

```
-----
--
-- Purpose:
--
--   Ripple Carry Adder
--
-- Discussion:
--
--
-- Licensing:
--
--   This code is distributed under the GNU LGPL license.
--
-- Modified:
--
--   2012.04.02
--
-- Author:
--
--   Young W. Lim
--
-- Parameters:
--
--   Input:
--
--   Output:
-----
```

```
library STD;
use STD.textio.all;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
```

```
entity adder is
  generic (
    WD      : in natural := 32);

  port (
    an      : in   std_logic_vector (WD-1 downto 0) := (others=>'0');
    bn      : in   std_logic_vector (WD-1 downto 0) := (others=>'0');
    ci      : in   std_logic := '0';
    cn      : out  std_logic_vector (WD-1 downto 0) := (others=>'0');
    co      : out  std_logic := '0');
```

```
end adder;
```

```
architecture rca of adder is
begin
  process (an, bn, ci)
    variable sn : std_logic_vector (WD-1 downto 0) := (others=>'0');
    variable c  : std_logic := '0';
  begin -- process
    c := ci;
    for i in 0 to WD-1 loop
      sn(i) := an(i) xor bn(i) xor c;
      c := (an(i) and bn(i)) or (an(i) and c) or (bn(i) and c);
    end loop; -- i

    cn <= sn;
    co <= c;
  end process;
end rca;
```