EEL 4712
Midterm 2 – Spring 2010
VERSION 1

Name: _____

UFID: _____

Sign your name here if you would like for your test to be returned in class:

_____

IMPORTANT:
• Please be neat and write (or draw) carefully. If we cannot read it with a reasonable effort, it is assumed wrong.
• **As always, the best answer gets the most points.**

COVER SHEET:

Problem#:	Points
1 (6 points)	
2 (6 points)	
3 (8 points)	
4 (10 points)	
5 (12 points)	
6 (6 points)	
7 (6 points)	
8 (12 points)	
9 (30 points)	
10 (4 points)	

Total:

Regrade Info:

```
ENTITY _entity_name IS
PORT(__input_name, __input_name : IN STD_LOGIC;
__input_vector_name : IN STD_LOGIC_VECTOR(__high downto __low);
__bidir_name, __bidir_name : INOUT STD_LOGIC;
__output_name, __output_name : OUT STD_LOGIC);
END __entity_name;

ARCHITECTURE a OF __entity_name IS
SIGNAL __signal_name : STD_LOGIC;
BEGIN
-- Process Statement
-- Concurrent Signal Assignment
-- Conditional Signal Assignment
-- Selected Signal Assignment
-- Component Instantiation Statement
END a;

__instance_name: __component_name PORT MAP (__component_port => __connect_port,
__component_port => __connect_port);

WITH __expression SELECT
__signal <= __expression WHEN __constant_value,
__expression WHEN __constant_value,
__expression WHEN __constant_value,
__expression WHEN __constant_value;
__signal <= __expression WHEN __boolean_expression ELSE
__expression WHEN __boolean_expression ELSE
__expression;

IF __expression THEN
__statement;
__statement;
ELSIF __expression THEN
__statement;
__statement;
ELSE
__statement;
__statement;
END IF;

CASE __expression IS
WHEN __constant_value =>
__statement;
__statement;
WHEN __constant_value =>
__statement;
__statement;
WHEN OTHERS =>
__statement;
__statement;
END CASE;

<generate_label>: FOR <loop_id> IN <range> GENERATE
-- Concurrent Statement(s)
END GENERATE;


type __identifier is type_definition;

subtype  __identifier is subtype_indication;
```

1) (6 points) For the memory entity given below, which of the answers best describes the memory structure that is inferred during synthesis?

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity ram is
  port (clk        : in  std_logic;
        we         : in  std_logic;
        addr1      : in  std_logic_vector(4 downto 0);
        addr2      : in  std_logic_vector(4 downto 0);
        data_in    : in  std_logic_vector(3 downto 0);
        read_data1 : out std_logic_vector(3 downto 0);
        read_data2 : out std_logic_vector(3 downto 0));
end ram;

architecture syn of ram is
  type ram_type is array (31 downto 0)
    of std_logic_vector (3 downto 0);
  signal RAM : ram_type;
begin
  process (clk)
  begin
    if (clk'event and clk = '1') then
      if (we = '1') then
        RAM(conv_integer(addr1)) <= di;
      end if;
    end if;
  end process;

  read_data1 <= RAM(conv_integer(addr1));
  read_data2 <= RAM(conv_integer(addr2));

end syn;
```

a) Single-port memory, synchronous writes, synchronous reads
b) Dual-port memory, synchronous writes, synchronous reads
c) Dual-port memory, synchronous writes, asynchronous reads
d) Dual-port memory, asynchronous writes, asynchronous reads

2) (6 points) What is the *minimum* number of adders and multipliers that are needed to create a datapath for the following pseudo-code, assuming that an appropriate controller exists?

```
for (i=0; i < 100; i++) {
    result = a[i]*b[i]*c[i]*d[i] + e[i]*16 + f[i]*8 + result;
}
```

3) a. (2 points) Write a VHDL type declaration called MY_ARRAY that creates a 2D array with 8 rows and 4 columns, where each element is a 32-bit std_logic_vector.

b. (2 points) Write a VHDL type declaration called MY_ARRAY that creates a 2D array with unconstrained ranges for each dimension, where each element is a 32-bit std_logic_vector.

c. (2 points) Using the type from part b, instantiate an object of type MY_ARRAY with 50 rows and 100 columns.

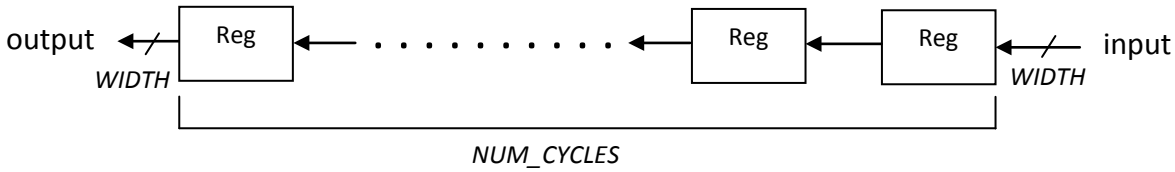
d. (2 points) What doesn't VHDL allow in the following type declaration?

```
type MY_ARRAY is array (natural range<>, natural range<>) of std_logic_vector
```

4) a. (5 points) Briefly explain the purpose of the horizon sync (h_sync) and vertical sync (v_sync) signals in the VGA lab.

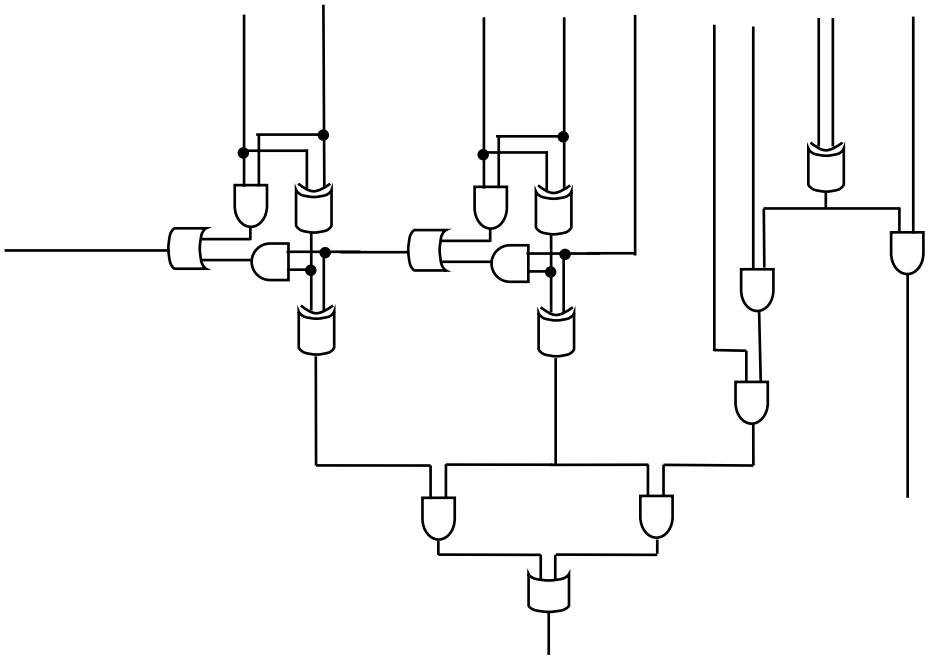
b. (5 points) Why are the color signals turned off at certain times during the drawing of the screen? When does this occur? What is the name of these intervals where the color is off?

5) (12 points) Fill in the code provided below to create a series of delay registers with generic width and number of delay cycles. *You must use a structural architecture with the provided generate loop* that connects together the register (reg) components. The circuit should look like this:




```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity delay is
  generic(num_cycles :     positive;
          width      :     positive);
  port( clk          : in  std_logic;
        rst          : in  std_logic;
        input        : in  std_logic_vector(width-1 downto 0);
        output       : out std_logic_vector(width-1 downto 0));
end delay;

architecture str of delay is

  type ARRAY_TYPE is array (0 to num_cycles) of
                              std_logic_vector(width-1 downto 0);

  component reg
    generic (width :     positive  := 32);
    port(clk       : in  std_logic;
         rst       : in  std_logic;
         input     : in  std_logic_vector(width-1 downto 0);
         output    : out std_logic_vector(width-1 downto 0));
  end component;

  signal reg_val : ARRAY_TYPE;

begin




  U_DELAY : for i in 0 to num_cycles-1 generate

    U_REG : reg generic map (width => width)
      port map (clk                => clk,
                rst                => rst,




                        );
  end generate U_DELAY;



end str;
```

6) (6 points) Map the following circuit onto 3-input, 2-output LUTs by drawing shapes around each portion of the circuit that is mapped to an individual LUT.



7) (6 points) What is the maximum number of gates that can be implemented in a 3-input, 2-output LUT?

8) (12 points) Briefly describe the components in an FPGA used for reconfigurable interconnect.

9) A. (12 points) For the following pseudo-code, create a datapath that is capable of performing all necessary behavior. Clearly show all inputs/outputs, multipliers, subtractors, comparators, registers, muxes, wires, and control signals.

```
n = input;
result = 1;
while (n > 0) {
  result = result*n;
  n --;
}
output = result;
```

B. (12 points) For the datapath in the previous problem, draw an FSM capable of controlling the datapath to perform the illustrated code. In the circle for each state of the FSM, show the statements from the code that are performed in that state.

C. (6 points) For each state in your FSM, list the values of the control signals that configure the datapath to perform the appropriate operations. Assume that the left input to a mux uses a select value of '1'. Specify default control values to avoid having to list every control signal for each state.

10) (4 free points) Who will replace Tebow as the starting QB for the football team next year?