THE UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
ELEC 379 : Design of Digital and Microcomputer Systems
1998/99 Winter Session Term 2

SUPPLEMENTAL EXAMINATION
August, 1999

This exam has five (5) questions. The marks for each question are as indicated. There are a total of 61 marks. Answer all questions. Write your answers in the exam book provided. Show your work. You may answer the questions in any order. Books, notes and calculators are allowed. You may keep this exam paper.

Question 1 (15 marks)

This question asks you to design a device that is part of a binary division circuit.

The device has the following VHDL entity declaration:

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

entity divider is
   port (
      n_in, d_in : in unsigned (15 downto 0) ;
      q_out : out std_logic ;
      load, clk : in std_logic ) ;
end divider ;
```

This device has two internal 8-bit registers: `n` and `d`.

The `q_out` output should always be set to '1' if `n` is less than or equal to `d`, and '0' otherwise.

On each rising edge of the clock if `load` is '1' your design should set `n` to `n_in` and `d` to `d_in`.

On each rising edge of the clock if `load` is not '1' your circuit should set the value of `n` as follows:

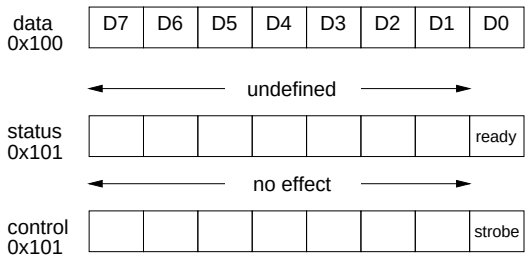
- if `q_out` is '1', it sets `n` to `n` minus `d` divided by two ($\frac{n-d}{2}$)
- if `q_out` is '0' it sets `n` to `n` divided by two ($\frac{n}{2}$)

Write an architecture that implements this device and is synthesizeable by MaxPlus+II. Use type conversion functions as necessary. Any `process` in your VHDL code must contain exactly one `if` statement surrounding one or two simple signal assignments. You need not include comments, `library` or `use` statements.

Hints: use "-" or "+". MaxPlus+II allows you to divide `unsigned` *values by 2.*

Question 2 (14 marks)

Write a subroutine, `bufout:`, in 8086 assembly language that write bytes from a buffer to a peripheral. The peripheral is accessed through an 8-bit output data register at I/O (not memory) address 100H, an 8-bit status register at I/O address 101H and an 8-bit control register also at address 101H:



Before writing each byte to the data register your program must wait until the LS bit of the status register (`ready`) is set to '1'.

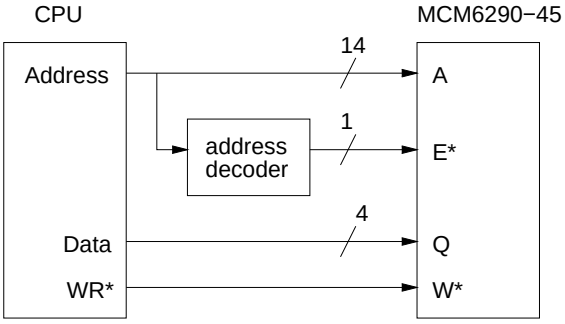
After writing each byte to the data register your program must set the LS bit of the control port (the `strobe` bit) to '0' and then to '1'.

Note that the other bits of the status register are undefined and only the LS bit of the control register has any effect.

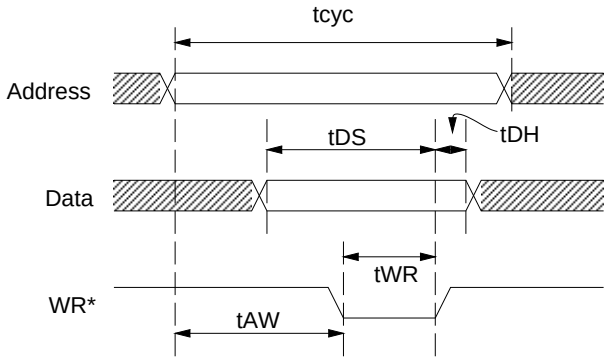
DS:BX will point to the start of the buffer when your function is called. The end of the buffer will be indicated by a byte with a zero (null) value. Your function must save and restore any registers it modifies before returning control to the calling function with a `RET` instruction. You must declare storage for any variables you use, but you need not include comments or assembler directives such as `segment`, `assume` or `org`.

Question 3 (10 marks)

The diagram below shows part of the interface between a hypothetical microprocessor and a Motorola MC6290-45 static RAM. An address decoder drives the RAM's chip enable (E*). The RAM's write enable (W*) is driven by the CPU's write strobe (WR*). The RAM's address and data buses are driven by the corresponding CPU buses.

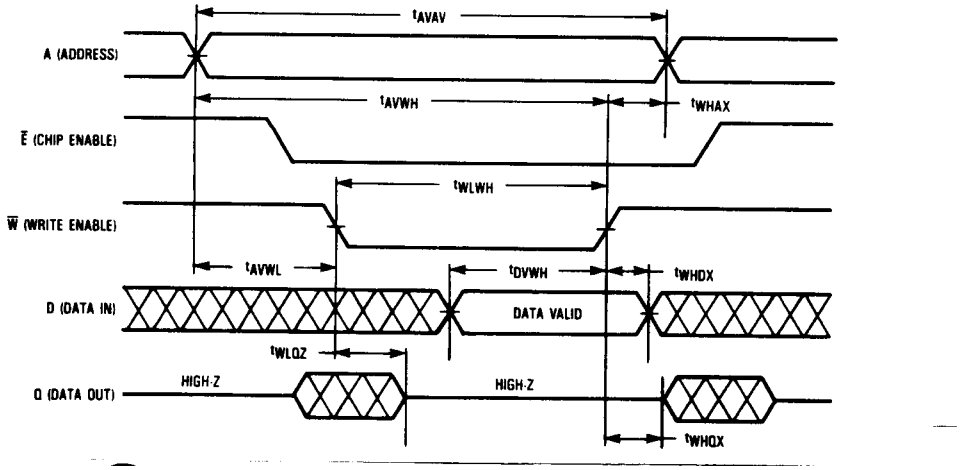


The following timing diagram and table gives the CPU's write-cycle timing specifications. Read-cycle timing specifications have been omitted.



Characteristic	Symbol	Min	Max	Unit
Cycle Time	t_{cyc}	100	100	ns
Data Setup	t_{DS}	25	-	ns
Data Hold	t_{DH}	0	-	ns
Write Width	t_{WR}	33	-	ns
Write Delay	t_{AW}	40	-	ns

The following timing diagram shows the SRAM write-cycle timing diagram.



The table below shows six SRAM write-cycle timing requirements that need to be met. Obtain expressions for these requirements in terms of the symbols for the CPU timing specifications given above. Obtain values for these expressions in nanoseconds. Find the amount by which the requirement is exceeded (in nanoseconds) and state whether the requirement is met or not (Y[es] or N[o]). Give your answer in the form of a table as shown below (you may omit the first column).

Requirement			Guaranteed		Margin	Met
Parameter	Symbol	Min. (ns)	Expression	Value	(ns)	(Y/N)
Write Cycle Requirements						
Write Cycle	t_{AVAV}	45				
Address Setup	t_{AVWL}	0				
Address Valid to End of Write	t_{AVWH}	35				
Write Pulse Width	t_{WLWH}	35				
Data Valid to End of Write	t_{DVWH}	20				
Data Hold	t_{WHDX}	0				

Question 4 (10 marks)

Draw the schematic of a memory system for an 8-bit CPU providing a total of 32 kBytes using 16k by 4 ROMs. Assume the CPU has a 20-bit address bus (`A0` to `A19`), and an 8-bit data bus (`D0` to `D7`). Each memory chip has data out pins (`Q0`, `Q1`, ...), address inputs (`A0`, `A1`, ...), and an active-low chip-select signal (`CS*`). The `CS*` pins should be driven from an address decoding circuit with an appropriate number of outputs. You may assume all memory chips' outputs are tri-stated when the `CS*` is not asserted.

Show the connections between the CPU, the decoder and the memory chip pins. The connections must be *unambiguous*. You may use the bus notation `X[n:m]` to indicate a subset of the signals of `X` consisting of bits n through m. For example, the following diagram shows that `D2` connects to `Q6` and `D3` connects to `Q7`:

To avoid ambiguity, place pin (signal) labels inside the rectangles representing the chips and the corresponding bus labels outside.

The details of the address decoder need not be shown.

Hint: Use a full page to draw your diagram.

Question 5 - "Best Match" (12 marks)

For each term in column A select the best matching entry in column B. There is one item in column A for each description in column B and one description in column B for each item in column A. Write your answers in numerical order and show the number and the selected letter unambiguously. No additional marks will be deducted for incorrect answers.

A		B	
1	DMA controllers	A	can be written to and retains contents when power is removed
2	partial decoding	B	is similar to the 8086 processor bus
3	SRAM	C	can mask individual interrupts
4	DRAM	D	is used to implement a serial interface
5	EEPROM	E	is based on the 8088 processor
6	HDLC	F	is a type of peripheral bus
7	ISA bus	G	can operate in cycle-stealing mode
8	SCSI	H	prevents use of the full memory space
9	ISA bus	I	has multiplexed address and data buses
10	UART	J	stores data as charge on a capacitor
11	Intel 8088	K	uses bit stuffing
12	8259 interrupt controller	L	can retain contents for long periods using a battery