**Review**

**4.1** Name the five components of the von Neumann model. For each component, state its purpose.

ALU-

Memory-

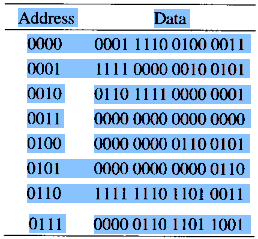
Control Unit-

Input Unit-

Output Unit

**4.2** Briefly describe the interface between the memory and the processing unit. That is, describe the method by which the memory and the processing unit communicate.

**4.5** The following table represents a small memory. Refer to this table for the following questions.



**A.** What binary value does location 3 contain? Location 6?

**B.** The binary value within each location can be interpreted in many

ways. We have seen that binary values can represent unsigned

numbers, 2's complement signed numbers, floating point numbers,

and so forth.

1. Interpret location 0 and location 1 as 2's complement integers.

**(2)** Interpret location 4 as an ASCII value.

**(3)** Interpret locations 6 and 7 as an IEEE floating point number. Location 6 contains number [15:0]. Location 7 contains number [31:16].

**(4)** Interpret location 0 and location 1 as unsigned integers.

**C.** In the von Neumann model, the contents of a memory location can also be an instruction. If the binary pattern in location 0 were interpreted as an instruction, what instruction would it represent?

**D.** A binary value can also be interpreted as a memory address. Say the value stored in location 5 is a memory address. To which location does it refer? What binary value does that location contain?

**4.6** What are the two components of an instruction? What informationdo these two components contain?

**4.8** Suppose a 32-bit instruction takes the following format If there are 225 opcodes and 120 registers,

**A.** What is the minimum number of bits required to represent the OPCODE?

**B.** What is the minimum number of bits required to represent the Destination Register (DR)?

**C.** What is maximum number of UNUSED bits in the instruction encoding?

**4.9** The FETCH phase of the instruction cycle does two important things.

One is that it loads the instruction to be processed next into the IR. What

is the other important thing?

**4.11**   State the phases of the instruction cycle and briefly describe what

operations occur in each phase.

**5.1** Given instructions ADD, JMP, LEA, and NOT, identify whether the

instructions are operate instructions, data movement instructions, or

control instructions. For each instruction, list the addressing modes that

can be used with the instruction

**5.2** A memory's addressability is 64 bits. What does that tell you about the

size of the MAR and MDR?

**5.3** There are two common ways to terminate a loop. One way uses a counter

to keep track of the number of iterations. The other way uses an element

called a \_\_\_\_\_\_\_\_\_. What is the distinguishing characteristic of this element?

**5.5**

*a.* What is an addressing mode?

*b.* Name three places an instruction's operands might be located.

c. List the five addressing modes of the LC-3, and for each one state

where the operand is located (from part b).

*d.* What addressing mode is used by the ADD instruction shown in

Section 5.1.2?

**5.7** What is the largest positive number we can represent literally (i.e., as an

immediate value) within an LC-3 ADD instruction?

**5.8** We want to increase the number of registers that we can specify in the

LC-3 ADD instruction to 32. Do you see any problem with that?

Explain.

**5.13**

***a*.** How might one use a single LC-3 instruction to move the value in R2

into R3?

***b.*** The LC-3 has no subtract instruction. How could one perform the

following operation using only three LC-3 instructions:

***c.*** Using only one LC-3 instruction and without changing the contents of

any register, how might one set the condition codes based on the value

that resides in Rl?

***d.*** Is there a sequence of LC-3 instructions that will cause the condition

codes at the end of the sequence to be N = 1, Z = 1, and P = 0?

Explain.

***e.*** Write an LC-3 instruction that clears the contents of R2

**5.16** Which LC-3 addressing mode makes the most sense to use under the following conditions. (There may be more than one correct answer to each of these; therefore, justify your answers with some explanation.)

**A.** You want to load one value from an address which is less than ±28

locations away. )

**B.** You want to load one value from an address which is more than 28 locations away.

**c.** You want to load an array of sequential addresses.

**5.17** How many times does the LC-3 make a read or write request to memory during the processing of the LD instruction?

How many times during the processing of the LDI instruction?

How many times during the processing of the LEA instruction?

Processing includes all phases of the instruction cycle.

**5.21** What is the maximum number of TRAP service routines that the LC-3 ISA can support? Explain.

**5.23** Suppose the following LC-3 program is loaded into memory starting at location x30FF:

x30FF 1110 0010 0000 0001

x3100 0110 0100 0100 0010

x3101 1111 0000 0010 0101

Won’t be executed

x3102 0001 0100 0100 0001

x3103 0001 0100 1000 0010

If the program is executed, what is the value in R2 at the end of

execution?

R1 = 3101

**5.42** The LC-3 macho-company had decided to use opcode 1101 to implement

a new instruction. They need you help to pick the most useful one from

the following:

a. MOVE Ri, Rj; The contents of Rj are copied into Ri.

b. NAND Ri, Rj, Rk; Ri is the bit-wise NAND of Rj, Rk

c. SHFL Ri, Rj, #2; The contents of Rj are shifted left 2 bits and stored

into Ri.

d. MUL Ri, Rj, Rk; Ri is the product of 2's complement integers

in Rj, Rk.

**6.2** The LC-3 has no Subtract instruction. If a programmer needed to subtract two numbers he/she would have to write a routine to handle it. Show the systematic decomposition of the process of subtracting two integers.

**6.4** Write a short LC-3 program that compares the two numbers in R1 and R2 and puts the value 0 in RO if R1 = R2, 1 if R1 > R2 and -1 if R1 < R2.

**7.3** What is the problem with using the string AND as a label?

Using An instruction as a label confuses the assembler Because it treats the label as the opcode itself so the label AND will not be entered into the symbol table. The assembler gives and error in the second pass.

**7.4** Create the symbol table entries generated by the assembler when translating the following routine into machine code:

**7.9** What is the purpose of the . END pseudo-op? How does it differ from the HALT instruction?

**7.10** The following program fragment has an error in it. Identify the error and explain how to fix it.