THE UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
EECE 379 : Design of Digital and Microcomputer Systems
1999/2000 Winter Session, Term 2

FINAL EXAMINATION
8:30 AM – 11:30 AM
April 14, 2000

This exam has five (5) questions on five (5) pages. The marks for each question are as indicated. There are a total of 50 marks. Answer all questions. Write all answers in the exam book provided. Show your work. You may answer the questions in any order. Books, notes and calculators are allowed. Show your work. You may keep this exam paper.

Question 1 (10 marks)

This question asks you to design *hardware* to implement the same car headlight controller described in the mid-term exam. This controller implements a delay feature so that the headlights stay on for 30 seconds after the driver turns the headlight switch off.

The device has the following VHDL entity declaration:

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

entity lightdelay is
   port (
      lights : out std_logic ;
      switch, clk : in std_logic ) ;
end lightdelay ;
```

The `lights` output controls the headlights: a high output turns them on and a low output turns them off. The headlight `switch` input is high when the driver turns the switch on and low when the driver turns the switch off. The clock input, `clk`, frequency is 10 Hz.

Your hardware should turn the headlights on when the driver turns the switch on. After the driver turns the switch off, there should be a delay of 30 seconds and then the headlights should be turned off. The driver should be able to turn the headlights back on during this delay time. The headlights should come on within a very short time ($\leq$ 100 ms) of the switch being turned on.

Write an architecture that implements this device and is synthesizeable by MaxPlus+II. You may use only `std_logic`, `std_logic_vector` or `unsigned` types. Use type conversion functions as necessary. Any `process` in your VHDL code must contain exactly one `if` statement controlling one or more simple signal assignment statements. You need not include

comments, `library` or `use` statements. You need not synchronize inputs or register outputs (for glitch removal). You may ignore the initial (uninitialized) state of the device.

Hints: Draw an (incomplete) state transition diagram. Use "-" or "+".

Question 2 (10 marks)

Write a function, `getdigits:`, in 8086 assembly language that reads characters and stores them in a buffer.

When your function is called, the register BX will contain the offset of the buffer where the characters are to be stored.

Your function should call a function, `getch`, that reads one character and returns it in AL. `getch` does not modify any registers except AL. You do not have to write the `getch` function.

Your function should return when the line-feed (0AH) character is read. This line-feed character should not be stored in the buffer.

Your function should also "throw away" (not store) any characters that are outside the range '0' to '9' (30H to 39H inclusive).

Your function should keep track of the number of characters that are stored in the buffer and place this value in AX when it returns.

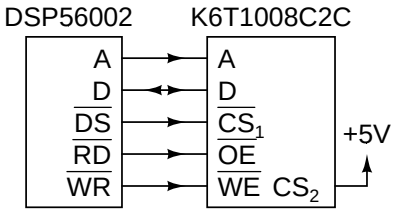
Your function must save any registers it modifies and restore them, except for AX, before returning with a `RET` instruction.

You must declare storage for any variables. You need not include comments or assembler directives such as `segment`, `assume` or `org`.

Hints: Write a C, flowchart, pseudo-code, or other high-level solution before you start coding.

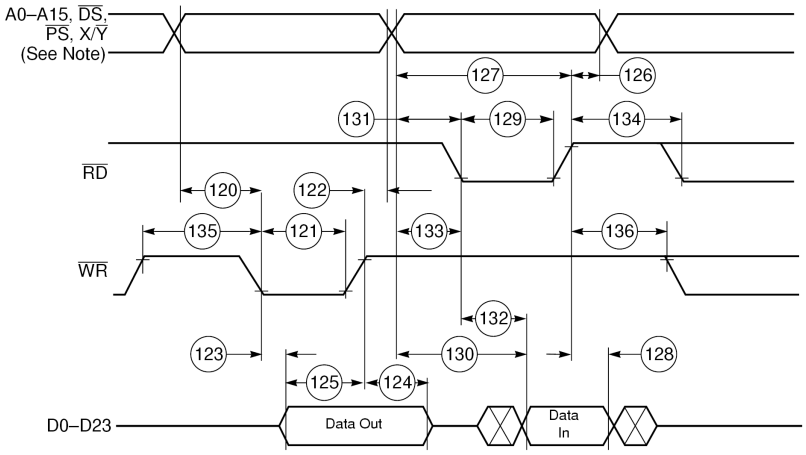
Question 3 (10 marks)

The diagram below shows part of the interface between a Motorola DSP56002 microprocessor and a Samsung K6T1008C2C SRAM:



The microprocessor's data strobe (DS*) signal drives the RAM's chip select (CS1*). The RAM's write enable (WE*) is driven by the CPU's write signal (WR*). The RAM's output enable (OE*) is driven by the CPU's read signal (RD*). The RAM's second chip select (CS2) is always enabled. The RAM's address and data buses are connected to the corresponding CPU buses.

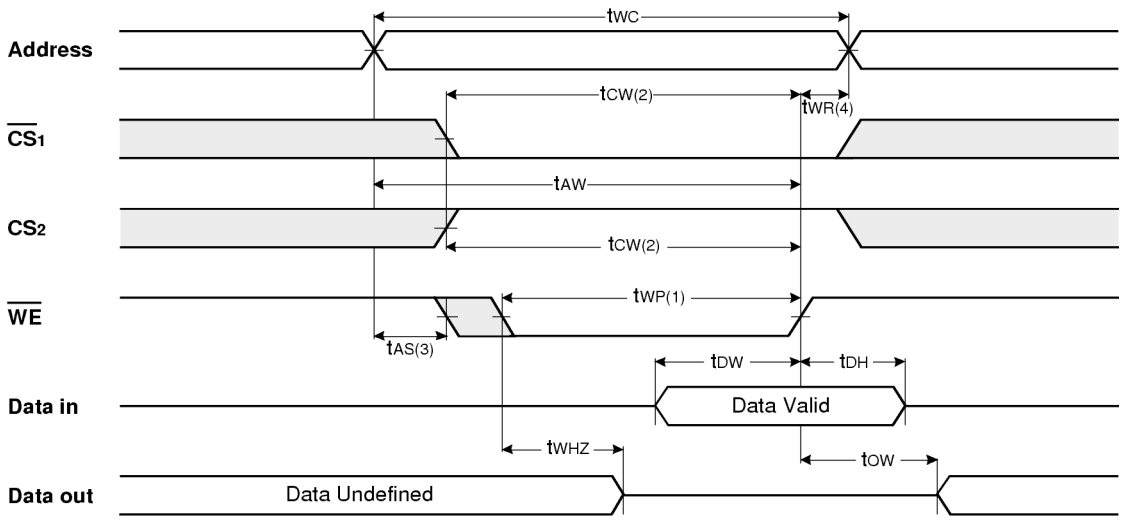
The following figure shows the CPU read and write cycle timing diagrams (both are shown on the same diagram).



The following table show some of the CPU timing specifications. The first column identifies the specification on the timing diagram. The third column is the minimum value and the fourth column is the maximum value. Assume T_C is 25ns, $T_L = T_H = 12.5$ns and $WS = 1$.

120	Address Valid to $\overline{WR}$ Assertion • WS = 0 • WS > 0	 $T_L - 6$ $T_C - 6$	 — —
121	$\overline{WR}$ Assertion Width • WS = 0 • WS > 0	 $T_C - 4$ $WS \times T_C + T_L$	 — —
122	$\overline{WR}$ Deassertion to Address Not Valid	$T_H - 6$	—
123	$\overline{WR}$ Assertion to Data Out Active From High Impedance • WS = 0 • WS > 0	 $T_H - 4$ 0	 — —
124	Data Out Hold Time from $\overline{WR}$ Deassertion (the maximum specification is periodically sampled, and not 100% tested)	$T_H - 7$	$T_H - 2.5$
125	Data Out Setup Time to $\overline{WR}$ Deassertion • WS = 0 • WS > 0	 $T_L - 0.8$ $WS \times T_C + T_L - 0.8$	 — —

The following diagram shows the SRAM write-cycle timing diagram. Note that the write cycle ends when the first of CS* or WE* goes inactive.



The table below shows five 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. Use symbol of the form t_n where n is the number from the diagram (e.g. t_{120}). 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. Write your answer in your exam book – *not on the exam paper*.

Requirement			Guaranteed		Margin	Met
Parameter	Symbol	Min. (ns)	Expression	Value	(ns)	(Y/N)
Write Cycle Requirements						
Write Pulse Width	t_{WP}	40				
Data to Write Time Overlap	t_{DW}	25				
Data Hold from Write	t_{DH}	0				
Address Valid to End of Write	t_{AW}	45				
Write Recovery	t_{WR}	0				

Question 4 (6 marks)

This question asks you to design a memory system for a CPU that has an 16-bit address bus, `A0` to `A15` and an 8-bit data bus (`D0` to `D7`). The memory system provides a total of 16 kBytes using 8k by 4 ROMs. Each memory chip has data out pins (`Q0`, `Q1`, ...), address inputs (`A0`, `A1`, ...), and an active-low chip-select signal (`CS*`). The `CS*` pins are driven from an address decoding circuit with an appropriate number of outputs.

(a) Draw the schematic of the memory system showing the connections between the CPU, the decoder and the memory chip pins. The connections must *unambiguously* show which CPU signals connect to which memory signals.

Hint: Use a complete page for your diagram.

(b) Given the following VHDL entity declaration:

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

entity decoder is
   port (
      a : in std_logic_vector (15 downto 0) ;
      cs0, cs1, ... : out std_logic ) ;
end decoder ;
```

write an architecture that sets the value(s) of the *active-low* chip-select signals `CS0*`, `CS1*`, `...` so that the 16 kByte memory region starts at address 8000H and addresses are fully decoded.

Question 5 (14 marks)

(a) For each bus in column "A" select the best matching entry in column "B". Write your answers in numerical order and show the number and the selected letter unambiguously. For this part you need not explain your answer. Make sure your answer is unambiguous.

A		B	
1	PC-104	A	synchronous
2	"Centronics"	B	standard device interfaces
3	SCSI	C	unidirectional
4	PCI	D	ISA-like
5	RS-232	E	uses bridges
6	USB	F	UART

(b) The first 4 bytes of memory in a PC have the following values (in hex): `00 10 20 30`. What are the segment and offset of the address of the divide-by-zero exception handler? What is the physical address (as a 5-digit hex number)?

(c) The following diagram shows the waveform as a character is transmitted over an RS-232 serial interface. What is the hex value of the character that was transmitted? What is the baud rate?