University of Toronto
Faculty of Applied Science and Engineering

Final Examination

December 2001
Exam Type: D

ECE341 – Computer Organization

Examiners: P. Anderson, A. Bilas, and A. Moshovos

Instructions

1. This is a type D exam. You are allowed to use the course textbook, the lab manual, one double-sided A4 sheet with personal notes, and a non-programmable calculator

2. There are 6 questions in this exam and one extra credit question. The total value of all questions is 100 marks. The value of each question is indicated. Answer all questions.

3. Use only the space provided on these sheets. Do not remove any sheets from this test book. An examination book is available for rough work and must also be handed in at the end of the exam. Only the additional aids specified in (1) are permitted.

4. Write your name and student number in the space below. Write your student number on the top of each sheet of this test book.

Last Name (Print): ________________

First Name: ______________________

Student Number: _________________

(Do not write below this line)

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Extra Credit	
Total	

Question 1 (15 marks): Addressing Modes and Simple Operations

Assume that memory is initialized as follows:

```
        org   $30000
        dc.b  $04,$ff
        dc.w  $abcd
FOO     dc.l  $0f2e3d4c
        dc.b  $00,$01,$f0,$a1
```

Show the values of the registers listed on the rightmost column after the execution of each of the following instructions (and the ones before it). Assume that all instructions execute in the order shown and that initially all registers are zero.

```
1.  move.l  #$30000, a0      a0 =

2.  move.b  $30000, d1       d1 =

3.  or.b    d1,d1            d1 =

4.  lsr.b   #1,d1            d1 =

5.  move.b  3(a0), d2        d2 =

6.  move.w  (a0,d1), d3      d3 =

7.  move.l  FOO, d4          d4 =

8.  move.l  #FOO, a0         a0 =

9.  move.l  (a0)+, d1        d1 =

10. move.w  (a0)+, d2        d2 =

11. move.b  (a0), d1         d1 =
```

Question 2 (15 marks): Alternate Assembly Instructions

The 68k instruction set was extended to include multimedia oriented instructions. These instructions operate on the existing 32-bit registers treating them as a collection of four individual bytes. For example, the "`add4b d0,d1`" instruction adds the contents of registers d0 and d1 treats this as four independent adds as follows:

	Byte 3	Byte 2	Byte 1	Byte 0
D0	$0f	$01	$02	$ff
	+	+	+	+
D1	$01	$02	$05	$01
	=	=	=	=
Result in D1	$10	$03	$07	$00

Notice that all four additions are done independently and do not affect each other. For example, while the addition of the bytes labeled "byte 0" overflows this does not affect the values next to them.

The following code was written in the original 68k assembly to add two byte arrays ARR1 and ARR2 of NELEMS bytes producing a third array OUT:

```
        move.l      #ARR1, a0
        move.l      #ARR2,a1
        move.l      #OUT,a2
        move.l      #NELEMS, d3
        subq.l      #1, d3
Loop:   move.b      (a0)+, d0
        add.b       (a1)+, d0
        move.b      d0,(a2)+
        dbf         d3, loop
```

How would you change the code to improve performance now that `add4b` is available? Assume that ARR1, ARR2 and OUT are always non-overlapping and are word aligned (i.e., you can do move.l safely). Assume that all instructions take a single cycle to complete including `add4b`. Write your new code in the space provided below.

Question 3 (20 marks): Caches

a) A cache holds a total of 2Mbytes (2 x 2^20) of data. Each block is 64 bytes long. The particular machine has a 256Gbyte (256 x 2^30) address space.

i) How many bits are used for memory addresses?

ii) How many blocks are there in the cache?

iii) How many bits are used for the TAG, SET and OFFSET fields for a direct mapped cache?

iv) How many bits are used for the TAG, SET and OFFSET fields for an 8-way set-associative cache? Remember, the total cache size remains the same even if the associativity changes.

b) A cache has only 4 blocks each of them holding a single byte. The address space of the particular machine is 256 bytes. You can organize the cache as either direct mapped or 4-way set associative. Report the **number of hits and the references** that hit for the following reference sequences assuming that the cache uses the LRU (least recently used) replacement policy. A reference sequence is a sequence of memory addresses generated by the CPU one after the other. Please circle the references that hit.

i) `Reference sequence: 0,4,8,12,0,4,8,12,0,4,8,12`

(1) Direct mapped case hits: `0,4,8,12,0,4,8,12,0,4,8,12`

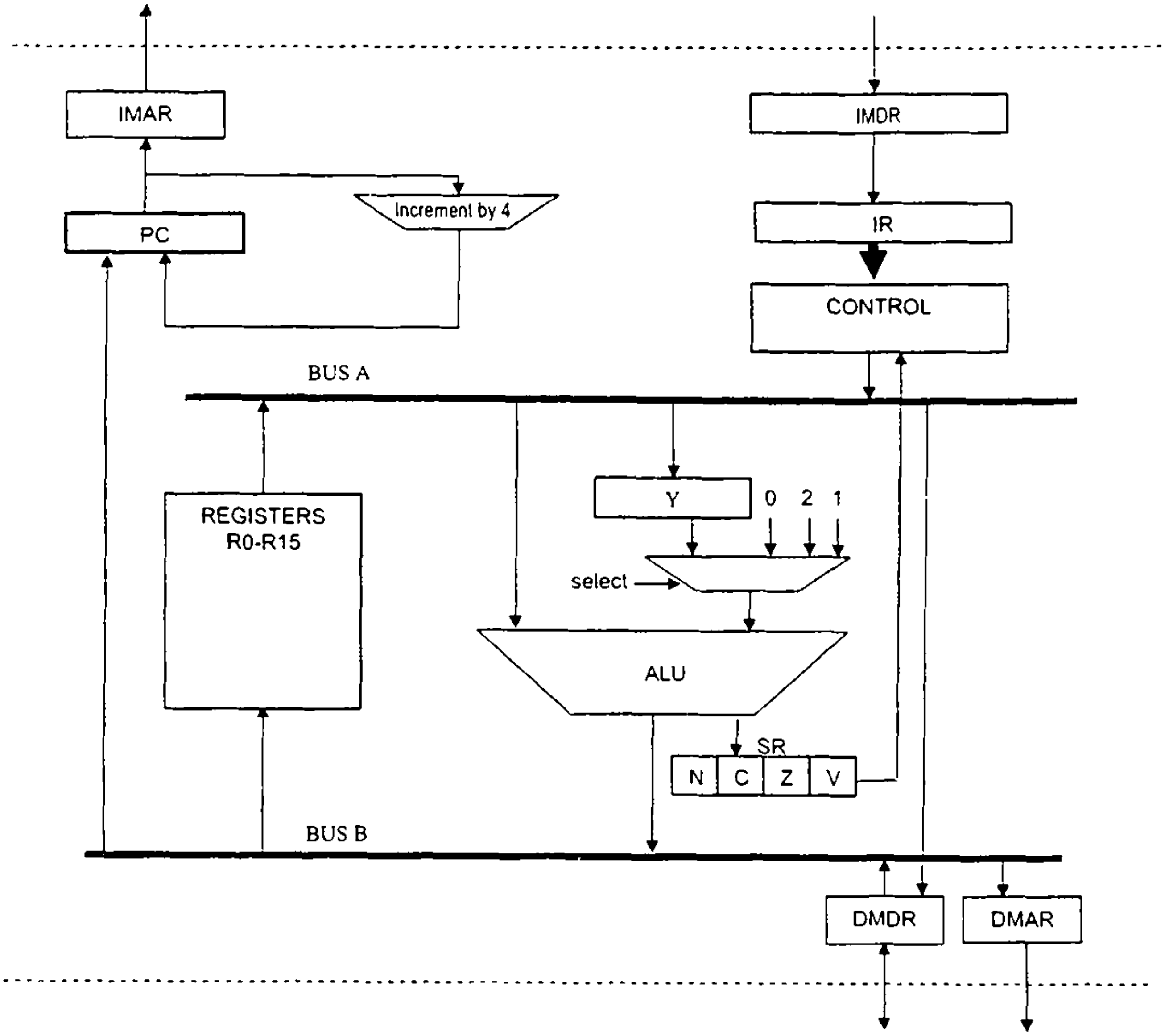
(2) 4-way set associative case hits: `0,4,8,12,0,4,8,12,0,4,8,12`

ii) `Reference sequence: 0,1,2,3,7,0,1,2,3,7,0,1,2,3,7`

(1) Direct mapped case hits: `0,4,8,12,0,4,8,12,0,4,8,12`

(2) 4-way set associative case hits: `0,4,8,12,0,4,8,12,0,4,8,12`

Question 4 (30 marks): Microprocessor Control



All of these questions are with regard to the architecture in the figure above. Note that there are two different sets of registers for memory access, the MDR_I / MAR_I and the MDR_D / MAR_D. There are also two separate memory paths. TMP1 and TMP2 are internal temporary registers that can be accessed directly by control signals. Assume all instructions are 4 bytes long. All registers shown have two control signals NAMEin and NAMEout. The arrows indicate unidirectional connections. For example, the ALU result may appear only on BUS B while a R0 may place its value only on BUS A. Assume that the ALU can perform addition, subtraction, logical, and arithmetic shifts and bitwise AND, OR, XOR, and NOT.

NAME:________________________________

a) (5 marks) Briefly, what might be the advantages of this organization?

b) (5 marks) Briefly (one or two sentences), what advantage might be gained by adding a queue where the IR is?

c) (20 marks) Write machine control sequences for the following two instrucıions. Include the instruction fetch steps and outline any assumptions you are making about memory coordination. Copy and label the drawing as you need to in order to clarify your answer, and add to the drawing (on your booklet) any additional datapath components (muxes, constants, paths/connections) you feel are required.

i) (10 marks) ADD (R1), R2:
R1 and R2 are registers in the register file

ii) (10 marks) CLRALL:
sets all registers R0 through R15 to zero.

NAME:________________________________

Question 5 (10 marks): System Timing

A computer has memory and devices that work at various speeds. Here we are using memory in its very general sense, not just as a synonym for 'main memory'. Using the numbers 1 through 10 order the operations in the following table by speed, where 1 is the fastest and 10 the slowest. Do not duplicate numbers. Assume a single byte or other convenient minimum piece of information is transferred in each case unless otherwise stated. Also, assume that the processor operates at 1Ghz. Calculate the times for the transfers over the physical links for f through i. Assume a 100 MHz memory bus. If you need to clarify your decisions by including other assumptions, do it below. Aid: Light travels about 30 cm in one nanosecond.

Item	*Speed Order (1=fastest)*	*Operation*
a		Memory fetch in a processor with no cache
b		Memory fetch from a cache
c		Memory fetch for a processor with cache where there is a cache miss
d		Transfer of data from a register to the ALU in the processor
e		Average time to fetch of a 256 byte block of information from a floppy drive with 94 msec access time and a 500 kbit/sec transfer rate
f		Transfer of a 256 byte block from one computer to another over a 10 MBit Ethernet link
g		Transfer of a character through a UART at 1200 baud
h		Transfer of a character through a UART at 9600 baud
i		Transfer of a character over a parallel interface that can be up to 100 m in distance.
j		Writing this exam.

Question 6 (10 marks): Buses/Memory

Suppose we have a memory system that uses a 50-MHz clock. The memory transmits 8-word requests at the rate of 1 word per cycle. For reads from memory, the accesses occur as follows:

1. 1 cycle to accept the address,
2. 3 cycles of delay, and
3. 8 clock cycles to transmit the 8 words

For writes to memory, accesses occur as follows:

1. 1 cycle to accept the address,
2. 2 cycles of delay
3. 8 clock cycles to transmit the 8 words, and
4. 3 cycles to complete the transaction (this includes the actual write in the memory and updating related information such as the error correction code).

The **bus bandwidth** is defined as the number of bytes that can be transferred in time T over T, and is usually measured in bytes/s: bandwidth = (# bytes transferred in T s)/T (bytes/s). For example, 16Kbytes/sec suggests that in one second 16Kbytes can be transferred, while it takes 4 seconds to transfer 64Kbytes. For the purposes of this question bus bandwidth includes memory latency.

Find the **maximum** bandwidth in megabytes per second for an access pattern consisting of:

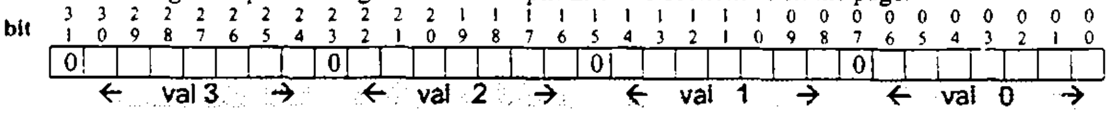
a) All reads from memory.

b) All writes to memory.

c) A mix of 60% reads from memory and 40% writes to memory.

EXTRA CREDIT Question (10 marks): Implementing Matrix Manipulation Instructions

Many modem machines have multimedia oriented instructions that treat registers as four independent bytes similarly to what we saw in question 3. There is a "poor man's" alternative that leverages existing instructions. The idea is to use a 32-bit register to store four independent 7-bit numbers. The four 7-bit numbers are separated using zero bits in between them as shown in the following example. The figure had to be split into two sections to fit the page.



a) Implement the equivalent of add4b. To do so, use existing 68k instructions such as add.l, and.l, or.l, etc. You will probably need more than one instruction. Provide the code for add4b d1,d2.

b) Some machines implemented saturated unsigned integer arithmetic where on overflow the resulting value is the maximum integer (this is useful, for example, when numbers represent force or other physical quantities). Implement the equivalent of add4b with saturated arithmetic using existing 68000 instructions. Provide the equivalent code for add4b d1,d2.

For both (a) and (b) try to minimize the number of instructions and registers you need. In addition, avoid using memory for storing values.