Print:

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

Student Number:...

University of Toronto
Faculty of Applied Science and Engineering

Final Examination – December 1998

ECE352F – Computer Organization

Examiner – Paul Chow

1. There are 6 questions and **15** pages. Do **all** questions. The total number of marks is 100.
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 [15]	
2 [10]	
3 [10]	
4 [25]	
5 [20]	
6 [20]	
Total [100]	

[2 marks] 1. (a) Give the properties of a fundamental mode asynchronous circuit.

(b) An asynchronous sequential circuit is described by the following excitation function:

$$Y = x_1\overline{x_2} + (x_1 + \overline{x_2})y$$

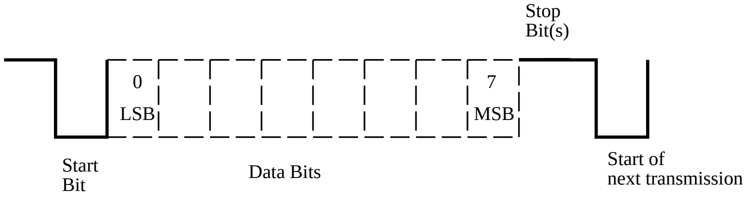
[6 marks] i. Draw the logic diagram of the circuit.

[4 marks] ii. Give the 2-state flow table for the circuit.

[3 marks] iii. Assume that the circuit starts with $(x_1x_2) = (01)$. Give a sequence of inputs in the form of ordered pairs for (x_1x_2) such that all stable positions in the flow table are visited.

[10 marks] 2. An asynchronous *start-stop* transmission scheme is shown in the figure below. Assuming that the receive clock is nominally 16 times the bit rate, recall that the basic operation on the receive side is to:

- Wait for the falling edge of the *start bit*.
- Count to 8 to get to the middle of the *start bit*.
- Count to 16 and sample the data, which is ideally at the middle of the data bit.
- Repeat until all bits have been sampled.
- The last bit sampled should be the stop bit, which should be high for at least one bit time. If not, then there is a *framing error*.



Assume that the frequency of the transmit clock is f_t. What is the range of frequencies for the receive clock, f_r that will still allow the data to be correctly sampled with no *framing error*? Give your answer by finding the values of m and n for the expression below. Be sure to explain how you are deriving your answer. The next page is blank if you need more space.

$$m \times f_t < f_r < n \times f_t$$

(The expressions should be easy enough to work out by hand. A result to one decimal place is sufficient or you may just give an expression.)

$m =$ □

$n =$ □

Question 2 continued…

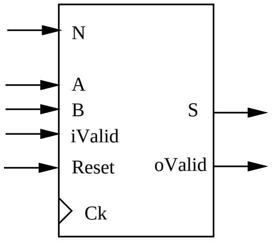
[6 marks] 3. (a) Assume that you have two 16-bit binary numbers to add. Give a circuit showing how these numbers can be added using only one single-bit full adder circuit (FA) whose inputs are labeled **A**, **B**, and $\mathbf{C_{in}}$, and whose outputs are **S** and $\mathbf{C_{out}}$. You may use any additional types of logic gates and registers that you need but make it clear what their functions are. You may NOT use these additional circuits to build other full-adder circuits!

The overall circuit should have two 16-bit parallel inputs, **X** and **Y**, a parallel output called **Sum**, and a **Carry** output. You may have any other inputs and outputs that you feel are necessary. Include a few words to explain how to use your circuit.

[4 marks] (b) Prove that the overflow condition for a 2's complement addition can be detected by using an exclusive-or of the carry-in and carry-out of the most-significant bit stage of the addition.

4. You are to design a circuit that can be used to add two arbitrary-length binary numbers presented one bit at a time (starting with the least-significant bit) at inputs **A** and **B**. The actual length of the numbers, **N** (**N** > 0), is loaded in parallel when the circuit is reset, so **N** must be stable for at least the last clock pulse before **Reset** goes low. **N** is eight bits, allowing numbers to be a maximum of 256 bits long. The circuit starts working on the first rising edge of the clock after the **Reset** signal goes low. The data inputs are first valid when **iValid** is high and the data changes on every clock pulse thereafter. There is a single bit output, **S** where the sum bits are produced in synchronization with the input bits. The **oValid** signal goes high on the first valid output of **S** and goes low after the Carry bit is output. The *N+1*st bit at the output will be the Carry bit. The initial carry-in to the adder is assumed to be 0.

Clearly state any assumptions you need to make.



[10 marks] (a) Draw a state diagram for the circuit, taking care to explain what happens in each state.

[15 marks] (b) Write a VHDL program that can be synthesized to produce the desired circuit. Use an *asynchronous* reset for your design. Grossly inefficient coding styles will be penalized.

You should try to write syntactically correct VHDL except that in the architecture part only the process that describes your state machine needs to be syntactically correct. For all other processes, the body of the process only needs to be comments or psuedo code describing the function of the process instead of detailed code. However, you must clearly indicate the inputs and outputs of each process.

5. A RISC processor has been developed with a 5-stage pipeline such that one instruction can be issued every clock cycle.

IF Instruction fetch

ID Instruction decode and register fetch

EX Execute

MEM Access data memory

WB Write the results to the register file

All load instructions use indexed addressing, where the effective address is computed during the **EX** stage and sent to the cache memory at the end of **EX**. The **MEM** stage then allows one cycle for accessing the cache. The results of any instructions are written during the end of the **WB** phase. The register file uses a write-before-read policy.

Assume the following initial conditions for register and memory values:

```
R1 = 6
R2 = 0
R3 = 5
R4 = 4
Mem = 10        This is a memory location
```

Consider the following instruction sequence:

```
1:        Load    R2 <-- Mem         ; R2 is loaded with the contents of Mem
2:        Add     R1 <-- R3 + R2     ; R1 gets the sum of R3 + R2
3:        Add     R1 <-- R1 + R2
4:        Add     R1 <-- R1 + R2
5:        Add     R1 <-- R1 + R2
```

Compute the value written into R1 for instructions 2–5 under the following conditions. There are no hardware interlocks.

[4 marks] (a) Using a non-pipelined machine where all instructions complete before the next one begins.

$R1_2 =$

$R1_3 =$

$R1_4 =$

$R1_5 =$

[4 marks] (b) Assuming the RISC pipeline described above is used but there is no bypassing logic.

$R1_2 =$

$R1_3 =$

$R1_4 =$

$R1_5 =$

[5 marks] (c) Assuming that the above RISC pipeline is used and full bypassing logic is available to minimize the latencies in the pipeline.

Explain why this results in a load delay of one cycle.

$R1_2 =$

$R1_3 =$

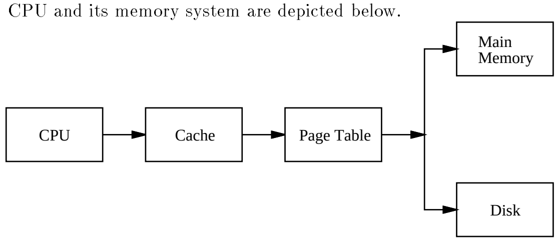
$R1_4 =$

$R1_5 =$

Question 5 continued…

[7 marks] (d) Sketch the datapath required to implement the version of the pipeline that has full bypassing logic. Be sure to include all necessary pipeline registers and bypassing logic. You may explain some things in words to simplify the diagram.

6. A simple CPU and its memory system are depicted below.



The computer features a 32-bit virtual address space, a 4-way set-associative cache with a total of 2^{10} blocks in the cache, 8-byte data blocks, and a write-through policy. The main memory is 2^{26} bytes and each page is 2^{12} bytes. Assume that a *word* is a byte.

[2 marks] (a) Does this system cache virtual or physical addresses? Explain.

[2 marks] (b) What is the data capacity of the cache in terms of bytes?

Cache size =

[3 marks] (c) How many bits are there in the tag, set, and word fields?

tag bits =

set bits =

word bits =

[2 marks] (d) How many tag comparators are needed? How many bits are in each comparator? Explain.

tag comparators =

bits/comparator =

[2 marks] (e) Assuming a one-level page table (like what was described in class) is used, how many entries are there in the page table?

page table entries =

[2 marks] (f) At any time, what is the greatest number of page-table entries that can have their valid bit set? Explain.

max valid page table entries = □

[7 marks] (g) Draw a figure of a one-level page-table scheme and explain how it works.