Undefined Instruction is when an attempt is made to preform an undefined instruction. This normally happens when there is a logical error in the program and the processor starts to execute data rather than program code.

Prefetch Abort occurs when the processor attempts to access memory that does not exist.

Data Abort occurs when attempting to access a word on a non-word aligned boundary. The lower two bits of a memory must be zero when accessing a word.

Interrupt occurs when an external device asserts the IRQ (interrupt) pin on the processor. This can be used by external devices to request attention from the processor. An interrupt can not be interrupted with the exception of a fast interrupt.

Fast Interrupt occurs when an external device asserts the FIQ (fast interrupt) pin. This is designed to support data transfer and has sufficient private registers to remove the need for register saving in such applications. A fast interrupt can not be interrupted.

When an exception occurs, the processor halts execution after the current instruction. The state of the processor is preserved in the *Saved Processor Status Register* (SPSR) so that the original program can be resumed when the exception routine has completed. The address of the instruction the processor was just about to execute is placed into the Link Register of the appropriate processor mode. The processor is now ready to begin execution of the exception handler.

The exception handler are located a pre-defined locations known as *exception vectors*. It is the responsibility of an operating system to provide suitable exception handling.

3.5 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.

See Chapter ?? and Appendix ?? for a detailed description of the individual instructions and chapters 7 through to 15 for 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.