1 alu_tb.vhd

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

entity alu_tb is
end alu_tb;


architecture tb of alu_tb is
    signal r1, r2, r3, o : std_logic_vector(127 downto 0);
    signal instrc : std_logic_vector(24 downto 0);
    constant clk : time := 5us;
begin
    UUT : entity alu
        port map (r1 => r1, r2 => r2, r3 => r3, instrc => instrc, o => o);

        process begin


--r4 register set up
--r1
--r1(127 downto 112)    <= std_logic_vector(to_signed(10, 16));
--r1(111 downto 96)     <= std_logic_vector(to_signed(10, 16));
--r1(95 downto 80)  <= std_logic_vector(to_signed(10, 16));
r1(127 downto 96)   <= std_logic_vector(to_signed(-2147483648, 32));--underfl
ow

r1(95 downto 64)    <= std_logic_vector(to_signed(2147483647, 32)); --overflo
w

r1(63 downto 48)    <= std_logic_vector(to_signed(10, 16));
r1(47 downto 32)    <= std_logic_vector(to_signed(10, 16));
r1(31 downto 16)    <= std_logic_vector(to_signed(10, 16));
r1(15 downto 0)     <= std_logic_vector(to_signed(10, 16));

--r2
r2(127 downto 112)  <= std_logic_vector(to_signed(32767, 16));  --for underfl
ow
r2(111 downto 96)   <= std_logic_vector(to_signed(32767, 16));  --for underfl
ow
r2(95 downto 80)    <= std_logic_vector(to_signed(32767, 16));  --for overflo
w
r2(79 downto 64)    <= std_logic_vector(to_signed(32767, 16));  --for overflo
w
r2(63 downto 48)    <= std_logic_vector(to_signed(-2, 16));
r2(47 downto 32)    <= std_logic_vector(to_signed(-2, 16));
r2(31 downto 16)    <= std_logic_vector(to_signed(-5, 16));
r2(15 downto 0)     <= std_logic_vector(to_signed(2, 16));

--r2
r3(127 downto 112)  <= std_logic_vector(to_signed(-32768, 16));--for underflo
w
r3(111 downto 96)   <= std_logic_vector(to_signed(-32768, 16));--for underflo
w
r3(95 downto 80)    <= std_logic_vector(to_signed(32767, 16)); --for overflow
r3(79 downto 64)    <= std_logic_vector(to_signed(32767, 16)); --for overflow
r3(63 downto 48)    <= std_logic_vector(to_signed(20, 16));
r3(47 downto 32)    <= std_logic_vector(to_signed(20, 16));
```

```
r3(31 downto 16)    <= std_logic_vector(to_signed(10, 16));
r3(15 downto 0)     <= std_logic_vector(to_signed(20, 16));


wait for clk;
instrc <= "0000011101001001001000000"; -- imm expected to load  0111010010010
010 = hex 7492, position from pos 16-0
wait for clk;


--R4 Instructions
--expected result
--last 127 to  bit field has overflow
--next 32 bit field has underflow
--next 32 bit field has regular negative
--next regular multiply add


-- 32 bit
-- 32 bit Signed Integer Multiply-Add Low with Saturation
instrc <= "1000000000000000000000000";
wait for clk;


-- 32 bit Signed Integer Multiply-Add high with Saturation

instrc <= "1000100000000000000000000";
wait for clk;




-- 32 bit Signed Integer Multiply-sub low with Saturation
instrc <= "1001000000000000000000000";
wait for clk;

-- 32 bit Signed Integer Multiply-sub high with Saturation
instrc <= "1001100000000000000000000";
wait for clk;


-- 64 bit
--nop to divide sections
instrc <= "1100000000000000000000000";  --"1800000" hex
wait for clk;

-- Long Signed Integer Multiply-Add Low with Saturation
instrc <= "1010000000000000000000000";
wait for clk;


-- Long Signed Integer Multiply-Add high with Saturation

instrc <= "1010100000000000000000000";
wait for clk;




-- Long Signed Integer Multiply-sub low with Saturation
instrc <= "1011000000000000000000000";
wait for clk;
```

```
-- Long Signed Integer Multiply-sub high with Saturation
instrc <= "1011100000000000000000000";
wait for clk;



--r3(instr) register set up
--r1
r1(127 downto 112)  <= std_logic_vector(to_signed(0, 16)); --for counting zer
os
r1(111 downto 96)   <= std_logic_vector(to_signed(0, 16));
r1(95 downto 80)    <= std_logic_vector(to_signed(-32768, 16));
r1(79 downto 64)    <= std_logic_vector(to_signed(10, 16));
r1(63 downto 48)    <= std_logic_vector(to_signed(10, 16));
r1(47 downto 32)    <= std_logic_vector(to_signed(10, 16));
r1(31 downto 16)    <= std_logic_vector(to_signed(10, 16));
r1(15 downto 0)     <= std_logic_vector(to_signed(10, 16));

--r2
r2(127 downto 112)  <= std_logic_vector(to_signed(0, 16));
r2(111 downto 96)   <= std_logic_vector(to_signed(10, 16));
r2(95 downto 80)    <= std_logic_vector(to_signed(32767, 16));
r2(79 downto 64)    <= std_logic_vector(to_signed(32767, 16));
r2(63 downto 48)    <= std_logic_vector(to_signed(-2, 16));
r2(47 downto 32)    <= std_logic_vector(to_signed(-2, 16));
r2(31 downto 16)    <= std_logic_vector(to_signed(2, 16));
r2(15 downto 0)     <= std_logic_vector(to_signed(2, 16));

--r3
r3(127 downto 112)  <= std_logic_vector(to_signed(-32768, 16));
r3(111 downto 96)   <= std_logic_vector(to_signed(-32768, 16));
r3(95 downto 80)    <= std_logic_vector(to_signed(32767, 16));
r3(79 downto 64)    <= std_logic_vector(to_signed(32767, 16));
r3(63 downto 48)    <= std_logic_vector(to_signed(20, 16));
r3(47 downto 32)    <= std_logic_vector(to_signed(20, 16));
r3(31 downto 16)    <= std_logic_vector(to_signed(20, 16));
r3(15 downto 0)     <= std_logic_vector(to_signed(20, 16));





--or
instrc <= "1100001011000000000000000";    --1858000
wait for clk;

--and
instrc <= "1100001010000000000000000";  -- 1828000
wait for clk;


--broadcast word
instrc <= "1100000110000000000000000"; -- 1830000
wait for clk;



----max signed word
instrc <= "1100000111000000000000000"; --1838000
wait for clk;

----min signed word
instrc <= "1100001000000000000000000";  --1840000
```

```
wait for clk;

--leading zeroes
instrc <= "1100000001000000000000000";   ---1800000
wait for clk;


--count 1s
instrc <= "1100000110000000000000000";   ---1860000
wait for clk;

--add word
instrc <= "1100000001000000000000000";   ---1810000
wait for clk;

--add half word
instrc <= "1100000001100000000000000";   ---1818000
wait for clk;

--add half word saturated
instrc <= "1100000100000000000000000";   ---1820000
wait for clk;

--sub half word
instrc <= "1100001110000000000000000";   ---1870000
wait for clk;

--sub half word saturated
instrc <= "1100001111000000000000000";   ---1878000
wait for clk;

--r1 values
r1(127 downto 96)   <= std_logic_vector(to_signed(2147483647, 32));

r1(95 downto 64)    <= std_logic_vector(to_signed(2147483647, 32));
r1(95) <= '1';

r2(111 downto 96)   <= std_logic_vector(to_signed(65535, 16));  --for underfl
ow
r3(111 downto 96)   <= std_logic_vector(to_signed(65535, 16));  --for underfl
ow
r3(111 downto 96)   <= std_logic_vector(to_signed(0, 16));  --for underflow
-- 64 bit signed   Integer Multiply-Add low with Saturation
instrc <= "1010000000000000000000000";
wait for clk;





    std.env.finish;

    end process;

end tb;
```