

4 *Instruction Set*

Why are a microprocessor's instructions referred to as an instruction set? Because the microprocessor designer selects the instruction complement with great care; it must be easy to execute complex operations as a sequence of simple events, each of which is represented by one instruction from a well-designed instruction set.

Assembler often frighten users who are new to programming. Yet taken in isolation, the operations involved in the execution of a single instruction are usually easy to follow. Furthermore, you need not attempt to understand all the instructions at once. As you study each of the programs in these notes you will learn about the specific instructions involved.

Table 4.1 lists the instruction mnemonics. This provides a survey of the processors capabilities, and will also be useful when you need a certain kind of operation but are either unsure of the specific mnemonics or not yet familiar with what instructions are available.

The appendix A gives a detailed description of the individual instructions while chapters 7 through to 15 provide a discussion on how to use them.

The ARM instruction set can be divided into six broad classes of instruction.

- Data Movement
- Arithmetic
- Memory Access
- Logical and Bit Manipulation
- Flow Control
- System Control / Privileged

Before we look at each of these groups in a little more detail there are a few ideas which belong to all groups worthy of investigation.

Important Note:

The ARM instruction set can be divided into six broad classes of instruction:

- Data-processing instructions (Data Movement)
- Branch instructions (Flow Control)
- Status register transfer instructions (Logic/Bit Bashing)
- Load and store instructions (Memory Access)
- Coprocessor instructions (System Control)
- Exception-generating instructions (Privileged)