**Moves/Assignments**

**Sources**

Numeric constants (including preceding – and $ hexadecimal format)

String constants (ASCIIZ)

BYTE variables

WORD variables

ARRAY variables

ARRAY elements with variable or constant indexes.

**Targets**

BYTE variables

WORD variables

ARRAY variables

ARRAY elements with variable or constant indexes.

**Non Standard Moves**

**STRING CONSTANT -> ARRAY AREA = “ABCDE”**

The string constant is copied into the array. If the text is longer than the array size it is truncated.

**ARRAY -> ARRAY AREA = TEXT**

One array is copied to a second array, limited to the smaller of the two array sizes.

**ARRAY -> ARRAYELEMENT AREA(3) = TEXT**

The array will be copied to the array element onwards. Note, no checking is done here for array overflow.

**Standard Moves**

**AE/BYTE/WORD/CONSTANT -> BYTE/WORD A = B or C = 7**

The second value is copied into the first. The type of the move is decided by the target. In a Z80 CPU this also determines whether we use HL or A. In moves, the whole move is done in one mode or the other.

**AE/BYTE/WORD/CONSTANT -> ARRAYELEMENT A(3) = 42 or A(5)=!42**

The second value is copied into the first. The type of the move is determined by the presence or absence of the preceding ! – the ! causes values to be written to A(5) and A(6) in the example.