David Elliott

Evan Yeh

EE125 Homework #2

<u>Problem 1: Multiple Detector</u>

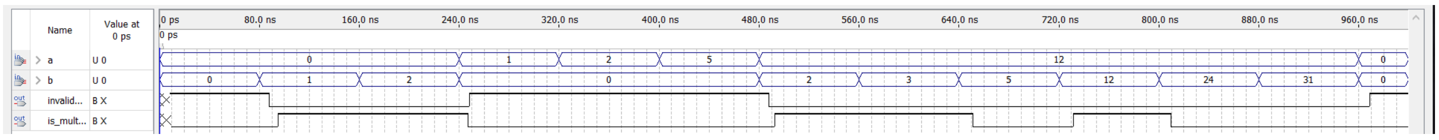
VHDL Code:

```vhdl
------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
------------------------------------------------------------------------
entity multiple_detector is
    generic (
        NUM_BITS: natural := 5);
    port (
        a,b: in std_logic_vector(NUM_BITS-1 downto 0);
        is_multiple: out boolean;
        invalid_inp: out boolean);
end entity;
------------------------------------------------------------------------
architecture multiple_detector of multiple_detector is
    -- internal unsigned operands
    signal a_unsig, b_unsig: unsigned(NUM_BITS-1 downto 0);
begin
    a_unsig <= unsigned(a);
    b_unsig <= unsigned(b);

    is_multiple <= (a_unsig mod b_unsig = 0) and (b_unsig /= 0);
    invalid_inp <= b_unsig = 0;

end architecture;
```

Simulation Results:

Tests cases:

- a = 0 : 0 is a multiple of all numbers (except 0)
- b = 0 : No number can be a multiple of 0
- a > b
- a < b
- random test cases



Problem 2: Absolute difference calculator

VHDL Code:

absolute_difference.vhd

```vhdl
--------------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;
use work.user_types.all;
--------------------------------------------------------------------------------
entity absolute_difference is
    generic (
        SIZE: natural := 6;
        BITS: natural := 5;
        EXTRABITS: natural := 3); --integer(ceil(log2(SIZE))));
    port (
        -- a, b: in slv_array(0 to SIZE - 1)(BITS - 1 downto 0);
        -- for simulation:
        a0, a1, a2, a3, a4, a5, b0, b1, b2, b3, b4, b5: in
            std_logic_vector(BITS - 1 downto 0);
        final_diff: out std_logic_vector(BITS + EXTRABITS - 1 downto 0));
end entity;
--------------------------------------------------------------------------------
architecture generic_chain of absolute_difference is
    type unsigned_array is array (natural range <>) of unsigned;
    -- internal array of differences
    signal abs_diffs: unsigned_array(0 to SIZE - 1)(BITS + EXTRABITS - 1 downto 0);
    -- cumulative sum of differences
    signal sums: unsigned_array(0 to SIZE - 1)(BITS + EXTRABITS - 1 downto 0);
    -- for simulation, input arrays
    signal a : slv_array(0 to SIZE - 1)(BITS - 1 downto 0) := (a0,a1,a2,a3,a4,a5);
    signal b : slv_array(0 to SIZE - 1)(BITS - 1 downto 0) := (b0,b1,b2,b3,b4,b5);
begin
    -- initializing first element of internal arrays
    abs_diffs(0) <= unsigned(abs(resize(signed(a(0)), BITS + EXTRABITS) -
                              resize(signed(b(0)), BITS + EXTRABITS)));
    sums(0) <= unsigned(abs_diffs(0));

    -- calculate each difference and cumulative sum
    gen: for i in 1 to SIZE - 1 generate
        abs_diffs(i) <= unsigned(abs(resize(signed(a(i)), BITS + EXTRABITS) -
                                  resize(signed(b(i)), BITS + EXTRABITS)));
        sums(i) <= sums(i-1) + unsigned(abs_diffs(i));
    end generate;

    -- final difference is last element of cumulative sum array
    final_diff <= std_logic_vector(sums(SIZE - 1));

end architecture;
```

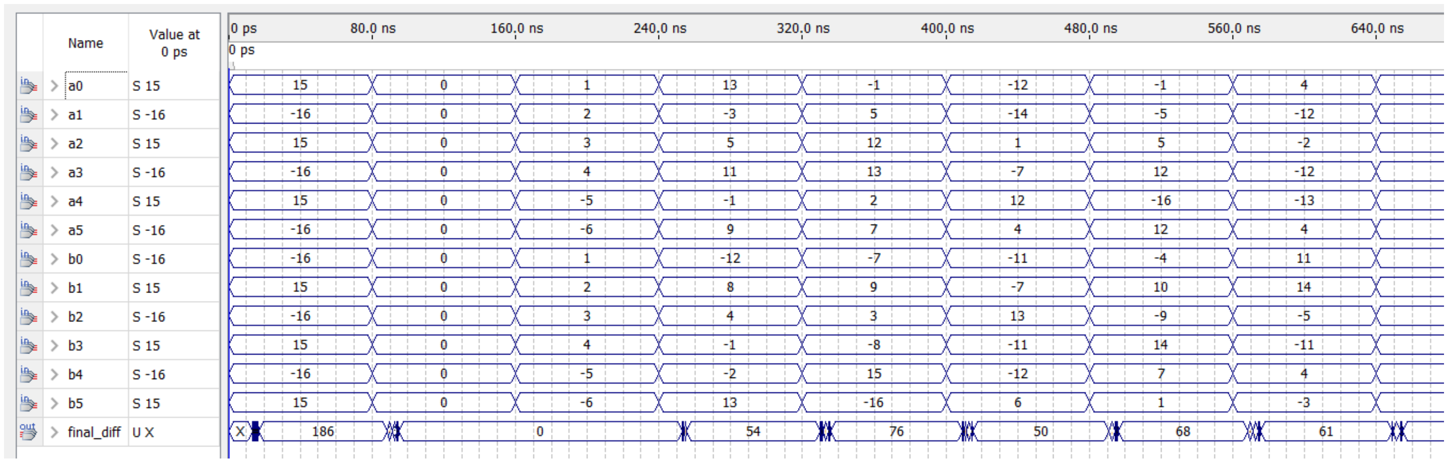
user_types.vhd

```vhdl
-----------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;

package user_types is
    type slv_array is array (natural range <>) of std_logic_vector;
end package;
```

Simulation Results:

Test cases:

- maximum absolute difference
- zero absolute difference and zero input
- zero absolute difference
- random test cases



Name	Value at 0 ps	0 ps	80.0 ns	160.0 ns	240.0 ns	320.0 ns	400.0 ns	480.0 ns	560.0 ns	640.0 ns
a0	S 15	15	0	1	13	-1	-12	-1	4	
a1	S -16	-16	0	2	-3	5	-14	-5	-12	
a2	S 15	15	0	3	5	12	1	5	-2	
a3	S -16	-16	0	4	11	13	-7	12	-12	
a4	S 15	15	0	-5	-1	2	12	-16	-13	
a5	S -16	-16	0	-6	9	7	4	12	4	
b0	S -16	-16	0	1	-12	-7	-11	-4	11	
b1	S 15	15	0	2	8	9	-7	10	14	
b2	S -16	-16	0	3	4	3	13	-9	-5	
b3	S 15	15	0	4	-1	-8	-11	14	-11	
b4	S -16	-16	0	-5	-2	15	-12	7	4	
b5	S 15	15	0	-6	13	-16	6	1	-3	
final_diff	U X	X 186	0		54	76	50	68	61	

Problem 3: Leading-ones counter

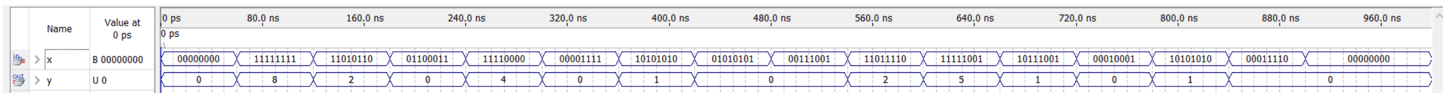
VHDL Code:

```vhdl
--------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
use IEEE.math_real.all;
--------------------------------------------------------------------------
ENTITY leading_ones IS
    GENERIC (N: INTEGER :=8 );  --number of input bits
    PORT (x: IN STD_LOGIC_VECTOR(N-1 DOWNTO 0);
            ssd: OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
            y: buffer integer);
END ENTITY;
--------------------------------------------------------------------------
ARCHITECTURE leading_ones OF leading_ones IS
    --define type int_vector to store a vector of integers
    type int_vector is array(N-1 downto 0) of integer;
    --first_ones is a copy of the input x, but with only the leading ones set
    signal first_ones: std_logic_vector(N-1 downto 0);
    signal ones_count: int_vector;  --vector for summing up the ones
BEGIN
    first_ones(N-1)<= x(N-1);
    ones_count(N-1)<=1 when x(N-1)='1' else 0;
    gen: for i in N-2 downto 0 generate
        first_ones(i)<=first_ones(i+1) and x(i);    --cut off ones when hit zero
        --increment sum if this element is one
        ones_count(i)<=ones_count(i+1) + 1 when(first_ones(i)='1') else
                      ones_count(i+1);
    end generate;
    y<=ones_count(0);   --the final sum is the last (0th) bit in the ones_count
                        --vector
    --Encode into ssd bits
    with y select
        ssd <=  "0000001" when 0,
                "1001111" when 1,
                "0010010" when 2,
                "0000110" when 3,
                "1001100" when 4,
                "0100100" when 5,
                "0100000" when 6,
                "0001111" when 7,
                "0000000" when 8,
                "0000100" when 9,
                "0110000" when others;
END ARCHITECTURE;
--------------------------------------------------------------------------
```

Simulation Results:

Test cases:

- given cases
- random cases



```
Name   Value at 0 ps   0 ps      80.0 ns   160.0 ns  240.0 ns  320.0 ns  400.0 ns  480.0 ns  560.0 ns  640.0 ns  720.0 ns  800.0 ns  880.0 ns  960.0 ns
x      B 00000000      00000000  11111111  11010110  01100011  11110000  00001111  10101010  01010101  00111001  11011110  11111001  10111001  00010001  10101010  00011110  00000000
y      U 0             0         8         2         0         4         0         1         0                   2         5         1         0         1         0
```