## CLA16bit.vhd

```
----------------------------------------
--      Author:      Roger Grayson
--      Date: Feb, 2003
--      Purpose:a 16-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 CLA16bit is
    port(   A,B: in std_logic_vector(15 downto 0);
            Cin: in std_logic;
            Sum: out std_logic_vector(15 downto 0);
            SGG,SGP: out std_logic);
end CLA16bit;

architecture behavior of CLA16bit is
signal C: std_logic_vector(16 downto 0);
signal G,P:std_logic_vector(15 downto 0);
signal GG,GP:std_logic_vector(3 downto 0);
begin
    C(0) <= Cin;
    CLA0:CLA4bit port map(A(3 downto 0),B(3 downto 0),C(0),Sum(3 downto 0),G(3 downto 0),P
(3 downto 0));
    C(1) <= G(0) or (P(0) and C(0));
    C(2) <= G(1) or (P(1) and C(1));
    C(3) <= G(2) or (P(2) and C(2));
    C(4) <= G(3) or (P(3) and C(3));
    CLA1:CLA4bit port map(A(7 downto 4),B(7 downto 4),C(4),Sum(7 downto 4),G(7 downto 4),P
(7 downto 4));
    C(5) <= G(4) or (P(4) and C(4));
    C(6) <= G(5) or (P(5) and C(5));
    C(7) <= G(6) or (P(6) and C(6));
    C(8) <= G(7) or (P(7) and C(7));
    CLA2:CLA4bit port map (A(11 downto 8),B(11 downto 8),C(8),Sum(11 downto 8),G(11 downto
 8),P(11 downto 8));
    C(9) <= G(8) or (P(8) and C(8));
    C(10) <= G(9) or (P(9) and C(9));
    C(11) <= G(10) or (P(10) and C(10));
    C(12) <= G(11) or (P(11) and C(11));
    CLA3:CLA4bit port map (A(15 downto 12),B(15 downto 12),C(12),Sum(15 downto 12),G(15
downto 12),P(15 downto 12));
    C(13) <= G(12) or (P(12) and C(12));
    C(14) <= G(13) or (P(13) and C(13));
    C(15) <= G(14) or (P(14) and C(14));
    C(16) <= G(15) or (P(15) and C(15));
    GG(0) <= G(3) or (P(3) and G(2)) or (P(3) and P(2) and G(1)) or (P(3) and P(2) and P(1
) and G(0));
    GG(1) <= G(7) or (P(7) and G(6)) or (P(7) and P(6) and G(5)) or (P(7) and P(6) and P(5
) and G(4));
    GG(2) <= G(11) or (P(11) and G(10)) or (P(11) and P(10) and G(9)) or (P(11) and P(10)
```

```
and P(9) and G(8));
    GG(3) <= G(15) or (P(15) and G(14)) or (P(15) and P(14) and G(13)) or (P(15) and P(14)
 and P(13) and G(12));
    GP(0) <= P(3) and P(2) and P(1) and C(0);
    GP(1) <= P(7) and P(6) and P(5) and C(4);
    GP(2) <= P(11) and P(10) and P(9) and C(8);
    GP(3) <= P(15) and P(14) and P(13) and C(12);
    SGG <= GG(3) or (GP(3) and GG(2)) or (GP(3) and GP(2) and GG(1)) or (GP(3) and GP(2)
and GP(1) and GG(0));
    SGP <= GP(3) and GP(2) and GP(1) and GP(0);
end behavior;
```