**Assignment 06**

Assigned: 03/19/15 Due: 04/12/15

1a. Show how to perform the operation (23 / 9) using the optimized division algorithm. Assume the operands are 6b instead of 32b (be sure that you adjust all of the registers appropriately). You must show each step, from initialization to the final product. ***(8)***

Remainder Quotient Divisor

|  |  |  |
| --- | --- | --- |
| 000000 010111 | 000000 | 001001 000000 |
| 110111 010111 | 000000 | 001001 000000 |
| 000000 010111 | 000000 | 000100 100000 |
| 111011 101011 | 000000 | 000100 100000 |
| 000000 010111 | 000000 | 000010 010000 |
| 111110 000111 | 000000 | 000010 010000 |
| 000000 010111 | 000000 | 000001 001000 |
| 111111 001111 | 000000 | 000000 100100 |
| 000000 010111 | 000000 | 000000 100100 |
| 111111 110011 | 000000 | 000000 100100 |
| 000000 010111 | 000000 | 000000 010010 |
| 000000 000101 | 000001 | 000000 001001 |
| 111111 111100 | 000010 | 000000 000100 |
| 000000 000101 | 000010 | 000000 000100 |

2a. What is the entity name in the design file? ***(1) or\_gate***

2b. What is the name of the architecture in the design file? ***(1) rtl***

2c. What is the name of the entity in the testbench file? ***(1) testbench***

2d. What is the name of the architecture in the testbench file? ***(1) tb***

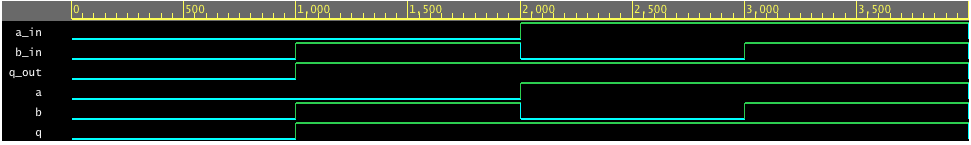
3a. What does an ‘X’ represent when using the std\_logic type? ***(1)***

***- string drive, unknown logic type***

3b. What does a ‘Z’ represent? ***(1)***

***- high impedance***

4a. Based on the VHDL in the source file, is *q\_out* what you would expect it to be? Paste a screenshot of your timing diagram below (note: do not paste the entire page, crop it so that only the relevant information is shown). ***(4)***

* ***Yes. Once when ‘a’ and ‘b’ started to switch back and forth between ‘0’ and ‘1’, ‘q’ remained constant. Even though for a short amount of time I felt like there would show a glitch in the timing diagram, I believe the wait time resolved that issue.***
* ******

5a. Run the simulation again. Is the timing diagram the same or different? ***(1)***

***-The same.***

5b. Based on the differences between sequential and concurrent operations in VHDL, is this what you would expect the behavior to be? ***(1)***

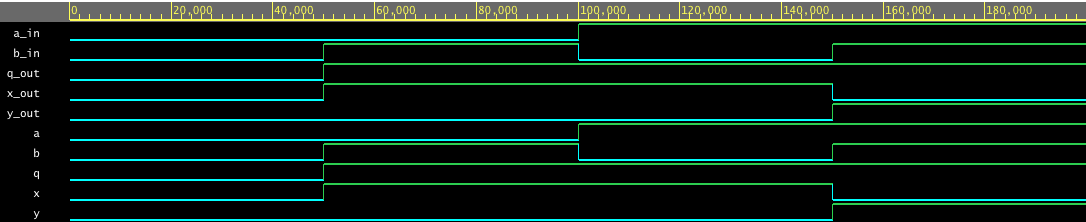
***- Assignment made inside a process are resolved at the conclusion of the current iteration of that process in sequential code in which the timing diagram explains exactly that. In sequential code, only the final assignment is valid.***

6a. Look at the code beneath the PROCESS statement. What is this going to do? ***(1)***

***- The process statement is going to count down 1 to 0, or 0 to 1 while the current count is less than the binary value “11”. After each iteration, the count increases by 1.***

7a. Are the results what you would expect? Paste a copy of your timing diagram below. ***(4***

***-Yes. When ‘a’ AND ‘b’ are both high, ‘y’ is also high. When only ‘a’ is high or only ‘b’ is high, ‘x’ is also high.***

******

7b. ------------------------------------------------------------------

-- author: Drew Smith

-- class: CDA 3100

-- id: dbs13

------------------------------------------------------------------

library IEEE;

use IEEE.std\_logic\_1164.all;

entity or\_gate is

port(

a: in std\_logic;

b: in std\_logic;

q: out std\_logic;

x: out std\_logic;

y: out std\_logic);

end or\_gate;

architecture rtl of or\_gate is

begin

q <= a or b;

x <= a XOR b;

y <= a AND b;

end rtl;

8a. What is a std\_logic\_vector? ***(1)***

***-Is used for arrays of type std\_logic variables and signals.***

b. Give an example on how to declare a standard logic vector that can hold 32 elements. ***(1)***

***- SIGNAL myvectSig: STD\_LOGIC\_VECTOR (1 to 32);***

c. Show how you would assign element 11 of that vector to a signal of type std\_logic, *a*. ***(1)***

- ***SIGNAL myvectSig: STD\_LOGIC\_VECTOR (1 to 32) := (11 => ‘a’, OTHERS => ‘0’);***

9a. What does the OTHERS keyword do? ***(1)***

***-You can use the keyword OTHERS to assign a value to variables, etc, that you initinally did not sign a value previously.***

b. What is the variable slt\_temp set to when (a < b) is true? ***(1)***

***- 0001***

c. What is the variable slt\_temp set to when (a < b) is false? ***(1)***

***- 0000***

d. Fill in the missing sections of VHDL to complete the ALU. Upload a copy of your solution to Blackboard and paste a copy of the source code here. Be sure that your solution has a header at the top that contains, at minimum, your name and your Blackboard user name. ***(8)***

***-*** ***-------------------------------------------------------------------------***

***-- @file assignment.06.alu.vhd***

***-- @author DREW SMITH***

***-- @date 4/12/2015***

***-- @brief file solution assignment 6, CDA3100***

***-------------------------------------------------------------------------***

***LIBRARY ieee;***

***USE ieee.std\_logic\_1164.all;***

***USE ieee.numeric\_std.all;***

***-----------------------------------------***

***ENTITY mips\_alu IS***

***GENERIC (N: INTEGER := 4);***

***PORT***

***(***

***alu\_ctrl: IN STD\_ULOGIC\_VECTOR(3 downto 0);***

***a, b: IN STD\_ULOGIC\_VECTOR(N-1 downto 0);***

***result: OUT STD\_ULOGIC\_VECTOR(N-1 downto 0);***

***zero: OUT STD\_ULOGIC***

***);***

***END ENTITY;***

***-----------------------------------------***

***ARCHITECTURE comb\_circuit OF mips\_alu IS***

***SIGNAL a\_temp: UNSIGNED(N-1 downto 0);***

***SIGNAL b\_temp: UNSIGNED(N-1 downto 0);***

***SIGNAL r\_temp: UNSIGNED(N-1 downto 0);***

***BEGIN***

***a\_temp <= unsigned(a);***

***b\_temp <= unsigned(b);***

***alu : process(alu\_ctrl, a\_temp, b\_temp)***

***variable slt\_temp: UNSIGNED(N-1 downto 0);***

***begin***

***-- tests to see if a is less than b***

***if( a\_temp < b\_temp ) then***

***slt\_temp := (0 => '1', OTHERS => '0');***

***else***

***slt\_temp := (OTHERS => '0');***

***end if;***

***case alu\_ctrl is***

***when "000" =>***

***slt\_temp := a AND b;***

***when "001" =>***

***slt\_temp := a OR b;***

***when "010" =>***

***slt\_temp := a + b;***

***when "011" =>***

***slt\_temp := a - b;***

***when "100" =>***

***slt\_temp := a < b;***

***when "101" =>***

***slt\_temp := a NOR b;***

***when OTHERS =>***

***slt\_temp := 'Z';***

***if slt\_temp = "000" then***

***zero <= '1';***

***else***

***zero <= '0';***

***end if;***

***r\_temp <= slt\_temp;***

***end process alu;***

***result <= std\_ulogic\_vector(r\_temp);***

***END comb\_circuit;***

10. Are the errors reported for *add* and *sub* for all combinations? Explain. ***(3)***

***-No, only when overflow occurs.***