a is used as input port
       b : IN  std_logic_vector(31 downto 0);                    -- b is used as input port
       carry_in : IN  std_logic;                                  -- carry_in is used as input port
       sum : OUT  std_logic_vector(31 downto 0);                 -- sum is used as output port
       carry_out : OUT  std_logic                              -- carry_out is used as output port
      );
   END COMPONENT;
  --Inputs
  signal a : std_logic_vector(31 downto 0) := (others => '0');
  signal b : std_logic_vector(31 downto 0) := (others => '0');
  signal carry_in : std_logic := '0';
 --Outputs
  signal sum : std_logic_vector(31 downto 0);
  signal carry_out : std_logic;
  BEGIN
 -- Instantiate the Unit Under Test (UUT)
  uut: thirty_two_bit_cla PORT MAP (
        a => a,
        b => b,
        carry_in => carry_in,
        sum => sum,
        carry_out => carry_out
      );
```

```
-- Stimulus process
stim_proc: process
begin
carry_in <= '0';
 a <= "00000000000000000000000000000000";
 wait for 10 ns;
 b <= "00000000000000000000000000000000";
 wait for 10 ns;
 carry_in <= '1';
 wait for 10 ns;
   a <= "00000000000000000000000000000010";
   wait for 10 ns;
   b <= "00000000000000000000000000001000";
   wait for 10 ns;
   a <= "11000000000000000000000000000010";
   wait for 10 ns;
   b <= "11000000000000000000000000001000";
   wait for 10 ns;
  A <= x"0AF70000";
  wait for 10 ns;
  B <= x"23640000";
  wait for 10 ns;
  end process;
  END;
```