UNIVERSITY OF TORONTO

Faculty of Applied Science and Engineering

Final Examination, December 2001

Department of Electrical and Computer Engineering

ECE 370F – Introduction to Microprocessors

Exam Type: A
(Closed Book, non-programmable calculators only)

Exam duration 2½ hours

Total marks: 100

INSTRUCTIONS
1. Answer all questions on the exam paper
2. There are 7 questions
3. READ all questions carefully
4. Be sure that your name is on all pages of the exam
5. All questions have the value indicated in [].

Name: _____

Student Number: _____

Question 1	Question 2	Question 3	Question 4	Question 5	Question 6	Question 7	Total

Name:_____

Question 1 - General Short Answers [13 marks]

1. Name two functions that the Status Register (SR) performs on the 68000? [2 marks]

2. What is the difference between the following two program fragments? [2 mark]

```
org $40000
DATA    dc.l    42                      move.l #42,$40000
```

3. Perform the operations indicated below, showing both the results and the resulting condition codes N, Z, V and C (as in the 68000). All the numbers are 6-bit signed integers in 2's complement representation. [5 marks]

   a. 011101 + 100011 = **N=** **Z=** **V=** **C=**

   b. 010101 + 011010 = **N=** **Z=** **V=** **C=**

4. What two things does the **rte** instruction do and why are each of the two operations necessary. [4 marks]

Name:_____

Question 2 – Debugging a Subroutine [10 marks]

A programmer has written a subroutine that is supposed to wait for the user to type in a line and then to print that line to the screen. The input line is guaranteed to be less than 80 characters and is terminated by a carriage return (#CR in the program). The number of times the line is to be output is an input parameter to the subroutine. All I/O is to be done using polling.

The code below is used to call the subroutine. It is known to be correct:

```
MOVE.W    #NUM_TIMES_TO_PRINT,-(A7)
BSR       SUB
ADD.L     #4,A7
```

The following subroutine is supposed to implement the above specification. However, as it is written this program contains several **logical** errors.

You are to correct each of the incorrect instructions on the line adjacent to that instruction. Do not "correct" any instructions that do not have mistakes in them or you will be penalized. You do not have to delete any instructions to make this program work; you only have to correct the flaws in the existing instructions.

Note: the "receive ready" bit of the UART is bit #0 and "transmit ready" is bit #2. Each "ready" bit is normaly 0 and becomes 1 when the UART is ready.

```
* The following are used by the program and are known to be correct.
* Don't bother to check

SRB       equ   $fffff7f3
RBB       equ   $fffff7f7
TBB       equ   $fffff7f7

* BUFER is used to hold the input characters
* Don't bother to check

BUFFER    ds.b  80
```

Name:_____

```
* Save all of the registers used in the subroutine onto the stack.
        MOVE.L   A0,-(A7)
        MOVE.L   D0,-(A7)
        MOVE.L   D2,-(A7)
* Retrieve the line from the user
        MOVEA.L  BUFFER,A0
LOOP    BRA      GETC
        MOVE.B   D0,(A0)+
        CMP.W    #CR,D0
        BNE      LOOP
* Get the number of times to output the line and output it that many times
        MOVE.W   6(A7),D0
OUTPUT  MOVEA.L  BUFFER,A0
LOOP1   MOVE.W   (A0)+,D2
        BRA      PUTC
        CMP.W    #CR,D2
        BNE      LOOP1
        SUB.W    #1,D0
        BGE      OUTPUT
*Restore the registers we used
DONE    MOVE.L   (A7)+,D0
        MOVE.L   (A7)+,D2
        MOVE.L   (A7)+,A0
        RTS
*Send a Character to the screen via polling
PUTC    BTST.B   #2,SRB
        BEQ      PUTC
        MOVE.B   D2,TBB
        RTS
*Read a Character via polling
GETC    BTST.B   #0,SRB
        BEQ      GETC
        MOVE.B   RBB,D2
        RTS
```

Name:_____

Question 3 -- Memory and Caching [38 marks]

1. What is Temporal Locality and how does it relate to cache design? [3 mark]

2. Consider a computer that is to be built with a block-set associative cache with 32 sets of 4 blocks where each block has 16 words. Assume that memory has 2048 blocks of 32 words. How many bits are there in each of the following: [4 marks]

Item	Number of Bits
Main Memory Address	
TAG field	
SET field	
Word field	

3. It is proposed to add a direct mapped cache to an existing CPU. The cache will be divided into 16 "word" blocks. It is expected that the hit rate will be 90% for memory reads (Ignore writes). When a "cache hit" occurs, one item is read from the cache and passed to the CPU taking one time unit. When a "cache miss" occurs, a block is read into the cache then one item is passed to the CPU. In this case the time taken is 1+10*Block_size time units. I.e., Main memory access is 10 times slower than cache access.
   a. Is it worth adding the cache? [2 marks]

   b. Suggest an improvement to the cache design [2 marks]

Name:_____

3. A byte-addressable computer has a small data cache that is implement as a block set associative cache with 4 sets each containing 2 blocks capable. Each cache block consists of one 32-bit word. When a given program is executed, the processor reads data from the following sequence of hex addresses:

200, 204, 208, 20C, 2F4, 2F0, 201, 206, 218, 21C, 24C, 264

This pattern is repeated four times.

a) Using the following table show the contents of the cache at the end of each pass. Assume that the cache is initially empty. Express the contents of the cache as the *first* address in the block. [20 marks]

Cache Set	Cache Block	Loop 1	Loop 2	Loop 3	Loop 4
0	0				
	1				
1	0				
	1				
2	0				
	1				
3	0				
	1				

b) Number of Cache Hits [1 mark]?

c) Number of Cache Misses [1 mark]?

d) What is the hit rate for this program [1 marks]?

Name:_____

4. A memory system has pages of size 4096 bytes. Given a part of the page table shown below:

Virtual Page	Real Page Slot
1	6
2	9
3	1
4	5
5	4
6	2
7	3
8	7
9	10
10	8

a) What is the real address corresponding to the virtual address 3231 (The address is in hex)? [2 marks]

b) What is the virtual address corresponding to real address 7890 (The address is in hex)? [2 marks]

Name:_____

Question 4 - CPU Control [10 marks]

Consider the two-bus architecture on the next page. It is being proposed for a new 68000 CPU. The direction of the arrows indicates which bus the register can read or write to. The controls signal G_{enable} ties the two busses together so that data can pass from one bus to the other. Note: G is not a register. It ties the two buses together.

You are to give the steps needed to fetch and execute the following instruction.

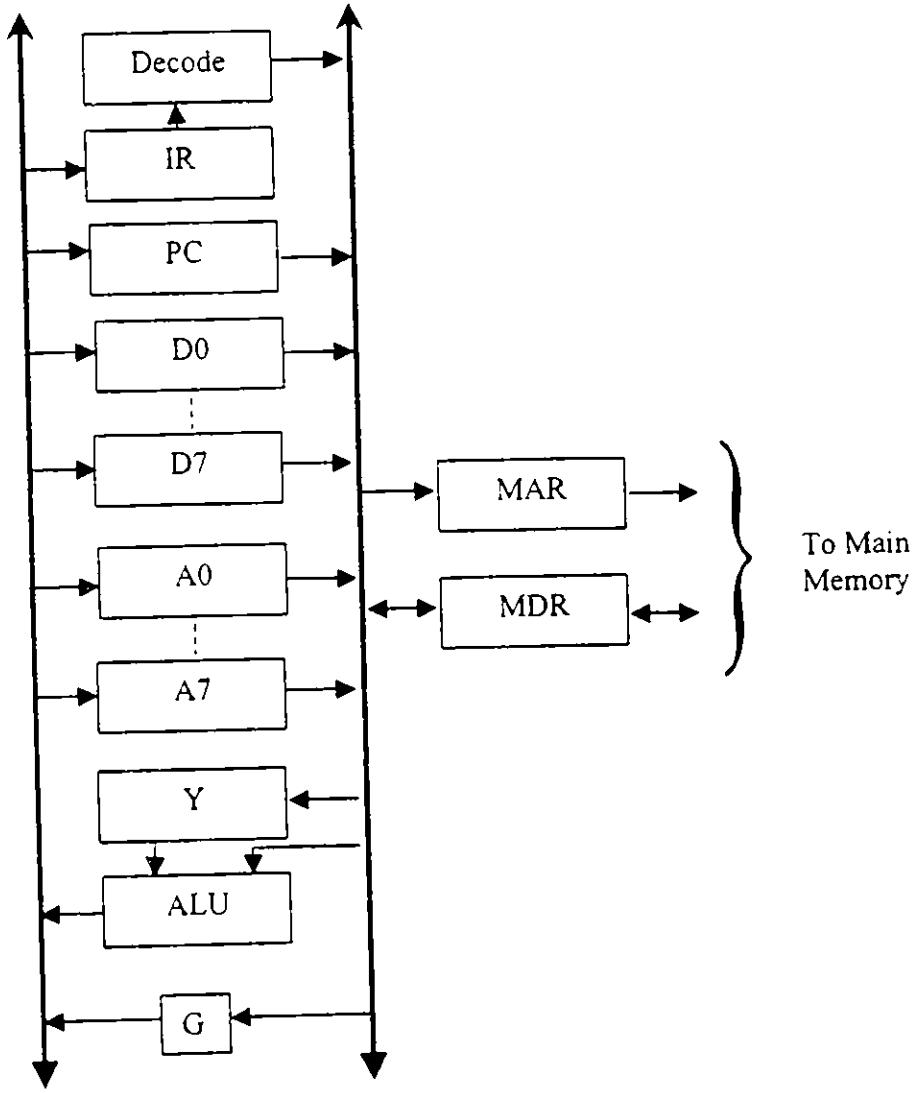
BSR LABEL (Branch to Subroutine)

Assume for each of the registers control signals exist for connecting any register onto the internal CPU bus – e.g. for D1, the control signals would be called $D1_{out}$ and $D2_{in}$.

NOTE:

1. The CPU should behave exactly like the 68000. This means, amongst other things, that the registers are 32 bits and the word size of the machine is 16 bits.
2. Assume that the instruction can be encoded in 16 bits
3. There is a signal called Field_in_IR_{out} that will put the operand to BSR onto the bus
4. You may assume that a register that outputting data on the right bus may also be set to accept data from the left bus in the same clock cycle.

Name:_____



Name:_____

When answering this question fill in the table below (Note: 10 steps are shown but it may not be necessary to use all of them):

Step	
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

Name:_____

Question 5 – Analyzing a Program's Execution [9 marks]

Given the following Motorola 68000 assembly language program you are to analyze its execution and show the contents of the registers and memory at every point in the program's execution. Assume that the contents of registers a0, a1, d0 and d1 before the program beings are listed in the first row of the table below. Complete the table by filling in the values of the various registers *after* the statement labeled State1 has been executed, *after* the statement labeled State2 has been executed, and so on up to State7. You can leave an entry blank if the value in it does not change at that step. Note that your table entries should be a symbol (e.g. DATA) or an offset from a symbol (e.g. DATA+1) whenever possible. Otherwise, your answer should be a number. Be sure to always record the complete long word (32 bit) value stored in each register, even after a word move.

```
        org     $20000
State1  move.l  -(a0),d1
State2  move.l  #2,d0
State3  movea.l -4(a0),a1
State4  move.w  (a1,d0.b),d0
State5  move.w  d0,d1
State6  move.b  -1(a1),d0
State7  movea.l (a0)+,a1
        trap    #15
NUM     dc.l    LIST
        dc.l    $deadbeef
DATA    dc.l    feedbacc
        dc.l    $cad
LIST    dc.b    $12
        dc.b    $34
        dc.b    $56
        dc.b    $78
```

State	Register a0	Register a1	Register d0	Register d1
Initial	Data	$0	$0	$0
State1				
State2				
State3				
State4				
State5				
State6				
State7				

Name:_____

Question 6 – Input/Output [10 marks]

A printer is to be connected to the Ultragizmo boards. The printer has three memory-mapped addresses as indicated below: All of the memory-mapped addresses are 8-bits long.

Address	Read/Write	Function
STATUS	Read only	MSB=1 when ready for a new character LSB=1 when interrupt is pending
CONTROL	Read or Write	LSB=1 to enable interrupts LSB=0 to disable interrupts
DATA	Write only	Write Character to this address to print it

Note: MSB is the Most Significant Bit (Bit 7). LSB is the Least Significant Bit (Bit 0). An interrupt is cleared by writing to DATA

You are to right an ISR that will output a series of characters to the printer up until the first carriage return (use #CR to identify the carriage return in your program). The pointer to the buffer is located in the memory location POINTER. To be clear, POINTER is defined as:

```
POINTER dc.l 0
```

The contents of POINTER are a pointer to the beginning of the string of text to print. You are free to change POINTER in your ISR. When you are done outputting the Buffer set the memory location FLAG to1 and don't forget to disable the printer interrupt.

The following program fragment that begins the printing:

```
        CLR.B   FLAG              ;Clear end of line flag
        MOVE.L  #BUFFER,POINTER   ;Initialize the pointer
        MOVE.B  #1,CONTROL        ;Enable interrupts
WAIT    TST.B   FLAG              ;Has all of the line been printed
        BEQ     WAIT              ;No, then WAIT
        TRAP    #15               ;We are done
```

Note: You are only responsible for writing the ISR. You may assume that all necessary setup to enable interrupts, setup the vector table, etc. has been done.

Put your answer on the following page. **You will be marked on the quality of your comments as well as your code.**

Name:_____

[Answer to Question 6]

Name:_____

Question 7 – CPU Pipeline Operation [10 marks]

1. Assume that there exists a CPU with four pipeline stages:

   - F : Fetch – read the instruction form memory
   - D: Decode – decode the instruction and fetch the source operands
   - E: Execute– Performing the operation
   - W: Write – store the result in the destination location

A diagram that shows the execution of five instructions (that have no dependencies) in the pipeline might look as shown below:

I_1	F_1	D_1	E_1	W_1					
I_2:		F_2	D_2	E_2	W_2				
I_3:			F_3	D_3	E_3	W_3			
I_4:				F_4	D_4	E_4	W_4		
I_5:					F_5	D_5	E_5	W_5	
Clock cycle:	1	2	3	4	5	6	7	8	9

Consider the following instructions, numbered from 1 to 3 for your convenience. Complete the chart on the next page that shows the pipeline contents when executing these instructions. Assume that no pipeline stalls are caused by instruction fetches. Whenever a pipeline stage has nothing useful to do because of a data-dependence pipeline stall, indicate the contents of that stage as being the same as in the previous clock cycle. Assume that an instruction never affects its destination (if applicable) until the end of its W pipeline stage. Also assume that the meanings of the instructions given are similar to the instructions with the same names in the M68000 CPU. Registers R0, R1, etc., are general-purpose CPU registers.

1. sub R1,R4
2. mul R1,R0
3. move R4,R0

Put your answer on the following page.

Name:_____

I_1:												
I_2:												
I_3:												
Clock cycle:	1	2	3	4	5	6	7	8	9	10	11	12

2. What are the 3 types of hazards that can prevent a pipeline from being as efficient as possible [3 marks]

3. How does branch prediction increase performance [2 marks]