UNIVERSITY OF VICTORIA
EXAMINATIONS APRIL 1997

C SC 230 Computer Architecture and Assembly Language

NAME (print): _____ REG NO. _____

SIGNATURE: _____ DURATION: 3 hours

INSTRUCTOR: D. M. Miller

TO BE ANSWERED ON THIS EXAMINATION PAPER.

STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER BEFORE BEGINNING TO WRITE, AND REPORT ANY DISCREPANCY IMMEDIATELY TO THE INVIGILATOR.

THIS EXAMINATION HAS 9 PAGES PLUS THIS COVER PAGE.

ATTEMPT EVERY QUESTION. ANSWER IN THE SPACES PROVIDED (YOU DO NOT NECESSARILY HAVE TO USE ALL LINES PROVIDED AND MAY USE OTHER AREAS ON THE **FRONTS** OF THE PAGES IF NECCESARY). USE THE BACKS OF THE PAGES FOR ROUGH WORK.

THIS IS AN OPEN TEXT BOOK EXAMINATION. YOU MAY REFER TO
H.W. Huang: "MC68HC11: an Introduction" West Publishing, 1996.

NO OTHER AIDS, E.G. COURSE NOTES OR CALCULATORS, ARE PERMITTED.

QUESTION	MAX. MARK	STUDENT'S MARK
1	**4**	
2	**8**	
3	**13**	
4	**12**	
5	**15**	
6	**10**	
7	**13**	
8	**6**	
9	**3**	
10	**7**	
11	**4**	
12	**5**	
TOTAL	100	

1. (4 marks)
Write the 8-bit 2's complement representation of each value below as a binary number and its hexadecimal equivalent:

	binary	hex		binary	hex
a) 0	_____	_____	b) 64	_____	_____
c) -35	_____	_____	d) -128	_____	_____

2. (8 marks)
Consider the following program:

```
        org   $8000
count   rmb   2
start   lds   #$dfff
        ldd   #0
        std   count
loop    jsr   update
        fdb   count
        ldd   count
        cpd   #25
        bls   loop
        jmp   $e000
```

(a) Write the subroutine "update" in as few instructions as possible so that variable "count" is incremented by one each time "update" is executed. Note that the subroutine uses program memory parameter passing. Your subroutine does not have to protect any registers.

_____

_____

_____

_____

_____

_____

_____

_____

_____

(b) How many times is subroutine "update" executed? _____

3. (13 marks)
Consider the following program:

```
;The sum of the first 4 elements of array arr is placed in location
;labelled "total".

        org     $8000
arr     fcb     121,144,169,75,38,205
total   rmb     2
        org     $9000
        lds     #$dfff
        ldx     #arr
        ldy     #4
        jsr     sum
        std     total
        jmp     $e000

;Subroutine sum receives the number of elements to be totaled in register Y
;and the array address in register X. It leaves the result in register D.
sum     ldd     #0
loop    cpy     #0
        beq     end
        addb    0,x
        adca    #0
        inx
        dey
        bra     loop
end     rts
```

(a) What is the content of accumulator A on exit from the call to subroutine "sum"?

_____

(b) What is the content of index register X on exit from the call to subroutine "sum"?

_____

(c) The above program passes its parameters, both input and output, via registers. You are to modify it to pass the parameters via the stack. There are to be three parameters: (1) the address of the array of values; (2) the number of values to be totaled; (3) the address of the location where the result is to be placed.
(continued on next page)

You are also to add any instructions necessary to ensure all registers used by the subroutine are protected i.e. upon return from the subroutine all registers (except the CCR) have the same contents they had when the jsr was executed/

```
        org     $8000
arr     fcb     121,144,169,75,38,205
total   rmb     2
        org     $9000
        lds     #$dfff

        _____
        _____
        _____
        _____
        _____
        _____
        _____
        jsr     sum
        jmp     $e000

;Subroutine sum receives three parameters on the stack: the address of an
;array, the number of values to be totaled and the address of the location
;to hold the result
sum     _____
        _____
        _____
        _____
        _____
        ldd     #0
        tsy
        ldx     ___,y
loop    tst     ___,y
        beq     end
        addb    0,x
        adca    #0
        inx
        dec     __,y
        ___     loop
end     ldx     __,y
        std     0,x
        _____
        _____
        _____
        _____
        _____
        rts
```

4. (12 marks)
Write a subroutine called DLYSEC that returns after a delay of the number of milliseconds specified by the value in the A accumulator when the subroutine is entered. For example, if A contains 15 on entering the subroutine should return after a delay of 15 milliseconds. The requested delay must be between 1 and 255 milliseconds. Assume the processor E clock is 2 MHz.

Your subroutine must use an output compare and polling in generating the requested delay. Your subroutine must protect all registers it uses.

5. (15 marks)
Frequently, a system has a task that must be executed at a fixed period. That is the situation here. You are given a subroutine called CHECKUP which performs a routine system check each time it is executed. You are to write the necessary 68HC11 code to ensure this task is executed at 10 second intervals i.e. the calls to CHECKUP are to occur as close to 10 seconds apart as possible. You must use the real-time interrupt in determining when to execute CHECKUP. CHECKUP has no parameters and returns via an RTS instruction with no result (in a real application it would abort the system if a problem is found). You do not have to write any code for checkup. Assume the E clock is 2 MHz.

VERY IMPORTANT: CHECKUP can take up to 500 milliseconds to execute so you must allow for a real time interrupt while CHECKUP is executing.

Initialization code required:

Real-time interrupt service routine:

6. (10 marks)
Suppose you are writing a monitor to control execution of several tasks using real-time interrupts. Part of the monitor is a priority list called PLIST where each list entry (3 bytes) contains:

- the address of the task in memory (2 bytes)
- the priority of the task, a value from 1 to 15 (1 byte)

The table is terminated by an entry with the priority byte set to 0.

The task address points to a stack area associated with that task. You can assume the top entries in the stack area are the register contents from the last time the task was interrupted and that initialization of the stack area is done appropriately.

You are to write only the 68HC11 code which searches the table for the highest priority task (in case of a tie, the first one in the list is selected), updates the stack appropriately, and transfers control to the selected task.

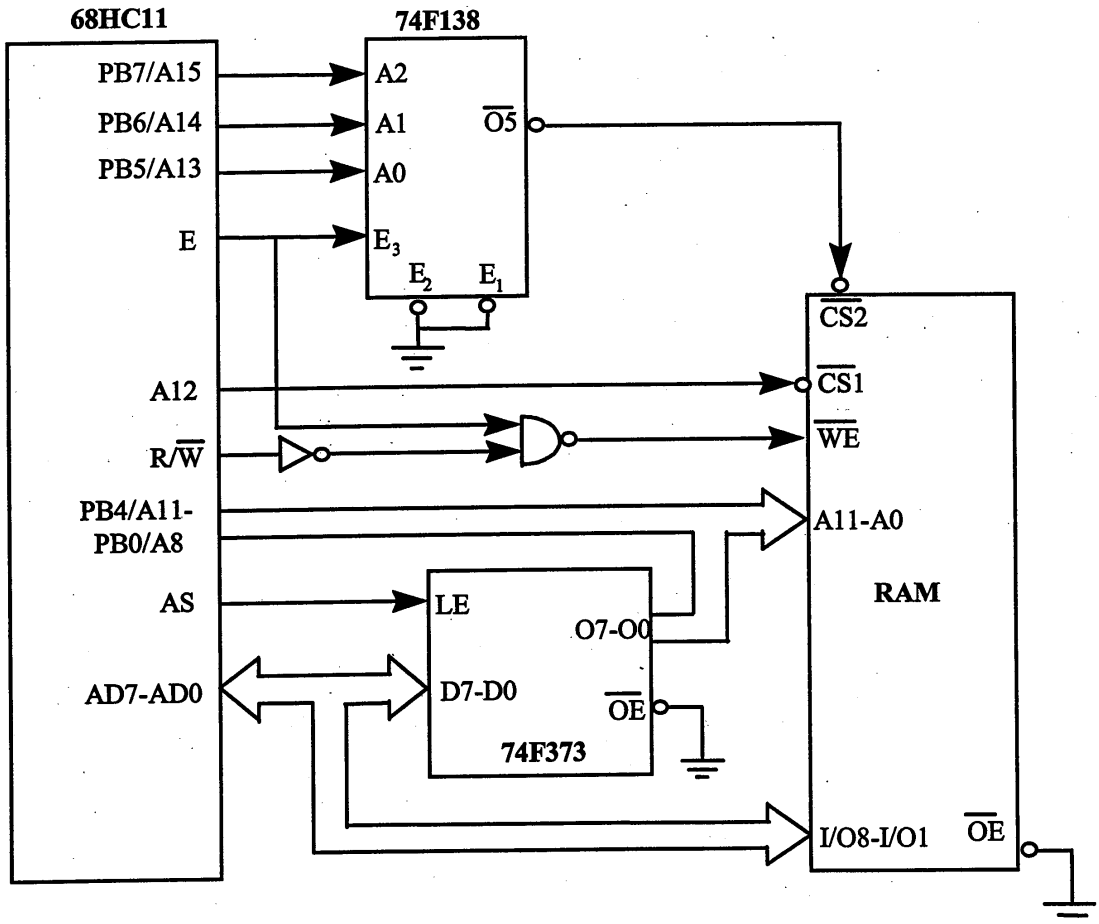
7. (13 marks)
Consider the diagram below (note it is **NOT** identical to the one in the text).

(a) How many bytes of memory should the RAM chip have? _____

(b) What range of addresses does the RAM chip occupy in the given diagram?

_____

(c) Show the changes necessary so that (a) the diagram represents the connection of a RAM chip of twice the number of bytes as the one given and (b) that RAM chip occupies the block of memory beginning at location $A000.



(d) Which component(s) would not be needed if the 68HC11 had separate address and data busses?

_____

8. (6 marks)
Indicate whether each of the following statements is true or false.

A processor using memory mapped I/O requires special I/O instructions. _____

On the 68HC11, external memory must respond within one E clock cycle. _____

RAM and ROM can overlap in the address space because they are different types of memory. _____

The PowerPC has separate fixed and floating point units only because the results of one are never needed by the other. _____

Good RISC design eliminates the need for a cache. _____

Effective use of the branch prediction mechanism on the PowerPC requires special coding on the part of the assembly language programmer or the compiler. _____

9. (3 marks)
State three significant factors that distinguish RISC designs from CISC designs:

_____

_____

_____

10. (7 marks)
(a) Briefly explain the difference between *write-through* and *write-back* with respect to cache management.

_____

_____

(b) Briefly explain what is meant by a *set associative cache.*

_____

_____

_____

11. (4 marks)
    (a) Briefly explain what is meant by pipelining and how it speeds up the execution of machine language instructions.

    (b) Why does a conditional branch cause difficulties for pipelining whereas an unconditional branch does not?

12. (5 marks)
    (a) Briefly explain the difference between SIMD and MIMD with reference to parallel processing.

    (b) What is the dominant factor as to whether a problem can be effectively 'parallelized' on a SIMD machine?

****** END OF EXAMINATION ******