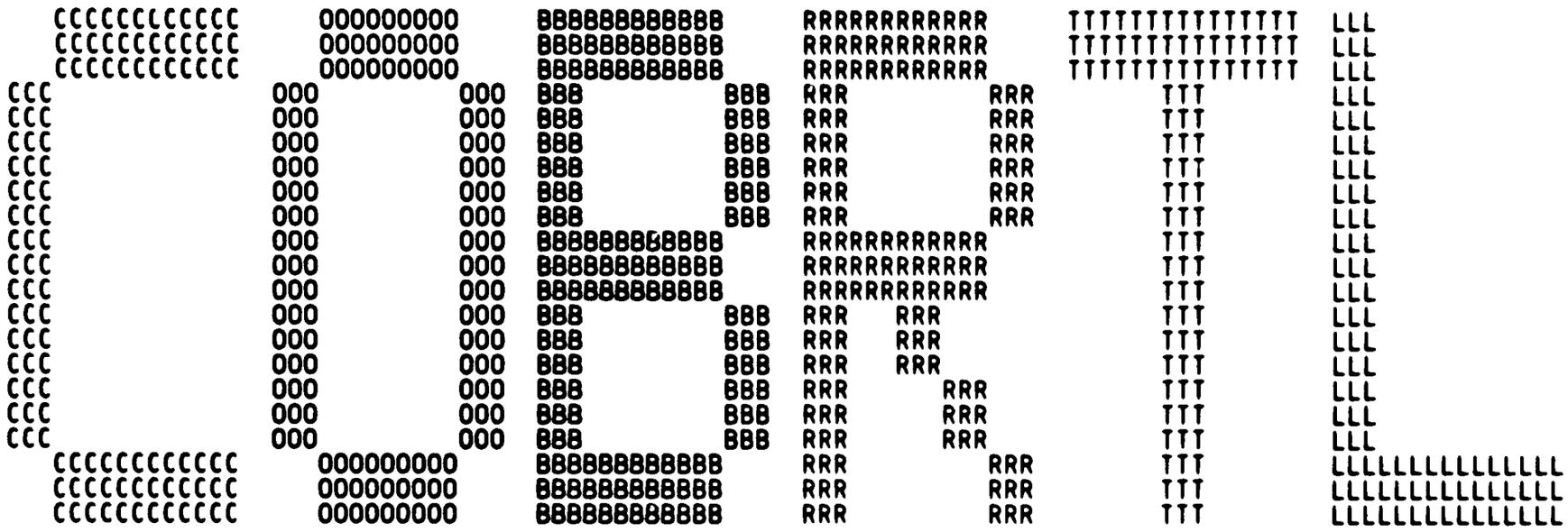
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

```
  CCCCCCCC    000000    BBBBBBBB    MM      MM  UU      UU  LL            QQQQQQ
  CCCCCCCC    000000    BBBBBBBB    MM      MM  UU      UU  LL            QQQQQQ
CC           00      00  BB      BB  MMMM  MMMM  UU      UU  LL          QQ      QQ
CC           00      00  BB      BB  MMMM  MMMM  UU      UU  LL          QQ      QQ
CC           00      00  BB      BB  MM  MM  MM  UU      UU  LL          QQ      QQ
CC           00      00  BB      BB  MM  MM  MM  UU      UU  LL          QQ      QQ
CC           00      00  BBBBBBBB    MM      MM  UU      UU  LL          QQ      QQ
CC           00      00  BBBBBBBB    MM      MM  UU      UU  LL          QQ      QQ
CC           00      00  BB      BB  MM      MM  UU      UU  LL          QQ  QQ  QQ
CC           00      00  BB      BB  MM      MM  UU      UU  LL          QQ  QQ  QQ
CC           00      00  BB      BB  MM      MM  UU      UU  LL          QQ    QQ      ....
  CCCCCCCC    000000    BBBBBBBB    MM      MM  UUUUUUUUUU  LLLLLLLLLL    QQQQ  QQ      ....
  CCCCCCCC    000000    BBBBBBBB    MM      MM  UUUUUUUUUU  LLLLLLLLLL    QQQQ  QQ      ....
    CCCCCCC    00000    BBBBBBB     MM      MM   UUUUUUUU   LLLLLLLLLL     QQQQ  QQ      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  COB$MULQ_R8     COBOL Multiply Quadwords
0000     2          .IDENT  /1-006/         ; File: COBMULQ.MAR
0000     3
0000     4 ;
0000     5 ;
0000     6 ;*****************************************************************************
0000     7 ;*                                                                           *
0000     8 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     9 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    10 ;*  ALL RIGHTS RESERVED.                                                     *
0000    11 ;*                                                                           *
0000    12 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    13 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    14 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    15 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    16 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    17 ;*  TRANSFERRED.                                                             *
0000    18 ;*                                                                           *
0000    19 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    20 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    21 ;*  CORPORATION.                                                             *
0000    22 ;*                                                                           *
0000    23 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    24 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    25 ;*                                                                           *
0000    26 ;*                                                                           *
0000    27 ;*****************************************************************************
0000    28 ;
0000    29 ;
0000    30 ;
0000    31 ; FACILITY: COBOL ARITHMETIC
0000    32 ;++
0000    33 ; ABSTRACT:
0000    34 ;       This module contains the routine which multiplies two
0000    35 ;       quadwords, producing a quadword result.
0000    36 ;
0000    37 ;
0000    38 ;--
0000    39 ;
0000    40 ; VERSION: 1
0000    41 ;
0000    42 ; HISTORY:
0000    43 ;
0000    44 ; AUTHOR:
0000    45 ;       John Sauter, 22-DEC-78
0000    46 ;
0000    47 ; MODIFIED BY:
0000    48 ;
0000    49 ;
0000    50 ;
```

```
0000     52          .SBTTL  HISTORY         ; Detailed Current Edit History
0000     53
0000     54
0000     55 ; Edit History for Version 1 of COBMULQ
0000     56 ;
0000     57 ; 1-001 - Original from Architecture Handbook, page 6-28.
0000     58 ; 1-002 - Correct some comments.  JBS 26-DEC-78
0000     59 ; 1-003 - Make the entry point symbol global.  JBS 03-JAN-1979
0000     60 ; 1-004 - Minor cleanups.  MLJ 10-Mar-1979
0000     61 ; 1-005 - Make overlapping input and output operands work correctly.
0000     62 ;         RKR 24-SEPT-79
0000     63 ; 1-006 - Cosmetic changes.  RKR 21-OCT-79
```

```
                        .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;

;
; EXTERNAL SYMBOLS:
;       NONE
;

;
; MACROS:
;       NONE
;

;
; PSECT DECLARATIONS:
;       .PSECT  _COB$CODE       PIC, SHR, LONG, EXE, NOWRT

;
; EQUATED SYMBOLS:
;       NONE
;

;
; OWN STORAGE:
;       NONE
;
```

```
                                        95          .SBTTL  COB$MULQ_R8
                                        96
                                        97 ;++
                                        98 ; FUNCTIONAL DESCRIPTION:
                                        99 ;
                                       100 ;       Multiplies two quadwords, producing a quadword result.
                                       101 ;       There is no check for overflow; the low-order 64 bits
                                       102 ;       are returned.
                                       103 ;
                                       104 ; CALLING SEQUENCE:
                                       105 ;
                                       106 ;       JSB COB$MULQ_R8 (multiplier.rq.r, multiplicand.rq.r, product.wq.r)
                                       107 ;
                                       108 ;       Arguments are passed in R6, R7 and R8.
                                       109 ;
                                       110 ; INPUT PARAMETERS:
                                       111 ;
                                       112 ;       MULTIPLIER.rq.r         Value to the right of the *
                                       113 ;       MULTIPLICAND.rq.r       Value to the left of the *
                                       114 ;
                                       115 ; IMPLICIT INPUTS:
                                       116 ;
                                       117 ;       All of the trap bits in the PSL are assumed off.
                                       118 ;
                                       119 ; OUTPUT PARAMETERS:
                                       120 ;
                                       121 ;       PRODUCT.wq.r            The result of the multiply.
                                       122 ;
                                       123 ; IMPLICIT OUTPUTS:
                                       124 ;
                                       125 ;       NONE
                                       126 ;
                                       127 ; COMPLETION CODES:
                                       128 ;
                                       129 ;       NONE
                                       130 ;
                                       131 ; SIDE EFFECTS:
                                       132 ;
                                       133 ;       Destroys registers R0 through R8.
                                       134 ;
                                       135 ;--
                                       136
                                       137 COB$MULQ_R8::
54   00   67   66   7A                 138         EMUL    (R6),(R7),#0,R4 ; Multiply low half - Result to R4,R5
  50   67     04 A6   C5               139         MULL3   4(R6),(R7),R0           ; Form cross products
  51   04 A7   66   C5                 140         MULL3   (R6),4(R7),R1           ;
          50   51   C0                 141         ADDL2   R1,R0                   ; Sum cross products
       03 66   1F   E1                 142         BBC     #31,(R6),10$            ; Compensate for unsigned bias
          50   67   C0                 143         ADDL2   (R7),R0                 ;
       03 67   1F   E1                 144 10$:    BBC     #31,(R7),20$            ; Compensate for unsigned bias
          50   66   C0                 145         ADDL2   (R6),R0                 ;
          55   50   C0                 146 20$:    ADDL2   R0,R5                   ; Add in cross product
          68   54   7D                 147         MOVQ    R4,(R8)                 ; Return result
                    05                 148         RSB                             ; Return
                                       149 ;
                                       150         .END
```

COB$MULQ_R8 00000000 RG 01

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_COB$CODE	00000027 (39.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:00.43
Command processing	116	00:00:00.30	00:00:02.21
Pass 1	66	00:00:00.24	00:00:02.12
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	41	00:00:00.20	00:00:00.89
Symbol table output	1	00:00:00.00	00:00:00.00
Psect synopsis output	3	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	258	00:00:00.79	00:00:05.67

The working set limit was 900 pages.
1644 bytes (4 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 1 non-local and 2 local symbols.
150 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:COBMULQ/OBJ=OBJ$:COBMULQ MSRC$:COBMULQ/UPDATE=(ENH$:COBMULQ)

COBPAUSE
LIS

COBMSG
LIS

COBHANDLE
LIS

COBINTAR
LIS

COBINTER
LIS

COBMULQ
LIS

COBPOSERA
LIS

COBIOEXCE
LIS

COBLINAGE
LIS

COBKEY
LIS

COBINVUSE
LIS