3 ARM Architecture

This chapter outlines the ARM processor's architecture and describes the syntax rules of the ARM assembler. Later chapters of this book describe the ARM's stack and exception processing system in more detail.

Figure 3.1 on the following page shows the internal structure of the ARM processor. The ARM is a *Reduced Instruction Set Computer* (RISC) system and includes the attributes typical to that type of system:

- A large array of uniform registers.
- A load/store model of data-processing where operations can only operate on registers and not directly on memory. This requires that all data be loaded into registers before an operation can be preformed, the result can then be used for further processing or stored back into memory.
- A small number of addressing modes with all load/store addresses begin determined from registers and instruction fields only.
- A uniform fixed length instruction (32-bit).

In addition to these traditional features of a RISC system the ARM provides a number of additional features:

- Separate *Arithmetic Logic Unit* (ALU) and shifter giving additional control over data processing to maximize execution speed.
- Auto-increment and Auto-decrement addressing modes to improve the operation of program loops.
- Conditional execution of instructions to reduce pipeline flushing and thus increase execution speed.

3.1 Processor modes

The ARM supports the seven processor modes shown in table 3.1.

Mode changes can be made under software control, or can be caused by external interrupts or exception processing.

Most application programs execute in User mode. While the processor is in User mode, the program being executed is unable to access some protected system resources or to change mode, other than by causing an exception to occur (see 3.4 on page 29). This allows a suitably written operating system to control the use of system resources.