```
-------------------------------------------
--      Author:      Roger Grayson
--      Date: Feb, 2003
--      Purpose:a 32-bit carry look ahead adder
-------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_signed.conv_integer;
use work.my_stuff.all;       --my package

entity CLA32bit is
    port(   A,B: in std_logic_vector(31 downto 0);
            AS,Cin: in std_logic;
            Sum: out std_logic_vector(31 downto 0);
            Cout: out std_logic);
end CLA32bit;

architecture behavior of CLA32bit is
constant allone: std_logic_vector(31 downto 0):="11111111111111111111111111111111";
signal c16:std_logic;
signal SGG,SGP:std_logic_vector(1 downto 0);
signal BB:std_logic_vector(31 downto 0);
begin
    process (A,B,AS,Cin)
    begin
        if (AS='1') then BB <=(B xor allone);else BB <= B; end if;
    end process;
    CLA0:CLA16bit port map(A(15 downto 0),BB(15 downto 0),Cin,Sum(15 downto 0),SGG(0),SGP(
0));
    c16 <= SGG(0) or (SGP(0) and Cin);
    CLA1:CLA16bit port map(A(31 downto 16),BB(31 downto 16),c16,Sum(31 downto 16),SGG(1),
SGP(1));
    Cout <= SGG(1) or (SGP(1) and C16);
end behavior;
```