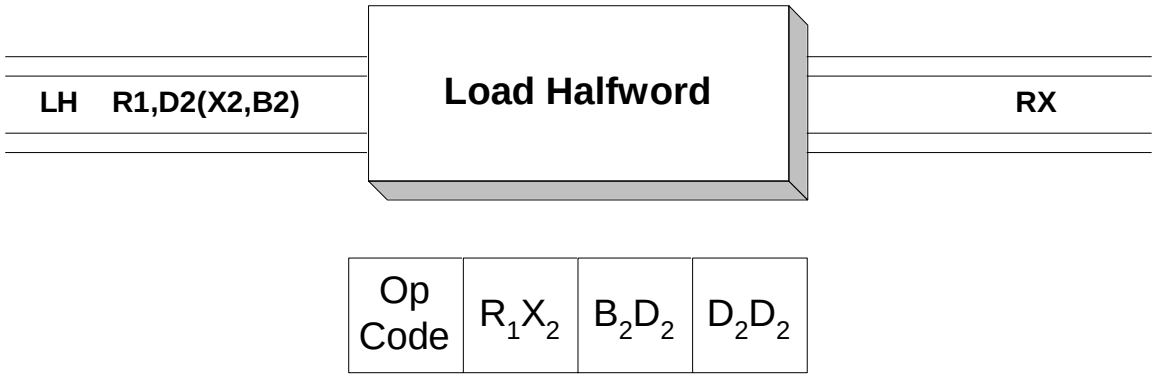
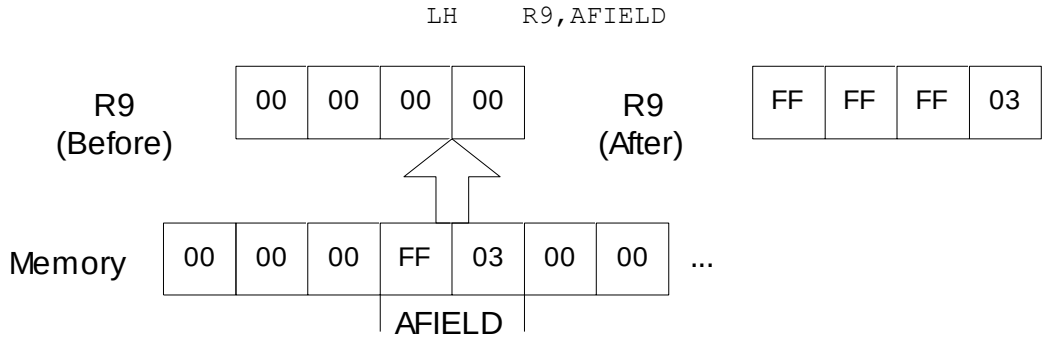


The **LH** instruction has two operands: the first operand is a general purpose register, and the second operand is a halfword in memory. The effect of the instruction is to place a 32-bit signed integer, algebraically equivalent to the 16-bit halfword, into Operand 1. The 32-bit integer can be obtained by extending the sign-bit of the halfword so that it occupies 32 bits. (Extending the sign bit of 2's complement data does not change its arithmetic value.) Consider the following example.



The halfword "AFIELD" contains a binary 1 in the sign-bit. A sign-extended version of the halfword is copied into register 9, destroying the previous values in the register. The halfword is unchanged by this operation.

Since **LH** is an RX instruction, an index register may be coded as part of operand 2 (see **Explicit Addressing**).

Examples

Some Unrelated Load Halfwords

```
            R4 =  X'00000000'
            R5 =  X'FFFFFFFF'
            R6 =  X'00000004'

AFIELD   DC  H'4'       AFIELD = X'0004'
BFIELD   DC  H'-1'      BFIELD = X'FFFF'
CFIELD   DC  H'0'       CFIELD = X'0000'

         LH   R4,AFIELD     R4 = X'00000004'
         LH   R4,BFIELD     R4 = X'FFFFFFFF'
```

```
LH   R4,CFIELD     R4 = X'00000000'
LH   R4,DFIELD     R4 = X'FFFF


CONSIDER THE NEXT TWO CONSECUTIVELY EXECUTED LOADS
LH   R5,AFIELD     R5 = X'00000004'
LH   R6,AFIELD(R5) R6 = X'FFFFFFFF'
```