EEAP 282

EXAM #1
SOLUTIONS

September 26, 1997

NAME: _____ CWRUnet ID:_____

IMPORTANT INFORMATION:

1. All questions are worth TEN (10) points apiece. There are NINE questions.
2. Exam is closed book, closed notes. Only the M68000 Programmer's Reference Manual and/or Programming Reference Card are allowed to be used.

Problem	Score
1	
2	
3	
4	
5	
6	
7	
8	
9	
TOTAL SCORE	

Notation used: $ indicates hex, % indicates binary, @ indicates octal.
The following logic functions may be needed at various points throughout the exam.

A	B	A OR B	A AND B	A EOR B
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

?? people took exam in 1995

1. Where are each of the following commands used (vi, UNIX, debugger, etc) and describe their function (a sentence or two):

		where	function
a)	dd	vi	delete current line
b)	:wq	vi	save file and quit
c)	x	vi	delete character under cursor
d)	cd ..	UNIX	change directory one level up

2. Represent -375_{10} in
(a) 16 bit two's complement representation (Give your answer in hex)

ANSWER:
Convert 375 to binary:
29_{10} = %0000 0001 0111 0111
1's complement = %1111 1110 1000 1000
2's complement = %1111 1110 1000 1001 = $FE89

(b) 16 bit signed magnitude representation

ANSWER:
29_{10} = %0000 0001 0111 0111
signed magnitude = %1000 0001 0111 0111 = $8177

3. (a) Add the two following 16-bit 2's complement numbers. What is the answer?

$DFEA
$A764

Did signed overflow occur? Yes or No

ANSWER:
$874E There is no signed overflow.

(b) Add the same two 16-bit numbers which are now unsigned. Does the result change?

$DFEA
$A764

Is there unsigned overflow? Yes or No

ANSWER: The result is still the same. There is unsigned overflow, i.e. a carry.

4. You bought a new video processor board for your computer. To set it for hardware MPEG you need set the board's firmware. Specifically, you must set bits 3, 4, and 5 of the byte at $14400 to 1,0 and 1 respectively. Furthermore, bit 0 must be set to zero. All other bits must remain unchanged.

Describe how to do this operation with masks. Specify each mask and logical operation that you use. **Give your masks in hex**.

ANSWER:
Do an AND to set bits to 0; an OR to set bits to 1.

To set bit 0 and 4 to zero.
ANDI.B #%1110 1110,$14400 or ANDI.B #$EE,$8000

This then sets bits 3 and 5 to one.
ORI.B #%0010 1000,$80000 or ORI.B #$28,$8000

5. Give the memory map (i.e., the contents of memory) corresponding to the following sequence of assembler directives.

```
        ORG     $5000
TABLE1  DC.L    $00609000
        DC.B    $82,12
        DC.L    $9008AB12
        DC.W    $0100,$0090
        DC.L    $FF00
```

Indicate memory contents using the following table.

$5000	00	60
$5002	90	00
$5004	82	0C (hex)
$5006	90	08
$5008	AB	12
$500A	01	00
$500C	00	90
$500E	00	00
$5010	FF	00
$5012		
$5014		
$5016		
$5018		

$501A		
$501C		
$501E		

6. Starting at address $6000, memory contains the following sequence of 16-bit values:

```
$D3EE   $01CA   $EEAA   $B00F   $F00D   $3214
```

(a) What is the hex value of the byte contents of address $6005?

(b) What is the hex value of the long word contents of address $6006?

ANSWERS: The contents of memory are:

$6000	$D3
$6001	$EE
$6002	$01
$6003	$CA
$6004	$EE
$6005	$AA
$6006	$B0
$6007	$0F
$6008	$F0
$6009	$0D
$600A	$0D
$600B	$0D

($6005.B) = $AA
($6006.L) = $B00F F00D

7. That crazy CWRU professor who designs all the strange processors is back. He has designed a new, specialized processor based upon a 0.35μm CMOS process which has an 8-bit data bus and a 24 bit address bus. The processor operates at 64 MHz and is interfaced to a 16 bit A/D converter which looks like a 16-bits of memory at $88000. Assume that the processor can perform a complete read/write in one clock cycle.

(a) What is the range of memory address (in hex) that this microprocessor can access?

ANSWER: The address range is defined by the address bus being all zero's and all one's, i.e. %0000 0000 0000 0000 0000 0000 to %1111 1111 1111 1111 1111 1111 ($000000 to $FFFFFF).

(b) What is the maximum number of bytes of memory this machine can have? Be exact.

ANSWER: Since the address bus is 24 bits the maximum amount of memory is 2^24 = 16,777,216 or 16 MB.

We will accept as correct 16,777,216, 16,777,215, or 16,777,214. This will give many people an extra 4 points.

8. What is the instruction(s) performed by the following machine code in memory? Express your answer in hex. Hint: You need only consider instructions of the form MOVE, ADD, SUB, ADDI or MOVE.

```
address     word        ANSWER
$8000     $3039       MOVE.W    $00300330,D0
$8002     $0030
$8004     $0330
$8006     $0485       SUBI.L    #$00209834,D5
$8008     $0020
$800A     $9834
```

ANSWER:
From the Programmer's Reference Manual we decode the instruction according to the format:

```
$3039
%0011 0000 0011 1001
%0011 000 0 00 11 1 001

op    destination    source
 code   reg  mode   mode  reg
%0011   000  000    111   001
```

This gives us a 00 op code which is a MOVE. The next two 11 bits indicate that it is a word length MOVE.W. The souce is 111 which gives us several choices. The register 001 identifies it as (xxx).L which means a long word address follows in two extension words, i.e. the long word source address is $0030 0330. The destination is mode 000 which is addressing mode Dn and the register number is 000. So the complete disassembled instruction is
MOVE.W $0030 0330,D0

This means that $0485 0020 9834 is a separate instruction to be disassembled.

```
$0485
%0000 0100  1000 0101
%0000   0100  10 000 101

  op             eff address
 code      size  mode  reg
%00000100  10    000   101
```

Since the destination effective address is specified by mode=000, reg=000 This specifies D5 as the destination register. The complete instruction is then

```
SUBI.L  #$00209834,D5
```

Machine code	assembly language	
0443 <number>	SUBI.W	#N,D3
0479 <number>,<long address>	SUBI.W	#N,<long address>
0478 <number>,<word address>	SUBI.W	#N,<word address>
9679<long address>	SUB.W	<long address>,D3
9678<word address>	SUB.W	<word address>,D3
9779 <long address>	SUB.W	D3,<long address>
9778 <word address>	SUB.W	D3,<word address>
0643 <number>	ADDI.W	#N,D3
0679 <number>,<long address>	ADDI.W	#N,<long address>
0678 <number>,<word address>	ADDI.W	#N,<word address>
D678 <word address>	ADD.W	<word address>,D3
D778 <word address>	ADD.W	D3,<word address>
D679 <long address>	ADD.W	<long address>,D3
D779 <long address>	ADD.W	D3,<long address>
363C <number>	MOVE.W	#N,D3
33FC <number>,<long address>	MOVE.W	#N,<long address>
3639 <long word>	MOVE.W	<long address>,D3
31C3 <word address>	MOVE.W	D3,<word address>
33C3 <long address>	MOVE.W	D3,<long address>
4EB9 <address>	BRA	<address>
4EF9 <address>	JMP	<address>

9. Using the table given above, decode the machine instructions shown below i.e. what is the assembly equivalent of the machine code?

Address	Machine Code	ANSWERS
$9000	$3639	MOVE.W $00009502,D3
$9002	$0000	
$9004	$9502	
$9006	$D678	ADD.W #$9504,D3
$9008	$9504	
$900A	$33C3	MOVE.W D3,$9010
$900C	$0000	
$900E	$9010	
$9010	$0443	SUBI.W D3,$9500
$9012	$9500	
$9014	$0000	
$9016	$0000	
. . .		
$9500	$0600	
$9502	$3000	
$9504	$01C3	

ANSWER:
MOVE.W $009502,D3
ADD.W $FF9504,D3 <--very subtle, actually sign extends

MOVE.W D3,$009010
SUBI.W #$9500,D3

You were not asked to execute this program and points were taken off if you gave MOVE.W D3,$9500 rather than SUBI.W #$9500,D3 If you actually execute this program it becomes:

```
Address     Machine Code      ANSWERS
$9000       $3639             MOVE.W $00009502,D3; (D3)=$xxxx3000
$9002       $0000
$9004       $9502
$9006       $D678             ADD.W     #$9504,D3;
(D3.W)=$3000+$01C3=$31C3
$9008       $9504
$900A       $33C3             MOVE.W    D3,$9010
$900C       $0000
$900E       $9010
$9010       $0443             Becomes $31C3; MOVE.W D3,$9500
$9012       $9500             Puts $31C3 into $9500
$9014       $0000
$9016       $0000
. . .
$9500       $0600             Becomes $31C3
$9502       $3000
$9504       $01C3
```