Chapter 16
More Binary Arithmetic

Objectives

Upon completion of this chapter you will be able to:

- Define even-odd pairs of registers,
- Use the M, MH, and MR instructions to perform binary multiplication, and
- Use the D and DR instructions to perform binary division.

Introduction

In chapter 14 we saw how to add and subtract binary numbers. Specifically, we looked at the A,
AH, AR, S, SH, and SR instructions. Recall that for each of these instructions, the first operand is
always a register, and the second operand is either a fullword, halfword, or register. In this
chapter we will complete our discussion of binary math by looking at the multiplication and
division instructions: M, MH, MR, D, and DR.

Even-odd Register Pairs

In order to discuss binary, or register, multiplication, we must introduce a new concept - that of
even-odd pairs of registers. Recall that there are sixteen registers in all, numbered 0 through 15.
This gives us eight even-odd pairs of registers:

```
R0  R1
R2  R3
R4  R5
R6  R7
R8  R9
R10 R11
R12 R13
R14 R15
```

Remember, an even-odd pair of registers starts with the even number register, which is the lower
of the two numbers. For example, (R2, R3) is an even-odd pair, but (R3, R4) and (R5, R6) are
not even-odd pairs.

Binary Multiplication: The M, MH, and MR Instructions

The M (multiply) instruction multiplies a register by a fullword. For example:

```
L    R9,subtotal
M    R8,discount
```

where SUBTOTAL DS F

where DISCOUNT DS F

- First, one of the values to be multiplied is placed in the odd numbered register of an even-odd
  pair.
- Second, multiply. Specify the even numbered register as the first operand, and the other
  fullword as the second operand. (If the first operand is not an even numbered register, you
  will get a specification exception at run time.)
Finally, the product will be a doubleword occupying the even-odd pair. (Usually the right most portion of the product; that is, the odd numbered register, will be sufficient to hold the product. Recall that a single register can hold a value of up to 2,147,483,647. If necessary, you can use the store multiple instruction to store very large products in a doubleword; for example,STM R4,R5,DBL. Likewise, you can use the load multiple instruction to put a doubleword into a register pair; for example,LM R4,R5,DBL.

We need not be concerned with what is in the even numbered register prior to the multiply. It does not need to be initialized: whatever is there will be replaced by the high order digits of the product.

Let’s look more closely at another example. Given:

| FULL1  DC  F’64’ | 00 00 00 40 |
| FULL2  DC  F’8’  | 00 00 00 08 |

To multiply FULL1 by FULL2 (64 x 8 = 512 = X’200’) and convert the product to a packed number we code (assume DBLWORD DC D’0’):

```
R4     R5
L    R5,FULL1
  ?? ?? ?? ?? 00 00 00 40
R4     R5
M    R4,FULL2
  00 00 00 00 00 00 02 00
DBLWORD
CVD  R5,DBLWORD
  00 00 00 00 00 00 51 2C
```

We can also use the MR (multiply register) instruction to multiply an even-odd pair by a (single) register. For example:

```
R4     R5
L    R5,FULL1
  ?? ?? ?? ?? 00 00 00 40
L    R6,FULL2
  ?? ?? ?? ?? 00 00 00 40
MR   R4,R6
  00 00 00 00 00 00 02 00
```

You Try It...

1. Given: X DC F’16’, Y DC F’3’, and Z DC F’0’. Supply the instructions to multiply X by Y, with the product in Z and in register 7. Show the intermediate results.
2. Given: A DC F'32', B DC F'4', and C DC F'0'. Supply the instructions to multiply A by B, with the product in C and in register 5. Show the intermediate results.

R4     R5
R4     R5
C

3. Given: register 9 contains 64. Use the M instruction to multiply this by 5, with the product in register 9. Show the intermediate results.

    (before)  R8     R9
    ______________________  R8     R9

4. Given: register 5 contains 4, register 6 contains 3, and register 7 contains 2. Supply the instruction to multiply the value in register 7 by the value in register 5, with the product in register 7. Show the intermediate results.

    (before)  R6     R7
    ______________________  R6     R7

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

In both of these examples, the second operand was small (8 and 3). Each would, in fact, fit in a halfword. If one or both of the operands for a multiply instruction is a halfword, the MH (multiply halfword) instruction can be used. The MH instruction is much simpler than the M or MR: it uses a single register only, and this register can be even or odd. For example:

L    R8, SUBTOTAL  where SUBTOTAL DS F
MH   R8, DISCOUNT  where DISCOUNT DS H

- First, one of the values to be multiplied is placed in a register. Reminder: use L if the value is a fullword, or LH if the value is a halfword.
- Second, multiply. Specify the register as the first operand, and the halfword as the second operand.
- Finally, the product will be a fullword occupying the register. (If the product will not fit in the register, truncation occurs without warning.)
Consider the following example:

```
FULL1  DC  F'64'
   00 00 00 40
HALF2 DC  H'8'
   00 08
```

To multiply `FULL1` by `HALF2` (64 x 8 = 512 = X'200') and convert the product to a packed number we code:

```
L R4,FULL1
  R4
MH R4,HALF2
  R4
CVD R4,DBLWORD
   DBLWORD
   00 00 00 00 00 00 51 2C
```

**You Try It...**

5. **Given:** R DC H'15', S DC H'4', and T DC H'0'. Supply the instructions to multiply R by S giving T. Show the intermediate results.

6. **Given:** A DC F'32', B DC H'4', and C DC H'0'. Supply the instructions to multiply A by B giving C. Show the intermediate results.

7. Given: register 6 contains 16. Use the `MH` instruction to multiply this by 4, with the product in register 6. Show the intermediate results.
Binary Division: The $D$ and $DR$ Instructions

Recall from our discussion of the $DP$ (divide packed) instruction, following the division the dividend was replaced by the quotient (on the left) and the remainder (on the right). Something similar occurs with register division. As with register multiplication, register division uses an even-odd pair of registers. Initially, the dividend will occupy an even-odd pair. Following the divide operation, the quotient will be in one register, and the remainder will be in the other register. There is one potential point of confusion: unlike the divide packed instruction, the quotient will be on the right (in the odd register) and the remainder will be on the left (in the even register).

The $D$ (divide) instruction divides the dividend (in an even-odd pair) by a fullword. For example:

\[
\begin{align*}
&L \quad R9, \text{SUM} \\
&M \quad R8, =F’1’ \\
&D \quad R8, \text{COUNT}
\end{align*}
\]

where \text{SUM} DS F

\[
\begin{align*}
&L \quad R9, \text{SUM} \\
&M \quad R8, =F’1’ \\
&D \quad R8, \text{COUNT}
\end{align*}
\]

where \text{COUNT} DS F

- First, the dividend must be placed in the odd numbered register of an even-odd pair. Reminder: use $L$ if the dividend is a fullword, or $LH$ if the dividend is a halfword.
- Second, unlike binary multiplication, the contents of the even numbered register is significant for division. Usually we want to clear (zero) it. We do so by multiplying the even-odd pair by a fullword with value of one. (Recall that to multiply we specify the even register of the even-odd pair, and the product will occupy the pair.)
- Third, divide. Specify the even numbered register as the first operand and a fullword containing the divisor are the second operand. (If the first operand is not an even numbered register, you will get a specification exception at run time.)
- Finally, the quotient will be in the odd numbered register and the remainder will be in the even numbered register.

As mentioned above, the contents of the even numbered register is significant; that is, both registers in the pair determine the value of the dividend. Recognizing this, it is not uncommon to see someone zero out the even numbered register by subtracting it from itself, as in $SR \ R8,R8$ but this will not work if the dividend is negative. Instead, multiply the pair by a fullword of one (as shown above) so as to maintain the integrity of the sign.

Let's look more closely at the above example. Given:

\[
\begin{align*}
&\text{SUM} \quad \text{DC} \quad F’214’ \\
&\text{COUNT} \quad \text{DC} \quad F’8’ \\
&\text{AVG} \quad \text{DC} \quad \text{PL3’0’}
\end{align*}
\]

Copyright © 2009 by Bill Qualls – All Rights Reserved
To divide \( \text{SUM} \) by \( \text{COUNT} \) giving \( \text{AVERAGE} \) (214 / 8 = 26 + remainder 6).

\[
\begin{align*}
L & \quad \text{R9, \text{SUM}} & \quad \text{R8} & \quad \text{R9} \\
M & \quad \text{R8,=F'1'} & \quad \text{R8} & \quad \text{R9} \\
D & \quad \text{R8, \text{COUNT}} & \quad \text{R8} & \quad \text{R9} \\
\text{CVD} & \quad \text{R9, \text{DBLWORD}} & \quad \text{DBLWORD} \\
\text{ZAP} & \quad \text{AVG, \text{DBLWORD}}
\end{align*}
\]

What if we want the result to be rounded? That is, 214 / 8 = 26.75 which we would like to show as 27. In order to do so, rather than multiply the dividend by one, we will multiply by ten. We then have 2140 / 8 = 267, which we can shift-and-round to get 27. For example:

\[
\begin{align*}
L & \quad \text{R9, \text{SUM}} & \quad \text{R8} & \quad \text{R9} \\
M & \quad \text{R8,=F'10'} & \quad \text{R8} & \quad \text{R9} \\
D & \quad \text{R8, \text{COUNT}} & \quad \text{R8} & \quad \text{R9} \\
\text{CVD} & \quad \text{R9, \text{DBLWORD}} & \quad \text{DBLWORD} \\
\text{SRP} & \quad \text{DBLWORD, 64-1, 5} & \quad \text{DBLWORD} \\
\text{ZAP} & \quad \text{AVG, \text{DBLWORD}}
\end{align*}
\]

We can also use the DR (divide register) instruction to divide an even-odd pair by a (single) register:

\[
\begin{align*}
L & \quad \text{R9, \text{SUM}} & \quad \text{R8} & \quad \text{R9} \\
M & \quad \text{R8,=F'10'} & \quad \text{R8} & \quad \text{R9} \\
L & \quad \text{R7, \text{COUNT}} & \quad \text{R8} & \quad \text{R9} \\
\text{DR} & \quad \text{R8, R7}
\end{align*}
\]
Note: there is no divide equivalent to \texttt{MH}; that is, there is no divide halfword. This is not to say a halfword cannot be used as a divisor, simply that it must be loaded to a register first (with \texttt{LH}), and then the \texttt{DR} instruction is used.

\textbf{You Try It...}

8. Given: \texttt{X DC F'17', Y DC H'3', and Z DC PL3'0'}. Supply the instructions to divide \( X \) by \( Y \) giving \( Z \) equal to \( X'00567C' \) (representing 5.67).

\begin{verbatim}
R4 R5
M
LH R3,Y
R4 R5
DBLWORD
DBLWORD
Z
\end{verbatim}

9. Given: \texttt{A DC F'20', B DC F'42', and C DC PL3'0'}. Supply the instructions to divide \( A \) by \( B \) giving \( C \) equal to \( X'00048C' \) (representing 48\%). Hint: \( 20,000 / 42 = 476 + R8 \).

\begin{verbatim}
R6 R7
M
L R8,B
R6 R7
DBLWORD
DBLWORD
C
\end{verbatim}
To illustrate these concepts we introduce two programs: COGS16A.MLC and COGS16B.MLC, which are modifications of COGS13A.MLC and COGS13B.MLC respectively. These programs were introduced in chapter 13. COGS16A.MLC will determine nationwide dollar sales for Cogsworth Industries, while COGS16B.MLC will produce a report showing California's contribution to sales. Both programs will read COGS.BIN, which is the binary equivalent to COGS.DAT. COGS.BIN was created by COGS14A.MLC as shown in chapter 14. The program listings follow. Changes to the earlier versions have been shaded. The execution and output are not shown as they are the same as was shown in chapter 13.

Sample Program: Cogsworth's Nationwide Dollar Sales

```
PRINT NOGEN
****************************************************************
* FILENAME: COGS16A.MLC *
* AUTHOR : Bill Qualls *
* SYSTEM : PC/370 R4.2 *
* REMARKS : Determine nationwide dollar sales for *
*            COGSWORTH INDUSTRIES. *
* This is a modification of COGS13A.MLC and *
* illustrates binary multiplication. *
****************************************************************
START 0
REGS
BEGIN BEGIN
WTO 'COGS16A ... Begin execution'
BAL R10,SETUP
MAIN EQU *
CLI EOFSW,C'Y'
BE EOJ
BAL R10,PROCESS
B MAIN
EOJ EQU *
BAL R10,WRAPUP
WTO 'COGS16A ... Normal end of program'
RETURN
****************************************************************
* SETUP - Those things which happen one time only, *
* before any records are processed. *
****************************************************************
SETUP EQU *
ST R10,SVSETUP
OPEN INVENTORY Input is EBCDIC, no CR/LF
BAL R10,READ
L R10,SVSETUP
BR R10
****************************************************************
* READ - Read a record. *
****************************************************************
READ EQU *
ST R10,SVREAD
GET INVENTORY,IREC Read a single product record
B READX
ATEND EQU *
MVI EOFSW,C'Y'
```

(continued)
### CHAPTER 16
### MORE BINARY ARITHMETIC

---

**READX** EQU *
L R10,SVREAD
BR R10

*PROCESS - Those things which happen once per record.*

**PROCESS** EQU *
ST R10,SVPROC

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LH R3,ICALIF</td>
<td>Determine total units</td>
</tr>
<tr>
<td>AH R3,IILL</td>
<td>sold for this product</td>
</tr>
<tr>
<td>AH R3,IUTAH</td>
<td></td>
</tr>
<tr>
<td>AH R3,IMISC</td>
<td></td>
</tr>
<tr>
<td>MH R3,ISELL</td>
<td>Multiply units by price</td>
</tr>
<tr>
<td>A R3,TOTAL</td>
<td>Add total thus far</td>
</tr>
<tr>
<td>ST R3,TOTAL</td>
<td>then save back.</td>
</tr>
</tbody>
</table>

BAL R10,READ
L R10,SVPROC
BR R10

**WRAPUP - Those things which happen one time only, after all records have been processed.*

**WRAPUP** EQU *
ST R10,SVWRAP

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L R3,TOTAL</td>
<td>Must put it in a register</td>
</tr>
<tr>
<td>CVD R3,DBLWORD</td>
<td>to convert it to packed.</td>
</tr>
<tr>
<td>ED ODOLLARS,DBLWORD+4</td>
<td></td>
</tr>
</tbody>
</table>

WTO OMSG
CLOSE INVENTORY
L R10,SVWRAP
BR R10

**File definitions**

**INVENTORY** DCB LRECL=28,RECFM=F,MACRF=G,EODAD=ATEND,DDNAME='COGS.BIN'

**RETURN ADDRESSES**

**Miscellaneous field definitions**

**EOFSW** DC CL1'N' End of file? (Y/N)

**TOTAL** DC F'0' Nationwide dollar sales

**DBLWORD** DC D'0'

**Input record definition**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IREC</td>
<td>0CL28 1-28 Inventory record</td>
</tr>
<tr>
<td>IDESC</td>
<td>CL10 1-10 Product description</td>
</tr>
</tbody>
</table>

---

(continued)

---

Copyright © 2009 by Bill Qualls – All Rights Reserved
Sample Program: California’s Contribution to Sales

PRINT NOGEN

*****************************************************************
* FILENAME:  COGS16B.MLC                                       *
* AUTHOR    :  Bill Qualls                                      *
* SYSTEM    :  PC/370 R4.2                                      *
* REMARKS   :  Produce report for COGSWORTH INDUSTRIES          *
*             California's contribution to sales.                *
* This is a modification of COGS13B.MLC and                    *
* illustrates binary division                                   *
*****************************************************************

START 0
REGS
BEGIN BEGIN
WTO 'COGS16B ... Begin execution'
BAL R10,SETUP
MAIN EQU *
CLI EOFSW,C'Y'
BE EOJ
BAL R10,PROCESS
B MAIN
EOJ EQU *
BAL R10,WRAPUP
WTO 'COGS16B ... Normal end of program'
RETURN

*****************************************************************
* SETUP - Those things which happen one time only,             *
* before any records are processed.                            *
*****************************************************************
SETUP EQU *
ST R10,SVSETUP
OPEN INVENTORY Input is EBCDIC, no CR/LF
  GI REPORT+10,X'08'
  PC/370 ONLY - Convert all
  output from EBCDIC to ASCII
  OPEN REPORT
  BAL R10,HDGS
  BAL R10,READ
  L R10,SVSETUP
  BR R10

(continued)
HDGS  EQU *
ST   R10,SVHDGS
PUT  REPORT,HD1
PUT  REPORT,HD2
PUT  REPORT,HD3
PUT  REPORT,HD4
PUT  REPORT,HD5
PUT  REPORT,HD6
L    R10,SVHDGS
BR   R10

PROCESS  EQU *
ST   R10,SVPROC
BAL  R10,FORMAT
BAL  R10,WRITE
BAL  R10,READ
L    R10,SVPROC
BR   R10

FORMAT  EQU *
ST   R10,SVFORM
MVC  OREC,BLANKS
MVC  ODESC,DESC
L     R3,ICALIF          Determine total units
AH    R3,IILL           sold for this product
AH    R3,IUTAH
AH    R3,IWISC          R3 = Nationwide
LR    R2,R3
A     R2,TTOTAL         Add nationwide so far
ST    R2,TTOTAL         and save it back.
CVD   R3,DLBWORD        Convert to packed
ZAP   PK2,DLBWORD       for printing.
MVC  OTOTAL,=X'40202120'
ED    OTOTAL,PK2
LH    R5,ICALIF         R5 = California only
LR    R2,R5
A     R2,TCALIF         Add California so far
ST    R2,TCALIF         and save it back.
CVD   R5,DLBWORD        Convert to packed
ZAP   PK2,DLBWORD       for printing.
MVC  OCALIF,=X'40202120'
ED    OCALIF,PK2
M    R4,=F'1000'        Dividend will be in (R4,R5)
DR    R4,R3             Divisor (nationwide) in R3
CVD   R5,DLBWORD        Quotient is in R5
SRP   DBLWORD,64-1,5
ZAP   PK2,DLBWORD
MVC  OPCT,=X'40202120'
ED    OPCT,PK2
MVI  OPCT+L'OPCT,PERCENT
MVC  OCRLF,WCRLF       PC/370 only.

(continued)
L   R10,SVFORM
BR   R10

****************************************************************
**        READ - Read a record.                                 **
****************************************************************
READ   EQU   *
ST   R10,SVREAD
GET   INVENTORY,IREC  Read a single product record
B   READX

****************************************************************
**        WRITE - Write a single detail line.                   **
****************************************************************
WRITE   EQU   *
ST   R10,SVWRITE
PUT   REPORT,OREC  Write report line
L   R10,SVWRITE
BR   R10

****************************************************************
**        WRAPUP - Those things which happen one time only,     **
**                 after all records have been processed.        **
****************************************************************
WRAPUP   EQU   *
ST   R10,SVWRAP
PUT   REPORT,HD6
MVC   OREC,BLANKS
MVC   ODESC(6),=C'TOTALS'
L   R3,TTOTAL  R3 = Nationwide total
CVD   R3,DBLWORD  Convert to packed
ZAP   PK2,DBLWORD  for printing.
MVC   OTOTAL,=X'40202120'  for printing.
ED   OTOTAL,PK2
L   R5,TCALIF  R5 = California only
CVD   R5,DBLWORD  Convert to packed
ZAP   PK2,DBLWORD  for printing.
MVC   OCALIF,=X'40202120'  for printing.
ED   OCALIF,PK2
M   R4,=F'1000'  Dividend will be in (R4,R5)
DR   R4,R3  Divisor (nationwide) in R3
CVD   R5,DBLWORD  Quotient is in R5
SRP   DBLWORD,64-1,5
ZAP   PK2,DBLWORD
MVC   OPCT,=X'40202120'  for printing.
ED   OPCT,PK2
MVI   OPCT+L'OPCT,PERCENT
MVC   OCRLF,WCRLF  PC/370 only.
BAL   R10,WRITE
CLOSE   INVENTORY
CLOSE   REPORT
WTO   'COGS16B ... Sales report on REPORT.TXT'
L   R10,SVWRAP
BR   R10

(continued)
CHAPTER 16
MORE BINARY ARITHMETIC

LTORG

* File definitions

INVENTORY DCB LRECL=28,RECFM=F,MACRF=G,EODAD=ATEND,
DDNAME='COGS.BIN'
REPORT DCB LRECL=62,RECFM=F,MACRF=P,
DDNAME='REPORT.TXT'

* RETURN ADDRESSES

SVSETUP DC F'0'               SETUP
SVHDGS  DC F'0'               HDGS
SVPROC  DC F'0'               PROCESS
SVREAD  DC F'0'               READ
SVFORM  DC F'0'               FORMAT
SVWRITE DC F'0'               WRITE
SVWRAP DC F'0'               WRAPUP

Miscellaneous field definitions

WCRLF    DC X'0D25'            PC/370 ONLY - EBCDIC CR/LF
EOFSW    DC CL1'N'             End of file? (Y/N)
BLANKS   DC CL62' '            
TCALIF   DC F'0'               Grand total for Calif
TTOTAL   DC F'0'               Grand total nationwide
DBLWORD DC D'0'
PK2      DC PL2'0'
PERCENT  EQU C'\%

Input record definition

DS 0H                 Force halfword alignment
IREC     DS 0CL28        1-28  Inventory record
IDESC    DS CL10         1-10  Product description
ICALIF   DS H           11-12  Units sold in Calif
IIIL    DS H           13-14  Units sold in Illinois
IUTAH    DS H           15-16  Units sold in Utah
IWISC    DS H           17-18  Units sold in Wisconsin
IBEGIN   DS H           19-20  Beginning inventory
IPURCH   DS H           21-22  Purchases throughout year
IQOH     DS H           23-24  Actual quantity on hand
ICOST    DS H           25-26  Cost (each) 99V99
ISELL    DS H           27-28  Sell for (each) 99V99

Output (line) definition

OREC     DS 0CL62        1-62
ODESC    DS CL10         1-10  Product description
DS CL7    11-17
OTOTAL   DS CL4         18-21  Units sold Nationwide
DS CL9    22-30
OCALIF   DS CL4         31-34  Units sold in Calif
DS CL8    35-42
OPCT     DS CL4         43-46  Percent sales from Calif
DS CL14   47-60
OCRLF    DS CL2         61-62  PC/370 only - CR/LF

(continued)
CHAPTER 16
MORE BINARY ARITHMETIC

******************************************************************************
*      Headings definitions                        *
******************************************************************************
HD1  DS 0CL62
     DC CL60' COGSWORTH INDUSTRIES '
     DC XL2'0D25'
HD2  DS 0CL62
     DC CL60' California's Contribution to Sales'
     DC XL2'0D25'
HD3  DS 0CL62
     DC CL60'
     DC XL2'0D25'
HD4  DS 0CL62
     DC CL40' Nationwide California '
     DC CL20'Percent of'
     DC XL2'0D25'
HD5  DS 0CL62
     DC CL40' Product Sales Sales '
     DC CL20' National '
     DC XL2'0D25'
HD6  DS 0CL62
     DC CL40'-------- -------- -------- '
     DC CL20'--------'
     DC XL2'0D25'
END BEGIN
Exercises

1. True or false.

   a. (R4, R3) is an even-odd pair of registers.  
   b. All binary multiplication instructions use at least one register.  
   c. The first operand of an M instruction must specify the odd register of an even-odd pair.  
   d. The first operand of an MH instruction must specify the even register of an even-odd pair.  
   e. Following an M instruction, the product will occupy an even-odd pair of registers.  
   f. Following an MH instruction, the product will occupy a halfword.  
   g. It is impossible to multiply a halfword by a fullword with the product occupying the halfword.  
   h. When performing binary division, in anticipation of rounding, multiply the even-odd pair containing the dividend by a power of ten.  
   i. Following M or D, the product or dividend must be converted to packed decimal in order to properly display its value.  
   j. If the dividend is in a register and the divisor is in a halfword, the DH instruction can be used.  
   k. The DR instruction uses a total of three registers.  
   l. The MR instruction uses a total of two registers.  
   m. Following the D instruction, the remainder will be in the even numbered register and the quotient will be in the odd numbered register.

2. Given the following field definitions:

   H1    DC    H'25'
   H2    DC    H'8'
   H3    DC    H'0'
   F1    DC    F'6'
   F2    DC    F'3'
   F3    DC    F'0'

   Find the error (one only) in each of the following:

   a. * Multiply F1 by F2 giving F3
      L     R5,F1
      M     R5,F2
      ST    R5,F3

   b. * Multiply F1 by F2 giving F3
      L     R5,F1
      L     R6,F2
      M     R4,R6
      ST    R5,F3
Exercises

c. * Multiply F1 by F2 giving F3
   L  R5,F1
   M  R4,F2
   ST R4,F3

d. * Multiply H1 by H2 giving H3
   LH R4,H1
   MH R4,H2
   STH R5,H3

e. * Multiply H1 by H2 giving H3
   LH R3,H1
   M  R3,H2
   STH R3,H3

f. * Multiply F1 by H2 giving F3
   LH R3,F1
   MH R3,H2
   ST R3,F3

3. Given the following field definitions:

   H1 DC H'25'
   H2 DC H'8'
   H3 DC H'0'
   F1 DC F'6'
   F2 DC F'3'
   F3 DC F'0'

Find the error (one only) in each of the following:

a. * Divide F1 by F2, quotient in F3
   L  R5,F1
   M  R4,F'1'
   D  R4,F2
   ST R4,F3

b. * Divide F1 by F2, quotient in F3
   L  R4,F1
   M  R4,F'1'
   L  R6,F2
   DR R4,R6
   ST R5,F3

c. * Divide H1 by H2, quotient in H3
   LH R5,H1
   M  R4,H'1'
   LH R6,H2
   DR R4,R6
   STH R5,H3

d. * Divide H1 by F2, quotient in H3
   LH R3,H1
   M  R2,F'1'
   D  R2,F2
   ST R3,H3
Exercises

4. Given the following field definitions:

\[
\begin{align*}
F1 & \text{ DC } F'16' \\
F2 & \text{ DC } F'8' \\
H1 & \text{ DC } H'4' \\
H2 & \text{ DC } F'3' \\
DBL & \text{ DC } D'0' \\
PK3 & \text{ DC } PL3'0'
\end{align*}
\]

Supply the instructions to perform each of the following. Show all intermediate results. Start with fresh data each time.

a. Multiply \( F1 \) by \( F2 \) giving \( F2 \).
b. Multiply \( H1 \) by \( H2 \) giving \( H1 \).
c. Multiply \( F1 \) by \( H1 \) giving \( F2 \).
d. Multiply \( H2 \) by \( F2 \) giving \( F1 \).
e. Multiply \( F1 \) by 2 giving \( F1 \).
f. Multiply \( H1 \) by 2 giving \( H1 \).
g. Divide \( F1 \) by \( F2 \) giving quotient in \( F1 \).
h. Divide \( F1 \) by \( H1 \) giving remainder in \( H2 \).
i. Divide \( F2 \) by \( H2 \) giving quotient in \( PK3 \).
j. Divide \( H2 \) by \( F2 \) giving remainder in \( PK3 \).
k. Divide \( F1 \) by 5 giving quotient in \( PK3 \).
l. Divide \( H1 \) by 5 giving remainder in \( H2 \).

5. Write a program which will read the binary version of the \texttt{TOOL} file (\texttt{TOOL.BIN}) produced in exercise 8 of chapter 14 and create the Markup report shown in exercise 8 of chapter 13. Do all arithmetic in binary; that is, use packed fields only as required for the \texttt{ED} command.