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:**

<table>
<thead>
<tr>
<th>Problem#</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (6 points)</td>
<td></td>
</tr>
<tr>
<td>2 (6 points)</td>
<td></td>
</tr>
<tr>
<td>3 (8 points)</td>
<td></td>
</tr>
<tr>
<td>4 (10 points)</td>
<td></td>
</tr>
<tr>
<td>5 (12 points)</td>
<td></td>
</tr>
<tr>
<td>6 (6 points)</td>
<td></td>
</tr>
<tr>
<td>7 (6 points)</td>
<td></td>
</tr>
<tr>
<td>8 (12 points)</td>
<td></td>
</tr>
<tr>
<td>9 (30 points)</td>
<td></td>
</tr>
<tr>
<td>10 (4 points)</td>
<td></td>
</tr>
</tbody>
</table>

**Total:**

**Regrade Info:**
ENTITY _entity_name IS
  PORT (_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?

```vhdl
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?

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

1 adder, 1 multiplier

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.

```plaintext
type MY_ARRAY is array (0 to 7, 0 to 3) of std_logic_vector(31 downto 0);
```

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.

```plaintext
type MY_ARRAY is array (natural range <>, natural range<> ) of std_logic_vector(31 downto 0);
```

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

```plaintext
signal sig : MY_ARRAY (0 to 49, 0 to 99);
```

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

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

VHDL doesn't allow unconstrained arrays of an unconstrained type.
4) a. (5 points) Briefly explain the purpose of the horizon sync (h_sync) and vertical sync (v_sync) signals in the VGA lab.

h_sync synchronizes the refresh of each row of pixels

v_sync synchronizes the refresh of the entire screen (i.e., it causes the monitor to begin refreshing at the top left)

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?

a) to avoid drawing while the electron beam is being moved to the left edge or from the bottom of the screen back to the top.

b) blanking intervals
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

    reg_val(0) <= input;

    U_DELAY : for i in 0 to num_cycles-1 generate
        U_REG : reg generic map (width => width)
            port map (clk => clk,
                      rst => rst,
                      input => reg_val(i),
                      output => reg_val(i+1));
    end generate U_DELAY;

    output <= reg_val(width)
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?

    no maximum

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

    routing tracks - wires between every row and column of CLBs
    connection box - connects CLB I/O to routing tracks
    switch box - connects routing tracks together
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.

\[
\begin{align*}
n &= \text{input}; \\
\text{result} &= 1; \\
\text{while} \ (n > 0) \ {} &\{} \\
\text{result} &= \text{result} \times n; \\
n &= --; \\
\text{output} &= \text{result};
\end{align*}
\]
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.

1) \( \text{result}_\text{-sel} = 0 \)
   \( \text{result}_\text{-id} = 1 \)
   \( n\text{-sel} = 1 \)
   \( n\text{-id} = 1 \)

2) \( \text{nope} \)

3) \( \text{result}_\text{-sel} = 1 \)
   \( \text{result}_\text{-id} = 1 \)
   \( n\text{-sel} = 0 \)
   \( n\text{-id} = 1 \)

4) \( \text{output}_\text{-id} = 1 \)

\( \underline{\text{defaults}} \)

all Id signals = 0

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