```
-------------------------------------------------------------------------------
--
-- Title       : ID_EX
-- Design      : ID_EX
-- Author      : Aaron Lin and Hang Chen
-- Company     : Stony Brook University
--
-------------------------------------------------------------------------------
--
-- File        : c:\my_designs\ID_EX\ID_EX\src\ID_EX.vhd
-- Generated   : Fri May  1 12:36:02 2020
-- From        : interface description file
-- By          : Itf2Vhdl ver. 1.22
--
-------------------------------------------------------------------------------
--
-- Description : Register between ID and EX stage for storing the W_EN signal, registers A, B, C, and D, the data of
--  registers A, B, and C, and ALUop.
--
-------------------------------------------------------------------------------

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.NUMERIC_STD.all;

entity ID_EX is
    port(
         CLK : std_logic;
         I_W_EN : in STD_LOGIC;
         O_W_EN : out STD_LOGIC;
         I_regA : in STD_LOGIC_VECTOR(4 downto 0);
         I_regB : in STD_LOGIC_VECTOR(4 downto 0);
         I_regC : in STD_LOGIC_VECTOR(4 downto 0);
         I_regD : in STD_LOGIC_VECTOR(4 downto 0);
         I_ALUop : in STD_LOGIC_VECTOR(19 downto 0);
         I_regA_Data : in STD_LOGIC_VECTOR(127 downto 0);
         I_regB_Data : in STD_LOGIC_VECTOR(127 downto 0);
         I_regC_Data : in STD_LOGIC_VECTOR(127 downto 0);
         O_regA : out STD_LOGIC_VECTOR(4 downto 0);
         O_regB : out STD_LOGIC_VECTOR(4 downto 0);
         O_regC : out STD_LOGIC_VECTOR(4 downto 0);
         O_regD : out STD_LOGIC_VECTOR(4 downto 0);
         O_ALUop : out STD_LOGIC_VECTOR(19 downto 0);
         O_regA_Data : out STD_LOGIC_VECTOR(127 downto 0);
         O_regB_Data : out STD_LOGIC_VECTOR(127 downto 0);
         O_regC_Data : out STD_LOGIC_VECTOR(127 downto 0)
         );
end ID_EX;

--}} End of automatically maintained section

architecture ID_EX of ID_EX is
```

```vhdl
begin

    process(CLK)
    variable W_EN : std_logic := '0';
    type stor is array (0 to 3) of std_logic_vector(4 downto 0);
    type stor2 is array (0 to 2) of std_logic_vector(127 downto 0);
    variable ALUop : std_logic_vector(19 downto 0) := std_logic_vector(to_unsigned(0, 20));
    variable reg : stor := (others => "00000");
    variable data : stor2 := (others => X"00000000000000000000000000000000");
    begin
        if rising_edge(CLK) then
            W_EN := I_W_EN;
            reg(0) := I_regA;
            reg(1) := I_regB;
            reg(2) := I_regC;
            reg(3) := I_regD;
            ALUop := I_ALUop;
            data(0) := I_regA_Data;
            data(1) := I_regB_Data;
            data(2) := I_regC_Data;

            O_W_EN <= W_EN;
            O_regA <= reg(0);
            O_regB <= reg(1);
            O_regC <= reg(2);
            O_regD <= reg(3);
            O_ALUop <= ALUop;
            O_regA_Data <= data(0);
            O_regB_Data <= data(1);
            O_regC_Data <= data(2);

        end if;
    end process;
end ID_EX;

```