EEL 4712
Midterm 2 – Spring 2013
VERSION 1

Name: Solution

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 (18 points)	
2 (8+ 7 extra credit)	
3 (12 points)	
4 (6 points)	
5 (6 points)	
6a (16 points)	
6b (16 points)	
6c (16 points)	

Total:

Regrade Info:

1:00 1:21

```
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) a. (6 points) For a VGA resolution of 800x600 and a pixel clock of 40 MHz, the h_sync pulse takes 3.2 us. Assuming you have a counter that starts counting at the beginning of the h_sync pulse, what count should represent the end of the h_sync pulse? Show your work.

time per count = $\frac{1}{40e6}$ s

pulse = end_count * time_per_count

$3.2e^{-6}$ s = end_count * $\frac{1}{40e6}$ s

end_count = $40e^{6} * 3.2e^{-6}$ = **128**

b. (6 points) The Cyclone III EP3C16 has 56 M9k block RAMs, which each contain 9,000 bits. Show whether or not it is possible to store a 640x480 12-bit image in the available block RAM. Assume the individual block RAMs can be combined to form larger memories. No points will be given unless the work is shown.

total RAM = 56 × 9000 = 504k bits

image size = 640 × 480 × 12 = 3,686,400 bits

not possible

c. (6 points) The ROM used in lab6 had a 1-cycle read latency. Consider a situation where the image is stored in off-chip SRAM that has a read latency of 20 cycles. Explain how the rest of the VGA circuit should be modified to account for this increased latency.

delay all control signals with 20-cycle shift register

2) (15 points) Identify the violations of the *synthesis coding guidelines* in the following 2-process FSM. Be sure to explain the violation and state if the violation is related to sequential or combinational logic. See the comments for the purpose of each region of code.

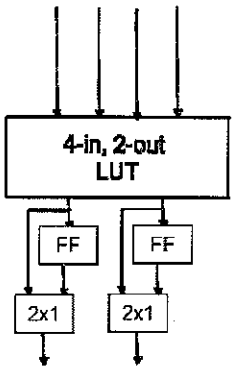
```
architecture PROC2 of FSM is

  type STATE_TYPE is (STATE_0, STATE_1, STATE_2);
  signal state, next_state : STATE_TYPE;

begin

  -- state register
  process(clk, rst)
  begin
    if (stall = '0') then
      if (rst = '1') then
        state <= STATE_0;
      elsif(clk'event and clk = '1') then
        state <= next_state;
      end if;
    end if;
  end process;

  -- next-state combinational logic
  process(en)
  begin

    case state is
      when STATE_0 =>

        output <= "001";

        if (en = '1') then
          next_state <= STATE_1;
        end if;

      when STATE_1 =>

        output <= "010";

        if (en = '1') then
          next_state <= STATE_2;
        end if;

      when STATE_2 =>

        output <= "100";

        if (en = '1') then
          next_state <= STATE_0;
        end if;

      when others => null;
    end case;
  end process;
end PROC2;
```

code outside of "if rst elsif clk ..."

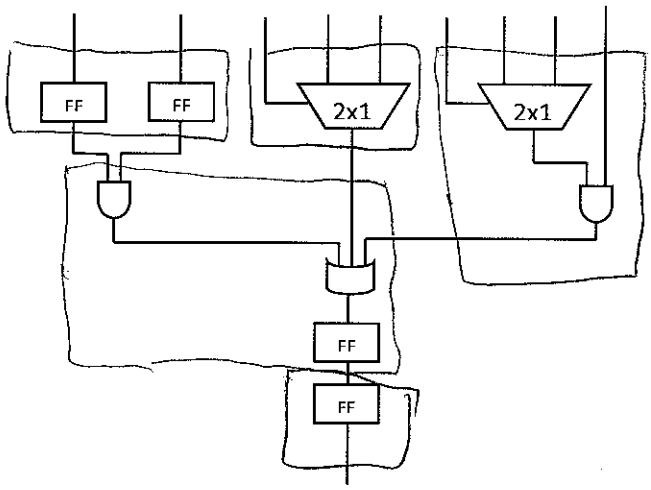
state missing from list

next_state not defined on all paths

3) (12 points) Assume you are given an FPGA that consists of the following CLB structures:



Map the following circuit onto these CLBs by drawing rectangles to represent CLBs. Use the minimum number of CLBs.



4) (6 points) Show the sequence of FPGA resources that establish a connection between two CLBs that are placed far apart on the FPGA.

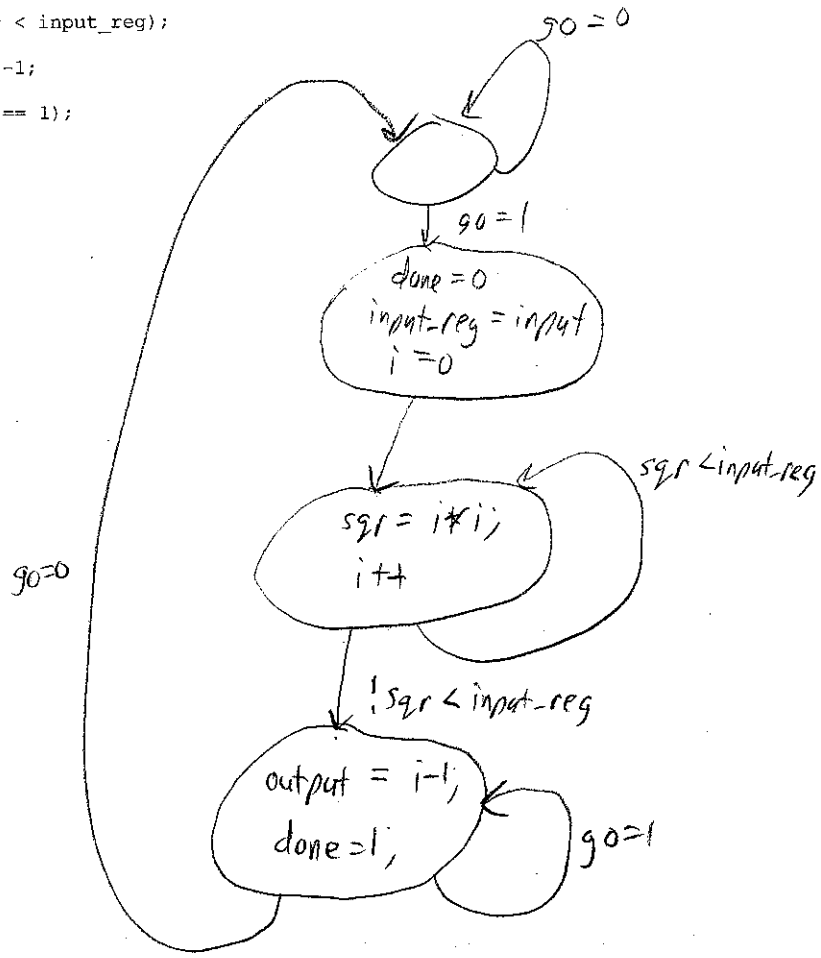
CLB → connection box → routing track → switch box → routing track → connection box → CLB

5) (6 points) Name two resources other than CLBs that commercial FPGAs include in the fabric. Do not list routing resources.

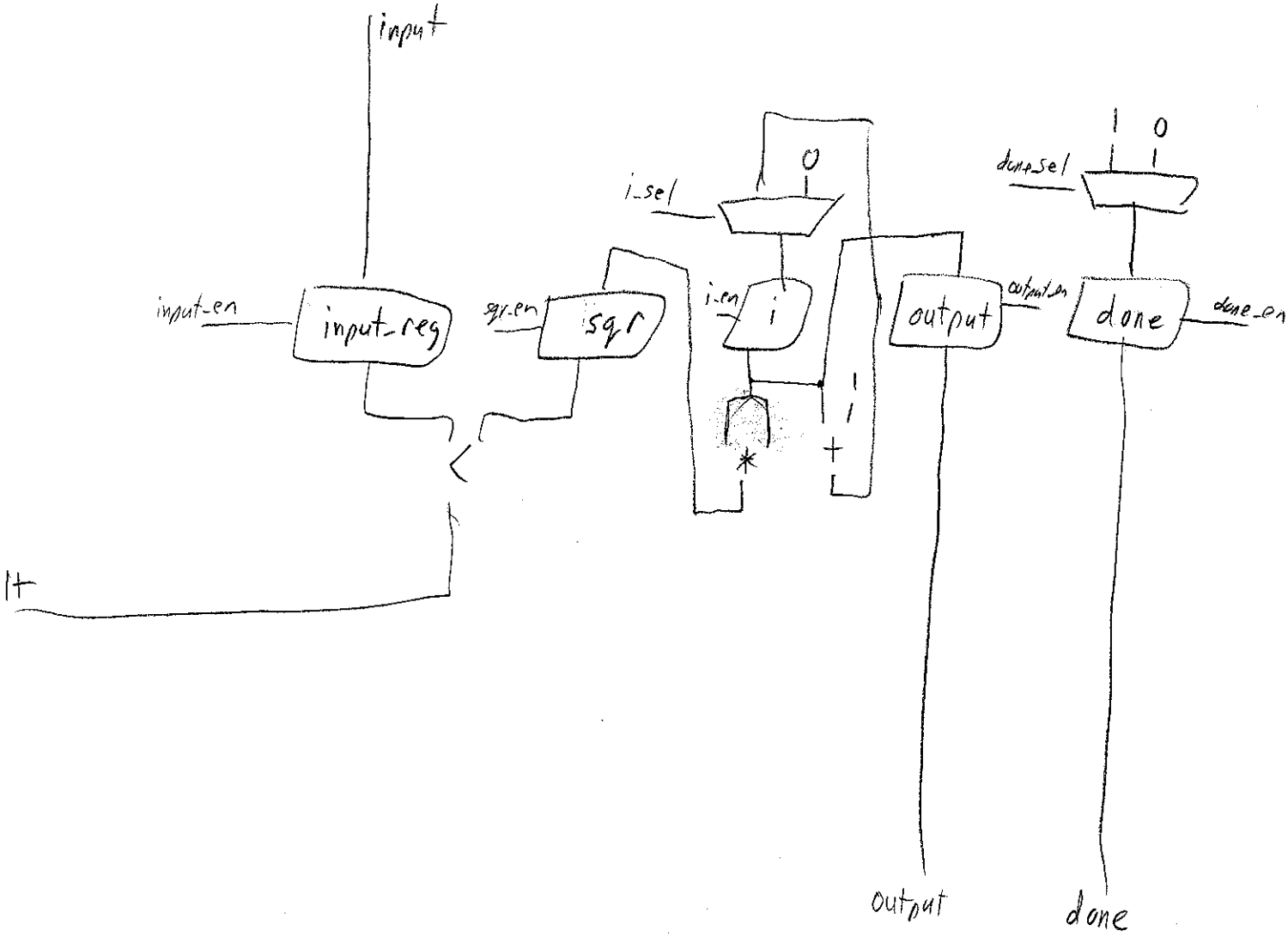
DSP, memory

6) a. (16 points) Create an FSMD that implements the following pseudo-code. **Do not write VHDL and instead leave the FSMD in graphical form** (i.e., state machine with corresponding operations in each state). Make sure to specify all operations and state transitions. Note that *input*, *output*, *go*, and *done* are I/O.

```
Inputs: go, input
Outputs: output, done

done = 0;   // reset values for outputs
output = 0; // reset values for outputs

while (1) {
        while (go == 0);
        done = 0;
        input_reg = input; // Store input in a register.
        i = 0;

        do {
                sqr = i*i;
                i ++;
        }
        while (sqr < input_reg);

        output = i-1;
        done = 1;
        while (go == 1);
}
```



b. (16 points) For the same pseudo-code, create a datapath capable of executing the code (ignore the controller in this step). Make sure to show all control signals (i.e., mux select signals, register load signals, comparator output signals). Make sure to include a register for *input, output,* and *done* in addition to other registers you might need. To make things easier, I don't recommend sharing resources. **Do not write any code, just show the datapath.** If you do any non-obvious optimization, make sure to explain.



c. (16 points) For the datapath in the previous step, draw an FSM capable of controlling the datapath to perform the pseudo-code. In each state of the FSM, show the values of your control signals from the previous step that configure the datapath to do the corresponding operations. Hint: to save yourself time, try to use the same states as the FSMD, and just change the operations to the corresponding control signals. **Do not write any VHDL code, just show the FSM and control signals. Be sure to mention default signal values to save space.**

Assume defaults of '0' for all signals

Assume '1' selects left mux input