

Here are some sample exams and questions that you can study for your first exam. These are actual problems that have been used on old exams. I have tried to eliminate all the problems of types which will not be on our Friday exam but might have missed a few.

Good luck studying!

Section 1--Computer architecture: organization of the cpu and memory, address and data bus, fetch-execute cycle

1. Consider the hypothetical CASE 1990 microprocessor. It has a 22 bit data bus and a 32 bit address bus. The memory is organized as 11 bit bytes and the smallest unit of memory that the CASE1990 can access is an 11 bit byte. WHAT IS THE RANGE OF MEMORY ADDRESSES THAT THE 1990 CAN ACCESS?

ANSWER: The data bus information is almost irrelevant to the problem. The CASE 1990 can access 2^{32} (\$100000000) memory addresses. This translates to into addresses between \$0000 0000 to \$ FFFF FFFF.

Section 2--UNIX, vi: simple questions about what UNIX and vi commands do

2. 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)	ls	UNIX	display current directory contents
d)	cd ..	UNIX	change directory to parent directory

Section 3--number systems:hexadecimal, binary

4. Using 8 bit binary numbers, represent the decimal number -18 in :

- a) two's complement representation
- b) signed magnitude representation

ANSWER:

- a) $18 = \$12 = \% 0001\ 0010$ which becomes in one's complement as $\%1110\ 1101$. Add 1 to get two's complement solution, which is $\%1110\ 1110$
- b) in signed magnitude, the answer is : $\%1001\ 0010$

5. What are the decimal equivalents of :

- a) $\%00101.1011$
- b) $\$-7A4C$

ANSWER:

- a) $1*2^{**2}+1*2^{**0}+1*2^{**-1}+1*2^{**-3}+1*2^{**-4} = 5.6875$
- b) $\$-7A4C = \% -0111\ 1010\ 0100\ 1100 = -31308$

Section 4--mathematics: addition, subtraction, 2's complement

6. Perform the indicated calculations (all values are 2's complement)

- a) $AF5D+50A3$
- b) $B33C+9555$
- c) $3D5C-D4EF$

ANSWER:

- a) $1\ 0000$
- b) $1\ 4891$
- c) $1\ 686D$ (negative)

Section 5--68000 instructions: assembly, disassembly of simple 68000 instructions (typically add, subtract and move)

7. What is the machine code for the following instruction? Express your answer in hex.

ADD.W D0, \$5000

ANSWER: D178 5000

8. Disassemble the machine code instruction 0479 FFFA 0010 000A. Express your answer as a 68000 assembly language mnemonic complete with appropriate size, extension, etc.

ANSWER: SUBI.W # \$FFFA, \$10000A

EEAP 282: SAMPLE EXAM #2 with solutions (originally used on 10/2/91)

Questions 10 and 11 would require that I give you a table of 68000 op words .

- 2) Exam is closed book, closed notes. Only the M68000 Programmer's Reference Manual and/or Programming Reference Card are allowed to be used.
- 3) \$ denotes hexadecimal numbers; % denotes binary numbers
- 4) x**y means, in this sample test, means x to the power y, while x*y means x times y
- 5) If you have any questions, please send e-mail to ee282

Section 1--Computer architecture: organization of the cpu and memory, address and data bus, fetch-execute cycle

1. Consider the hypothetical CASE 1991 microprocessor. It has a 24 bit data bus and a 17 bit address bus. The memory is organized as 8 bit bytes and the smallest unit of memory that the CASE1991 can access is a 8 bit byte.

- a) The CASE 1991 is a _____ bit machine.
- b) What is the range of memory addresses taht the 1991 can access?

ANSWER: a) 24-bit; only the data bus is relevant
 b) \$0 0000 to \$1 FFFF. The address range is determined by the width of the address bus.

Section 2--UNIX, vi: simple questions about what UNIX and vi commands do, also included will be how to invoke the assembler, linker and debugger and what each uses as input and output

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

- | | where | function |
|--------|-------|---|
| a) lp | UNIX | direct the following file to the printer |
| b) :q! | vi | in command mode, exit without saving |
| c) man | UNIX | display documentation about following subject |
| d) x | vi | delete character under cursor |

Section 3--number systems:hexadecimal, binary

4. Represent the decimal number -96 in :
- 16 bit two's complement representation
 - 16 bit signed magnitude representation

ANSWER:

- \$FFA0 = %1111 1111 1010 0000
- 96 = \$60; in signed magnitude representation this becomes
\$8060 = %1000 0000 0110 0000

5. What are the decimal equivalents of :

- %1111 1111.1010 0000 (this is in sign-extended 2's complement representation.
- \$+04F B

ANSWER:

- 0.375. The number must be treated as a fixed length (16-bit) binary number, i.e. 1111 1111.1010 0000. You then proceed with the reverse of 2's complementing. Instead of adding one, you subtract 1 from 1111 1111.1010 0000 to get 1111 1111 1001 1111, and then complement to get 0000 0000.0110 0000 which is 0.375
- Simple evaluation leads to +1275

Section 4--mathematics: addition, subtraction, 2's complement

6. Perform the indicated calculations (all values are 2's complement 16 bit numbers)

- A5A5+5A5A
- FFFF+0001
- 1234+4321

ANSWER:

- | | | | |
|----------|---------|----------|---------|
| | a) FFFF | b) 10000 | c) 5555 |
| CARRY | NO | YES | NO |
| OVERFLOW | NO | NO | NO |

Section 5--68000 instructions: assembly, disassembly of simple 68000 instructions (typically add, subtract and move)

7. What is the machine code for the following instruction?
Express your answer in hex.

MOVEA.L \$5000,A5

- ANSWER: 0010 101 001 111 000
Using (xxx).W we get: 0010 1010 0111 1000 = \$2A78
The answer is 2A78 5000

0010 1010 0111 1001 to use (xxx).L = \$2A79 would also be acceptable giving us a different answer:
2A79 0000 5000

Section 7--68000 programming: assembler format, how to write short programs using a very small number of 68000 instructions (mostly add, subtract and move), what a set of given instructions will do.

9. Assume (D0)=\$0005 3F02, (D1)=\$FFFF 0110

```

                                ORG                $1000
                                ADD.L              #DATA,D0          (a)
                                MOVE.W          DATA,D1          (b)

                                ORG                $2000
DATA                            DC.B              $0A,$EE,$83,$82
                                DC.B              $0A,$EE,$30,$00

```

- a.) What is in D0 after the instruction labeled (a) is executed?
- b.) What is in D1 after the instruction labeled (b) is executed?

ANSWER: a) \$00 05 5F 02 b) \$FF FF 0A EE

These following two problems would be tougher than I would give you. Assume that your assembler outputs the following program segment and answer the the following two questions:

```

                                ;ANSWER
000500  3A39 0000 5000  ;MOVE.W $0000 5000,D5
000506  0645 0045      ;ADD.W  #$45,D5
                                ;ADDI.W is also acceptable
00050A  33C5 0000 0510  ;MOVE.W D5,$0000 0510
000510  0445 0002      ;SUB.W  #$2,D5
000514  0000 0000      ;NOT AN INSTRUCTION(*)

005000  0600          ;JUST DATA
005002  0777

```

10. Disassemble the first five lines of the above program listing. Express the program in 68000 mnemonics being sure to explicitly show the size extension.

EXAM #1 SOLUTIONS
October 2, 1992

computer architecture:

1. Consider the hypothetical Case 1992 microprocessor. It has a 10 bit data bus and a 16 bit address bus. The memory is organized as 5-bit bytes and the smallest unit of memory that the 1992 can access is a 5 bit byte.

(a) What is the range of memory addresses that the 1992 can access?

ANS: \$0000 to \$FFFF. The address range is determined by the width of the address bus.

The following coded program data (i.e. instructions) is stored in the 1992's memory beginning at memory location \$1000.

binary data

10010 <----\$1000, first instruction begins here

10001

11110

10111

00100

11101

00010

00001

01010<----next instruction begins here

11000

(b) How many fetch cycles are necessary to completely fetch the instruction? How many execute cycles?

ANS: Four (4) fetch cycles since the data bus fetches 10 bits at a time; only one execute cycle. There was no explicit reference to clock cycles which some students tried to answer the question with.

(c) What is the address of the next instruction?

ANS: \$1008

2. Starting at memory address \$2000, memory contains the following sequence of 16-bit values in consecutive memory locations.

\$1000--> \$3039
\$9070
\$D079
\$33C0
\$4E40

(a) The byte at \$2003 is:
ANS: \$70.

(b) The word at \$2006 is:
ANS: \$33C0.

UNIX, vi:

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

- (a) ESC vi exit text entry mode
- (b) :wq vi go to command mode, write the current buffer to a file (assigned when vi was invoked), and exit vi
- (c) cd UNIX return to the user's home directory

number systems:

5. What are the decimal equivalents of
(a) 1011 01112 (this is in sign-extended 2's complement notation) =

ANS: -73 base 10

The trick is to simply reverse the 2's complement procedure. Since the number is in binary, subtract 0000 0001 from it to get 1011 0110. Complement the result, 0100 1001. This is the hexadecimal number \$49. The original number was \$-49 which is -73 decimal.

(b) 67.2 base 16 =

ANS: This is simple evaluation
 $6 \times 16^1 + 7 \times 16^0 + 2 \times 16^{-1} = 96 + 7 + 0.125 = 103.125$

mathematics:

6. Perform the indicated calculations (IMPORTANT: All values, including the results, are two's complement 16-bit numbers!)

		RESULT	V	C
(a)	\$1732+\$4631	\$5D63	NO	NO
(b)	\$F1A5+\$B8D5	\$(1)AA7A	NO	YES
(c)	\$F1A5+\$D5	\$(1)F17A	NO	YES

The carries are explicitly shown as (1) but will not appear in the 16-bit answer.

NOTE: There was some discussion about whether the wording of the question requires sign extension in part (c). This is still being considered.

68000 instructions: assembly, disassembly of simple 68000 instructions

7. Disassemble the machine code instruction(s)

\$23F9 0007 FFFF 0640 002C

Express your answer in terms of 68000 assembly language mnemonics complete with appropriate size extension.

ANS:

The instruction word in binary is:

0010 0011 1111 1001

This can be decoded as

0010 001 111 111 001

where the 0010 indicates a long word MOVE. The destination is specified by mode=111, register=001 which corresponds to (xxx).L.

The source is specified by mode=111, register=001 which also corresponds to (xxx).L The correct response is then MOVE.L \$0007FFFF,\$0640002C

68000 programming:

9. A MC68000 system has 32k of memory.

(a) What is its address range? Give your answer in hex.

ANS: \$000000-\$007FFF

(b) Indicate if the following instructions will execute without error.

(i) MOVE.W #\$8000,D0

ANS: No error, \$8000 is not an address

(ii) MOVE.W \$008000,D0

ANS: Error, no memory above \$7FFF

NOTE: This was virtually identical to a homework

problem.

10. What is in D0 after the following machine code executes?

(D0.L) = _____

ANSWER: \$XXXX 043A where xxxx indicates that the contents are unchanged.

The following table may be of use:

Machine code	assembly language
0440 <number>	SUBI.W #N,D0
0479 <number>,<address>	SUBI.W #N,<address>
0640 <number>	ADDI.W #N,D0
0679 <number>,<address>	ADDI.W #N,<address>
303C <number>	MOVE.W #N,D0
33FC <number>,<address>	MOVE.W #N,<address>
3039 <address>	MOVE.W <address>,D0
33C0 <address>	MOVE.W D0,<address>
4EB9 <address>	BRA <address>
4EF9 <address>	JMP <address>
9079 <address>	SUB.W <address>,D0
9179 <address>	SUB.W D0,<address>
D079 <address>	ADD.W <address>,D0
D179 <address>	ADD.W D0,<address>

NOTE: All numbers are in hex!!!!

address	code
1000	303C
1002	0020
1004	D079
1006	0000
1008	0020
100A	0640
100C	0020
100E	33C0
1010	0000
1012	1014
1014	0640
1016	0005

address	data
0020	0400

0022 0002

ANSWER:

Including the extension words, this becomes:

```
00001000 303C 0020            MOVE.W    #20,D0
00001004 D079 0000 0020        ADD.W     $000020,D0
0000100A 0640 0020            ADDI.W    #20,D0
0000100E 33C0 0000 1014        MOVE.W    D0,$001014
00001014 0640 0005            ADDI.W    #5,D0
00001018 4E72 4E72            STOP      #4E72
```

This program puts \$0020 into D0, adds \$0400 to it to get \$0420, adds another \$0020 to it to get \$0440, and then puts the contents of D0 into memory location \$1014. \$1014 is the location of the next instruction word and this changes the instruction to \$0440 0005 which, when disassembled, is SUBI.W #\$0005,D0 which changes the final value of D0 to \$0440-\$0005=\$043A. Note that the upper word of D0 was not changed by this program.

11. Assume (D0)=\$ 0005 3F02, (D1) = \$ FFFF 0110

```
          ORG        $1000
          MOVE.W    #A,D0    (a)
          MOVE.B    CNT,D1   (b)

          ORG        $2000
CNT    DC.W        $1
BLANK DC.L        $FFA0
SUM    DS.W        1
ICNT   DC.W        3
```

(a) What is in memory beginning at memory location \$2000. Any memory contents not explicitly stated by the code may be indicated by "xx," i.e. if the program doesn't put anything in that location its contents are "xx."

address	contents
\$2000	\$00
\$2001	\$01
\$2002	\$00
\$2003	\$00
\$2004	\$FF
\$2005	\$A0
\$2006	xx
\$2007	xx

\$2008	\$00
\$2009	\$03
\$200A	xx
\$200B	xx
\$200C	xx
\$200D	xx

(b) What is in D1 after the instruction labeled (b) is executed?

(D1) = _____

ANS: (D1)=FFFF0100