File: C:/my_designs/ALU/ALU/src/R4_6_tb.vhd

```
1    ------------------------------------------------------------------------
     -----
2    --
3    -- Title       : R4_6_tb
4    -- Design      : ALU
5    -- Author      : Aaron Lin and Hang Chen
6    -- Company     : Stony Brook University
7    --
8    ------------------------------------------------------------------------
     -----
9    --
10   -- File        : C:\my_designs\ALU\ALU\src\R4_6_tb.vhd
11   -- Generated   : Thu Apr 23 08:54:40 2020
12   -- From        : interface description file
13   -- By          : Itf2Vhdl ver. 1.22
14   --
15   ------------------------------------------------------------------------
     -----
16   --
17   -- Description :  Testbench for the R4 110 Instruction. Checks if
     overflows and underflows are saturated properly, and checks if the normal
     case
18   --      is calculated correctly. Each of the 2 64 bit fields are tested.
19   --
20   ------------------------------------------------------------------------
     -----
21
22   library work;
23   use work.all;
24   library IEEE;
25   use IEEE.std_logic_1164.all;
26   use IEEE.NUMERIC_STD.all;
27
28   entity R4_6_tb is
29   end R4_6_tb;
30
31   --}} End of automatically maintained section
32
33   architecture R4_6_tb of R4_6_tb is
34   signal ALUop : std_logic_vector(19 downto 0);
35   signal Res, regA_Data, regB_Data, regC_Data : std_logic_vector(127 downto
     0);
36   constant period : time := 20ns;
37   begin
38
39        uut : entity ALU
40           port map (ALUop => ALUop, Res => Res, regA_Data => regA_Data,
     regB_Data => regB_Data, regC_Data => regC_Data);
41
42       tb : process
43       begin
44           wait for period/2;
45
46           ALUop <= "10110XXXXXXXXXXXXXX";
47
48           --Right most 64 bits
49           --Testing max negative - (-1 * -1)
```

```
50          regB_Data <= (63 downto 0 => '1', others => '0');
51          regC_Data <= (63 downto 0 => '1', others => '0');
52          regA_Data <= (63 => '1', others => '0');
53          wait for period;
54
55          --Testing max positive - (1 * -1)
56          regB_Data <= (63 downto 0 => '1', others => '0');
57          regC_Data <= (0 => '1', others => '0');
58          regA_Data <= (62 downto 0 => '1', others => '0');
59          wait for period;
60
61          --Testing (3 * 3) + 0
62          regB_Data <= (1 downto 0 => '1', others => '0');
63          regC_Data <= (1 downto 0 => '1', others => '0');
64          regA_Data <= (others => '0');
65          wait for period;
66
67          --Testing (-3 * 3) + 0
68          regB_Data <= (1 downto 0 => '1', others => '0');
69          regC_Data <= (31 downto 2 => '1', 1 => '0', 0 => '1', others =>
    '0');
70          regA_Data <= (others => '0');
71          wait for period;
72
73          --Left most 64 bits
74          --Testing max negative - (-1 * -1)
75          regB_Data <= (127 downto 64 => '1', others => '0');
76          regC_Data <= (127 downto 64 => '1', others => '0');
77          regA_Data <= (127 => '1', others => '0');
78          wait for period;
79
80          --Testing max positive - (1 * -1)
81          regB_Data <= (127 downto 64 => '1', others => '0');
82          regC_Data <= (64 => '1', others => '0');
83          regA_Data <= (126 downto 64 => '1', others => '0');
84          wait for period;
85
86          --Testing (3 * 3) + 0
87          regB_Data <= (65 downto 64 => '1', others => '0');
88          regC_Data <= (65 downto 64 => '1', others => '0');
89          regA_Data <= (others => '0');
90          wait for period;
91
92          --Testing (-3 * 3) + 0
93          regB_Data <= (65 downto 64 => '1', others => '0');
94          regC_Data <= (127 downto 66 => '1', 65 => '0', 64 => '1', others
    => '0');
95          regA_Data <= (others => '0');
96          wait for period;
97
98          wait;
99      end process;
100
101 end R4_6_tb;
102
```