**Chapter 13 - Exercises**

13.1 Briefly define immediate addressing.

**In immediate addressing, the value of the operand is in the instruction.**

13.2 Briefly define direct addressing.

**In direct addressing, the address field contents the effective address of the operand.**

13.3 Briefly define indirect addressing.

**In Register indirect addressing, the address field refers to a register, which in turn contains the effective address of the operand.**

13.4 Briefly define register addressing.

**In register addressing, the address field refers to a register that contains the operand.**

13.5 Briefly define register indirect addressing.

**In Register indirect addressing, the address field refers to a register, which in turn contains the effective address of the operand.**

13.6 Briefly define displacement addressing.

**In Displacement addressing, the instruction has two address fields, at least one of which is explicit. The value contained in one address field (value = A) is used directly. The other address field refers to a register whose contents are added to A to produce the effective address.**

13.7 Briefly define relative addressing.

**In Relative addressing, the implicitly referenced register is the program counter (PC). That is, the current instruction address is added to the address field to produce the EA.**

13.8 What is the advantage of autoindexing?

**It is typical that there is a need to increment or decrement the index register after each reference to it. Because this is such a common operation, some systems will automatically do this as part of the same instruction cycle, using what is known as autoindexing.**

13.9 What is the difference between postindexing and preindexing?

**These are two forms of addressing, both of which involve indirect addressing and indexing.**

**With preindexing, the indexing is performed before the indirection. With postindexing, the indexing is performed after the indirection.**

13.10 What facts go into determining the use of the addressing bits of an instruction?

**Below are some of the facts that go into determining the use of the addressing bits of an instruction.**

**Number of addressing modes: Sometimes an addressing mode can be indicated implicitly. In other cases, the addressing modes must be explicit, and one or more mode bits will be needed.**

**Number of operands: Typical instructions on today’s machines provide for two operands. Each operand address in the instruction might require its own mode indicator, or the use of a mode indicator could be limited to just one of the address fields.**

**Register versus memory: The more that registers can be used for operand references, the fewer bits are needed.**

**Number of register sets: One advantage of using multiple register sets is that, for a fixed number of registers, a functional split requires fewer bits to be used in the instruction.**

**Address range: For addresses that reference memory, the range of addresses that can be referenced is related to the number of address bits. Because this imposes a severe limitation, direct addressing is rarely used. With displacement addressing, the range is opened up to the length of the address register.**

**Address granularity: In a system with 16- or 32-bit words, an address can reference a word or a byte at the designer’s choice. Byte addressing is convenient for character manipulation but requires, for a fixed-size memory, more address bits.**

13.11 What are the advantages and disadvantages of using a variable-length instruction format?

**One of the advantages of using a variable-length instruction format is that it easy to provide a large repertoire of opcodes, with different opcode lengths. Addressing can be more flexible, with various combinations of register and memory references plus addressing modes.**

**A disadvantage however is an increase in the complexity of the CPU.**