

Op Code	LL ₁	B ₁ D ₁	D ₁ D ₁	B ₂ D ₂	D ₂ D ₂
---------	-----------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------

MVN performs in a way that is analagous to **MVC**. While **MVC** works on entire bytes, **MVN** only processes the numeric parts (rightmost 4 bits) of the bytes it references. The purpose of a move numeric instruction is to move the numeric parts of a consecutive collection of bytes from one location in memory to another location. As you can see from the instruction format above, the instruction carries with it the number of bytes to be copied (LL₁), as well as the beginning addresses of the source (B₂D₂D₂D₂) and target (B₁D₁D₁D₁) fields. Notice that the instruction does not specify the ending addresses of either field - the instruction is no respecter of fields. **MVN** copies the numeric parts of LL₁ + 1 consecutive bytes from the storage location designated by B₂D₂D₂D₂ to the storage location designated by B₁D₁D₁D₁.

The length (LL₁) determines the number of "half-bytes" which will be copied. The length is usually determined implicitly from the length of operand 1 but the programmer can provide an explicit length. Consider the two example MVN's below,

Object code	Assembler code		
	FIELDA	DS	CL8
	FIELDDB	DS	CL5
	...		
D107C008C010	MVN	FIELDA, FIELDDB	Implicit length
D102C008C010	MVN	FIELDA(3), FIELDDB	Explicit length

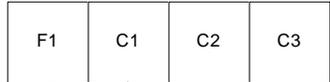
In the first example, the length implicitly defaults to 8, the length of FIELDA. In the second example, the length is explicitly 3. Notice that the assembled length (LL₁) is one less than the implicit or explicit length. This can be seen in the object code above where the assembled lengths are x'07' and x'02'.

The copying operation is usually straightforward, but can be more complicated by overlapping the source and target fields. Keep in mind that the copy is made one byte at a time. Consider the following examples,

Object code	Assembler code		
	ONE	DC	C'1' ONE = X'F1'
	FIELDA	DC	CL3'ABC' FIELDA = X'C1C2C3'
	FIELDDB	DC	XL3'123456' FIELDDB = X'123456'
	...		
D102C008C00B	MVN	FIELDA, FIELDDB	After FIELDA = X'C2C4C6'
D102C00EC008	MVN	FIELDDB, FIELDA	After FIELDDB = X'113253'
D102C008C007	MVN	FIELDA, ONE	After FIELDA = X'C1C1C1'

In the first **MVN** above, 3 consecutive numeric half-bytes in **FIELD B** are simply copied to the numeric portions of **FIELD A**. The half-bytes are copied, one at a time moving left to right within both operands. In the second example, 3 consecutive bytes are copied into **FIELD B** (implicit length = 3) from **FIELD A**. The third **MVN** is complicated by the fact that the source and target fields overlap. We will examine the third move in some detail.

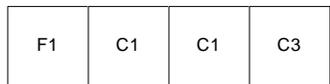
MVN FIELD A, ONE THIS IS A 3 BYTE MOVE



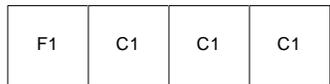
First half-byte of source copied to first half-byte of target.



Second half-byte of source copied to second half-byte of target.



Third half-byte of source copied to third half-byte of target.



Examples

Some Unrelated MVN's:

A	DC	X'123456'		
B	DC	X'ABCDEF'		
C	DC	X'A1B2'		
		...		
	MVN	A, B	A = X'1B3D5F'	B = X'ABCDEF'
	MVN	A+1, B	A = X'123B5D'	B = X'AFCDEF'
	MVN	A+1(2), B	A = X'123B5D'	B = X'ABCDEF'
	MVN	B, =X'D1E2'	B = X'A1C2E?'	One byte copied from the literal pool
	MVN	B, B+1	B = 'ADCFE1'	Left shift
	MVN	B+1(2), B	B = 'ABCBE?'	1st byte is propagated
	MVN	C, A	C = 'A2B4'	A = X'123456'
	MVN	A(L'C), C	A = '113256'	Explicit Length
attribute	MVN	A(1000), B	Assembly Error - max length is 256	
bytes	MVN	A, B(20)	Assembly Error - Op-1 determines length	

Tips

1. Pay attention to the lengths of the fields involved in any **MVN** statement. If the target field is longer than the source field, bytes following the source may be transferred. If the target field is shorter than the source field, bytes in the source may may be truncated.