

---

# ASSEMBLY LANGUAGE ADDRESSING MODES

---

Low level instructions often contain an operand, which is usually a memory address. It normally refers to the location of a data item in the computers memory. There are several ways in which the processor might interpret the operand, and these are called addressing modes. The examples below are the five basic modes available in some form on most processors.

## **Absolute (Direct) addressing**

The operand is the address in memory where the data item can be found.

LDA     \$6F55                      This instruction will access memory location \$6F55, and copy the contents into the accumulator register.

## **Immediate addressing**

The operand is the data item.

LDA     #41                        This instruction will load the number 41 (\$29) into the accumulator register.

## **Indexed addressing**

The operand is added to the contents of the index register, and this gives the location in memory of the data.

LDA     \$4F10,X                      

\$34
------

                      This instruction will access the memory location \$4F44, and copy the contents into the accumulator.

Index Register  
X

## **Indirect addressing**

The operand is the address of a memory location which contains the address of the data item.

LDA     \$2B57                      2B57                      

3C
6A
44

                      This instruction will load the data item 44 into the accumulator register.

2B58

6A3C

## **Relative addressing**

This is used in conjunction with branches. The operand is a 2's comp number (-128 - +127) which is added to the address of the instruction following the branch instruction to give the address of the instruction to be jumped to.

4545                      

BEQ	\$13
LDA	\$9333

                      This BEQ command will cause control to be passed to the instruction in location \$4547 + \$13 = \$455A, provided that the zero flag is set.

4547