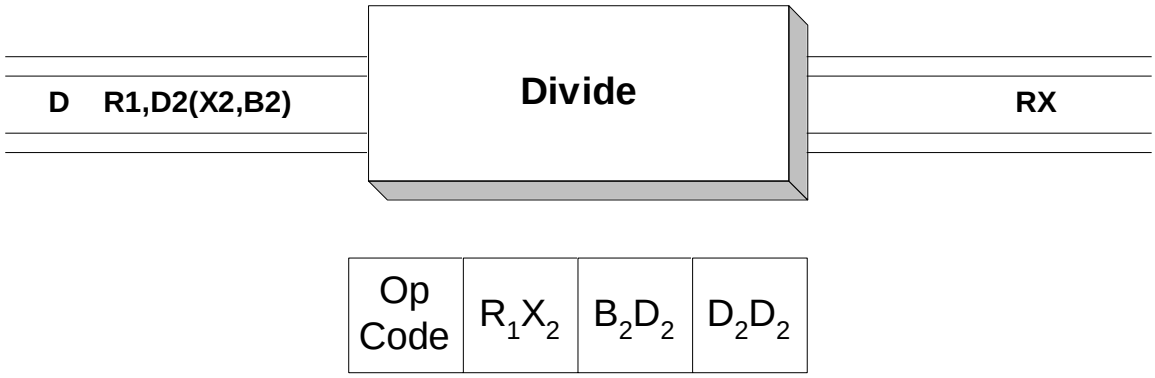
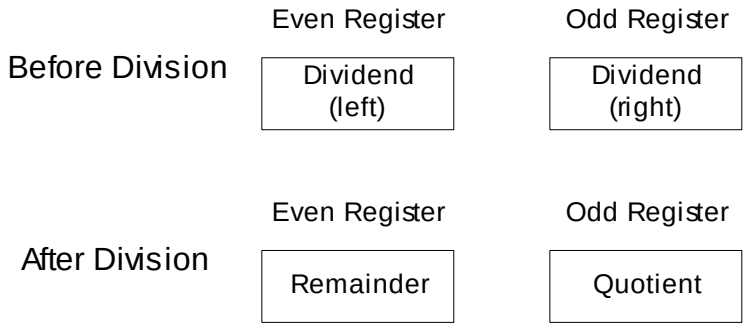


The Divide instruction performs 2's complement binary integer division and returns a quotient and a remainder. Operand 1 names an even register of an "even-odd" consecutive register pair. For instance, R2 would be used to name the R2 / R3 even-odd register pair, and R8 would be used to name the R8 / R9 even-odd register pair. Operand 2 is the name of a fullword in memory containing the divisor. Before the division, the even-odd register pair must be initialized with the dividend, which is effectively a 64-bit 2's complement integer. After the division, the remainder is contained in the even register and the quotient is contained in the odd register. The sign of the quotient is determined by the rules of algebra. The sign of the remainder will be the same as the dividend.

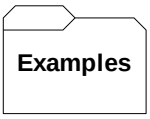


In preparing to divide a fullword, a common practice is to load the fullword in the even register and algebraically shift it to the odd register. This practice propagates the appropriate sign bit throughout the even register and successfully initialized the even/odd register pair. Here is an example where we compute A/B where A and B are fullwords.

```
        L    R8,A       PUT THE DIVIDEND IN THE EVEN REGISTER
        SRDA R8,32      ALGEBRAICALLY SHIFT R8 INTO R9
        D    R8,B       DIVIDE A BY B
        ...
A       DC   F'19'      DIVIDEND
B       DC   F'5'       DIVISOR
```

The diagram below illustrates the above division just after R8 has been shifted.

The diagram illustrates that the result of the integer division of 19 by 5 is a remainder of 4 in R8, and a quotient of 3 in R9.



Some Unrelated Divide Instructions

```
L    R6,=F'100'   DIVIDEND INITIALLY GOES IN THE EVEN REGISTER
SRDA R6,32        ... AND IS SHIFTED TO THE ODD REGISTER
D    R6,=F'10'    ... BEFORE DIVIDING
                  R6 (REMAINDER) = X'00000000',
                  R7 (QUOTIENT) = X'0000000A'

L    R6,=F'100'   DIVIDEND INITIALLY GOES IN THE EVEN REGISTER
SRDA R6,32        ... AND IS SHIFTED TO THE ODD REGISTER
D    R6,=F'8'     ... BEFORE DIVIDING
                  R6 (REMAINDER) = X'00000004',
                  R7 (QUOTIENT) = X'0000000C'

L    R6,=F'100'   DIVIDEND INITIALLY GOES IN THE EVEN REGISTER
SRDA R6,32        ... AND IS SHIFTED TO THE ODD REGISTER
D    R6,=F'0'     ... BEFORE DIVIDING
                  ABEND - DIVISION BY 0 NOT ALLOWED

L    R6,=F'-100'  DIVIDEND INITIALLY GOES IN THE EVEN REGISTER
SRDA R6,32        ... AND IS SHIFTED TO THE ODD REGISTER
D    R6,=F'-8'     ... BEFORE DIVIDING
                  R6 (REMAINDER) = X'FFFFFFFC' = -4
                  R7 (QUOTIENT)  = X'0000000C'
```

Tips

1) Know your data! If the divisor might be zero, you must protect your divisions by testing the divisor beforehand.

```
CLC   DIVISOR,=F'0    IS THE DIVISOR 0?
BE    ZERODIV         BRANCH IF DIVISOR IS 0
D     R8,DIVISOR      O.K. TO DIVIDE NOW
...
```

```
ZERODIV  EQU   *
         (CODE TO HANDLE A ZERO DIVISOR)
```