

The Load and Test Register instruction copies the rightmost 32 bits (4 bytes) of the register specified by Operand 2, into the rightmost 32 bits of the register specified by Operand 1. The contents of Operand 2 are unchanged by this operation. In this respect **LTR** is equivalent to the **LR** instruction. The difference between these instructions is that **LTR** sets the condition code based on the final contents of the Operand 1 register.

Condition Code	Meaning	Test With
0 (Zero)	Operand 1 = 0	BE, BZ
1 (Negative)	Operand 1 < 0	BL, BM
2 (Positive)	Operand 1 > 0	BH, BP

Consider the instruction below.

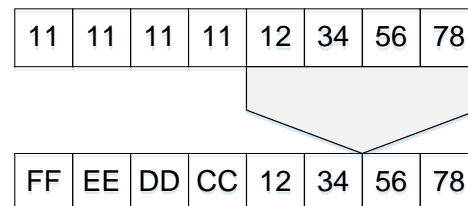
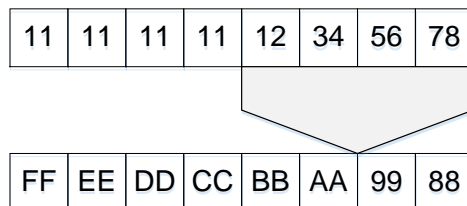
LTR R5, R10

The contents of register 10 are copied to register 5, destroying the previous value in register 5. Register 10 is unaffected by the operation. Since the contents of R5 is a positive binary integer after completion of the operation, the condition code is set to 2 (High/Positive). The diagram below illustrates this operation.

LTR R5, R10

R10 (Before)

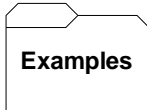
R10 (After)



R5 (Before)

R5 (After)

CC = High/Positive



Some Unrelated LTR's

```

R4 = X'FFFFFFFF'
R5 = X'00000028'
R6 = X'00000004'
R7 = X'00000000'

```

```

LTR   R4,R5   R4 = X'00000028' R5 = X'00000028'
          Cond. Code = 2 High/Positive
LTR   R5,R4   R5 = X'FFFFFFFF' R4 = X'FFFFFFFF'
          Cond. Code = 1 Low/Negative
LTR   R5,R6   R5 = X'00000004' R6 = X'00000004'
          Cond. Code = 2 High/Positive
LTR   R6,R5   R6 = X'00000028' R5 = X'00000028'
          Cond. Code = 2 High/Positive
LTR   R6,R7   R6 = X'00000000' R7 = X'00000000'
          Cond. Code = 0 Zero
LTR   R4,R4   R4 = X'FFFFFFFF'
          Cond. Code = 1 Low/Negative

```

Tips

1) **LTR** is commonly used to test the contents of a single register in order to determine if the binary number in the register is positive, negative or zero. For example, the following code illustrates how to test the contents of register 5.

```

          LTR   R5,R5           SET THE CONDITION CODE
          BM   NEGATIVE        IS R5 < 0 ?
          BP   POSITIVE        IS R5 > 0?
ZERO     EQU   *
          ...
NEGATIVE EQU   *
          ...
POSITIVE EQU   *

```