Addressing Modes and Assembly Code

L.O.: To understand the difference between Direct and Immediate addressing modes

## Assembly Language

Assembly language is code made up of a set of mnemonics which directly translate into the binary machine code.

## Instruction Sets

Each processor has its own instruction set; either a **RISC (reduced instruction set)** or a **CICS (complex instruction set)**.

An instruction set is the patterns of binary that a given processor recognises as commands, along with their meanings.

## Types of Operation Code

The operation codes for an assembly language can be put into 4 categories:

* **Data transfer:** e.g. move (MOV), store (STR) and load (LDR)
* **Arithmetic operations:** e.g. add, subtract, multiply, divide and shift.
* **Logical operation:** e.g. bitwise operations such as AND, OR, NOT and XOR.
* **Branch operations:** e.g. conditional and unconditional operations to move to other parts of the code (control flow), such as BNA and BEG.

Example statement:

CMP r1, #10 ‘compare the value in register 1 with the value 10’

* r implies a **register**
* # implies **immediate addressing** (#10 means the literal value 10)
* A plain number implies **direct addressing** (10 means the data in memory location 10)