**Chapter 12 - Exercises**

12.1 What are the typical elements of a machine instruction?

**The essential elements of a computer instruction are the opcode, which specifies the operation to be performed, the source and destination operand references, which specify the input and output locations for the operation, and a next instruction reference, which is usually implicit.**

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

**Source and destination operands can be held by Registers and memory.**

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

**If an instruction contains four addresses, each address serves to represent the two operands, one result, and the address of the next instruction respectively.**

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

**The five important instruction set design issues are as listed below;**

**1/Operation repertoire: How many and which operations to provide, and how complex operations should be.**

**2/Data types: The various types of data upon which operations are performed.**

**3/Instruction format: Instruction length (in bits), number of addresses, size of 4/various fields, and so on.**

**5.Registers: Number of CPU registers that can be referenced by instructions, and their use.**

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

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

**The types of operands used mainly in machine instruction sets are addresses, numbers, characters, and logical data.**

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

**For the IRA bit pattern 011XXXX, the digits 0 through 9 are represented by their binary equivalents, 0000 through 1001, in the right-most 4 bits. This is the same code as packed decimal.**

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

**With a logical shift, the bits of a word are shifted left or right. On one end, the bit shifted out is lost. On the other end, a 0 is shifted in.**

**The arithmetic shift operation treats the data as a signed integer and does not shift the sign bit. On a right arithmetic shift, the sign bit is replicated into the bit position to its right. On a left arithmetic shift, a logical left shift is performed on all bits but the sign bit, which is retained.**

12.8 Why are transfer of control instructions needed?

**Transfer of control instructions are needed because:**

**1. In the practical use of computers, it is essential to be able to execute each instruction more than once and perhaps many thousands of times. It may require thousands or perhaps millions of instructions to implement an application. This would be unthinkable if each instruction had to be written out separately. If a table or a list of items is to be processed, a program loop is needed. One sequence of instructions is executed repeatedly to process all the data.**

**2. Virtually all programs involve some decision making. We would like the computer to do one thing if one condition holds, and another thing if another condition holds.**

**3. To compose correctly a large or even medium-size computer program is an exceedingly difficult task. It helps if there are mechanisms for breaking the task up into smaller pieces that can be worked on one at a time.**

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

**There are two common ways of generating the condition to be tested in a conditional branch instruction. First, most machines provide a 1-bit or multiple-bit condition code that is set as the result of some operations. Another approach that can be used with a three-address instruction format is to perform a comparison and specify a branch in the same instruction.**

12.10 What is meant by the term nesting of procedures?

**Nesting of procedures refers to the occurrence of a procedure call inside a procedure.**

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

**The return address for a procedure return can be stored in three places:**

* **Register**
* **Start of procedure**
* **Top of stack.**