**Chapter 12 - Exercises**

**12.1 What are the typical elements of a machine instruction?**

Opcodes, source operand reference, result operand reference and next instruction reference.

**12.2 What types of locations can hold source and destination operands?**

Main or virtual memory, processor register, immediate (contained in a field in the instruction being executed), and an I/O device.

**12.3 If an instruction contains four addresses, what might be the purpose of each address?**

The purpose could be to access or store information in memory or registers, or to hold data from arithmetic operations.

**12.4 List and briefly explain five important instruction set design issues.**

Operation repertoire – How many and which operations to provide.

Data types: various types of data upon which operations are performed.

Instruction format: Instruction length(in bits), number of addresses, size of various fields, etc.

Registers: Number of processor registers that can be referenced by instructions, and their use

Addressing: mode or modes by which address of an operand is specified.

**12.5 What types of operands are typical in machine instruction sets?**

**Addresses, numbers, characters, and logical data.**

**12.6 What is the relationship between the IRA character code and the packed decimal representation?**

The last four bits in IRA bit pattern 011XXXX, the numbers 0-9 are represented using the same 4-bits as packed decimal

**12.7 What is the difference between an arithmetic shift and a logical shift?**

In a logical shift, the bits of a word are shifted left or right. On one end, the bit that is shifted out is lost, while on the other end a zero is added in. With arithmetic shift, treats the data as a signed integer and does not shift the sign bit. On a right shift, the sign bit is replicated into the bit position to its right, while with a left shift, a logic shift is performed on all bits but the sign bit

**12.8 Why are transfer of control instructions needed?**

They are required for looping (allowing for repeated calls to instructions), allow for decision making, and it allows for breaking up programs into smaller pieces for easier programming.

**12.9 List and briefly explain two common ways of generating the condition to be tested in a conditional branch instruction.**

One way is to test for some condition caused by an arithmetic operation, and checking for 0, positive, negative, or overflow. A three-address instruction format can also be used, with two addresses used for condition testing

**12.10 What is meant by the term nesting of procedures?**

Being able to call a procedure from within a procedure

**12.11 List three possible places for storing the return address for a procedure return.**

Register, start of called procedure, or the top of the stack.