ECE 590 Midterm Exam

Name: NetID:

There are 6 questions, with the point values as shown below. You have 75 minutes with a total of 75 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. You may use 1 sheet of notes you created, but no other external resources.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

#	Question	Points Earned	Points Possible
1	Combinatorial Logic		15
2	Sequential Logic		10
3	FSMs		10
4	Asm Programming		20
5	Datapaths		10
6	Memory Hierarchy		10
	Extra Credit		
	Total		75
	Percent		100

Question 1: Combinatorial Logic [15 pts]

Given the following truth-table:

a	b	c	x
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. Write the sum-of-product formula

2. Simplify the formula

3. Write VHDL that implements the formula

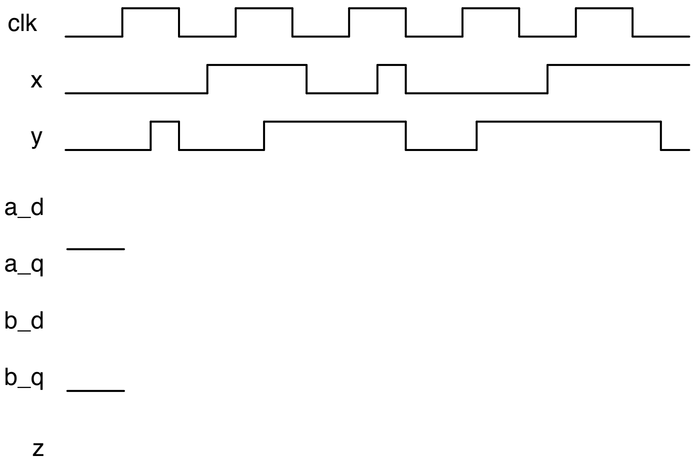
```
entity q1 is
  port (
    a : in  std_logic;
    b : in  std_logic;
    c : in  std_logic;
    x : out std_logic);
end q1;
architecture basic of q1 is
begin


end basic;
```

Question 2: Sequential Logic [10 pts]

Consider the following VHDL fragment, in which x, and y are inputs, z is an output, and there are two DFFs (a and b) whose d inputs are a_d and b_d respectively, and whose q outputs are a_q and b_q respectively:

```
b_d <= (a_q and y)  or x;
a_d <= (b_q xor a_q) xor y;
z   <= not (a_q xor b_q);
```

Complete the waveform below (assume that the DFFs are triggered by the rising edge of clk): Note that a_q is initially 1, and b_q is initially 0:



Question 3: FSMs [10 pts]

Draw a state machine diagram for a finite state machine which accepts a single bit input (either 0 or 1—you can just label each edge with 0 or 1). This state machine also has a single bit of output, which is initally 0.

- Whenever the FSM receives an input sequence of 1001 or 1100 the output bit goes to 1.
- The output remains at 1 until the FSM receives an input sequence of 110.
- Once the FSM receives the input sequence of 110, the output bit returns to 0.
- The output now remains 0 until 1001 or 1100 are input again, in which case the output bit goes to 1 and the process repeats.

Label each state with the bit it outputs. Be sure to indicate your start state (with an arrow to it from nowhere).

Question 4: Asm Programming [20 pts]

Write MIPS assembly for the following C function.

```
int someFun(int * ptr, int n) {
  int i = 0;
  int x = 2;
  while (i < n) {
      int temp = *ptr;
      if (temp != 0) {
         x  = x + f(temp);
      }
      else  {
        x = x - 3;
      }
      i++;
  }
  return x;
}
```

You will answer on the next page, where the code is re-produced in a format where you can show a line-by-line translation. Feel free to use this page for scratch work.

```
   .globl someFun
   .ent   someFun
   .text
someFun:
# int someFun(int * ptr, int n) {



#   int i = 0;

#   int x = 2;

#   while (i < n) {


#       int temp = *ptr;


#       if (temp != 0) {


#          x  = x + f(temp);
#       }

#       else  {

#         x = x - 3;

#       }

#       i++;

#   }

#   return x;





# }

.end someFun
```

Question 5:Datapaths [10 pts]

1. If a single cycle datapath has a clock period of 100ns, and is split into a multi-cycle data path with 5 stages, the clock will be slower than 20ns. Give *two* reasons why:

   -

   -

2. In the 1990s, processor design was about "performance at all cost," however now, performance must be balanced against other design considerations. Name two such design considerations, and for each one explain (briefly) why it is important.

   -

   -

Question 6:Memory Hierarchy [10 pts]

A processor uses 12-bit addresses, and has a 64-byte direct-mapped cache. The cache is organized in 8 sets, with 8-bytes per block. The contents of the cache are shown below:

	Tag	Data
Set 0	0F	12 34 56 78 99 43 59 B3
Set 1	15	11 22 33 44 B4 C6 A9 D2
Set 2	31	2F 3E 4D 5C A0 C9 DA 9C
Set 3	15	37 83 A9 C0 3F F3 2C AE
Set 4	3E	FF EE DD CC F9 F8 7C 8D
Set 5	1C	1F 1E 1D 1C 1B 1A 10 19
Set 6	0A	F6 D7 C8 E9 00 01 02 03
Set 7	0A	77 66 55 AA 05 06 07 08

Note that the least-significant byte in each block is on the right, and the most-significant is on the left (so in the first block, byte 0 is B3 and byte 7 is 12).

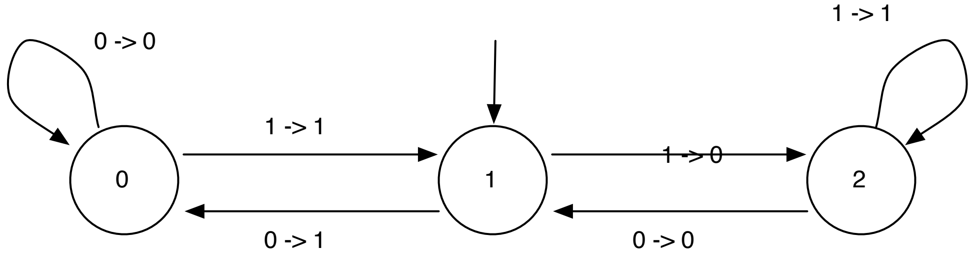
For each of the following addresses (a) split the address into tag/index/offset and write them in hex (b) determine if the access is a hit or miss (c) if the access is a hit, write the data that a 1-byte load to that address returns (if it is a miss, leave that field blank):

1. 0x55E
    - Tag:
    - Index:
    - Offset:
    - Hit or miss?
    - Value loaded:

2. 0x297
    - Tag:
    - Index:
    - Offset:
    - Hit or miss?
    - Value loaded:

Question Extra Credit [?? pts]

Consider the following finite state machine:



This FSM starts in state 1, and receives a binary number as input, starting at the least significant bit. The input contains sufficient implicit always has sufficient zeros after the most significant bit for the FSM to reach state 0 (so the FSM always ends in state 0). The FSM's edges are labelled with the input and output for that transition (so "1 -¿ 0" means input is a 1, output a 0). For example, 13 would be input into the FSM as 1 0 1 1 0 0, and would result in an output of 0 0 0 1 0 1.

If the number input to the FSM is the binary encoding of a number,`x`, what number does the binary string output by the FSM represent (as a mathematical function of `x`)?

Explain how/why this FSM computes this function of `x`.