E&CE 327 Midterm

2010t1 (Winter)

Instructions and General Information

- 100 marks total. Time limit: 1 hour, 20 minutes
- No books, no notes, no computers. Calculators are allowed
- If you need extra paper, request some from a proctor.
- Write neatly. To earn partial credit, you must show the formulas you are using and all of your work.
- **The proctors and instructors will not answer questions, except in cases where an error on the exam is suspected. If you are confused about a question, write down your assumptions or interpretation.**
- **Justifications of answers will be marked according to correctness, clarity, and conciseness.**

			Total Marks	**Approx. Time**	**Page**
Q0	!!Almost Free!!		2	2	2
Q1	Short Answer		8	5	3
Q2	VHDL Semantics		23	18	4
Q3	Dataflow Diagram		23	18	6
Q4	Code Review		23	20	9
Q5	Performance		23	15	12
Totals			100	78	

Name:_____

UWuserid:_____

Q0 (2 Marks) !!Almost Free!!

(*estimated time: 2 minutes*)

Q0a (1 Mark) Best part

What is the best part of the course?

Q0b (1 Mark) Most improve

What one thing could be done to most improve the course for the remainder of the term?

Potentially useful information

$$\log_x y = \frac{\log y}{\log x}$$

Q1 (8 Marks) Short Answer

(*estimated time: 5 minutes*)

What are the two goals of zero-delay simulation?

1)

2)

How does VHDL delta-cycle simulation achieve these goals?

1)

2)

Q2 (23 Marks) VHDL Semantics

(*estimated time: 18 minutes*)

In this question, you will describe the behaviour of the code below based on the delta-cycle simulation semantics of VHDL.

NOTES:

1. The clock period is 33 ns.
2. The code is legal VHDL.

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

entity sim_cyc is
  port (
    clk, a, b, c : in std_logic;
    z            : out std_logic
  );
end entity;

architecture main of sim_cyc is
  signal d, e, f, g, h, i, j, k : std_logic;
begin

  proc_1 : process begin
             wait until rising_edge(clk);
             d <= a;
             e <= b;
             f <= c;
             z <= k;
           end process;

  proc_2 : process ( d, k ) begin
             g <= not d;
           end process;

  proc_3 : process (d, e, g, j) begin
             h <= d xor e;
             k <= g nand j;
           end process;

  proc_4 : process (h, f, i) begin
             i <= h and f;
             j <= not i;
           end process;

end architecture;
```

For the code on the previous page, what is the maximum number of simulation cycles that can occur in a single simulation round? Answer the question by giving a **brief** description of what happens in each simulation cycle of the longest simulation round. Also, for each simulation cycle, answer whether the simulation cycle is a delta cycle.

NOTES:

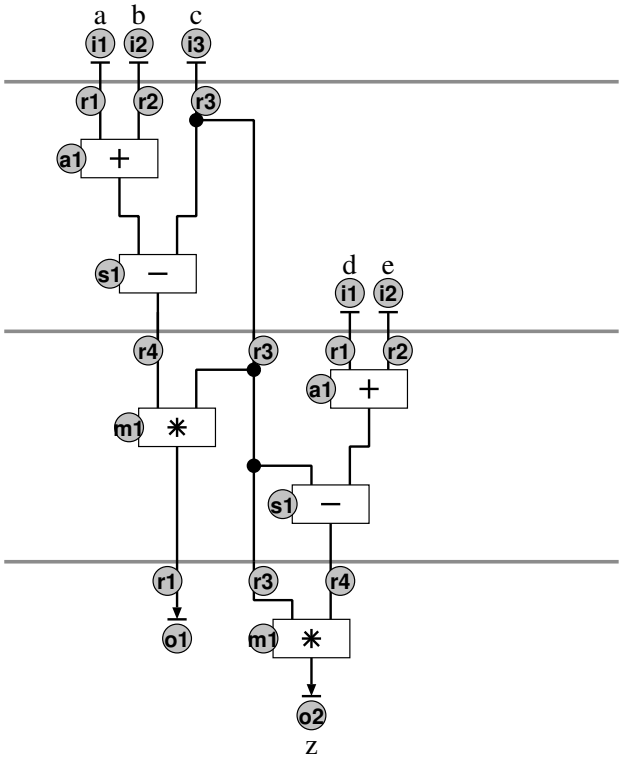
1. Your descriptions should be brief: do *not* include all of the information that is in a waveform diagram. The goal of this question is to evaluate your understanding of delta-cycle simulation semantics without the tedious details of working through an actual delta-cycle simulation.

Simulation cycle count	**Delta cycle Yes**	**No**	**Description**
example	√	☐	*Time jumps to 2012; all midterms are cancelled due to impending destruction of Earth*
1)	☐	☐	
2)	☐	☐	
3)	☐	☐	
4)	☐	☐	
5)	☐	☐	
6)	☐	☐	
7)	☐	☐	
8)	☐	☐	
9)	☐	☐	
10)	☐	☐	

Q3 (23 Marks) Dataflow Diagram

(*estimated time: 18 minutes*)

Before leaving for his annual ski trip to St. Moritz, your manager, Ilov H. Del, assigned you the task of designing the control signals for a dataflow diagram that he was working on.



Q3a (12 Marks) Control Signals

List the control signals (multiplexer select, chip-enable, *etc*) needed for each register and datapath component in the dataflow diagram. If a register or datapath component does not need any control signals, write "NONE". For each signal, give a *brief* description of its purpose.

NOTES:

1. The primary purpose of the description is to enable the marker to understand Q3b, where you will draw waveforms for the signals. You will *not* be marked upon the completeness of the descriptions, so do not waste time with detailed descriptions.
2. You may perform optimizations that simplify the control signals, so long as they do *not* affect: allocation, scheduling, or latency.
3. The state encoding scheme has not been chosen yet.

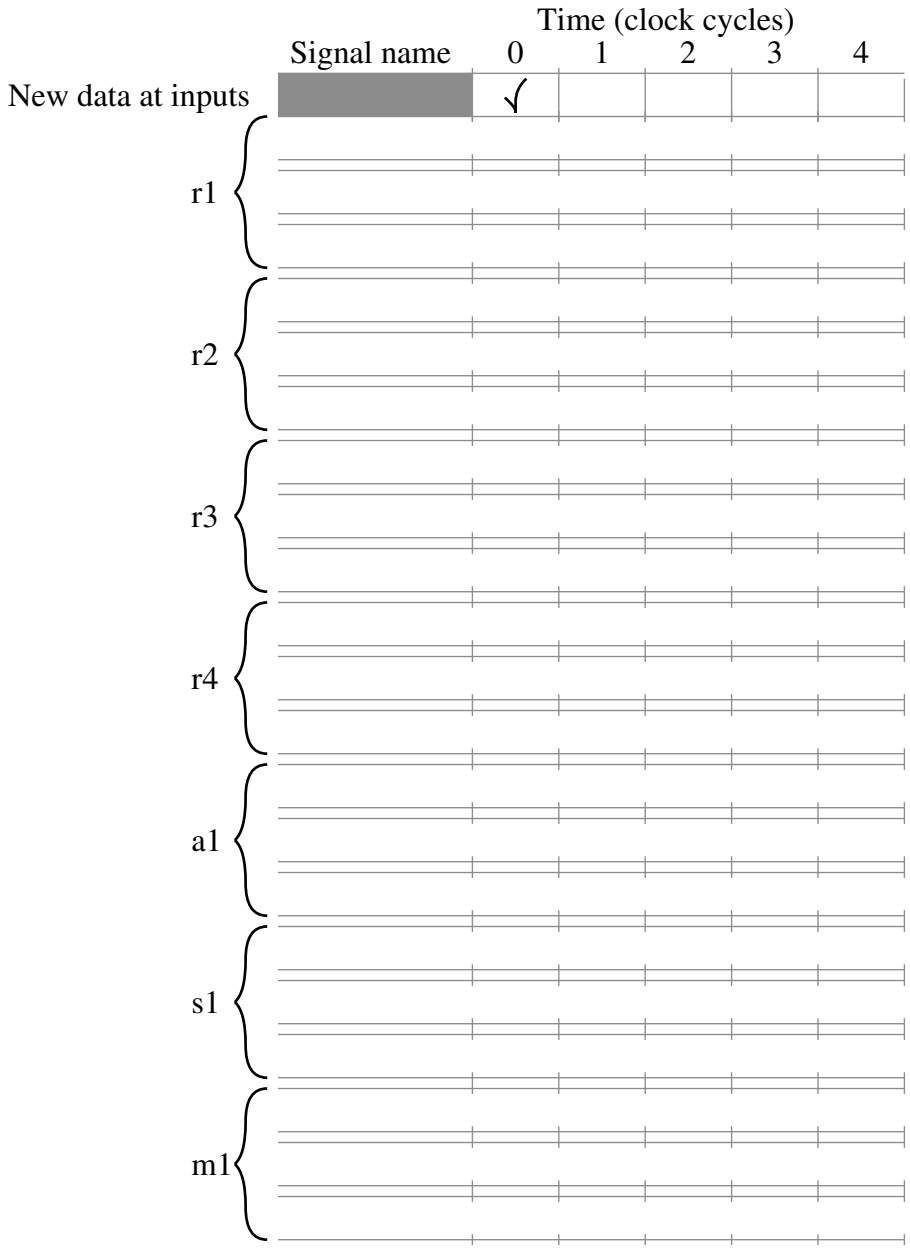
Register or component	Signal Name	Description
r1		
r2		
r3		
r4		
a1		
s1		
m1		

Q3b (11 Marks) Signal Behaviour

Draw a waveform diagram showing the behaviour of the control signals for 5 clock cycles beginning when a new set of data is available on the input pins.

NOTES:

1. Put a $\surd$ at each clock cycle where a new set of data can enter the circuit on the input pins.
2. Do *not* select values for don't cares; write "DC" as the value for "don't care".
3. For the values of multiplexer select signals, do *not* use "0" and "1", instead use the name of the component, register, or input that is selected; or "DC".
4. If you are unsure of what notation to use, choose one that is simple, and then describe it.



Q4 (23 Marks) Code Review

(*estimated time: 20 minutes*)

You are supervising an intern from some UTher school and are responsible for reviewing the intern's VHDL code. The intern has compiled the `count_gt` process on the next page, but has not simulated or synthesized it yet. Your task is to write comments that describe the five most important changes that should be made to the code.

NOTES:

1. The purpose of the code is to receive a sequence of 256 data values and count the number of data values that are greater than 63.
2. The input data are unsigned 8-bit numbers.
3. For each data value, `i_valid` is asserted ('1') for exactly one clock cycle, followed by 0 or more clock cycles of bubbles.
4. The input data `i_data` is valid only when `i_valid` is asserted.
5. The types of `i_data` and `o_count` must be `std_logic_vector`.
6. If a comment applies to multiple signals or processes, list the comment just once.
7. Keep the comments focused and try to preserve as much of the intern's code as possible. Try to avoid large structural changes to the code, such as "Combine the three processes into a single process."
8. If you cannot find five changes that will improve the code, then give positive comments about the good features in the code.

This area is blank intentionally

```
1)  library ieee;
2)  use ieee.std_logic_1164.all;
3)  use ieee.numeric_std.all;
4)
5)  entity count_gt is
6)    port (
7)      reset, clk, i_valid : in  std_logic;
8)      i_data              : in  std_logic_vector( 7 downto 0 );
9)      o_done              : out std_logic;
10)     o_count             : out std_logic_vector( 7 downto 0 )
11)   );
12) end entity;
13)
14) architecture main of count_gt is
15)   signal gt_count, data_count : unsigned( 7 downto 0 );
16)   signal done : std_logic;
17) begin
18)
19)   process (clk) begin
20)     if rising_edge( clk ) then
21)       if reset = '1' then
22)         data_count <= (others => '0');
23)       elsif i_valid = '1' then
24)         data_count <= data_count + 1;
25)       else
26)         data_count <= data_count;
27)       end if;
28)     end if;
29)   end process;
30)
31)   process (clk) begin
32)     if rising_edge( clk ) then
33)       if reset = '1' then
34)         gt_count <= (others => '0');
35)       elsif unsigned(i_data) > 63 then
36)         gt_count <= gt_count + 1;
37)       end if;
38)     end if;
39)   end process;
40)
41)   process begin
42)     wait until rising_edge( clk );
43)     if reset = '1' then
44)       o_done = '0';
45)     elsif data_count >= 256 then
46)       o_done <= '1';
47)     end if;
48)   end process;
49)
50)   o_count <= std_logic_vector(gt_count);
51)
52) end architecture;
```

The table below lists the different categories of comments in their order of importance, from most important to least important.

Each category has a key (B, S, A, C, or P). For each comment, write the key of the comment's category. If a comment fits into more than one category, write down the most important category that the comment fits into.

Key	Category
B	*bug* fixes
S	making the code *synthesizable*
A	decrease the *area*
C	making the *code* simpler and more elegant
P	a *positive* comment

	Key	Comment
example	*C*	*WRITE YOUR CODE IN ALL CAPS — IT WILL LOOK MORE IMPORTANT*
1)	☐	
2)	☐	
3)	☐	
4)	☐	
5)	☐	

Q5 (23 Marks) Performance

(*estimated time: 15 minutes*)

You work for Dark Silicon on their Waterluvian filter project, which run on FPGAs. At last year's Wonderful Waterluvian World Conference (W3C), you demonstrated the D1 Waterluvian filter. You are now working on the D2, which will be demonstrated at this year's W3C.

Another company, Fluffville Software, makes a software based Waterluvian filter.

Your project leader has gathered some data and rumours about Fluffville's Waterluvian filter. Assuming that the data and rumours that she has are true, she has asked you to predict the performance that the D2 must achieve to be faster than Fluffville's Waterluvian filter at this year's W3C.

NOTES:

1. At last year's W3C, Fluffville's filter (the F1) was 10% faster than the D1.
2. There's a rumour that Fluffville will demonstrate their new F2 filter a month before this year's W3C and will demonstrate the even faster F3 at W3C.
3. The rumours are that the F2 will be 1.80 times the speed of the F1. The performance of the F3 is still unknown.
4. Assume that Fluffville is able to sustain their rate of performance increase from the F2 to the F3.
5. Your goal is to develop the D2 Waterluvian filter in time for W3C such that the D2 will be faster than the F3.

How much faster than the D1 does the D2 need to be, in order for the D2 to be faster than the F3 at this year's W3C?

The D2 needs to be [ ] faster than the D1.