```
library IEEE;
      use IEEE.STD_LOGIC_1164.all;

entity CLA_block is
      port(
            a, b  : in std_logic_vector(3 downto 0);
            cin   : in std_logic;
            s     : out std_logic_vector(3 downto 0);
            cout  : out std_logic
      );
end entity CLA_block;

architecture mixed of CLA_block is
      component fulladder is
            port(
                  a, b  : in std_logic;
                  cin   : in std_logic;
                  s     : out std_logic;
                  cout  : out std_logic
            );
      end component;

      signal p, g : std_logic_vector(3 downto 0);
      signal c    : std_logic_vector(4 downto 0);
      signal p30, g30   : std_logic;

begin

      -- kobler sammen 4 fulladder
      addder:
            for i in 0 to 3 generate
                  ny_adder : fulladder
                  port map(
                        a => a(i),
                        b => b(i),
                        cin => c(i),
                        --
                        s => s(i), -- sum
                        cout => open -- vi bryr oss ikke om cout
                  );
            end generate addder;

      c(0) <= cin;
      -- propagate og generate per sum-adder
      cla : for i in 0 to 3 generate
            p(i) <= a(i) or b(i);
            g(i) <= a(i) and b(i);
            c(i+1) <= g(i) or (p(i) and c(i));

      end generate cla;

  -- dette er crry out
      p30 <= and p;
      g30 <= (((((((( g(0) and p(1) ) or g(1) ) and p(2) ) or g(2) ) and p(3) ) or
g(3) ) and p(3) ) or g(3));
      cout <= (p30 and cin) or g30;


end architecture;
```