Print:

First Name:............................Last Name:............................

Student Number:...

University of Toronto
Faculty of Applied Science and Engineering

Final Examination – December 1997

ECE352F – Computer Organization

Examiner – Paul Chow

1. There are 6 questions and **17** pages. Do **all** questions. The total number of marks is 90.
2. **ALL WORK IS TO BE DONE ON THESE SHEETS!** Use the back of the pages if you need more space. Be sure to indicate clearly if your work continues elsewhere.
3. Please put your final solution in the appropriate spaces. A big space **does not** necessarily mean a long answer is required.
4. **No calculators or other computing devices allowed.**
5. **Paper Type D** – You may use a VHDL reference manual.
6. Place your student card on your desk

1 [10]	
2 [20]	
3 [15]	
4 [10]	
5 [15]	
6 [20]	
Total [90]	

[5 marks] 1. (a) Give a minimum-row, reduced-flow-table description of a two-input (x_1,x_2), one-output (z) fundamental-mode sequential circuit that operates in the following manner:

The output is 1 if and only if the input is $x_1 = x_2 = 1$ and the next-to-last input variable change was a change of x_1.

Example:

Input	00	10	11	10	11	01	11	01	00
Output	0	0	1	0	0	0	1	0	0

Question 1 continued…

[5 marks] (b) Consider the reduced flow table shown below.

	00	01	11	10
a	ⓐ	c	ⓐ	d
b	a	ⓑ	c	ⓑ
c	ⓒ	ⓒ	ⓒ	d
d	ⓓ	b	a	ⓓ

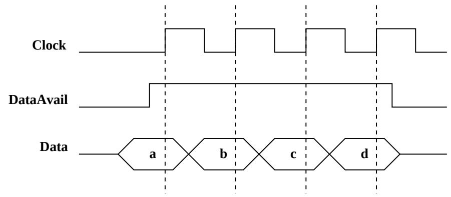
i. Obtain the transition diagram and show that three state variables are needed for a race-free binary state assignment.

ii. Make a state assignment and show a table that is free of critical races.

2. Your mission, which you will accept, is to design a complex multiplier. Recall that for two complex numbers, $C_1 = a + bj$ and $C_2 = c + dj$, the product is $P = R + Ij$, where

$$\begin{aligned} R &= (a * c) - (b * d) \\ I &= (a * d) + (b * c) \end{aligned}$$

The data will be made available to your circuit in four sequential clock cycles through one data port as indicated in the timing diagram below. Note that the rising edge of the **DataAvail** signal indicates that the first word of data is available. The outputs are to be generated in parallel, i.e., they should be available at the same time. When the outputs are generated, a **Done** signal is set high. Each of the input data words is 8 bits and each of the output words is 16 bits.



There are two versions of this product that you would like to sell. Your design will be implemented in a standard cell technology using VHDL to specify the design. As part of your technology library you have available parallel multipliers and parallel adder/subtracters that the synthesis tool will know how to use. The multiplier is about 8 times the size of the adder/subtracter.

Before starting this question, you might want to look at the next two questions since they are also related.

[5 marks] (a) Draw a diagram of what your datapath would look like for the smallest design in terms of area. Assume that the area of random logic and registers is negligible compared to the arithmetic units. You may use as many clock cycles as necessary.

[5 marks] (b) Describe the sequencing of data through your circuit on a cycle-by-cycle basis. A psuedo-code table format is acceptable, such as:

```
cycle               action                Comments
-----               ------                --------
  0               wait for DataAvail
  1               load register A         load inputs
  2               load register B
```

Question 2 continued…

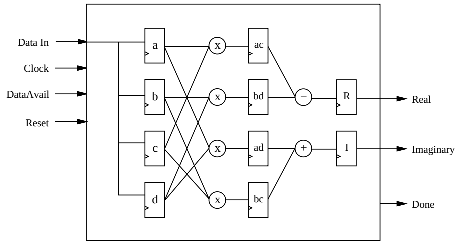
[10 marks] (c) To control your design, you are to build a controller using microcode.

- Show the format of your microinstruction and explain what each field is to be used for. To simplify your task, use a horizontal microinstruction encoding.

- Draw a block diagram that clearly shows the organization of your design. You do not have to give a detailed gate-level design, but you should indicate the function of each block, such as `adder`, `multiplexer`, or `decoder`, and how all the blocks connect.
  To make it easier to give part marks, please add a few words to describe how you want it to operate.

[15 marks] 3. Continuing with the theme of the complex multiplier in Question 2, you will now explore faster designs. All the specifications given in Question 2 are to be used here.

Your boss believes that the design shown below is the fastest possible, so you must implement it.



Write a VHDL program that can be synthesized to produce the design shown. Use a *synchronous* reset for your design.

The **Done** signal indicates that the output is available.

Question 3 continued...

4. You really know better than your boss and realize that the design in Question 3 is not the best you can do. For example, in that design the input stream is stopped until the outputs are generated and there is time when various resources are idle.

[7 marks] (a) Draw a datapath that once started, can accept an input every clock cycle, and generates a set of `R` and `I` every four clock cycles, i.e., in the same time that it takes to collect four input values.

[1 mark] (b) What must you assume about the speed of the multiplier relative to the input clock frequency to make this work?

[1 mark]

(c) What is the latency of your design?

1 mark]

(d) What is the throughput of your design?

5. The MIPS-X microprocessor has a five-stage pipeline:

IF Instruction fetch

ID Instruction decode and register fetch

EX Execute

MEM Access data memory

WB Write the results.

MIPS-X uses a compare and branch instruction with two branch delay slots. The instruction

```
bgt r1,r2,label
```

will branch to `label` when `r1` is greater than `r2`. The comparison is done in the Execute stage of the pipeline. The instruction encoding for the branch consists of the branch opcode, the condition to test, and a 16-bit signed offset.

For all instructions, the next value of the PC is computed by modifying the value of the PC currently being used in the **IF** stage. For example, if an instruction at address 100 is currently being fetched, the next instruction to fetch is either at $100 + 1$ or $100 +$ the branch offset, if a branch is to occur.

[2 marks] (a) What is the next value of the PC in the following sequence of instructions assuming that the branch is not taken? Explain/show how you got your answer.

```
1000    bge r1,r2,offset=20
1001    add r3,r4,r5
1002    sub r3,r4,r6
????
```

???? = □

[2 marks] (b) For the sequence in the previous part, what is the next value of the PC if the branch is taken? Explain/show how you got your answer.

???? = □

[2 marks] (c) One of the trickiest issues is to make sure that putting branches in branch delay slots works. Why is it desirable to do this?

[4 marks] (d) Now consider the following program:

```
i1    add  ...
i2    sub  ...
i3    xor  ...
i4    br offset = -1
i5    br offset = -1
i6    br offset = -1
i7    and  ...
```

The `br` instructions are unconditional branches, so they are always taken. The offset fields contain -1. Assuming that execution starts at `i1`, what are the first 15 instructions executed? The first three are shown already. To get part marks it would be helpful to add comments showing how you are determining the instructions that should be fetched.

```
i1, i2, i3,
```

Question 5 continued…

[5 marks] (e) In a non-pipelined processor, when an exception, such as a page fault, occurs, the action is to abort the instruction causing the exception, handle the exception, and then restart the instruction.

For the MIPS-X pipeline, and the sequence of code in Part (a), assume that the instruction at 1002 causes an exception during its execute phase. When this happens, all instructions in the pipeline, except the **WB** phase, are aborted. Why is it not sufficient to just restart the instruction at 1002?

What is the solution?

[3 marks] 6. (a) A set-associative cache is to have 128 sets with two blocks per set. A block is 64 bytes and an address is 32 bits. The word size is also 32 bits.

How many bits are there in the tag, set, and word fields?

#tag bits =

#set bits =

#word bits =

[1 mark] (b) How large is the cache in bytes?

#bytes =

[5 marks] (c) Devise a *pathological* example of a sequence of addresses that will make this cache ineffective.

[5 marks] (d) When doing research in cache design, the current approach is to *execute* a program and then monitor the sequence of addresses that it generates to memory. These addresses are then fed into a cache simulator program that models the particular cache being studied. The simulator can then generate various statistics.

The figure on the next page represents a small cache that has 8 sets. Each set contains 4 blocks and each block holds two words. When the cache is empty, the blocks are filled starting with `Block 0`. Assume that when one word in a block is accessed, then both words in the block are loaded into the cache on a miss. An *LRU* replacement policy is used.

The following sequence of **hexadecimal** addresses represents the stream of addresses generated by a program. Assume that all addresses are word addresses. Enter all addresses that are put into the cache into the appropriate boxes in the figure and write them starting from the left side in each box. If a block is replaced, then write the new address beside the old one separated by a comma, so that the most recent address is the one on the right.

```
NOTE: These are in HEXADECIMAL

0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 24 25 26 27
28 29 3A 100 101 102 103 105 105 106 225 226 227 228 1A
1B 1C 1D 1E 1F 10 11 12 13 14 304 305 30A 24 25 26 27 225
405 406 407 0A 0B 0C 30A 30B 19 1A 5A 5C
```

		Block 0	Block 1	Block 2	Block 3
Set 0	Word 0				
	Word 1				
Set 1	Word 0				
	Word 1				
Set 2	Word 0				
	Word 1				
Set 3	Word 0				
	Word 1				
Set 4	Word 0				
	Word 1				
Set 5	Word 0				
	Word 1				
Set 6	Word 0				
	Word 1				
Set 7	Word 0				
	Word 1				

[6 marks] (e) The average memory access time in the presence of a cache is defined as,

$$t_{ave} = hC + (1 - h)M$$

where,

$$\begin{aligned} h &= \text{hit ratio in the cache} \\ C &= \text{access time to the cache} \\ M &= \text{access time when the cache misses} \end{aligned}$$

- Derive the expression for t_{ave} when there are two levels of caches.

- As the number of transistors on a chip continues to increae, it becomes possible to put larger and larger caches on the same chip as the processor. Why might it be better to put two levels of cache on-chip rather than one large single-level cache? Hint: Think about how a memory is built.