ECE 448
Spring 2019
Final Exam (25 points)

First Name:
Last Name:

Problem 1

Task 1:

Analyze the memory map and draw the detailed internal block diagram of an FPro MMIO unit with the following interface and the memory map shown on the second page.

Interface:

```
entity quiz_mmio is
   port(
      clk     : in  std_logic;
      reset   : in  std_logic;
      cs      : in  std_logic;
      write   : in  std_logic;
      read    : in  std_logic;
      addr    : in  std_logic_vector(4 downto 0);
      rd_data : out std_logic_vector(31 downto 0);
      wr_data : in  std_logic_vector(31 downto 0)
   );
end quiz_mmio;
```

Assumptions:

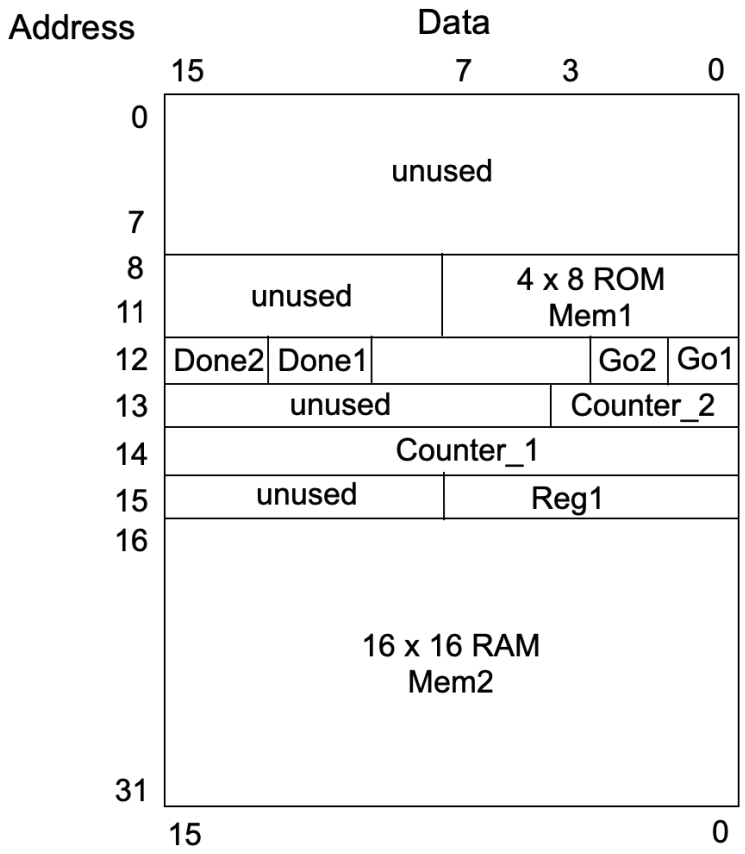
- **Registers must be implemented using actual registers, not memory. The register Reg1 can be written to and read from.**
- **Counter_1 is a 16-bit counter, Counter_2 is a 4-bit counter. Writing to a counter initializes this counter, reading from a counter, reads its current value. Assume that initializing a counter requires active values of inputs en and ld.**
- **Go2 and Go1 are 1-bit write-only flags located at the two least significant bit locations at address 12.**
- **Done2 and Done1 are 1-bit read-only flags located at the two most significant bit locations (15 and 14) at address 12.**

Task 2: Attempt only after finishing Task 1

Extend your circuit with the following additional internal functionality. Clearly mark with * any additional components you are adding to your circuit to realize this functionality.

- **When Go1 is equal to 1, the 16-bit Counter_1 should count down every clock cycle. When it reaches 0, the next value should be 2^{16}-1 (all ones).**
- **When Go2 is equal to 1, the 4-bit Counter_2 should count up every clock cycle. When it reaches 15 (all ones), the next value should be 0.**
- **Initializations of counters have priority over counting.**
- **When Go2 is equal to 1, the following operation should be performed: Mem2[Counter_2]=Counter_1, where Mem2 is the 16x16 RAM shown in the memory map. This operation has a priority over an external write to Mem2.**

Memory Map:



Solution to Problem 1:

Analysis of Memory Map (recommended):

Block diagram (if you need more space use the back side of this page, or ask for an extra blank page):

Problem 2

Draw a block diagram of the circuit performing the computations of the Julia Fractal algorithm, based on the pseudocode given below. Assume that you can use the MUL, ADD, and SUB units performing the multiplication, addition, and subtraction of fixed-point numbers, respectively. These units accept inputs and produce outputs in the Q4.28 representation, i.e., numbers with 4 bits in the integer part and 28 bits in the fractional part.

Color_table is an internal RAM of the size equal to the smallest power of 2 greater than or equal to 640*480.

When Go=0, the external circuit can initialize values of internal registers cx and cy (storing numbers in the Q4.28 representation), using ports DataIn, WrInit, XYchoice. When Go=1, the circuit performs computations. When Go=0 again, the external circuit can read the contents of internal memory color_table using ports Addr, Read, ColorOut.

Please clearly mark widths of all buses in your circuit.

Pseudocode:

```
begin:
wait for Go=1
zy0 = -1.5
zx0 = -2
for i = 0 to 479 do
  for j = 0 to 639 do
     zx = zx0
     zy = zy0
     iteration = 0
     limit = 0
     while( (limit < 4) and (iteration < 1000)) do
     {
      zxtemp = zx*zx - zy*zy + zx + cx
      zytemp = 2*zx*zy + zy + cy
      limit = zx*zx + zy*zy
      zx = zxtemp
      zy = zytemp
      iteration++
     }
     if (limit < 4)
       color = 1
     else
       color = 0
     endif
     color_table[640*i+j] = color
     zx0 = zx0 + 1/160
  endfor
  zy0 = zy0 + 1/160
endfor
Done=1
wait for Go=0
go to begin
```

Interface:

Assume the following interface to your circuit:

Port	Width	Meaning
Clk	1	System clock.
Reset	1	System reset – clears all internal registers and counters. Active high.
Go	1	Operating mode: 0 = waiting for data/reading results, 1 = processing.
DataIn	32	Input data bus.
WrInit	1	Synchronous write control signal
XYchoice	1	Selection between initializing cx and cy. 0 – initializing cx, 1 – initializing cy.
Addr	Please determine by yourself	Address of a memory location in color_table.
Read	1	Read enable. 0 = high impedance on the output bus ColorOut, 1 = valid output on the output bus ColorOut.
ColorOut	1	Output data bus used to read results. If Read = 1, ColorOut = color_table[Addr], otherwise ColorOut = 'Z' (high-impedance).
Done	1	Asserted when Go=1 and all computations are completed, zero otherwise.

Solution to Problem 2:

Analysis of variables (recommended):

Block diagram (if you need more space use the back side of this page, or ask for an extra blank page):

Problem 3

Perform the following three tasks:

A. **Draw in interface of the Julia Fractal unit, designed as a part of Problem 2, with the division into the Datapath and Controller.**
B. **Draw an ASM chart describing the Controller of the Julia Fractal unit.**
   Use actions and expressions corresponding (as much as practical) to the operations and conditions of the pseudocode.
C. **Draw a second version of the same ASM chart, expressing**
   a. **operations in terms of active values of control signals generated by the Controller.**
   b. **conditions in terms of values of status signals generated by the Datapath.**

<u>Solutions to Problem 3:</u>

Part A: Interface with the division into the Datapath and Controller:

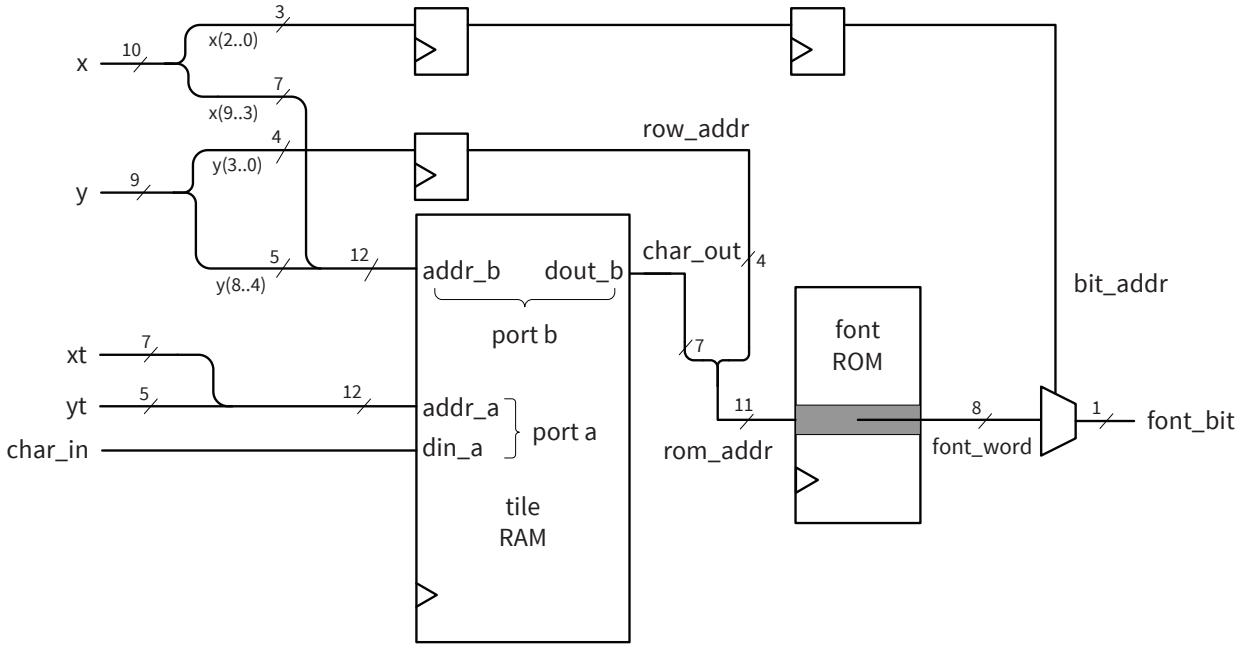
Part B: ASM chart based on the pseudocode

Part C: ASM chart based on active values of control and status signals

Problem 4

Based on the given below block diagram of the Text Generation Circuit with Tile Memory, please provide the following information about the <u>tile RAM</u>:

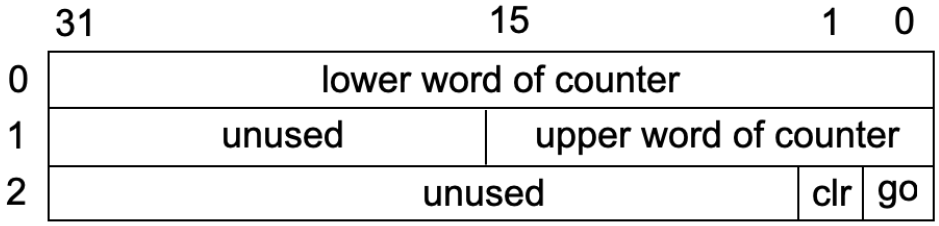
A. **Type of memory (single-port or dual-port):** …….
B. **Total number of words (number of locations in memory):** …….
C. **Size of each word (a single location in memory):** …….
D. **Total size of memory in kbits:** …….
E. **Number of words used to hold ASCII codes of tiles:** …….
F. **Utilization of memory in % (E/B*100%)** …….
G. **Address of the location holding ASCII code of the tile with coordinates (x=10, y=16):** …….



How would your answers to questions A.-G. changed if an additional control bit, used to conditionally reverse the foreground and background color of every character of the VGA screen is stored in the tile RAM? Which answers would remain the same, and which would change?

Problem 5

Fill in the blanks in the code of the FPro MMIO Timer unit, described by the following memory map:




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

entity chu_timer is
   port(
      clk     : in  std_logic;
      reset   : in  std_logic;
      cs      : in  std_logic;
      write   : in  std_logic;
      read    : in  std_logic;
      addr    : in  std_logic_vector(..... downto 0);
      rd_data : out std_logic_vector(..... downto 0);
      wr_data : in  std_logic_vector(..... downto 0)
   );
end chu_timer;

architecture arch of chu_timer is
   signal count_reg  : unsigned(..... downto 0);
   signal count_next : unsigned(..... downto 0);
   signal ctrl_reg   : std_logic;
   signal wr_en      : std_logic;
   signal clear, go  : std_logic;
begin
   process(clk, reset)
   begin
      if reset = '1' then
         count_reg <= .....;
```

```
      elsif (clk'event and clk = '1') then
         count_reg <= ……
      end if;
   end process;
   -- next-state logic
   count_next <= …… when clear = '1' else
                 …… when go = '1' else
                 ……;
   process(clk, reset)
   begin
      if reset = '1' then
         ctrl_reg <= '0';
      elsif (clk'event and clk = '1') then
         if wr_en = '1' then
            ctrl_reg <= ……;
         end if;
      end if;
   end process;
   -- decoding logic
   wr_en <= '1' when write=…… and cs=…… and ……
          else '0';
   clear <= '1' when wr_en=…… and …… else '0';
   go    <= ……;
   rd_data <=
      std_logic_vector(count_reg(31 downto 0)) when ……
      else …… when ……
      else ……;
end arch;
```