File: C:/my_designs/register_128b/register_128b/src/reg128b_32.vhd

```
1   ------------------------------------------------------------------------
    ----
2   --
3   -- Title       : reg128b_32
4   -- Design      : register_128b
5   -- Author      : Aaron Lin and Hang Chen
6   -- Company     : Stony Brook University
7   --
8   ------------------------------------------------------------------------
    ----
9   --
10  -- File        :
    C:\my_designs\register_128b\register_128b\src\reg128b_32.vhd
11  -- Generated   : Mon Apr 20 12:01:06 2020
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ------------------------------------------------------------------------
    ----
16  --
17  -- Description : 32 128 bits Register File that reads on a rising clock
    edge, and writes on a rising clock edge and W_EN on high.
18  --  3 registers can be read at a time, while 1 can be written to.
19  --
20  ------------------------------------------------------------------------
    ----
21
22  library IEEE;
23  use IEEE.std_logic_1164.all;
24  use IEEE.NUMERIC_STD.all;
25
26  entity reg128b_32 is
27       port(
28           CLK : in STD_LOGIC;
29           W_EN : in STD_LOGIC;
30           I_regA : in STD_LOGIC_VECTOR(4 downto 0);
31           I_regB : in STD_LOGIC_VECTOR(4 downto 0);
32           I_regC : in STD_LOGIC_VECTOR(4 downto 0);
33           I_regD : in STD_LOGIC_VECTOR(4 downto 0);
34           I_DataIn : in STD_LOGIC_VECTOR(127 downto 0);
35           O_regA : out STD_LOGIC_VECTOR(127 downto 0);
36           O_regB : out STD_LOGIC_VECTOR(127 downto 0);
37           O_regC : out STD_LOGIC_VECTOR(127 downto 0)
38           );
39  end reg128b_32;
40
41  --}} End of automatically maintained section
42
43  architecture reg128b_32 of reg128b_32 is
44
45  begin
46      process(CLK)
47      type stor is array (0 to 31) of std_logic_vector(127 downto 0);
48      variable reg: stor := (others => X"00000000000000000000000000000000");
49      begin
50          if rising_edge(CLK) then
51              if W_EN = '1' then
```

```
52                  reg(to_integer(unsigned(I_regD))) := I_DataIn;
53              end if;
54              O_regA <= reg(to_integer(unsigned(I_regA)));
55              O_regB <= reg(to_integer(unsigned(I_regB)));
56              O_regC <= reg(to_integer(unsigned(I_regC)));
57
58          end if;
59      end process;
60  end reg128b_32;
61
```