ALU.vhd

```
----------------------------------------------
--      Author:      Roger Grayson
--      Date: Feb, 2003
--      Purpose:An arithmetic logic unit
--                   implemented using a CLA and
--                   a barrel shifter
--      Updated Mar, 2003 to implement immediate
--                   addressing mode
----------------------------------------------
library IEEE;
library STD;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_signed.conv_integer;
use IEEE.STD_LOGIC_arith.all;
use work.my_stuff.all;       --my package

entity ALU is
    port(   control:in std_logic_vector(1 downto 0);
            eq:out std_logic;
            ir:in std_logic_vector(31 downto 0);
            A,B:in std_logic_vector(31 downto 0);
            R:out std_logic_vector(31 downto 0));
end ALU;

architecture behavior of ALU is
signal sumi,immzx,immsx,sum,diff,ShR,ShL: std_logic_vector(31 downto 0);
signal shamt: std_logic_vector(4 downto 0);
signal temp1: std_logic;
signal iA,iB: integer;
--lowest 6 bits of machine code
constant ADD_op:    std_logic_vector(5 downto 0) := "100000";
constant SUB_op:    std_logic_vector(5 downto 0) := "100010";
constant OR_op:     std_logic_vector(5 downto 0) := "100101";
constant AND_op:    std_logic_vector(5 downto 0) := "100100";
constant SLT_op:    std_logic_vector(5 downto 0) := "101010";
constant SLL_op:    std_logic_vector(5 downto 0) := "000000";
constant SRL_op:    std_logic_vector(5 downto 0) := "000010";
--one and zero constants
constant zero:  std_logic := '0';
constant one:   std_logic := '1';

begin
    addd:CLA32bit port map (A,B,zero,zero,sum,temp1);
    shamt<=(IR(10 downto 6));
    --sign extension of lowest 16 bits of ir
    immsx <=ir(15)&ir(15)&ir(15)&ir(15)&ir(15)&ir(15)&ir(15)&ir(15)&ir(15)&ir(15)
        &ir(15)&ir(15)&ir(15)&ir(15)&ir(15)& ir(15 downto 0);
    --zero extension of lowest 16 bits of ir
    immzx <="0000000000000000" & ir(15 downto 0);
    --port maps
    addi:CLA32bit port map (A,immsx,zero,zero,sumi,temp1);
    subb:CLA32bit port map (A,B,one,one,diff,temp1);
```

```
    shift_right:barrel_shift port map (A,one,shamt,ShR);
    shift_left:barrel_shift port map (A,zero,shamt,ShL);
    process (control,ir,A,B,sum,sumi,diff,ShR,ShL,immsx,immzx)
    begin
        iA<=conv_integer(A);
        iB<=conv_integer(B);
        if(A=B) then eq <= '1'; else eq <='0'; end if;
        case (control) is
            when "00" =>
                case (ir(5 downto 0)) is
                    when ADD_op => R <= sum;
                    when SUB_op => R <= diff;
                    when SLL_op => R <= ShL;
                    when SRL_op => R <= ShR;
                    when SLT_op =>
                        if (iA<iB) then R <=x"00000001"; else R <= (others=>'0'); end if;
                    when AND_op => R <= (A and B);
                    when OR_op => R <= (A or B);
                    when others => report("Lower 6 bits do not define a legal ALU op.");
                end case;
            when "01" => R <= diff;
            when "10" =>
                case (ir(29 downto 26)) is
                    when "1000" => R <= sumi;
                    when "1100" => R <= (A and immzx);
                    when "1101" => R <= (A or immzx);
                    when others => report("Illegal IR[29-26] in immediate mode");
                end case;
            when "11" => R <= sum;
            when others => report("Illegal control setting to ALU");
        end case;
    end process;
end behavior;
```