THE UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL ENGINEERING
ELEC 464 : Microcomputer System Design (Fall 1996)

FINAL EXAMINATION

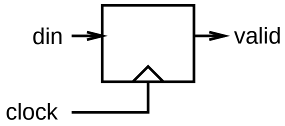
3:30 – 6:30 PM
December 16, 1996
MacMillan 166

This exam has four (4) questions on five (5) pages. The marks for each question are as indicated and there are a total of 64 marks. Answer all questions in the exam book provided. You may answer the questions in any order. Books, notes and calculators are allowed. You may keep this exam paper.

Question 1 (20 marks)

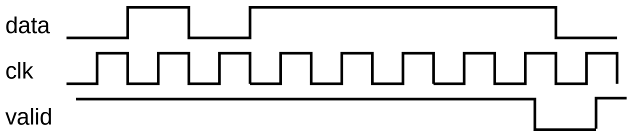
This question asks you to design a circuit to detect the zero bit which is inserted following a sequence of five consecutive ones in an HDLC serial data stream.

Your circuit should have a clock input (`clk`), a data input (`data`), and a "valid data" output (`valid`) as shown below:



The data input is valid on each rising clock edge and there is one rising clock edge per data bit. Your design must be a Moore state machine that changes state on the rising edge of the clock input. The output should be set to '1' if the *next* bit will be valid data and '0' if the *next* bit is a "stuffed" bit.

Your design should assume that all bits following five '1's will be zero bits (i.e. you may ignore end-of-frame flags and invalid sequences). The following diagram shows sample input and output waveforms:



(a) Choose an appropriate number of state variables and the encoding for the state variables and describe the state transition behaviour of the state machine in the form of a table as shown below. The table should have three columns: (1) the values of the state variables for the current state, (2) the value of the `data` input, and (3) the values of the state variables for the *next* state. You may use 'X' to indicate a "don't care" value.

State Variables	`data`	Next State

Also give the value of the `valid` output for each state in the form of a table as shown below. You may *not* combine the two tables.

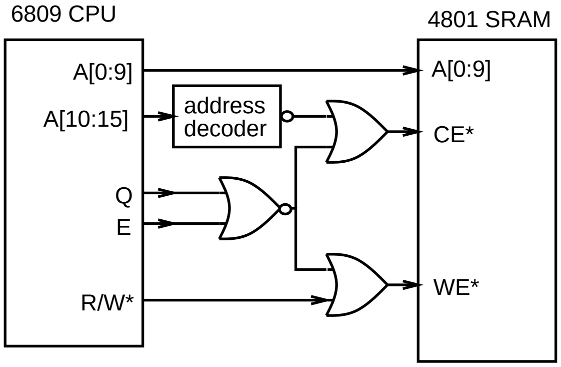
State Variables	`valid`

(b) Given the following VHDL entity, write a VHDL architecture that implements the above state machine. Your architecture must be synthesizable (e.g. by Synopsys Design Compiler). Use separate processes for the combinational and sequential parts of your design.

```
entity unstuff is
   port ( data, clk : in  bit ;
          valid : out bit ) ;
end unstuff ;
```

Question 2 (18 marks)

The diagram below shows part of the interface between a Motorola MC68B09 microprocessor and a Mostek MK4801A-3 static RAM. The 6809's Q and E outputs are OR'ed and used to enable both the SRAM's chip-enable (CE*) and write enable (WE*) inputs. This means that during a write cycle both CE* and WE* will be asserted when either E or Q is asserted. *Hint: draw CE* and WE* on the 6809 timing diagram.*



The following timing diagram and table gives the MC68B09 write-cycle timing specifications. Read-cycle timing specifications have been omitted. You may assume parameter $\boxed{4}$, *Clock Rise and Fall Time*, is zero.

No.	Characteristic	Symbol	Min	Max	Unit
$\boxed{1}$	Cycle Time	t_{cyc}	500	500	ns
$\boxed{2}$	Pulse Width, E high	PW_{EH}	210	—	ns
$\boxed{3}$	Pulse Width, E low	PW_{EL}	220	—	ns
$\boxed{4}$	Clock Rise and Fall Time	t_r,t_f	0	0	ns
$\boxed{5}$	Pulse Width, Q high	PW_{QH}	210	—	ns
$\boxed{6}$	Pulse Width, Q low	PW_{QL}	220	—	ns
$\boxed{7}$	Delay Time, E to Q high	t_{AVS}	80	125	ns
$\boxed{9}$	Address Hold Time	t_{AH}	20	—	ns
$\boxed{10}$	R/W* and Address Valid to Q Rise	t_{AQ}	15	—	ns
$\boxed{20}$	Data Delay Time from Q	t_{DDQ}	—	110	ns
$\boxed{21}$	Write Data Hold Time	t_{DHW}	30	—	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 6809 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.

Requirement			Guaranteed		Margin	Met
Parameter	Symbol	Min. (ns)	Expression	Value	(ns)	(Y/N)
Write Cycle Requirements						
Write Cycle	t_{WC}	250				
Address Setup	t_{AS}	0				
Address Hold	t_{AH}	80				
Data to Write Setup	t_{DSW}	20				
Data from Write Hold	t_{DHW}	10				
Write Pulse Duration	t_{WD}	70				

Question 3 (12 marks)

This question asks you to design a RAM memory system for a CPU with a 16-bit address bus and an 8-bit data bus. The memory system must contain a total of 16 kBytes of RAM and is to be built using 8k by 4 RAM chips. This memory should start at address 04000H and extend to 7FFFH.

(a) Draw a block diagram showing how the CPU's address and data bus pins connect to the RAM chips' address and data pins. Label the CPU address lines as A15 to A0 and the CPU data lines as D7 to D0. Label the RAM address lines as A12 to A0 and the RAM data lines as Q3 to Q0. You may use the bus notation as described in Assignment 5 (e.g. A[23:20]) to label buses.

(b) Add an address decoder to the diagram using a 74LS138 3-to-8 decoder IC. This address decoder must fully decode memory addresses. The 74LS138 decoder has 3 inputs (A, B and C) and 8 active-low outputs (Y0* through Y7*).

The truth table for a 74LS138 decoder chip is as follows:

C	B	A	Y0*	Y1*	Y2*	Y3*	Y4*	Y5*	Y6*	Y7*
L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H
L	H	L	H	H	L	H	H	H	H	H
L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H
H	L	H	H	H	H	H	H	L	H	H
H	H	L	H	H	H	H	H	H	L	H
H	H	H	H	H	H	H	H	H	H	L

Each RAM has an active-low chip select line, CS*. Show which CPU address bus signals are connected to which decoder inputs and which decoder outputs are connected to which RAM CS* inputs.

The connections in your diagram must be unambiguous. Please place chip pin labels (e.g. Q[0:3]) inside the boxes and labels for parts of a bus (e.g. A[15:14]) outside and next to the relevant chip. *Hint: use a whole sheet of paper for the diagram and leave plenty of room to draw the connections.*

Question 4 (14 marks)

This question asks you to complete an 8088 assembly-language program that outputs the string "Hello, world" through an IBM PC's serial port.

The serial port uses an 8250 serial interface chip. This chip has a data register which appears at I/O address 3F8H and a status register at I/O address 3FEH. Writing a byte to the data register causes it to be sent over the serial interface. Before writing a byte to the data register your program wait until bit 5 [1] of the status register is a one. Bit 5 is the “transmitter buffer empty” bit and it is a ’1’ when the serial interface transmit buffer is empty and a new byte may be written to the data register.

Your answer should be the missing part of the following 8088 assembly-language program:

```
code    segment public
        assume  cs:code,ds:code
        org     100h

[your code goes here]

; the message to print

msg     db     ’Hello, world’,0
tmp     db      ?

code    ends
        end     start
```

You may assume all serial port initialization (baud rate, parity, etc.) has already been done. Your code must send each of the characters in the `msg` string over the serial interface. Your code must test bit 5 of the status register and wait until the transmit buffer is empty before sending each character. Your code must return control to DOS using an `INT 20H` instruction when it is done.

You must use Intel-syntax 8088 assembly language as in the course notes and solutions. Minor syntax errors will be excused but marks will be deducted for ambiguous code. Please try to restrict yourself to the instructions and addressing modes covered in the course.

[1] bit 0 is the LS bit