**IC** is used to copy a single byte from storage into the rightmost byte of a register. The register is specified in operand 1, and the one-byte storage location is denoted by operand 2. Only the right-most byte of the register is changed. All other bytes in the register remain unchanged. Only the first byte of the field specified in storage is copied to the register.

A common use of the **IC** instruction is to move a one-byte binary length into the right-most byte of a register. The register will subsequently be used by an **EX** (Execute) instruction in order to move a variable number of bytes. Here is an example of this technique.

```
TARGET    MVC   FIELDA(0),FIELDB
LENGTH    DC    AL1(8)   A ONE-BYTE LENGTH = 8
...```

```
IC R8,LENGTH  (INITIALLY 8, LENGTH MIGHT CHANGE)
EX R8,TARGET  EXECUTE THE TARGET INSTRUCTION

Examples

For the following examples, assume that R8 contains x’11223344’.

FIELDA DS X’AABBCCDD’
FIELDB DS C’ABCD’
FIELDC DC AL1(8)
FIELDD DC AL1(20)

Result:
IC R8,FIELDA R8 = x’112233AA’
IC R8,FIELDB R8 = x’112233C1’
IC R8,FIELDC R8 = x’11223308’
IC R8,FIELDDD R8 = x’11223314’

Tips

1. It is a standard practice to use IC in conjunction with EX for moving variable length fields. Remember that the inserted length of “X” is treated as length “X + 1” when the MVC is executed. In other words the assembled length in a MVC instruction is 1 less than the actual length.