1 alu.vhd

```
-------------------------------------------------------------------------------
--
--
-- Title       : ALU
-- Design      : ALU
-- Author      : Robert Bacigalupo and Tyler Ovenden
-- Company     :
--
-------------------------------------------------------------------------------
--
--
-- File        : B:\Stony Brook\ESE345\Project\ESE345Project\ALU\src\ALU.vhd
-- Generated   : Sun Oct  2 17:41:31 2022
-- From        : interface description file
-- By          : Itf2Vhdl ver. 1.22
--
-------------------------------------------------------------------------------
--
--
-- Description :
--
-------------------------------------------------------------------------------
--

--{{ Section below this comment is automatically maintained
--   and may be overwritten
--{entity {ALU} architecture {ALU_Behavior}}


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity ALU is
    port(

         r1 : in STD_LOGIC_VECTOR(127 downto 0);
         r2 : in STD_LOGIC_VECTOR(127 downto 0);
         r3 : in STD_LOGIC_VECTOR(127 downto 0);
         instrc: in STD_LOGIC_VECTOR(24 downto 0);
         o : out STD_LOGIC_VECTOR(127 downto 0)
         );
end ALU;

--}} End of automatically maintained section

architecture ALU_Behavior of ALU is

begin
process(r1, r2, r3, instrc)




variable index : integer;   --temp variable used for storing load index
variable tempRes: signed (63 downto 0);    --need to be 64 for possible overf
low
variable temp_int0:integer:=0;
variable temp_int1:integer:=0;
variable temp_int2:integer:=0;
variable temp_int3:integer:=0;


variable tempRes128: signed (127 downto 0);



variable testMult: std_logic_vector(130 downto 0);
variable testRes: std_logic_vector(130 downto 0);
variable testRes1: std_logic_vector(32 downto 0);
variable testRes2: std_logic_vector(64 downto 0);
variable testRes3: std_logic_vector(17 downto 0);

variable testR1: std_logic_vector (17 downto 0);
variable testR2: signed (16 downto 0);

variable testMult1: std_logic_vector(32 downto 0);
variable testMult2: std_logic_vector(64 downto 0);
variable rd: std_logic_vector(127 downto 0);

constant max16 : std_logic_vector(15 downto 0) := "0111111111111111";
constant min16 : std_logic_vector(15 downto 0) := "1000000000000000";
constant max32: std_logic_vector(31 downto 0):= "0111111111111111111111111111
1111";
constant min32: std_logic_vector(31 downto 0) := X"80000000";
constant max64: std_logic_vector(63 downto 0):= X"7FFFFFFFFFFFFFFF";
constant min64: std_logic_vector(63 downto 0) := X"8000000000000000";


constant max16_int: integer:= 32767;
constant min16_int: integer:= -32768;
constant max32_int : integer := 2147483647;
constant min32_int : integer :=  -2147483648;




variable counter: integer:=0;       --counter for counting ones
variable tempPos : integer;
variable temp : std_logic_vector(31 downto 0); -- general temp variable, curr
ently using in ROTW


    begin


        --load immeditate
        if instrc(24) = '0' then


            o <= r1;

            --convert load index into a multiple of 16
            index := to_integer(signed(instrc(23 downto 21))) * 16;

            --load immeditate
            o(index + 15 downto index) <= instrc(20 downto 5);



            --r4
                elsif   (instrc(24 downto 23) = "10") then
```

```
                    --Signed Integer Multiply-Add Low with Saturation
                --y low 16-bit-fields for rs3, rs2
                if (instrc(22 downto 20) = "000")   then


                    for i in 0 to 3 loop

                    tempPos := 32 * i;


                    testMult1(32 downto 0) := std_logic_vector(resize(signed(
r3(tempPos+15 downto tempPos))* signed(r2(tempPos+15 downto tempPos)), 33));
                    testRes1(32 downto 0) := std_logic_vector(resize(signed(t
estMult1(32 downto 0))+ signed(r1(tempPos+31 downto tempPos)), 33));


                    if(signed(testRes1(32 downto 0)) >   signed( max32)) then


                        o(tempPos+31 downto tempPos)<= "011111111111111111111
11111111111";

                    elsif(signed(testRes1(32 downto 0)) <  signed( min32)) th
en
                        o(tempPos+31 downto tempPos)<= "100000000000000000000
00000000000";

                    else

                        o(tempPos+31 downto tempPos)<= testRes1(31 downto 0);

                     end if;
                end loop;


                elsif (instrc(22 downto 20) = "001")    then
                    for i in 0 to 3 loop
                        tempPos := 32 * i;



                    testMult1(32 downto 0) := std_logic_vector(resize(signed(
r3(tempPos+31 downto tempPos+16))* signed(r2(tempPos+31 downto tempPos+16)),
33));
                    testRes1(32 downto 0) := std_logic_vector(resize(signed(t
estMult1(32 downto 0))+ signed(r1(tempPos+31 downto tempPos)), 33));


                        if(signed(testRes1(32 downto 0)) >   signed( max32))
then


                    --  if testRes1(32) = '1' then
                        o(tempPos+31 downto tempPos)<= "011111111111111111111
11111111111";

                    elsif(signed(testRes1(32 downto 0)) <  signed( min32)) th
en

                        o(tempPos+31 downto tempPos)<= "100000000000000000000
```

```
00000000000";
                    else
                        o(tempPos+31 downto tempPos)<= testRes1(31 downto 0);


                        end if;

                end loop;



                elsif (instrc(22 downto 20) = "010")    then


                    for i in 0 to 3 loop
                    tempPos := 32 * i;

                    testMult(tempPos+32 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+15 downto tempPos))* signed(r2(tempPos+15 downto temp
Pos)), 33));
                    testRes(tempPos+32 downto tempPos) := std_logic_vector(re
size(signed(r1(tempPos+31 downto tempPos))- signed(testMult(tempPos+32 downto
 tempPos)), 33));

                        if(signed(testRes(tempPos+32 downto tempPos)) >   sig
ned( max32)) then
                        o(tempPos+31 downto tempPos)<= "01111111111111111111
11111111111";
                    elsif(signed(testRes(tempPos+32 downto tempPos)) <  signe
d( min32)) then
                        o(tempPos+31 downto tempPos)<= "10000000000000000000
00000000000";
                    else
                        o(tempPos+31 downto tempPos)<= testRes(tempPos+31 dow
nto tempPos);
                     end if;
                end loop;


            elsif (instrc(22 downto 20) = "011")    then


                for i in 0 to 3 loop
                    tempPos := 32 * i;

                    testMult(tempPos+32 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+31 downto tempPos+16))* signed(r2(tempPos+31 downto t
empPos+16)), 33));
                    testRes(tempPos+32 downto tempPos) := std_logic_vector(re
size(signed(r1(tempPos+31 downto tempPos))- signed(testMult(tempPos+32 downto
 tempPos)), 33));


                        if(signed(testRes(tempPos+32 downto tempPos)) >   sig
ned( max32)) then
                        o(tempPos+31 downto tempPos)<= "011111111111111111111
11111111111";

                    elsif(signed(testRes(tempPos+32 downto tempPos)) <  signe
d( min32)) then
                        o(tempPos+31 downto tempPos)<= "100000000000000000000
00000000000";

                    else

                        o(tempPos+31 downto tempPos)<= testRes(tempPos+31 dow
nto tempPos);

                    end if;







                end loop;

            elsif (instrc(22 downto 20) = "100")    then


                    for i in 0 to 1 loop


                        tempPos := 64 * i;


                    testMult(tempPos+64 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+31 downto tempPos))* signed(r2(tempPos+31 downto temp
Pos)), 65));
                    testRes(tempPos+64 downto tempPos) := std_logic_vector(re
size(signed(testMult(tempPos+64 downto tempPos))+ signed(r1(tempPos+63 downto
 tempPos)), 65));

                        if(signed(testRes(tempPos+64 downto tempPos)) >   sig
ned( max64)) then
                        o(tempPos+63 downto tempPos)<= "011111111111111111111
1111111111111111111111111111111111111111111";

                    elsif(signed(testRes(tempPos+64 downto tempPos)) <  signe
d( min64)) then
                        o(tempPos+63 downto tempPos)<= "100000000000000000000
0000000000000000000000000000000000000000000";


                    else
```

```
                        o(tempPos+63 downto tempPos)<= testRes(tempPos+63 dow
nto tempPos);
                    end if;


                end loop;
                elsif (instrc(22 downto 20) = "101")    then
                        for i in 0 to 1 loop

                        tempPos := 64 * i;


                    testMult(tempPos+64 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+63 downto tempPos+32))* signed(r2(tempPos+63 downto t
empPos+32)), 65));
                    testRes(tempPos+64 downto tempPos) := std_logic_vector(re
size(signed(testMult(tempPos+64 downto tempPos))+ signed(r1(tempPos+63 downto
 tempPos)), 65));


                    if(signed(testRes(tempPos+64 downto tempPos)) >   signed(
 max64)) then
                        o(tempPos+63 downto tempPos)<= "011111111111111111111
1111111111111111111111111111111111111111111";
                    elsif(signed(testRes(tempPos+64 downto tempPos)) <  signe
d( min64)) then
                        o(tempPos+63 downto tempPos)<= "100000000000000000000
0000000000000000000000000000000000000000000";

                    else
                        o(tempPos+63 downto tempPos)<= testRes(tempPos+63 dow
nto tempPos);
                    end if;



                end loop;
                elsif (instrc(22 downto 20) = "110")    then
                        for i in 0 to 1 loop

                        tempPos := 64 * i;


                    testMult(tempPos+64 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+31 downto tempPos))* signed(r2(tempPos+31 downto temp
Pos)), 65));
                    testRes(tempPos+64 downto tempPos) := std_logic_vector(re
size(signed(r1(tempPos+63 downto tempPos))- signed(testMult(tempPos+64 downto
 tempPos)), 65));


                    if(signed(testRes(tempPos+64 downto tempPos)) >   signed(
 max64)) then
                        o(tempPos+63 downto tempPos)<= "011111111111111111111
1111111111111111111111111111111111111111111";

                    elsif(signed(testRes(tempPos+64 downto tempPos)) <  signe
d( min64)) then
                        o(tempPos+63 downto tempPos)<= "100000000000000000000
0000000000000000000000000000000000000000000";


                    else

                        o(tempPos+63 downto tempPos)<= testRes(tempPos+63 dow
nto tempPos);

                    end if;


                end loop;




                elsif (instrc(22 downto 20) = "111")    then
                            for i in 0 to 1 loop

                        tempPos := 64 * i;


                    testMult(tempPos+64 downto tempPos) := std_logic_vector(r
esize(signed(r3(tempPos+63 downto tempPos+32))* signed(r2(tempPos+63 downto t
empPos+32)), 65));
                    testRes(tempPos+64 downto tempPos) := std_logic_vector(re
size(signed(r1(tempPos+63 downto tempPos))- signed(testMult(tempPos+64 downto
 tempPos)), 65));


                    if(signed(testRes(tempPos+64 downto tempPos)) >   signed(
 max64)) then
                        o(tempPos+63 downto tempPos)<= "011111111111111111111
1111111111111111111111111111111111111111111";

                    elsif(signed(testRes(tempPos+64 downto tempPos)) <  signe
d( min64)) then
                        o(tempPos+63 downto tempPos)<= "100000000000000000000
0000000000000000000000000000000000000000000";


                    else

                        o(tempPos+63 downto tempPos)<= testRes(tempPos+63 dow
nto tempPos);

                    end if;


                end loop;
```

```
            end if;



---------------------------------------------------------------------------------------
-----------
---------------------r3 instrcs----------------------------------------------
------------------

                        elsif   (instrc(24 downto 23) = "11") then
                                if instrc(18 downto 15) = "0000" then  --nop


                                    Null;

                                elsif (instrc(18 downto 15) = "0001") then --lead
ing zeros

                                    for i in 0 to 3 loop
                                        tempPos := (32 * i)+31;
                                        counter := 0;


                                        for j in 0 to 31 loop
                                            if r1(tempPos - j) = '0' then
                                                counter := counter + 1;
                                            else
                                                exit;
                                            end if;
                                        end loop;
                                    o((tempPos) downto (tempPos-31)) <=  std_logi
c_vector(to_unsigned(counter,32));

                                end loop;



                                elsif (instrc(18 downto 15) = "0010") then      -
-add word
                                    for i in 0 to 3 loop
                                        tempPos := 32 * i;
                                        o(tempPos+31 downto tempPos)<=  std_logic
_vector(unsigned(r1(tempPos+31 downto tempPos)) + unsigned(r2(tempPos+31 down
to tempPos)));
                                    end loop;


                                elsif (instrc(18 downto 15) = "0011") then  --add
 half word
                                    for i in 0 to 7 loop
                                        tempPos := 16 * i;
                                        o(tempPos+15 downto tempPos)<=  std_logic
_vector(unsigned(r1(tempPos+15 downto tempPos)) + unsigned(r2(tempPos+15 down
to tempPos)));
                                    end loop;



                                    elsif(instrc(18 downto 15) = "0100") then
--add half word saturated
```

```
                                          for i in 0 to 7 loop
                        tempPos := 16 * i;

                        temp_int0 := to_integer(signed(r1(tempPos+15 downto t
empPos)))+to_integer(signed(r2(tempPos+15 downto tempPos)));

                        if temp_int0>max16_int then
                            o(tempPos+15 downto tempPos)<= "0111111111111111"
;
                        elsif   temp_int0<min16_int then
                            o(tempPos+15 downto tempPos)<= "1000000000000000"
;
                        else
                            o(tempPos+15 downto tempPos)<= std_logic_vector(t
o_signed(temp_int0,16));


                             end if;
                        end loop;


                            elsif (instrc(18 downto 15) = "0101") then  --and
 r1 r2
                                o <= r1 and r2;

                            elsif (instrc(18 downto 15) = "0110") then  --  b
roadcast word
                                    for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    o((31 + tempPos) downto (tempPos)) <=  r1
(31 downto 0);

                                    end loop;



                            elsif (instrc(18 downto 15) = "0111") then  --  m
ax signed word
                                for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    temp_int1 := to_integer(signed(r1(tempPos
+31 downto tempPos)));

                                    temp_int2 := to_integer(signed(r2(tempPos
+31 downto tempPos)));

                                    if(temp_int1 > temp_int2) then
                                        o((31 + tempPos) downto tempPos) <=
 r1(tempPos+31 downto tempPos);
                                    else
                                        o((31 + tempPos) downto tempPos) <=
 r2(tempPos+31 downto tempPos);
                                    end if;
                                end loop;



                            elsif (instrc(18 downto 15) = "1000") then  --  m
in signed word
                                for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    temp_int1 := to_integer(signed(r1(tempPos
+31 downto tempPos)));
```

```
                                    temp_int2 := to_integer(signed(r2(tempPos
+31 downto tempPos)));
                                    if(temp_int1 < temp_int2) then
                                        o((31 + tempPos) downto tempPos) <=
 r1(tempPos+31 downto tempPos);
                                    else
                                        o((31 + tempPos) downto tempPos) <=
 r2(tempPos+31 downto tempPos);
                                    end if;
                                end loop;


                            elsif (instrc(18 downto 15) = "1001") then  --mul
tiply low bits of r1, r2
                                    for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    o((31 + tempPos) downto (tempPos)) <=  st
d_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) * unsigned(r2((1
5 + tempPos) downto (tempPos))));
                                    end loop;

                            elsif (instrc(18 downto 15) = "1010") then  --mul
tiply by constant
                                    for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    o((31 + tempPos) downto (tempPos)) <= std
_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) * resize(unsigned
(instrc(14 downto 10)),16));
                                    --switch out if need to resize
                        --          o((31 + tempPos) downto (tempPos)) <=  st
d_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) * unsigned(instr
c(14 downto 10)));
                                    end loop;

                            elsif (instrc(18 downto 15) = "1011") then  --or
r1 r2
                                o <=  r1 or r2;


                            elsif (instrc(18 downto 15) = "1100") then      -
-counts 1s in word

                                for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    counter := 0;
                                    for j in 0 to 31 loop
                                        if r1(j + tempPos) = '1' then
                                            counter := counter + 1;
                                        end if;
                                    end loop;
                                o((31 + tempPos) downto (tempPos)) <=  std_lo
gic_vector(to_unsigned(counter,32));

                            end loop;

                            elsif (instrc (18 downto 15) = "1101") then  --RO
TW rotate bits in word
                                for i in 0 to 3 loop
                                    tempPos := 32 * i;
                                    temp := r1(temppos + 31 downto temppos);
                                    temp_int1 := to_integer(unsigned(r2(tempp
os + 5 downto temppos)));

                                    o(temppos+31 downto temppos)<= std_logic_
```

```
vector(unsigned(temp) ror temp_int1);
                                end loop;

                            elsif (instrc(18 downto 15) = "1110") then      -
-sub word


                                    for i in 0 to 3 loop
                                        tempPos := 32 * i;
                                        o(tempPos+31 downto tempPos)<=  std_l
ogic_vector(unsigned(r1(tempPos+31 downto tempPos)) - unsigned(r2(tempPos+31
downto tempPos)));
                                    end loop;


                            elsif(instrc(18 downto 15) = "1111") then       -
-sub saturated



                                            for i in 0 to 7 loop

                        tempPos := 16 * i;

                        temp_int0 := to_integer(signed(r1(tempPos+15 downto t
empPos)))-to_integer(signed(r2(tempPos+15 downto tempPos)));

                        if temp_int0>max16_int then
                            o(tempPos+15 downto tempPos)<= "0111111111111111"
;
                        elsif   temp_int0<min16_int then
                            o(tempPos+15 downto tempPos)<= "1000000000000000"
;
                        else
                            o(tempPos+15 downto tempPos)<= std_logic_vector(t
o_signed(temp_int0,16));


                            end if;
                        end loop;

                  end if;
             end if;


    end process;
end ALU_Behavior;
```