```
FFFFFFFFFFFFFFF         111            111      XXX         XXX
FFFFFFFFFFFFFFF         111            111      XXX         XXX
FFFFFFFFFFFFFFF         111            111      XXX         XXX
FFF                  111111         111111      XXX         XXX
FFF                  111111         111111      XXX         XXX
FFF                  111111         111111      XXX         XXX
FFF                     111            111         XXX   XXX
FFF                     111            111         XXX   XXX
FFF                     111            111         XXX   XXX
FFFFFFFF.FFF            111            111            XXX
FFFFFFFFFFFF            111            111            XXX
FFFFFFFFFFFF            111            111            XXX
FFF                     111            111         XXX   XXX
FFF                     111            111         XXX   XXX
FFF                     111            111         XXX   XXX
FFF                     111            111      XXX         XXX
FFF                     111            111      XXX         XXX
FFF                     111            111      XXX         XXX
FFF                  111111111      111111111   XXX         XXX
FFF                  111111111      111111111   XXX         XXX
FFF                  111111111      111111111   XXX         XXX
```

```
_$25
Symb
----
IOC
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAK
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

```
RRRRRRRR    DDDDDDDD    BBBBBBBB    LL            000000    KK      KK
RRRRRRRR    DDDDDDDD    BBBBBBBB    LL            000000    KK      KK
RR      RR  DD      DD  BB      BB  LL          00      00  KK      KK
RR      RR  DD      DD  BB      BB  LL          00      00  KK      KK
RR      RR  DD      DD  BB      BB  LL          00      00  KK    KK
RR      RR  DD      DD  BB      BB  LL          00      00  KK    KK
RRRRRRRR    DD      DD  BBBBBBBB    LL          00      00  KKKKKK
RRRRRRRR    DD      DD  BBBBBBBB    LL          00      00  KKKKKK
RR  RR      DD      DD  BB      BB  LL          00      00  KK    KK
RR  RR      DD      DD  BB      BB  LL          00      00  KK    KK
RR    RR    DD      DD  BB      BB  LL          00      00  KK      KK      ....
RR    RR    DD      DD  BB      BB  LL          00      00  KK      KK      ....
RR      RR  DDDDDDDD    BBBBBBBB    LLLLLLLLLL    000000    KK      KK      ....
RR      RR  DDDDDDDD    BBBBBBBB    LLLLLLLLLL    000000    KK      KK      ....


LL            IIIIII      SSSSSSS
LL            IIIIII      SSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSS
LL              II        SSSSS
LL              II            SS
LL              II            SS
LL              II            SS
LL              II            SS
LLLLLLLLLL    IIIIII    SSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSS
```

```
;     1          0001  0 MODULE RDBLOK (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !*****************************************************************************
;     9          0009  1 !*                                                                           *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                      *
;    13          0013  1 !*                                                                           *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    19          0019  1 !*  TRANSFERRED.                                                              *
;    20          0020  1 !*                                                                           *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    23          0023  1 !*  CORPORATION.                                                              *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*****************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module contains routines for basic block I/O, as well
;    38          0038  1 !       as the buffer management mechanism.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  13-Dec-1976  22:48
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-032 CDS0015         Christian D. Saether    30-Aug-1984
;    53          0053  1 !               Allow for multi-header directory files.
;    54          0054  1 !
;    55          0055  1 !       V03-031 CDS0014         Christian D. Saether    24-Aug-1984
;    56          0056  1 !               Removing an entry from the directory index cache
;    57          0057  1 !               free list had one too many levels of indirection,
```

```
;    58          0058  1 !                occasionally messing up the queue.
;    59          0059  1 !
;    60          0060  1 !        V03-030 CDS0013         Christian D. Saether     7-Aug-1984
;    61          0061  1 !                Add KILL_DINDX routine.  Remove tests for dir_fcb
;    62          0062  1 !                and primary_fcb in unhook_bfrd - the fcb$v_dir flag
;    63          0063  1 !                handles those conditions correctly.
;    64          0064  1 !
;    65          0065  1 !        V03-029 CDS0012         Christian D. Saether     6-Aug-1984
;    66          0066  1 !                Fix bug in CDS0011 causing infinite loop when
;    67          0067  1 !                mounting disk /nocache, or while marked for dismount.
;    68          0068  1 !                Use L_NORM linkage for UNHOOK_BFRD.
;    69          0069  1 !
;    70          0070  1 !        V03-028 CDS0011         Christian D. Saether    15-Jul-1984
;    71          0071  1 !                Add support for maintaining directory index blocks.
;    72          0072  1 !                Add MAKE_DIRINDX routine.
;    73          0073  1 !                Remove intermediate blocks from CDS0010.  The
;    74          0074  1 !                problem was in the form of the bind_common
;    75          0075  1 !                declarations and was resolved there.
;    76          0076  1 !
;    77          0077  1 !        V03-027 CDS0010         Christian D. Saether     7-Jul-1984
;    78          0078  1 !                Break up READ_BLOCK and FIND_BUFFER into smaller
;    79          0079  1 !                blocks to stop the absurd amount of cse that the
;    80          0080  1 !                compiler generates.
;    81          0081  1 !
;    82          0082  1 !        V03-026 CDS0009         Christian D. Saether     2-Jul-1984
;    83          0083  1 !                Remove CACHELOCK consistency check so that cell
;    84          0084  1 !                can be used for STS_DISKREAD flag.
;    85          0085  1 !                Raise minimum file header requirements to 3 buffers
;    86          0086  1 !                from 2 buffers.
;    87          0087  1 !
;    88          0088  1 !        V03-025 CDS0008         Christian D. Saether    24-Jun-1984
;    89          0089  1 !                Restore multi-block read capability.
;    90          0090  1 !
;    91          0091  1 !        V03-024 CDS0007         Christian D. Saether    20-Jun-1984
;    92          0092  1 !                Raise/lower process diocnt around qio so that file
;    93          0093  1 !                system i/o is not blocked for lack of quota.  Do
;    94          0094  1 !                same for ASTCNT so it does not fail for lack of
;    95          0095  1 !                ast quota either.
;    96          0096  1 !
;    97          0097  1 !        V03-023 CDS0006         Christian D. Saether    25-May-1984
;    98          0098  1 !                When not cluster accessible, do not bump
;    99          0099  1 !                sequence number on modifies.
;   100          0100  1 !                Add routine to scan appropriate pool and kick
;   101          0101  1 !                out buffers of a given type.
;   102          0102  1 !
;   103          0103  1 !        V03-022 Expand test for clusterness to test whether we are
;   104          0104  1 !                are a cluster at all.
;   105          0105  1 !
;   106          0106  1 !        V03-021 CDS0004         Christian D. Saether     8-May-1984
;   107          0107  1 !                Do not leave lock on buffer if not cluster device.
;   108          0108  1 !                Modify buffer validation for bitmap, index, and quota
;   109          0109  1 !                type buffers.
;   110          0110  1 !
;   111          0111  1 !        V03-020 CDS0003         Christian D. Saether    10-Apr-1984
;   112          0112  1 !                Bump appropriate pool hit/miss counters.
;   113          0113  1 !                Don't set VALID, DIRTY in bfrd in CREATE_BLOCK if
;   114          0114  1 !                for special lbn -1.  Set VALID, DIRTY in bfrd in
```

```
;  115          0115  1 !                   RESET_LBN routine.
;  116          0116  1 !
;  117          0117  1 !       V03-019 CDS0002         Christian D. Saether     4-Apr-1984
;  118          0118  1 !               On lockbasis mismatch, when tolerated at all (for
;  119          0119  1 !               blocks in the data or headers), toss the buffer out
;  120          0120  1 !               to get the lock dissociated because it's the wrong
;  121          0121  1 !               one.
;  122          0122  1 !
;  123          0123  1 !       V03-018 CDS0001         Christian D. Saether     20-Mar-1984
;  124          0124  1 !               Establish BASE as AST parameter in SERIAL_CACHE routine.
;  125          0125  1 !
;  126          0126  1 !       V03-017 ACG0408         Andrew C. Goldstein,     23-Mar-1984  14:47
;  127          0127  1 !               Add AST parameter so that impure storage is fully based
;  128          0128  1 !
;  129          0129  1 !       V03-016 ACG0403         Andrew C. Goldstein,     15-Mar-1984  17:35
;  130          0130  1 !               Correct test for new buffer in WRONG_LOCKBASIS;
;  131          0131  1 !               fix buffer address computation in TOSS_CACHE_DATA.
;  132          0132  1 !
;  133          0133  1 !       V03-015 CDS     Christian D. Saether     9-Mar-1984
;  134          0134  1 !               Rewrite cache routines for shared paged pool caching.
;  135          0135  1 !
;  136          0136  1 !**
;  137          0137  1
;  138          0138  1
;  139          0139  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  140          0140  1 REQUIRE 'SRC$:FCPDEF.B32';
;  141          1131  1
;  142          1132  1 FORWARD ROUTINE
;  143          1133  1         GET_REQD_BFR_CREDITS : L_NORM NOVALUE,
;  144          1134  1         RETURN_BFRD     : L_JSB_TARG NOVALUE,
;  145          1135  1         FIND_BUFFER     : L_NORM,
;  146          1136  1         INVALIDATE      : L_NORM NOVALUE,
;  147          1137  1         UNHOOK_BFRD     : L_NORM NOVALUE,
;  148          1138  1         WRITE_BLOCK     : L_NORM NOVALUE,
;  149          1139  1         FREE_ONE        : L_JSB NOVALUE,
;  150          1140  1         WAKE_WAITER     : NOVALUE,
;  151          1141  1         RELEASE_CACHE   : L_JSB NOVALUE,
;  152          1142  1         SERIAL_CACHE    : L_JSB NOVALUE;
;  153          1143  1
;  154          1144  1 EXTERNAL ROUTINE
;  155          1145  1         WAIT_FOR_AST    : NOVALUE,      ! exit thread until completion ast
;  156          1146  1         CONTINUE_THREAD : NOVALUE;      ! completion AST to resume thread
;  157          1147  1
;  158          1148  1 EXTERNAL
;  159          1149  1         CTL$GL_PHD      : REF BBLOCK ADDRESSING_MODE (GENERAL),
;  160          1150  1         CTL$GL_PCB      : REF BBLOCK ADDRESSING_MODE (GENERAL);
;  161          1151  1
;  162          1152  1 MACRO
;  163          1153  1         QFLNK = 0,0,32,0 %,
;  164          1154  1         QBLNK = 4,0,32,0 %,
;  165          1155  1
;  166        M 1156  1         LOOKUP_LBN (LBN) =
;  167        M 1157  1
;  168        M 1158  1                 BEGIN
;  169        M 1159  1                 BIND
;  170        M 1160  1                     HSHTBL = .CACHE_HDR [F11BC$L_LBNHSHBAS] : VECTOR [,WORD];
;  171        M 1161  1
```

```
;  172        M 1162  1                HSHTBL [ABS (LBN MOD .CACHE_HDR [F11BC$W_LBNHSHCNT])]
;  173          1163  1                END %,
;  174          1164  1
;  175        M 1165  1        LOCKUP_LOCK (LOCKBASIS, PARLKID) =
;  176        M 1166  1
;  177        M 1167  1                BEGIN
;  178        M 1168  1                BIND
;  179        M 1169  1                    HSHTBL = .CACHE_HDR [F11BC$L_BLHSHBAS] : VECTOR [,WORD];
;  180        M 1170  1
;  181        M 1171  1                HSHTBL [ABS ((LOCKBASIS + PARLKID) MOD .CACHE_HDR [F11BC$W_BLHSHCNT])]
;  182          1172  1                END %,
;  183          1173  1
;  184          1174  1 ! Calculate address of given descriptor within 1-based descriptor blockvectors.
;  185          1175  1 !
;  186          1176  1
;  187        M 1177  1        BFRD_ADDR (BAS1INDX) =
;  188          1178  1                .CACHE_HDR [F11BC$L_BFRDBAS] + (BAS1INDX - 1)*BFRD$S_BFRDDEF %,
;  189          1179  1
;  190        M 1180  1        BFRLD_ADDR (BAS1INDX) =
;  191          1181  1                .CACHE_HDR [F11BC$L_BFRLDBAS] + (BAS1INDX - 1)*BFRL$S_BFRLDEF %;
;  192          1182  1
;  193          1183  1 BIND
;  194          1184  1        POOL_TABLE      = UPLIT BYTE (  2,      ! file headers
;  195          1185  1                                        0,      ! storage map
;  196          1186  1                                        1,      ! directories
;  197          1187  1                                        2,      ! index file blocks
;  198          1188  1                                        1,      ! random data blocks
;  199          1189  1                                        1,      ! quota file blocks
;  200          1190  1                                        3       ! directory index blocks
;  201          1191  1                        ) : VECTOR [,BYTE];
```

```
;  203          1192  1 GLOBAL ROUTINE MAKE_DIRINDX (FCB) : L_NORM =
;  204          1193  1
;  205          1194  1 !++
;  206          1195  1 !
;  207          1196  1 ! FUNCTIONAL DESCRIPTION:
;  208          1197  1 !
;  209          1198  1 ! This routine validates the directory index pointed to from the
;  210          1199  1 ! given FCB, under the cache interlock.
;  211          1200  1 !
;  212          1201  1 ! If the given FCB does not have a directory index, attempt to
;  213          1202  1 ! attach one to it.
;  214          1203  1 !
;  215          1204  1 ! ROUTINE VALUE:
;  216          1205  1 !
;  217          1206  1 !       0 - if the given FCB is invalid
;  218          1207  1 !       ' - if the given FCB is valid
;  219          1208  1 !
;  220          1209  1 !--
;  221          1210  1
;  222          1211  2 BEGIN
;  223          1212  2
;  224          1213  2 MAP
;  225          1214  2         FCB     : REF BBLOCK;
;  226          1215  2
;  227          1216  2 BIND_COMMON;
;  228          1217  2
;  229          1218  2 EXTERNAL
;  230          1219  2         PMS$GL_DIRHIT   : ADDRESSING_MODE (GENERAL),
;  231          1220  2         PMS$GL_DIRMISS  : ADDRESSING_MODE (GENERAL);
;  232          1221  2 LOCAL
;  233          1222  2         DIRINDX         : REF BBLOCK FIELD (DIRC),
;  234          1223  2         DNDX_BFRD       : REF BBLOCK,
;  235          1224  2         FCBVALID        : INITIAL (0);
;  236          1225  2
;  237          1226  2 IF .LB_HDRSEQ [.DIR_LCKINDX] EQL 0
;  238          1227  2 THEN
;  239          1228  2     LB_HDRSEQ [.DIR_LCKINDX] = .LB_HDRSEQ [.DIR_LCKINDX] + 1;
;  240          1229  2
;  241          1230  2 IF .LB_DATASEQ [.DIR_LCKINDX] EQL 0
;  242          1231  2 THEN
;  243          1232  2     LB_DATASEQ [.DIR_LCKINDX] = .LB_DATASEQ [.DIR_LCKINDX] + 1;
;  244          1233  2
;  245          1234  2 SERIAL_CACHE ();
;  246          1235  2
;  247          1236  2 IF (DIRINDX = .FCB [FCB$L_DIRINDX]) EQL 0
;  248          1237  2 THEN
;  249          1238  3     BEGIN
;  250          1239  3     LOCAL
;  251          1240  3         INDX0,
;  252          1241  3         POOL_LRU        : REF BBLOCK;
;  253          1242  3
;  254          1243  3     POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + 3*8;
;  255          1244  3
;  256          1245  3     REMQUE (.POOL_LRU [QFLNK], DNDX_BFRD);
;  257          1246  3
;  258          1247  3     INDX0 = ((.DNDX_BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF);
;  259          1248  3
```

```
;  260          1249  3      UNHOOK_BFRD (.INDX0 + 1, .DNDX_BFRD);
;  261          1250  3
;  262          1251  3      DIRINDX = (.INDX0*512) + .CACHE_HDR [F11BC$L_BUFBASE];
;  263          1252  3
;  264          1253  3      DNDX_BFRD [BFRD$L_LBN] = .FCB;
;  265          1254  3      DNDX_BFRD [BFRD$B_BTYPE] = DIRINDX_TYPE;
;  266          1255  3      DNDX_BFRD [BFRD$L_LOCKBASIS] = .LB_BASIS [.DIR_LCKINDX];
;  267          1256  3      DNDX_BFRD [BFRD$L_UCB] = .CURRENT_UCB;
;  268          1257  3      FCB [FCB$L_DIRINDX] = .DIRINDX;
;  269          1258  3      DNDX_BFRD [BFRD$V_VALID] = 1;
;  270          1259  3      DIRINDX [DIRC$W_INUSE] = 0;
;  271          1260  3      PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
;  272          1261  3      END
;  273          1262  3  ELSE
;  274          1263  3      BEGIN
;  275          1264  3
;  276          1265  4      DNDX_BFRD = (((.DIRINDX - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF)
;  277          1266  3                  + .CACHE_HDR [F11BC$L_BFRDBAS];
;  278          1267  3
;  279          1268  3      REMQUE (.DNDX_BFRD, DNDX_BFRD);
;  280          1269  3
;  281          1270  3      IF .DNDX_BFRD [BFRD$L_SEQNUM] EQL .LB_HDRSEQ [.DIR_LCKINDX]
;  282          1271  3      THEN
;  283          1272  4          BEGIN
;  284          1273  4          FCBVALID = 1;
;  285          1274  4
;  286          1275  4          IF .DIRINDX [DIRC$L_DATASEQ] NEQ .LB_DATASEQ [.DIR_LCKINDX]
;  287          1276  4          THEN
;  288          1277  5              BEGIN
;  289          1278  5              DIRINDX [DIRC$W_INUSE] = 0;
;  290          1279  5              PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
;  291          1280  5              END
;  292          1281  4          ELSE
;  293          1282  4              PMS$GL_DIRHIT = .PMS$GL_DIRHIT + 1;
;  294          1283  4
;  295          1284  4          END
;  296          1285  3      ELSE
;  297          1286  4          BEGIN
;  298          1287  4          DIRINDX [DIRC$W_INUSE] = 0;
;  299          1288  4          PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
;  300          1289  3          END;
;  301          1290  3
;  302          1291  2      END;
;  303          1292  2
;  304          1293  2  IF .BFRS_USED [3] NEQ 0
;  305          1294  2  THEN
;  306          1295  3      BUG_CHECK (XQPERR, 'should not have any in use')
;  307          1296  2  ELSE
;  308          1297  2      BFRS_USED [3] = 1;
;  309          1298  2
;  310          1299  2  IF .DNDX_BFRD [BFRD$W_CURPID] NEQ 0
;  311          1300  2  THEN
;  312          1301  3      BUG_CHECK (XQPERR, 'directory index buffer should not be in use')
;  313          1302  2  ELSE
;  314          1303  2      DNDX_BFRD [BFRD$W_CURPID] = .CTL$GL_PCB [PCB$L_PID];
;  315          1304  2
;  316          1305  2  INSQUE (.DNDX_BFRD, .BFR_LIST [3, QBLNK]);
```

```
;  317          1306  2
;  318          1307  2 RELEASE_CACHE ();
;  319          1308  2
;  320          1309  2 .FCBVALID
;  321          1310  2
;  322          1311  1 END;              ! of routine MAKE_DIRINDX


                                                        .TITLE  RDBLOK
                                                        .IDENT  \V04-000\

                                                        .PSECT  $CODE$,NOWRT,2

                    03  01  01  02  01  00  02  00000 P.AAA:  .BYTE   2, 0, 1, 2, 1, 1, 3                          ;

                                                 POOL_TABLE=        P.AAA
                                                        .EXTRN  WAIT_FOR_AST, CONTINUE_THREAD
                                                        .EXTRN  CTL$GL_PHD, CTL$GL_PCB
                                                        .EXTRN  PMS$GL_DIRHIT, PMS$GL_DIRMISS
                                                        .EXTRN  BUG$_XQPERR

                                      01FC 00000        .ENTRY  MAKE_DIRINDX, Save R2,R3,R4,R5,R6,R7,R8       ; 1192
                58        CC    AA  9E 00002        MOVAB   -52(BASE), R8                                 ; 1214
                55        F4    AA  9E 00006        MOVAB   -12(BASE), R5                                 ;
                56      00D4    CA  9E 0000A        MOVAB   212(BASE), R6                                 ;
                              57    D4 0000F        CLRL    FCBVALID                                      ;
                50              66    D0 00011        MOVL    (R6), R0                                      ; 1226
                50      0094 CA40   DE 00014        MOVAL   148(BASE)[R0], R0                             ;
                              60    D5 0001A        TSTL    (R0)                                          ;
                              02    12 0001C        BNEQ    1$                                            ;
                              60    D6 0001E        INCL    (R0)                                          ; 1228
                50              66    D0 00020 1$:    MOVL    (R6), R0                                      ; 1230
                50      00A8 CA40   DE 00023        MOVAL   168(BASE)[R0], R0                             ;
                              60    D5 00029        TSTL    (R0)                                          ;
                              02    12 0002B        BNEQ    2$                                            ;
                              60    D6 0002D        INCL    (R0)                                          ; 123c
                          0000V 30    00021 2$:    BSBW    SERIAL_CACHE                                  ; 1234
                50        04    AC  D0 00032        MOVL    FCB, R0                                       ; 1236
                54      00B0    C0  D0 00036        MOVL    176(R0), DIRINDX                              ;
                              54    12 0003B        BNEQ    3$                                            ;
            50        FC    AA 00000040 8F  C1 0003D        ADDL3   #64, -4(BASE), POOL_LRU                       ; 1243
                52        00    B0  0F 00046        REMQUE  @0(POOL_LRU), DNDX_BFRD                       ; 1245
                50        FC    AA  D0 0004A        MOVL    -4(BASE), R0                                  ; 1247
            50  52        18    A0  C3 0004E        SUBL3   24(R0), DNDX_BFRD, R0                         ;
            53  50                20  C7 00053        DIVL3   #32, R0, INDX0                                ;
                              52    DD 00057        PUSHL   DNDX_BFRD                                     ; 1249
                        01    A3  9F 00059        PUSHAB  1(INDX0)                                      ;
                0000V   CF          02  FB 0005C        CALLS   #2, UNHOOK_BFRD                               ;
            53  53                09  78 00061        ASHL    #9, R3, R3                                    ; 1251
            54  53          FC    BA  C1 00065        ADDL3   @-4(BASE), R3, DIRINDX                        ;
                08    A2        04  AC  D0 0006A        MOVL    FCB, 8(DNDX_BFRD)                             ; 1253
                19    A2              06  90 0006F        MOVB    #6, 25(DNDX_BFRD)                             ; 1254
                50              66    D0 00073        MOVL    (R6), R0                                      ; 1255
                10    A2      0080 CA40   D0 00076        MOVL    128(BASE)[R0], 16(DNDX_BFRD)                  ;
                0C    A2        94    AA  D0 0007D        MOVL    -108(BASE), 12(DNDX_BFRD)                     ; 1256
                50        04    AC  D0 00082        MOVL    FCB, R0                                       ; 1257
            00B0    C0              54  D0 00086        MOVL    DIRINDX, 176(R0)                              ;
```

```
                 18   A2              08  88 0008B          BISB2   #8, 24(DNDX_BFRD)                       ; 1258
                                      3D  11 0008F          BRB     4$                                      ; 1259
                      50        FC    AA  D0 00091 3$:      MOVL    -4(BASE), R0                            ; 1265
51                    54              60  C3 00095          SUBL3   (R0), DIRINDX, R1
                      51 00000200     8F  C6 00099          DIVL2   #512, R1
                      51              20  C4 000A0          MULL2   #32, R1
52                    51        18    A0  C1 000A3          ADDL3   24(R0), R1, DNDX_BFRD                   ; 1266
                      52              62  0F 000A8          RFMQUE  (DNDX_BFRD), DNDX_BFRD                  ; 1268
                      50              66  D0 000AB          MOVL    (R6), R0                                ; 1270
                0094 CA40        14   A2  D1 000AE          CMPL    20(DNDX_BFRD), 148(BASE)[R0]
                                      17  12 000B5          BNEQ    4$
                      57              01  D0 000B7          MOVL    #1, FCBVALID                            ; 1273
                      50              66  D0 000BA          MOVL    (R6), R0                                ; 1275
                00A8 CA40        08   A4  D1 000BD          CMPL    8(DIRINDX), 168(BASE)[R0]
                                      08  12 000C4          BNEQ    4$
                           00000000G  00  D6 000C6          INCL    PMS$GL_DIRHIT                           ; 1282
                                      08  11 000CC          BRB     5$                                      ; 1270
                                      64  B4 000CE 4$:      CLRW    (DIRINDX)                               ; 1287
                           00000000G  00  D6 000D0          INCL    PMS$GL_DIRMISS                          ; 1288
                                 06   A5  B5 000D6 5$:      TSTW    6(R5)                                   ; 1293
                                      06  13 000D9          BEQL    6$
                                        FEFF 000DB          BUGW                                            ; 1295
                                       0000* 000DD          .WORD   <BUG$_XQPERR!4>
                                      04  11 000DF          BRB     7$
                 06   A5                01  B0 000E1 6$:      MOVW    #1, 6(R5)                               ; 1297
                                 1C   A2  B5 000E5 7$:      TSTW    28(DNDX_BFRD)                           ; 1299
                                      06  13 000E8          BEQL    8$
                                        FEFF 000EA          BUGW                                            ; 1301
                                       0000* 000EC          .WORD   <BUG$_XQPERR!4>
                                      0C  11 000EE          BRB     9$
                      50 00000000G    00  D0 000F0 8$:      MOVL    CTL$GL_PCB, R0                          ; 1303
                 1C   A2        60    A0  B0 000F7          MOVW    96(R0), 28(DNDX_BFRD)
                 1C   B8              62  0E 000FC 9$:      INSQUE  (DNDX_BFRD), @28(R8)                    ; 1305
                                    0000V 30 00100          BSBW    RELEASE_CACHE                           ; 1307
                      50              57  D0 00103          MOVL    FCBVALID, R0                            ; 1311
                                          04 00106          RET                                             ;

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0007
```

```
;  324     1312  1 GLOBAL ROUTINE KILL_DINDX (FCB) : L_NORM NOVALUE =
;  325     1313  1
;  326     1314  1 !++
;  327     1315  1 !
;  328     1316  1 ! FUNCTIONAL DESCRIPTION:
;  329     1317  1 !
;  330     1318  1 !       Invalidate the directory index block pointed to by
;  331     1319  1 !       the given fcb, if any.  This is done under the cache
;  332     1320  1 !       serialization lock.
;  333     1321  1 !
;  334     1322  1 !--
;  335     1323  1
;  336     1324  2 BEGIN
;  337     1325  2
;  338     1326  2 MAP
;  339     1327  2         FCB     : REF BBLOCK;
;  340     1328  2
;  341     1329  2 BIND_COMMON;
;  342     1330  2
;  343     1331  2 LOCAL
;  344     1332  2         INDXO,
;  345     1333  2         BFRD    : REF BBLOCK,
;  346     1334  2         DIRINDX : REF BBLOCK,
;  347     1335  2         PIDINDX : WORD;
;  348     1336  2
;  349     1337  2 SERIAL_CACHE ();
;  350     1338  2
;  351     1339  2 IF (DIRINDX = .FCB [FCB$L_DIRINDX]) NEQ 0
;  352     1340  2 THEN
;  353     1341  3     BEGIN
;  354     1342  3     INDXO = (.DIRINDX - .CACHE_HDR [F11BC$L_BUFBASE])/512;
;  355     1343  3     BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + BFRD$S_BFRDDEF*(.INDXO);
;  356     1344  3
;  357     1345  3     PIDINDX = .CTL$GL_PCB [PCB$L_PID];
;  358     1346  3
;  359     1347  3     IF .BFRD [BFRD$W_CURPID] NEQ 0
;  360     1348  3     THEN
;  361     1349  4         BEGIN
;  362     1350  4         IF .BFRD [BFRD$W_CURPID] NEQ .PIDINDX
;  363     1351  4         THEN
;  364     1352  5             BUG_CHECK (XQPERR, 'should not belong to anyone else')
;  365     1353  4         ELSE
;  366     1354  5             BEGIN
;  367     1355  5
;  368     1356  5 ! If this one is in-process, simply clear valid to cause it to
;  369     1357  5 ! be cleaned up when locks are run down.  Also clear the pointer
;  370     1358  5 ! to the fcb in the bfrd.
;  371     1359  5 !
;  372     1360  5
;  373     1361  5             BFRD [BFRD$V_VALID] = 0;
;  374     1362  5             BFRD [BFRD$L_LBN] = 0;
;  375     1363  4             END;
;  376     1364  4         END
;  377     1365  3     ELSE
;  378     1366  4         BEGIN
;  379     1367  4         UNHOOK_BFRD (.INDXO+1, .BFRD);
;  380     1368  4         RETURN_BFRD (.BFRD);
```

```
;  381         1369  3            END;
;  382         1370  3
;  383         1371  3        FCB [FCB$L_DIRINDX] = 0;
;  384         1372  3
;  385         1373  2        END;
;  386         1374  2
;  387         1375  2  RELEASE_CACHE ();
;  388         1376  2
;  389         1377  1  END;                  ! of routine KILL_DINDX


                                   000C 00000          .ENTRY  KILL_DINDX, Save R2,R3          ; 1312
                             0000V  30 00002           BSBW    SERIAL_CACHE                    ; 1337
             50        04     AC    D0 00005           MOVL    FCB, R0                         ; 1339
             50      00B0     C0    D0 00009           MOVL    176(R0), DIRINDX                ;
                              54    13 0000E           BEQL    4$                              ;
             50        FC     BA    C2 00010           SUBL2   @-4(BASE), R0                   ; 1342
             50 00000200      8F    C6 00014           DIVL2   #512, INDX0                     ;
             51        FC     AA    D0 0001B           MOVL    -4(BASE), R1                    ; 1343
      52     50               05    78 0001F           ASHL    #5, INDX0, R2                   ;
             52        18     A1    C0 00023           ADDL2   24(R1), BFRD                    ;
             51 00000000G     00    D0 00027           MOVL    CTL$GL_PCB, R1                  ; 1345
             53        60     A1    B0 0002E           MOVW    96(R1), PIDINDX                 ;
                       1C     A2    B5 00032           TSTW    28(BFRD)                        ; 1347
                              15    13 00035           BEQL    2$                              ;
             53        1C     A2    B1 00037           CMPW    28(BFRD), PIDINDX               ; 1350
                              06    13 0003B           BEQL    1$                              ;
                                  FEFF 0003D           BUGW                                    ; 1352
                                 0000* 0003F           .WORD   <BUG$_XQPERR!4>                 ;
                              19    11 00041           BRB     3$                              ;
          18 A2               08    8A 00043  1$:      BICB2   #8, 24(BFRD)                    ; 1361
                       08     A2    D4 00047           CLRL    8(BFRD)                         ; 1362
                              10    11 0004A           BRB     3$                              ; 1347
                              52    DD 0004C  2$:      PUSHL   BFRD                            ; 1367
                       01     A0    9F 0004E           PUSHAB  1(INDX0)                        ;
       0000V CF               02    FB 00051           CALLS   #2, UNHOOK_BFRD                 ;
             50               52    D0 00056           MOVL    BFRD, R0                        ; 1368
                             0000V  30 00059           BSBW    RETURN_BFRD                     ;
             50        04     AC    D0 0005C  3$:      MOVL    FCB, R0                         ; 1371
                     00B0     C0    D4 00060           CLRL    176(R0)                         ;
                             0000V  30 00064  4$:      BSBW    RELEASE_CACHE                   ; 1375
                                    04 00067           RET                                     ; 1377

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 010E
```

```
;  391     1378  1 GLOBAL ROUTINE READ_BLOCK (LBN, COUNT, TYPE) : L_NORM =
;  392     1379  1
;  393     1380  1 !++
;  394     1381  1 !
;  395     1382  1 ! FUNCTIONAL DESCRIPTION:
;  396     1383  1 !
;  397     1384  1 !       This routine reads the desired block(s) from the disk.
;  398     1385  1 !       Blocks are categorized by type to aid buffer management.
;  399     1386  1 !       Note that the caller assumes only one block is ever read; multiple
;  400     1387  1 !       blocks read ahead are acquired through cache hits on subsequent calls.
;  401     1388  1 ! CALLING SEQUENCE:
;  402     1389  1 !       READ_BLOCK (ARG1, ARG2, ARG3)
;  403     1390  1 !
;  404     1391  1 ! INPUT PARAMETERS:
;  405     1392  1 !       ARG1: LBN of block(s)
;  406     1393  1 !       ARG2: number of blocks to read
;  407     1394  1 !       ARG3: block type code
;  408     1395  1 !
;  409     1396  1 ! IMPLICIT INPUTS:
;  410     1397  1 !
;  411     1398  1 ! OUTPUT PARAMETERS:
;  412     1399  1 !       NONE
;  413     1400  1 !
;  414     1401  1 ! IMPLICIT OUTPUTS:
;  415     1402  1 !       IO_STATUS receives status of I/O transfer
;  416     1403  1 !
;  417     1404  1 ! ROUTINE VALUE:
;  418     1405  1 !       address of buffer containing block
;  419     1406  1 !
;  420     1407  1 ! SIDE EFFECTS:
;  421     1408  1 !       BLOCK READ
;  422     1409  1 !
;  423     1410  1 !--
;  424     1411  1
;  425     1412  2 BEGIN
;  426     1413  2
;  427     1414  2
;  428     1415  2 LOCAL
;  429     1416  2         I,                              ! index of buffer used
;  430     1417  2         BFRD            : REF BBLOCK,   ! pointer to buffer descriptor
;  431     1418  2         STATUS,                         ! QIO service status
;  432     1419  2         FOUND_COUNT;                    ! count of buffers gotten
;  433     1420  2
;  434     1421  2 EXTERNAL
;  435     1422  2         ACP$GB_DATACHK  : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
;  436     1423  2                                         ! ACP datacheck enable flags
;  437     1424  2
;  438     1425  2 BIND_COMMON;
;  439     1426  2
;  440     1427  2 EXTERNAL LITERAL
;  441     1428  2         ACP$V_READCHK   : UNSIGNED (6); ! read check enable flag
;  442     1429  2
;  443     1430  2 ! Find a suitable block buffer. If it does not already contain the block,
;  444     1431  2 ! read it.
;  445     1432  2 !
;  446     1433  2
;  447     1434  2 STSFLGS [STS_DISKREAD] = 0;
```

```
;  448          1435  2
;  449          1436  2 I = FIND_BUFFER (.LBN, .TYPE, .COUNT, FOUND_COUNT);
;  450          1437  2
;  451          1438  2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + (.I*BFRD$S_BFRDDEF);
;  452          1439  2
;  453          1440  2 IF .BFRD [BFRD$V_VALID]
;  454          1441  2 THEN
;  455          1442  2     RETURN .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512);
;  456          1443  2
;  457          1444  3     BEGIN
;  458          1445  3     LOCAL
;  459          1446  3         PTR     : REF BBLOCK,
;  460          1447  3         SAVE_PRIV : VECTOR [4];
;  461          1448  3
;  462          1449  3     STSFLGS [STS_DISKREAD] = 1;
;  463          1450  3
;  464          1451  3     PTR = .CTL$GL_PCB;
;  465          1452  3     PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] + 1;
;  466          1453  3     PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] + 1;
;  467          1454  3     SAVE_PRIV [0] = .(PTR [PCB$Q_PRIV]);
;  468          1455  3     SAVE_PRIV [1] = .(PTR [PCB$Q_PRIV]+4);
;  469          1456  3     BBLOCK [PTR [PCB$Q_PRIV], PRV$V_LOG_IO] = 1;
;  470          1457  3     BBLOCK [PTR [PCB$Q_PRIV], PRV$V_BYPASS] = 1;
;  471          1458  3     PTR = .CTL$GL_PHD;
;  472          1459  3     SAVE_PRIV [2] = .(PTR [PHD$Q_PRIVMSK]);
;  473          1460  3     SAVE_PRIV [3] = .(PTR [PHD$Q_PRIVMSK]+4);
;  474          1461  3     BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_LOG_IO] = 1;
;  475          1462  3     BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_BYPASS] = 1;
;  476          1463  3
;  477          1464  3     PMS_TOT_READ = .PMS_TOT_READ + 1;
;  478        P 1465  3     STATUS = $QIO (
;  479        P 1466  3                 EFN    = EFN,
;  480        P 1467  3                 ASTADR = CONTINUE_THREAD,
;  481        P 1468  3                 ASTPRM = .BASE,
;  482        P 1469  3                 CHAN   = .IO_CHANNEL,
;  483        P 1470  3                 FUNC   = (IO$_READLBLK
;  484        P 1471  3                           OR .ACP$GB_DATACHK[ACP$V_READCHK]
;  485        P 1472  3                           ^ $BITPOSITION (IO$V_DATACHECK)),
;  486        P 1473  3                 IOSB   = IO_STATUS,
;  487        P 1474  3                 P1     = .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512),
;  488        P 1475  3                 P2     = .FOUND_COUNT*512,
;  489        P 1476  3                 P3     = .LBN
;  490          1477  3                 );
;  491          1478  3
;  492          1479  3     (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
;  493          1480  3     (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
;  494          1481  3     PTR = .CTL$GL_PCB;
;  495          1482  3     PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] - 1;
;  496          1483  3     PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
;  497          1484  3     (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
;  498          1485  3     (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];
;  499          1486  3
;  500          1487  3     IF NOT .STATUS
;  501          1488  3     THEN IO_STATUS = .STATUS
;  502          1489  3     ELSE WAIT_FOR_AST();
;  503          1490  3
;  504          1491  3     IF NOT .IO_STATUS
```

```
;  505          1492  3      THEN
;  506          1493  4          BEGIN
;  507          1494  4          INCR J FROM 0 TO .FOUND_COUNT-1
;  508          1495  4          DO
;  509          1496  4              INVALIDATE (.CACHE_HDR [F11BC$L_BUFBASE] + (.I+.J)*512);
;  510          1497  4          ERR_EXIT (.IO_STATUS[0]);
;  511          1498  3          END;
;  512          1499  3
;  513          1500  3      INCR J FROM 0 TO .FOUND_COUNT - 1
;  514          1501  3      DO
;  515          1502  4          BEGIN
;  516          1503  4          BFRD [BFRD$V_VALID] = 1;
;  517          1504  4          BFRD = .BFRD + BFRD$S_BFRDEF;
;  518          1505  3          END;
;  519          1506  3
;  520          1507  2      END;
;  521          1508  2
;  522          1509  3 RETURN .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512)
;  523          1510  3
;  524          1511  1 END;                                    ! end of routine READ_BLOCK


                                                        .EXTRN  ACP$GB_DATACHK, ACP$V_READCHK
                                                        .EXTRN  SYS$QIO

                                        00FC 00000      .ENTRY  READ_BLOCK, Save R2,R3,R4,R5,R6,R7              ; 1378
                57 00000000G    00      9E 00002        MOVAB   CTL$GL_PCB, R7                                  ;
                5E                      14 C2 00009     SUBL2   #20, SP                                         ;
                54          FC          AA 9E 0000C     MOVAB   -4(BASE), R4                                    ; 1422
             A6 AA                      01 8A 00010     BICB2   #1, -90(BASE)                                   ; 1434
                                        5E DD 00014     PUSHL   SP                                              ; 1436
                                    08  AC DD 00016     PUSHL   COUNT                                           ;
                                    0C  AC DD 00019     PUSHL   TYPE                                            ;
                                    04  AC DD 0001C     PUSHL   LBN                                             ;
                    0000V   CF          04 FB 0001F     CALLS   #4, FIND_BUFFER                                 ;
                            56          50 D0 00024     MOVL    R0, I                                           ;
                            51          64 D0 00027     MOVL    (R4), R1                                        ; 1438
             50             56          05 78 0002A     ASHL    #5, I, R0                                       ;
             53             50      18  A1 C1 0002E     ADDL3   24(R1), R0, BFRD                                ;
             03     18      A3          03 E1 00033     BBC     #3, 24(BFRD), 1$                                ; 1440
                                00CC    31 00038        BRW     9$                                              ;
                    A6      AA          01 88 0003B 1$: BISB2   #1, -90(BASE)                                   ; 1449
                            52          67 D0 0003F     MOVL    CTL$GL_PCB, PTR                                 ; 1451
                                    3E  A2 B6 00042     INCW    62(PTR)                                         ; 1452
                                    38  A2 B6 00045     INCW    56(PTR)                                         ; 1453
                    04      AE      0084 C2 7D 00048    MOVQ    132(PTR), SAVE_PRIV                             ; 1454
                    0084    C2 20000080 8F C8 0004E     BISL2   #536871040, 132(PTR)                            ; 1457
                            52 00000000G    00 D0 00057 MOVL    CTL$GL_PHD, PTR                                 ; 1458
                    0C      AE          62 7D 0005E     MOVQ    (PTR), SAVE_PRIV+8                              ; 1459
                            62 20000080 8F C8 00062     BISL2   #536871040, (PTR)                               ; 1462
                                08E8    CA D6 00069     INCL    2280(BASE)                                      ; 1464
                                        7E 7C 0006D     CLRQ    -(SP)                                           ; 1477
                                        7E D4 0006F     CLRL    -(SP)                                           ;
                                    04  AC DD 00071     PUSHL   LBN                                             ;
             7E     10      AE          09 78 00074     ASHL    #9, FOUND_COUNT, -(SP)                          ;
                            51      00  B4 D0 00079     MOVL    @0(R4), RT                                      ;
```

```
            50          56              09  78 0007D         ASHL     #9, I, R0                                  ;
                                      6041  9F 00081         PUSHAB   (R0)[R1]                                   ;
                                        5A  DD 00084         PUSHL    BASE                                       ;
                                  0000G CF  9F 00086         PUSHAB   CONTINUE_THREAD                            ;
                                     88 AA  9F 0008A         PUSHAB   -120(BASE)                                 ;
   50 00000000G 9F          01         00G  EF 0008D         EXTZV    S^ACP$V_READCHK, #1, @#ACP$GB_DATACHK, R0  ;
            50          50              0E  78 00096         ASHL     #14, R0, R0                                ;
            7E          50              21  C9 0009A         BISL3    #33, R0, -(SP)                             ;
                                  FF78  CA  DD 0009E         PUSHL    -136(BASE)                                 ;
                                        1E  DD 000A2         PUSHL    #30                                        ;
                  00000000G 00          0C  FB 000A4         CALLS    #12, SYS$QIO                               ;
                            62      0C  AE  7D 000AB         MOVQ     SAVE_PRIV+8, (PTR)                         ; 1479
                            52          67  D0 000AF         MOVL     CTL$GL_PCB, PTR                            ; 1481
                                    3E  A2  B7 000B2         DECW     62(PTR)                                    ; 1482
                                    38  A2  B7 000B5         DECW     56(PTR)                                    ; 1483
                      0084  C2      04  AE  7D 000B8         MOVQ     SAVE_PRIV, 132(PTR)                        ; 1484
                            06          50  E8 000BE         BLBS     STATUS, 2$                                 ; 1487
                        88  AA          50  D0 000C1         MOVL     STATUS, -120(BASE)                         ; 1488
                                        05  11 000C5         BRB      3$                                         ;
                     0000G  CF          00  FB 000C7 2$:     CALLS    #0, WAIT_FOR_AST                           ; 1489
                            24      88  AA  E8 000CC 3$:     BLBS     -120(BASE), 6$                             ; 1491
                            55          6E  D0 000D0         MOVL     FOUND_COUNT, R5                            ; 1494
                            52          01  CE 000D3         MNEGL    #1, J                                      ;
                                        14  11 000D6         BRB      5$                                         ;
                            51      00  B4  D0 000D8 4$:     MOVL     @0(R4), R1                                 ; 1496
            50              56          52  C1 000DC         ADDL3    J, I, R0                                   ;
            50              50          09  78 000E0         ASHL     #9, R0, R0                                 ;
                                      6041  9F 000E4         PUSHAB   (R0)[R1]                                   ;
                     0000V  CF          01  FB 000E7         CALLS    #1, INVALIDATE                             ;
            E8              52          55  F2 000EC 5$:     AOBLSS   R5, J, 4$                                  ;
                                    88  AA  BF 000F0         CHMU     -120(BASE)                                 ; 1497
                                            04 000F3         RET                                                 ;
                            51          6E  D0 000F4 6$:     MOVL     FOUND_COUNT, R1                            ; 1500
                            50          01  CE 000F7         MNEGL    #1, J                                      ;
                                        07  11 000FA         BRB      8$                                         ;
                        18  A3          08  88 000FC 7$:     BISB2    #8, 24(BFRD)                               ; 1503
                            53          20  C0 00100         ADDL2    #32, BFRD                                  ; 1504
            F5              50          51  F2 00103 8$:     AOBLSS   R1, J, 7$                                  ; 1500
            50              56          09  78 00107 9$:     ASHL     #9, I, R0                                  ; 1509
                            50      00  B4  C0 0010B         ADDL2    @0(R4), R0                                 ;
                                            04 0010F         RET                                                 ; 1511

; Routine Size:  272 bytes,    Routine Base:  $CODE$ + 0176
```

```
;   526          1512  1 ROUTINE SERIAL_CACHE : L_JSB NOVALUE =
;   527          1513  1
;   528          1514  1 !++
;   529          1515  1 !
;   530          1516  1 ! FUNCTIONAL DESCRIPTION:
;   531          1517  1 !
;   532          1518  1 ! Serialize cache processing '  queuing the CDRP part of our
;   533          1519  1 ! IO packet onto the AQB queue.  Go to sleep if someone else
;   534          1520  1 ! is already there.
;   535          1521  1 !
;   536          1522  1 !--
;   537          1523  1
;   538          1524  2 BEGIN
;   539          1525  2
;   540          1526  2 BIND_COMMON;
;   541          1527  2
;   542          1528  2 EXTERNAL
;   543          1529  2         PMS$GL_XQPCACHEWAIT     : ADDRESSING_MODE (GENERAL);
;   544          1530  2
;   545          1531  2 BUILTIN
;   546          1532  2         TESTBITSS,
;   547          1533  2         INSQUE;
;   548          1534  2
;   549          1535  2 LOCAL
;   550          1536  2         AQB     : REF BBLOCK,
;   551          1537  2         ACB     : REF BBLOCK;
;   552          1538  2
;   553          1539  2 AQB = .CURRENT_VCB [VCB$L_AQB];
;   554          1540  2
;   555          1541  2 IF (ACB = .ACB_ADDR) EQL 0
;   556          1542  2 THEN
;   557          1543  3     BEGIN
;   558          1544  3     ACB_ADDR = (ACB = .IO_PACKET + IRP$C_CDRP);
;   559          1545  3     ACB [ACB$L_PID] = .CT[$GL_PCB [PCB$L_PID];
;   560          1546  3     ACB [ACB$L_AST] = CONTINUE_THREAD;
;   561          1547  3     ACB [ACB$L_ASTPRM] = .BASE;
;   562          1548  3     ACB [ACB$B_RMOD] = PSL$C_KERNEL + ACB$M_NODELETE;
;   563          1549  3     ACB [ACB$B_TYPE] = DYN$C_ACB;
;   564          1550  3     ACB [ACB$W_SIZE] = 0;
;   565          1551  2     END;
;   566          1552  2
;   567          1553  2 IF INSQUE (.IO_PACKET, .AQB [AQB$L_ACPQBL])
;   568          1554  2 THEN
;   569          1555  2     RETURN
;   570          1556  2 ELSE
;   571          1557  3     BEGIN
;   572          1558  3     PMS$GL_XQPCACHEWAIT = .PMS$GL_XQPCACHEWAIT + 1;
;   573          1559  3     WAIT_FOR_AST ();
;   574          1560  2     END;
;   575          1561  2
;   576          1562  1 END;


                                                .EXTRN  PMS$GL_XQPCACHEWAIT

                                    52  DD 00000 SERIAL_CACHE:
```

```
                                                        PUSHL   R2                                        ; 1512
                   50         98   AA  D0 00002         MOVL    -104(BASE), R0                            ; 1539
                   52         10   A0  D0 00006         MOVL    16(R0), AQB                               ;
                   50         C8   AA  D0 0000A         MOVL    -56(BASE), ACB                            ; 1541
                                   2B  12 0000E         BNEQ    1$                                        ;
     50       90   AA 00000060     8F  C1 00010         ADDL3   #96, -112(BASE), ACB                      ; 1544
              C8   AA              50  D0 00019         MOVL    ACB, -56(BASE)                            ;
                   51 00000000G    00  D0 0001D         MOVL    CTL$GL_PCB, R1                            ; 1545
              0C   A0         60   A1  D0 00024         MOVL    96(R1), 12(ACB)                           ;
              10   A0       0000G  CF  9E 00029         MOVAB   CONTINUE_THREAD, 16(ACB)                  ; 1546
              14   A0              5A  D0 0002F         MOVL    BASE, 20(ACB)                             ; 1547
              08   A0 20020000     8F  D0 00033         MOVL    #537001984, 8(ACB)                        ; 1550
              04   B2         90   BA  0E 0003B 1$:     INSQUE  @-112(BASE), @4(AQB)                      ; 1553
                                   0B  13 00040         BEQL    2$                                        ;
                      00000000G    00  D6 00042         INCL    PMS$GL_XQPCACHEWAIT                       ; 1558
           0000G  CF               00  FB 00048         CALLS   #0, WAIT_FOR_AST                          ; 1559
                                   04  BA 0004D 2$:     POPR    #^M<R2>                                   ; 1562
                                       05 0004F         RSB                                               ;
```

; Routine Size: 80 bytes,   Routine Base: $CODE$ + 0286

```
;  578          1563  1 GLOBAL ROUTINE FIND_BUFFER (LBN, TYPE, COUNT, FOUND_COUNT) : L_NORM =
;  579          1564  1
;  580          1565  1 !++
;  581          1566  1 !
;  582          1567  1 ! FUNCTIONAL DESCRIPTION:
;  583          1568  1 !
;  584          1569  1 !
;  585          1570  1 ! CALLING SEQUENCE:
;  586          1571  1 !
;  587          1572  1 ! INPUT PARAMETERS:
;  588          1573  1 !
;  589          1574  1 ! IMPLICIT INPUTS:
;  590          1575  1 !
;  591          1576  1 ! OUTPUT PARAMETERS:
;  592          1577  1 !
;  593          1578  1 ! IMPLICIT OUTPUTS:
;  594          1579  1 !
;  595          1580  1 ! ROUTINE VALUE:
;  596          1581  1 !       index of first buffer found
;  597          1582  1 !
;  598          1583  1 ! SIDE EFFECTS:
;  599          1584  1 !       LRU list relinked, buffers may be written
;  600          1585  1 !
;  601          1586  1 !--
;  602          1587  1
;  603          1588  2 BEGIN
;  604          1589  2
;  605          1590  2 MAP
;  606          1591  2         TYPE    : BYTE;
;  607          1592  2
;  608          1593  2 BIND_COMMON;
;  609          1594  2
;  610          1595  2 EXTERNAL
;  611          1596  2         ACP$GB_MAXREAD          : BYTE ADDRESSING_MODE (ABSOLUTE),
;  612          1597  2         PMS$GL_FILHDR_HIT       : ADDRESSING_MODE (ABSOLUTE),
;  613          1598  2         PMS$GL_FILHDR_MISS      : ADDRESSING_MODE (ABSOLUTE),
;  614          1599  2         PMS$GL_DIRDATA_HIT      : ADDRESSING_MODE (ABSOLUTE),
;  615          1600  2         PMS$GL_DIRDATA_MISS     : ADDRESSING_MODE (ABSOLUTE),
;  616          1601  2         PMS$GL_STORAGMAP_HIT    : ADDRESSING_MODE (ABSOLUTE),
;  617          1602  2         PMS$GL_STORAGMAP_MISS   : ADDRESSING_MODE (ABSOLUTE);
;  618          1603  2
;  619          1604  2 LOCAL
;  620          1605  2         FNDCNT,
;  621          1606  2         INDX,
;  622          1607  2         POOL,
;  623          1608  2         INDX_ADDR,
;  624          1609  2         LOCKBASIS,
;  625          1610  2         SEQNUM,
;  626          1611  2         PIDINDX         : WORD,
;  627          1612  2         BFRD            : REF BBLOCK,
;  628          1613  2         BFRL            : REF BBLOCK,
;  629          1614  2         POOL_LRU        : REF BBLOCK;
;  630          1615  2
;  631          1616  2 LABEL
;  632          1617  2         GOT_ONE:
;  633          1618  2
;  634          1619  2 .FOUND_COUNT = 1;
```

```
;  635          1620  2
;  636          1621  2 SERIAL_CACHE ();
;  637          1622  2
;  638          1623  2 ! If this is for LBN = -1, we just want a buffer.  Don't look for it,
;  639          1624  2 ! because other processes may be doing this also, and we don't want
;  640          1625  2 ! to find theirs.
;  641          1626  2 !
;  642          1627  2
;  643          1628  2 IF (.LBN+1) EQL 0
;  644          1629  2 THEN
;  645          1630  2     INDX = 0
;  646          1631  2 ELSE
;  647          1632  3     BEGIN
;  648          1633  3
;  649          1634  3 ! Get initial index by hashing.  Follow links, if any, until we
;  650          1635  3 ! get match on both LBN and UCB, or we run out of links to follow.
;  651          1636  3 !
;  652          1637  3
;  653          1638  3     INDX = .LOOKUP_LBN (.LBN)<0,16>;
;  654          1639  3
;  655          1640  3     WHILE .INDX NEQ 0
;  656          1641  3     DO
;  657          1642  4         BEGIN
;  658          1643  4
;  659          1644  4         BFRD = BFRD_ADDR (.INDX);
;  660          1645  4
;  661          1646  4 ! Determine if this in fact matches on both LBN and UCB.
;  662          1647  4 !
;  663          1648  4
;  664          1649  4         IF .BFRD [BFRD$L_LBN] EQL .LBN
;  665          1650  4             AND .BFRD [BFRD$L_UCB] EQL .CURRENT_UCB
;  666          1651  4         THEN
;  667          1652  4             EXITLOOP
;  668          1653  4         ELSE
;  669          1654  4             INDX = .BFRD [BFRD$W_NXTBFRD];
;  670          1655  3         END;
;  671          1656  2     END;
;  672          1657  2
;  673          1658  2 POOL = .POOL_TABLE [.TYPE];
;  674          1659  2
;  675          1660  2 PIDINDX = .CTL$GL_PCB [PCB$L_PID];
;  676          1661  2
;  677          1662  2 ! Validate that we have a lockbasis for the buffer type requested.
;  678          1663  2 ! We deliberately do not include DIRINDX_TYPE buffers here, because
;  679          1664  2 ! they are not hashed at all and are not handled with this routine.
;  680          1665  2 !
;  681          1666  2
;  682          1667  2 CASE .TYPE FROM 0 TO 5 OF
;  683          1668  2     SET
;  684          1669  2     [HEADER_TYPE, DATA_TYPE]:
;  685          1670  3         BEGIN
;  686          1671  3
;  687          1672  3         IF .CURR_LCKINDX EQL 0
;  688          1673  3         THEN
;  689          1674  3             BUG_CHECK (XQPERR, 'no current lock index lock basis');
;  690          1675  3
;  691          1676  3         LOCKBASIS = .LB_BASIS [.CURR_LCKINDX];
```

```
;  692          1677  2          END;
;  693          1678  2
;  694          1679  2      [DIRECTORY_TYPE]:
;  695          1680  3          BEGIN
;  696          1681  3          IF .DIR_LCKINDX EQL 0
;  697          1682  3          THEN
;  698          1683  3              BUG_CHECK (XQPERR, 'No dir lock basis');
;  699          1684  3
;  700          1685  3          LOCKBASIS = .LB_BASIS [.DIR_LCKINDX];
;  701          1686  2          END;
;  702          1687  2
;  703          1688  2      [INDEX_TYPE, BITMAP_TYPE, QUOTA_TYPE]:
;  704          1689  2          IF (LOCKBASIS = .LB_BASIS [0]) EQL 0
;  705          1690  2          THEN
;  706          1691  2              BUG_CHECK (XQPERR, 'no allocation lock for lock basis');
;  707          1692  2
;  708          1693  2      [OUTRANGE]:
;  709          1694  2          BUG_CHECK (BADBUFTYP, 'Bad buffer type code');
;  710          1695  2
;  711          1696  2      TES;
;  712          1697  2
;  713          1698  2  ! Retrieve the lock basis and current buffer sequence number.  The
;  714          1699  2  ! lock basis is the same as used for the synchronization lock, and
;  715          1700  2  ! the sequence number was retrieved from the value block of that lock
;  716          1701  2  ! when it was raised to synchronize this operation.
;  717          1702  2  ! The appropriate lock depends on the type of buffer being requested.
;  718          1703  2  !
;  719          1704  2
;  720          1705  2  CASE .TYPE FROM 0 TO 5 OF
;  721          1706  2      SET
;  722          1707  2      [HEADER_TYPE]:
;  723          1708  3          BEGIN
;  724          1709  3          IF .LB_HDRSEQ [.CURR_LCKINDX] EQL 0
;  725          1710  3          THEN
;  726          1711  3              LB_HDRSEQ [.CURR_LCKINDX] = 1;
;  727          1712  3
;  728          1713  3          SEQNUM = .LB_HDRSEQ [.CURR_LCKINDX];
;  729          1714  2          END;
;  730          1715  2
;  731          1716  2      [DIRECTORY_TYPE]:
;  732          1717  3          BEGIN
;  733          1718  3          IF .LB_DATASEQ [.DIR_LCKINDX] EQL 0
;  734          1719  3          THEN
;  735          1720  3              LB_DATASEQ [.DIR_LCKINDX] = 1;
;  736          1721  3
;  737          1722  3          SEQNUM = .LB_DATASEQ [.DIR_LCKINDX];
;  738          1723  2          END;
;  739          1724  2
;  740          1725  2      [DATA_TYPE]:
;  741          1726  3          BEGIN
;  742          1727  3          IF .LB_DATASEQ [.CURR_LCKINDX] EQL 0
;  743          1728  3          THEN
;  744          1729  3              LB_DATASEQ [.CURR_LCKINDX] = 1;
;  745          1730  3
;  746          1731  3          SEQNUM = .LB_DATASEQ [.CURR_LCKINDX];
;  747          1732  2          END;
;  748          1733  2
```

```
;  749     1734  2 ! The storage bitmap, index file bitmap, and quota file data blocks
;  750     1735  2 ! get their sequence numbers from the volume lock value block.
;  751     1736  2 ! Check with the code in allocation_lock and allocation_unlock before
;  752     1737  2 ! changing any of this.
;  753     1738  2 !
;  754     1739  2
;  755     1740  2     [BITMAP_TYPE]:
;  756     1741  3         BEGIN
;  757     1742  3         IF .(LB_DATASEQ [0])<0,16,0> EQL 0
;  758     1743  3         THEN
;  759     1744  3             (LB_DATASEQ [0])<0,16,0> = 1;
;  760     1745  3
;  761     1746  3         SEQNUM = .(LB_DATASEQ [0])<0,16,0>;
;  762     1747  2         END;
;  763     1748  2
;  764     1749  2     [INDEX_TYPE]:
;  765     1750  3         BEGIN
;  766     1751  3         IF .(LB_DATASEQ [0])<16,16,0> EQL 0
;  767     1752  3         THEN
;  768     1753  3             (LB_DATASEQ [0])<16,16,0> = 1;
;  769     1754  3
;  770     1755  3         SEQNUM = .(LB_DATASEQ [0])<16,16,0>;
;  771     1756  2         END;
;  772     1757  2
;  773     1758  2     [QUOTA_TYPE]:
;  774     1759  3         BEGIN
;  775     1760  3         SEQNUM = .SAVE_VC_FLAGS<1,15,0>;
;  776     1761  3
;  777     1762  3         IF .SEQNUM EQL 0
;  778     1763  3         THEN
;  779     1764  4             BEGIN
;  780     1765  4             SEQNUM = .SEQNUM + 1;
;  781     1766  4             SAVE_VC_FLAGS<1,15,0> = .SEQNUM;
;  782     1767  3             END;
;  783     1768  3
;  784     1769  2         END;
;  785     1770  2
;  786     1771  2     TES;
;  787     1772  2
;  788     1773  2 IF .INDX NEQ 0
;  789     1774  2 THEN
;  790     1775  2
;  791     1776  2 ! We found a buffer that matches the LBN and UCB desired.
;  792     1777  2 ! It may either be on our in-process queue (curpid will be us),
;  793     1778  2 ! or in the general buffer cache.
;  794     1779  2 !
;  795     1780  2
;  796     1781  2 GOT_ONE:
;  797     1782  3     BEGIN
;  798     1783  3
;  799     1784  3     IF .BFRD [BFRD$W_CURPID] NEQ 0
;  800     1785  3     THEN
;  801     1786  3         IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
;  802     1787  3         THEN
;  803     1788  3
;  804     1789  3 ! This is a buffer we've already put onto our in-process queue.
;  805     1790  3 ! Move to the head of the LRU list and return.
```

```
;  806          1791  3
;  807          1792  3
;  808          1793  4              BEGIN
;  809          1794  4              REMQUE (.BFRD, BFRD);
;  810          1795  4              INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
;  811          1796  4              RELEASE_CACHE ();
;  812          1797  4              RETURN .INDX - 1
;  813          1798  4              END
;  814          1799  4
;  815          1800  3          ELSE
;  816          1801  4              bug_check (xqperr,'');
;  817          1802  !           RESOLVE_AMBIGUITY ();
;  818          1803  3
;  819          1804  3
;  820          1805  3      ! Verify lock basis and that things are in the right pool.
;  821          1806  3      !
;  822          1807  3      ! Blocks in the data block pool may legitimately have the wrong
;  823          1808  3      ! lockbasis as a result of having been deallocated to free storage
;  824          1809  3      ! and reallocated to another file.
;  825          1810  3      !
;  826          1811  3      ! File headers may legitimately have the wrong lockbasis if they
;  827          1812  3      ! have been deleted and are being treated as primary headers when
;  828          1813  3      ! they used to be extension headers or vice-versa.  This will
;  829          1814  3      ! force a read from disk.  Further checks will be made in read_header
;  830          1815  3      ! against the actual header to see if things really make sense.
;  831          1816  3      !
;  832          1817  3      ! Blocks can also cross pools (rare in practice) if directory or quota
;  833          1818  3      ! file data blocks, for example, are deallocated to free storage, and
;  834          1819  3      ! the index file is then extended, causing them to become file headers.
;  835          1820  3      !
;  836          1821  3      ! In all cases, unhook and return the bfrd to destroy the association
;  837          1822  3      ! of this buffer with the lock basis and lock that is backing it.
;  838          1823  3      !
;  839          1824  3
;  840          1825  3          IF .BFRD [BFRD$L_LOCKBASIS] NEQ .LOCKBASIS
;  841          1826  3              OR .BFRD [BFRD$V_POOL] NEQ .POOL
;  842          1827  3          THEN
;  843          1828  3              IF .POOL EQL 1
;  844          1829  3                  OR .TYPE EQL HEADER_TYPE
;  845          1830  3                  OR .BFRD [BFRD$V_POOL] EQL 1
;  846          1831  3              THEN
;  847          1832  4                  BEGIN
;  848          1833  4
;  849          1834  4                  UNHOOK_BFRD (
;  850          1835  4                      ((.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF) + 1,
;  851          1836  4                      .BFRD);
;  852          1837  4
;  853          1838  4                  RETURN_BFRD (.BFRD);
;  854          1839  4
;  855          1840  4                  LEAVE GOT_ONE;
;  856          1841  4                  END
;  857          1842  3              ELSE
;  858          1843  3                  BUG_CHECK (XQPERR, 'invalid lock basis');
;  859          1844  3
;  860          1845  3      ! This was not on our in-process queue.  We'll need to account for the
;  861          1846  3      ! fact that we are taking another buffer out of general circulation for
;  862          1847  3      ! this operation.
```

```
;  863          1848  3 !
;  864          1849  3
;  865          1850  3     IF .BFRS_USED [.POOL] EQL .BFR_CREDITS [.POOL]
;  866          1851  3     THEN
;  867          1852  4         BEGIN
;  868          1853  4         BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + .POOL*4;
;  869          1854  4
;  870          1855  4         IF .POOLAVAIL GTR 4
;  871          1856  4         THEN
;  872          1857  5             BEGIN
;  873          1858  5             POOLAVAIL = .POOLAVAIL - 1;
;  874          1859  5             BFR_CREDITS [.POOL] = .BFR_CREDITS [.POOL] + 1;
;  875          1860  5             END
;  876          1861  4         ELSE
;  877          1862  4             FREE_ONE (.POOL);
;  878          1863  3         END;
;  879          1864  3
;  880          1865  3     BFRS_USED [.POOL] = .BFRS_USED [.POOL] + 1;
;  881          1866  3
;  882          1867  3     IF .BFRD [BFRD$L_SEQNUM] NEQ .SEQNUM
;  883          1868  3     THEN
;  884          1869  3         BFRD [BFRD$V_VALID] = 0;
;  885          1870  3
;  886          1871  3 ! The sequence number will be stored in the BFRD when the buffer is
;  887          1872  3 ! released back to the cache, otherwise it would be stored here.
;  888          1873  3 !
;  889          1874  3
;  890          1875  3     BFRD [BFRD$W_CURPID] = .PIDINDX;
;  891          1876  3
;  892          1877  3 ! Pull this buffer out of the cache list and insert into our in-process queue.
;  893          1878  3 !
;  894          1879  3
;  895          1880  3     REMQUE (.BFRD, BFRD);
;  896          1881  3     INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
;  897          1882  3
;  898          1883  3     RELEASE_CACHE ();
;  899          1884  3
;  900          1885  3 ! Count finding a valid buffer as a hit, an invalid one as a miss.
;  901          1886  3 ! We are deliberately not counting hits on buffers already on our
;  902          1887  3 ! in-process list to get a more meaningful hit ratio.
;  903          1888  3 !
;  904          1889  3
;  905          1890  3     IF .BFRD [BFRD$V_VALID]
;  906          1891  3     THEN
;  907          1892  4         BEGIN
;  908          1893  4         PMS_TOT_CACHE = .PMS_TOT_CACHE + 1;
;  909          1894  4         CASE .POOL FROM 0 TO 2 OF
;  910          1895  4         SET
;  911          1896  4         [0]:    PMS$GL_STORAGMAP_HIT = .PMS$GL_STORAGMAP_HIT + 1;
;  912          1897  4         [1]:    PMS$GL_DIRDATA_HIT = .PMS$GL_DIRDATA_HIT + 1;
;  913          1898  4         [2]:    PMS$GL_FILHDR_HIT = .PMS$GL_FILHDR_HIT + 1;
;  914          1899  4         TES;
;  915          1900  4         END
;  916          1901  3     ELSE
;  917          1902  3         CASE .POOL FROM 0 TO 2 OF
;  918          1903  3         SET
;  919          1904  3         [0]:    PMS$GL_STORAGMAP_MISS = .PMS$GL_STORAGMAP_MISS + 1;
```

```
;  920          1905  3          [1]:    PMS$GL_DIRDATA_MISS = .PMS$GL_DIRDATA_MISS + 1;
;  921          1906  3          [2]:    PMS$GL_FILHDR_MISS = .PMS$GL_FILHDR_MISS + 1;
;  922          1907  3          TES;
;  923          1908  3
;  924          1909  3      RETURN .INDX - 1;
;  925          1910  2      END;                            ! of block GOT_ONE
;  926          1911  2
;  927          1912  2
;  928          1913  2  ! Failed to find a buffer matching desired LBN and UCB in the cache.
;  929          1914  2  ! Account for us using another buffer from the cache.
;  930          1915  2  !
;  931          1916  2
;  932          1917  2  FNDCNT = 1;
;  933          1918  2
;  934          1919  2  IF .BFRS_USED [.POOL] EQL .BFR_CREDITS [.POOL]
;  935          1920  2  THEN
;  936          1921  3      BEGIN
;  937          1922  3      BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + .POOL*4;
;  938          1923  3
;  939          1924  3      IF .POOLAVAIL GTR 4
;  940          1925  3      THEN
;  941          1926  4          BEGIN
;  942          1927  4          POOLAVAIL = .POOLAVAIL - 1;
;  943          1928  4          BFR_CREDITS [.POOL] = .BFR_CREDITS [.POOL] + 1;
;  944          1929  4          END
;  945          1930  3      ELSE
;  946          1931  3          FREE_ONE (.POOL);
;  947          1932  2      END;
;  948          1933  2
;  949          1934  2  POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
;  950          1935  2
;  951          1936  2  REMQUE (.POOL_LRU [QFLNK], BFRD);
;  952          1937  2
;  953          1938  2  IF .COUNT GTRU 1
;  954          1939  2      AND .POOL EQL 1
;  955          1940  2  THEN
;  956          1941  3      BEGIN
;  957          1942  3      LABEL CHK_BFRDS;
;  958          1943  3
;  959          1944  3      BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + 4;          ! Pool 1
;  960          1945  3      BIND POOLCNT = CACHE_HDR [F11BC$W_POOLCNT] : VECTOR [,WORD];
;  961          1946  3
;  962          1947  3      LOCAL
;  963          1948  3          DOWN,
;  964          1949  3          TRY_COUNT,
;  965          1950  3          LO_BFRD,
;  966          1951  3          HI_BFRD,
;  967          1952  3          CUR_BFRD : REF BBLOCK;
;  968          1953  3
;  969          1954  3      TRY_COUNT = .ACP$GB_MAXREAD;
;  970          1955  3
;  971          1956  3      IF .COUNT LSSU .TRY_COUNT
;  972          1957  3      THEN
;  973          1958  3          TRY_COUNT = .COUNT;
;  974          1959  3
;  975          1960  3      LO_BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + (.POOLCNT [0])*BFRD$S_BFRDDEF;
;  976          1961  3      HI_BFRD = .LO_BFRD + (.POOLCNT [1])*BFRD$S_BFRDDEF;
```

```
;  977          1962  3
;  978          1963  3      DOWN = 0;
;  979          1964  3      CUR_BFRD = .BFRD;
;  980          1965  3
;  981          1966  3      WHILE .FNDCNT LSSU .TRY_COUNT
;  982          1967  3      DO
;  983          1968  3          CHK_BFRDS:
;  984          1969  4          BEGIN
;  985          1970  4
;  986          1971  4          IF NOT .DOWN
;  987          1972  4          THEN
;  988          1973  5              BEGIN
;  989          1974  5
;  990          1975  5              CUR_BFRD = .CUR_BFRD + BFRD$S_BFRDDEF;
;  991          1976  5
;  992          1977  5              IF .CUR_BFRD GEQA .HI_BFRD
;  993          1978  5              THEN
;  994          1979  6                  BEGIN
;  995          1980  6                  CUR_BFRD = .BFRD;
;  996          1981  6                  DOWN = 1;
;  997          1982  6                  LEAVE CHK_BFRDS;
;  998          1983  5                  END;
;  999          1984  5
; 1000          1985  5              END
; 1001          1986  4          ELSE
; 1002          1987  5              BEGIN
; 1003          1988  5
; 1004          1989  5              CUR_BFRD = .CUR_BFRD - BFRD$S_BFRDDEF;
; 1005          1990  5
; 1006          1991  5              IF .CUR_BFRD LSSA .LO_BFRD
; 1007          1992  5              THEN
; 1008          1993  5                  EXITLOOP;
; 1009          1994  5
; 1010          1995  4              END,
; 1011          1996  4
; 1012          1997  4          IF .CUR_BFRD [BFRD$W_CURPID] NEQ 0
; 1013          1998  4          THEN
; 1014          1999  4              EXITLOOP;
; 1015          2000  4
; 1016          2001  4 ! NOTE: FNDCNT has not been bumped yet.
; 1017          2002  4
; 1018          2003  4          INDX = .LOOKUP_LBN ((.LBN + .FNDCNT))<0,16>;
; 1019          2004  4
; 1020          2005  4          WHILE .INDX NEQ 0
; 1021          2006  4          DO
; 1022          2007  5              BEGIN
; 1023          2008  5
; 1024          2009  5              LOCAL TMPBFRD : REF BBLOCK;
; 1025          2010  5
; 1026          2011  5              TMPBFRD = BFRD_ADDR (.INDX);
; 1027          2012  5
; 1028          2013  6              IF .TMPBFRD [BFRD$L_LBN] EQL (.LBN + .FNDCNT)
; 1029          2014  5                  AND .TMPBFRD [BFRD$L_UCB] EQL .CURRENT_UCB
; 1030          2015  5              THEN
; 1031          2016  5                  EXITLOOP
; 1032          2017  5              ELSE
; 1033          2018  5                  INDX = .TMPBFRD [BFRD$W_NXTBFRD];
```

```
;  1034          2019  4              END;
;  1035          2020  4
;  1036          2021  4          IF .INDX NEQ 0
;  1037          2022  4          THEN
;  1038          2023  4              EXITLOOP;
;  1039          2024  4
;  1040          2025  4          IF .POOLAVAIL GTR 4
;  1041          2026  4          THEN
;  1042          2027  5              BEGIN
;  1043          2028  5              POOLAVAIL = .POOLAVAIL - 1;
;  1044          2029  5              BFR_CREDITS [1] = .BFR_CREDITS [1] + 1;
;  1045          2030  5              END
;  1046          2031  4          ELSE
;  1047          2032  4              EXITLOOP;
;  1048          2033  4
;  1049          2034  4      ! This one is ok.  Yank off LRU and count it.
;  1050          2035  4      !
;  1051          2036  4
;  1052          2037  4          REMQUE (.CUR_BFRD, CUR_BFRD);
;  1053          2038  4          FNDCNT = .FNDCNT + 1;
;  1054          2039  4
;  1055          2040  4          IF .CUR_BFRD LSSA .BFRD
;  1056          2041  4          THEN
;  1057          2042  4              BFRD = .CUR_BFRD;
;  1058          2043  4
;  1059          2044  3          END;              ! of block CHK_BFRDS
;  1060          2045  3
;  1061          2046  3      .FOUND_COUNT = .FNDCNT;
;  1062          2047  3
;  1063          2048  3      ! Set BFRD to the highest one in the range found.
;  1064          2049  3      !
;  1065          2050  3
;  1066          2051  3      BFRD = .BFRD + (.FNDCNT - 1)*BFRD$S_BFRDDEF;
;  1067          2052  3
;  1068          2053  2      END;          ! of consider multi-block read
;  1069          2054  2
;  1070          2055  2  BFRS_USED [.POOL] = .BFRS_USED [.POOL] + .FNDCNT;
;  1071          2056  2
;  1072          2057  2  DECR I FROM (.FNDCNT - 1) TO 0
;  1073          2058  2  DO
;  1074          2059  3      BEGIN
;  1075          2060  3
;  1076          2061  3      INDX = ((.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF) + 1;
;  1077          2062  3
;  1078          2063  3      UNHOOK_BFRD (.INDX, .BFRD);
;  1079          2064  3
;  1080          2065  3      ! Insert the new buffer into the hash list using the new LBN.
;  1081          2066  3      !
;  1082          2067  3
;  1083          2068  3      INDX_ADDR = LOOKUP_LBN ((.LBN + .I));
;  1084          2069  3      BFRD [BFRD$W_NXTBFRD] = .(.INDX_ADDR)<0,16>;
;  1085          2070  3      (.INDX_ADDR)<0,16> = .INDX;
;  1086          2071  3
;  1087          2072  3      ! Fill in our new or recycled BFRD, as the case may be, and
;  1088          2073  3      ! insert it onto the appropriate in-process queue.
;  1089          2074  3      !
;  1090          2075  3
```

```
;  1091          2076  3      BFRD [BFRD$L_LBN] = .LBN + .I;
;  1092          2077  3      BFRD [BFRD$L_UCB] = .CURRENT_UCB;
;  1093          2078  3      BFRD [BFRD$W_CURPID] = .PIDINDX;
;  1094          2079  3
;  1095          2080  3      BFRD [BFRD$L_LOCKBASIS] = .LOCKBASIS;
;  1096          2081  3      BFRD [BFRD$L_SEQNUM] = .SEQNUM;
;  1097          2082  3      BFRD [BFRD$B_BTYPE] = .TYPE;
;  1098          2083  3
;  1099          2084  3      INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
;  1100          2085  3
;  1101          2086  3      BFRD = .BFRD - BFRD$S_BFRDDEF;
;  1102          2087  3
;  1103          2088  2      END;
;  1104          2089  2
;  1105          2090  2 RELEASE_CACHE ();
;  1106          2091  2
;  1107          2092  2 ! We didn't find the buffer in the cache so count a miss.
;  1108          2093  2 !
;  1109          2094  2
;  1110          2095  2 CASE .POOL FROM 0 TO 2 OF
;  1111          2096  2 SET
;  1112          2097  2 [0]:    PMS$GL_STORAGMAP_MISS = .PMS$GL_STORAGMAP_MISS + 1;
;  1113          2098  2 [1]:    PMS$GL_DIRDATA_MISS = .PMS$GL_DIRDATA_MISS + 1;
;  1114          2099  2 [2]:    PMS$GL_FILHDR_MISS = .PMS$GL_FILHDR_MISS + 1;
;  1115          2100  2 TES;
;  1116          2101  2
;  1117          2102  2 RETURN .INDX - 1;
;  1118          2103  2
;  1119          2104  1 END;


                                        .EXTRN  ACP$GB_MAXREAD, PMS$GL_FILHDR_HIT
                                        .EXTRN  PMS$GL_FILHDR_MISS
                                        .EXTRN  PMS$GL_DIRDATA_HIT
                                        .EXTRN  PMS$GL_DIRDATA_MISS
                                        .EXTRN  PMS$GL_STORAGMAP_HIT
                                        .EXTRN  PMS$GL_STORAGMAP_MISS
                                        .EXTRN  BUG$_BADBUFTYP

                          0BFC 00000    .ENTRY  FIND_BUFFER, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1563
                                                R11                                         ;
                    5E          1C  C2 00002    SUBL2   #28, SP                             ;
                    56      EC  AA  9E 00005    MOVAB   -20(BASE), R6                       ; 1591
                    58      FC  AA  9E 00009    MOVAB   -4(BASE), R8                        ;
                    55      14  AA  9E 0000D    MOVAB   20(BASE), R5                        ;
                    57    0080  CA  9E 00011    MOVAB   128(BASE), R7                       ;
                    52    00A8  CA  9E 00016    MOVAB   168(BASE), R2                       ;
                    59    00D4  CA  9E 0001B    MOVAB   212(BASE), R9                       ;
              10    BC          01  D0 00020    MOVL    #1, @FOUND_COUNT                    ; 1619
                                8A  10 00024    BSBB    SERIAL_CACHE                        ; 1621
          50  04    AC          01  C1 00026    ADDL3   #1, LBN, R0                         ; 1628
                                04  12 0002B    BNEQ    1$                                  ;
                                5B  D4 0002D    CLRL    INDX                                ; 1630
                                43  11 0002F    BRB     5$                                  ;
                    50          68  D0 00031 1$: MOVL    (R8), R0                        ; 1638
                    51      14  A0  3C 00034    MOVZWL  20(R0), R1                          ;
```

```
      7E        00      04  AC        01  7A 00038          EMUL    #1, LBN, #0, -(SP)              :
      51        51          8E        51  7B 0003E          EDIV    R1, (SP)+, R1, R1               :
                                      51  D5 00043          TSTL    R1                              :
                                      03  18 00045          BGEQ    2$                              :
                            51        51  CE 00047          MNEGL   R1, R1                          :
                            5B   10 B041  3C 0004A 2$:      MOVZWL  @16(R0)[R1], INDX               :
                                      23  13 0004F 3$:      BEQL    5$                              : 1640
                            51        68  D0 00051          MOVL    (R8), R1                        : 1644
                50          5B        05  78 00054          ASHL    #5, INDX, R0                    :
                            50   18   A1  C0 00058          ADDL2   24(R1), R0                      :
                            54   E0   A0  9E 0005C          MOVAB   -32(R0), BFRD                   :
                    04      AC   08   A4  D1 00060          CMPL    8(BFRD), LBN                    : 1649
                                      07  12 00065          BNEQ    4$                              :
                    94      AA   0C   A4  D1 00067          CMPL    12(BFRD), -108(BASE)            : 1650
                                      06  13 0006C          BEQL    5$                              :
                            5B   1E   A4  3C 0006E 4$:      MOVZWL  30(BFRD), INDX                  : 1654
                                      DB  11 00072          BRB     3$                              : 1640
                            50   08   AC  9A 00074 5$:      MOVZBL  TYPE, R0                        : 1658
                            53 FCAD CF40  9A 00078          MOVZBL  POOL_TABLE[R0], POOL            :
                            51 00000000G  00  D0 0007E      MOVL    CTL$GL_PCB, R1                  : 1660
                    18      AE   60   A1  B0 00085          MOVW    96(R1), PIDINDX                 :
                05          00        50  8F 0008A          CASEB   R0, #0, #5                      : 1667
    0031      001F        0031      0012     0008E 6$:      .WORD   7$-6$,-                         :
                          0031      0012     00096                  12$-6$,-                        :
                                                                    9$-6$,-                         :
                                                                    12$-6$,-                        :
                                                                    7$-6$,-                         :
                                                                    12$-6$                          :
                                    FEFF 0009A              BUGW                                    : 1694
                                   0000* 0009C              .WORD   <BUG$_BADBUFTYP!4>              :
                                  29  11 0009E              BRB     13$                             :
                                  65  D5 000A0 7$:          TSTL    (R5)                            : 1672
                                  04  12 000A2              BNEQ    8$                              :
                                    FEFF 000A4              BUGW                                    : 1674
                                   0000* 000A6              .WORD   <BUG$_XQPERR!4>                 :
                        50        65  D0 000A8 8$:          MOVL    (R5), R0                        : 1676
                                  0B  11 000AB              BRB     11$                             :
                                  69  D5 000AD 9$:          TSTL    (R9)                            : 1681
                                  04  12 000AF              BNEQ    10$                             :
                                    FEFF 000B1              BUGW                                    : 1683
                                   0000* 000B3              .WORD   <BUG$_XQPERR!4>                 :
                        50        69  D0 000B5 10$:         MOVL    (R9), R0                        : 1685
                    10  AE      6740  D0 000B8 11$:         MOVL    (R7)[R0], LOCKBASIS             :
                                  0A  11 000BD              BRB     13$                             : 1667
                    10  AE        67  D0 000BF 12$:         MOVL    (R7), LOCKBASIS                 : 1689
                                  04  12 000C3              BNEQ    13$                             :
                                    FEFF 000C5              BUGW                                    : 1691
                                   0000* 000C7              .WORD   <BUG$_XQPERR!4>                 :
                05          00   08   AC  8F 000C9 13$:     CASEB   TYPE, #0, #5                    : 1705
    0059      0027        004D      000C     000CE 14$:     .WORD   15$-14$,-                       :
                          0068      0038     000D6                  22$-14$,-                       :
                                                                    17$-14$,-                       :
                                                                    24$-14$,-                       :
                                                                    19$-14$,-                       :
                                                                    26$-14$                         :
                        50        65  D0 000DA 15$:         MOVL    (R5), R0                        : 1709
                           0094 CA40  D5 000DD              TSTL    148(BASE)[R0]                   :
```

```
                                        06  12 000E2          BNEQ    16$                                       ;
                      0094 CA40         01  D0 000E4          MOVL    #1, 148(BASE)[R0]                         ; 1711
                                50      65  D0 000EA  16$:    MOVL    (R5), R0                                  ; 1713
                                57  0094 CA40  D0 000ED       MOVL    148(BASE)[R0], SEQNUM                     ;
                                        51  11 000F3          BRB     27$                                       ; 1705
                                50      69  D0 000F5  17$:    MOVL    (R9), R0                                  ; 1718
                                      6240  D5 000F8          TSTL    (R2)[R0]                                  ;
                                        04  12 000FB          BNEQ    18$                                       ;
                              6240      01  D0 000FD          MOVL    #1, (R2)[R0]                              ; 1720
                                50      69  D0 00101  18$:    MOVL    (R9), R0                                  ; 1722
                                        0F  11 00104          BRB     21$                                       ;
                                50      65  D0 00106  19$:    MOVL    (R5), R0                                  ; 1727
                                      6240  D5 00109          TSTL    (R2)[R0]                                  ;
                                        04  12 0010C          BNEQ    20$                                       ;
                              6240      01  D0 0010E          MOVL    #1, (R2)[R0]                              ; 1729
                                50      65  D0 00112  20$:    MOVL    (R5), R0                                  ; 1731
                                57    6240  D0 00115  21$:    MOVL    (R2)[R0], SEQNUM                          ;
                                        2B  11 00119          BRB     27$                                       ; 1705
                                        62  B5 0011B  22$:    TSTW    (R2)                                      ; 1742
                                        03  12 0011D          BNEQ    23$                                       ;
                                62      01  B0 0011F          MOVW    #1, (R2)                                  ; 1744
                                57      62  3C 00122  23$:    MOVZWL  (R2), SEQNUM                              ; 1746
                                        1F  11 00125          BRB     27$                                       ; 1705
                                    02  A2  B5 00127  24$:    TSTW    2(R2)                                     ; 1751
                                        04  12 0012A          BNEQ    25$                                       ;
                           02   A2      01  B0 0012C          MOVW    #1, 2(R2)                                 ; 1753
                                57  02  A2  3C 00130  25$:    MOVZWL  2(R2), SEQNUM                             ; 1755
                                        10  11 00134          BRB     27$                                       ; 1705
      57        A4   AA         0F      01  EF 00136  26$:    EXTZV   #1, #15, -92(BASE), SEQNUM                ; 1760
                                        08  12 0013C          BNEQ    27$                                       ; 1762
                                        57  D6 0013E          INCL    SEQNUM                                    ; 1765
  A4   AA         0F            01      57  F0 00140          INSV    SEQNUM, #1, #15, -92(BASE)                ; 1766
                                        5B  D5 00146  27$:    TSTL    INDX                                      ; 1773
                                        5E  13 00148          BEQL    32$                                       ;
                                    1C  A4  B5 0014A          TSTW    28(BFRD)                                  ; 1784
                                        1D  13 0014D          BEQL    29$                                       ;
                           18   AE  1C  A4  B1 0014F          CMPW    28(BFRD), PIDINDX                         ; 1786
                                        12  12 00154          BNEQ    28$                                       ;
                                54      64  0F 00156          REMQUE  (BFRD), BFRD                              ; 1794
                                50  D0 AA43  7E 00159         MOVAQ   -48(BASE)[POOL], R0                       ; 1795
                           00   B0      64  0E 0015E          INSQUE  (BFRD), @0(R0)                            ;
                                     0000V  30 00162          BSBW    RELEASE_CACHE                             ; 1796
                                      0282  31 00165          BRW     69$                                       ; 1797
                                        FEFF 00168    28$:    BUGW                                              ; 1801
                                       0000* 0016A            .WORD   <BUG$_XQPERR!4>                           ;
                           10   AE  10  A4  D1 0016C  29$:    CMPL    16(BFRD), LOCKBASIS                       ; 1825
                                        08  12 00171          BNEQ    30$                                       ;
      53        18   A4         02      00  ED 00173          CMPZV   #0, #2, 24(BFRD), POOL                    ; 1826
                                        34  13 00179          BEQL    34$                                       ;
                                01      53  D1 0017B  30$:    CMPL    POOL, #1                                  ; 1828
                                        0D  13 0017E          BEQL    31$                                       ;
                                    08  AC  95 00180          TSTB    TYPE                                      ; 1829
                                        08  13 00183          BEQL    31$                                       ;
      01        18   A4         02      00  ED 00185          CMPZV   #0, #2, 24(BFRD), #1                      ; 1830
                                        1E  12 0018B          BNEQ    33$                                       ;
                                        54  DD 0018D  31$:    PUSHL   BFRD                                      ; 1836
                                50      68  D0 0018F          MOVL    (R8), R0                                  ; 1835
```

```
                50              54        18    A0  C3 00192          SUBL3   24(R0), BFRD, R0                        ;
                                50              20  C6 00197          DIVL2   #32, R0                             ;
                                          01    A0  9F 0019A          PUSHAB  1(R0)                               ;
                        0000V   CF              02  FB 0019D          CALLS   #2, UNHOOK_BFRD                     ;
                                50              54  D0 001A2          MOVL    BFRD, R0                            ; 1838
                                              0000V 30 001A5          BSBW    RETURN_BFRD                         ;
                                              0084  31 001A8 32$:     BRW     45$                                 ; 1840
                                                FEFF 001AB 33$:       BUGW                                        ; 1843
                                               0000* 001AD            .WORD   <BUG$_XQPERR!4>                     ;
                                    6643        F4 AA43  B1 001AF 34$:  CMPW    -12(BASE)[POOL], (R6)[POOL]         ; 1850
                                                  1D  12 001B5          BNEQ    36$                                 ;
                                50              00 B843  DE 001B7       MOVAL   @0(R8)[POOL], R0                    ; 1853
                                50              68    A0  9E 001BC      MOVAB   104(R0), R0                         ;
                                04              60    D1 001C0          CMPL    (R0), #4                            ; 1855
                                                07    15 001C3          BLEQ    35$                                 ;
                                                60    D7 001C5          DECL    (R0)                                ; 1858
                                              6643    B6 001C7          INCW    (R6)[POOL]                          ; 1859
                                                08    11 001CA          BRB     36$                                 ; 1855
                                                53    DD 001CC 35$:     PUSHL   POOL                                ; 1862
                                              0000V   30 001CE          BSBW    FREE_ONE                            ;
                                5E              04    C0 001D1          ADDL2   #4, SP                              ;
                                            F4 AA43   B6 001D4 36$:     INCW    -12(BASE)[POOL]                     ; 1865
                                57              14    A4  D1 001D8      CMPL    20(BFRD), SEQNUM                    ; 1867
                                                04    13 001DC          BEQL    37$                                 ;
                            18    A4                08    8A 001DE      BICB2   #8, 24(BFRD)                        ; 1869
                            1C    A4          18    AE    B0 001E2 37$:   MOVW    PIDINDX, 28(BFRD)                   ; 1875
                                54                64    0F 001E7        REMQUE  (BFRD), BFRD                        ; 1880
                                50              D0 AA43   7E 001EA        MOVAQ   -48(BASE)[POOL], R0                 ; 1881
                            00  B0                64    0E 001EF        INSQUE  (BFRD), @0(R0)                      ;
                                              0000V   30 001F3          BSBW    RELEASE_CACHE                       ; 1883
                    27      18    A4                03    E1 001F6      BBC     #3, 24(BFRD), 43$                   ; 1890
                                        08F0        CA    D6 001FB      INCL    2288(BASE)                          ; 1893
                    02              00              53    CF 001FF      CASEL   POOL, #0, #2                        ; 1894
                  0016            000E                  0006  00203 38$:  .WORD   39$-38$,-                        ;
                                                                                40$-38$,-                           ;
                                                                                41$-38$                             ;
                                    00000000G   9F    D6 00209 39$:     INCL    @#PMS$GL_STORAGMAP_HIT              ; 1896
                                                0E    11 0020F          BRB     42$                                 ;
                                    00000000G   9F    D6 00211 40$:     INCL    @#PMS$GL_DIRDATA_HIT                ; 1897
                                                06    11 00217          BRB     42$                                 ;
                                    00000000G   9F    D6 00219 41$:     INCL    @#PMS$GL_FILHDR_HIT                 ; 1898
                                              01C8    31 0021F 42$:     BRW     69$                                 ; 1890
                    02              00              53    CF 00222 43$:     CASEL   POOL, #0, #2                      ; 1902
                  01BE            01B6            01AE    00226 44$:     .WORD   66$-44$,-                         ;
                                                                                67$-44$,-                           ;
                                                                                68$-44$                             ;
                                              01A5    31 0022C          BRW     66$                                 ; 1904
                                52              01    D0 0022F 45$:     MOVL    #1, FNDCNT                          ; 1917
                                              6643    F4 AA43  B1 00232       CMPW    -12(BASE)[POOL], (R6)[POOL]         ; 1919
                                                  1D  12 00238          BNEQ    47$                                 ;
                                50              00 B843  DE 0023A       MOVAL   @0(R8)[POOL], R0                    ; 1922
                                50              68    A0  9E 0023F      MOVAB   104(R0), R0                         ;
                                04              60    D1 00243          CMPL    (R0), #4                            ; 1924
                                                07    15 00246          BLEQ    46$                                 ;
                                                60    D7 00248          DECL    (R0)                                ; 1927
                                              6643    B6 0024A          INCW    (R6)[POOL]                          ; 1928
                                                08    11 0024D          BRB     47$                                 ; 1924
```

```
                        53   DD 0024F 46$:    PUSHL   POOL                                    ; 1931
                   0000V 30   00251           BSBW    FREE_ONE
                5E      04   C0 00254           ADDL2   #4, SP
                50   00 B843 7E 00257 47$:    MOVAQ   @0(R8)[POOL], POOL_LRU                  ; 1934
                50      28   C0 0025C           ADDL2   #40, POOL_LRU
                54   00   B0 0F 0025F           REMQUE  @0(POOL_LRU), BFRD                      ; 1936
                01   0C   AC D1 00263           CMPL    COUNT, #1                               ; 1938
                        03   1A 00267           BGTRU   49$
                   00E6 31   00269 48$:          BRW     61$
                01      53   D1 0026C 49$:      CMPL    POOL, #1                                ; 1939
                        F8   12 0026F           BNEQ    48$
             59      68 0000006C   8F C1 00271   ADDL3   #108, (R8), R9                        ; 1944
             51      68 00000078   8F C1 00279   ADDL3   #120, (R8), R1                        ; 1945
             6E 00000000G   9F 9A 00281         MOVZBL  @#ACP$GB_MAXREAD, TRY_COUNT             ; 1954
             6E      0C   AC D1 00288           CMPL    COUNT, TRY_COUNT                        ; 1956
                     04   1E 0028C           BGEQU   50$
             6E      0C   AC D0 0028E           MOVL    COUNT, TRY_COUNT                        ; 1958
             50      68   D0 00292 50$:        MOVL    (R8), R0                                ; 1960
             55      61   3C 00295           MOVZWL  (R1), R5
             55      20   C4 00298           MULL2   #32, R5
          08 AE   18 B045 9E 0029B           MOVAB   @24(R0)[R5], LO_BFRD
             51   02   A1 3C 002A1           MOVZWL  2(R1), R1                               ; 1961
             51      20   C4 002A5           MULL2   #32, R1
          14 AE   08 BE41 9E 002A8           MOVAB   @LO_BFRD[R1], HI_BFRD
                0C   AE D4 002AE           CLRL    DOWN                                    ; 1963
             55      54   D0 002B1           MOVL    BFRD, CUR_BFRD                          ; 1964
             6E      52   D1 002B4 51$:      CMPL    FNDCNT, TRY_COUNT                       ; 1966
                     03   1F 002B7           BLSSU   52$
                   0089 31   002B9           BRW     60$
             12   0C   AE E8 002BC 52$:      BLBS    DOWN, 54$                               ; 1971
             55      20   C0 002C0           ADDL2   #32, CUR_BFRD                           ; 1975
          14 AE      55   D1 002C3           CMPL    CUR_BFRD, HI_BFRD                       ; 1977
                     12   1F 002C7           BLSSU   55$
             55      54   D0 002C9           MOVL    BFRD, CUR_BFRD                          ; 1980
          0C AE      01   D0 002CC           MOVL    #1, DOWN                                ; 1981
                     E2   11 002D0 53$:      BRB     51$                                     ; 1982
             55      20   C2 002D2 54$:      SUBL2   #32, CUR_BFRD                           ; 1989
          08 AE      55   D1 002D5           CMPL    CUR_BFRD, LO_BFRD                       ; 1991
                     6A   1F 002D9           BLSSU   60$
                1C   A5   B5 002DB 55$:      TSTW    28(CUR_BFRD)                            ; 1997
                     65   12 002DE           BNEQ    60$
             50      68   D0 002E0           MOVL    (R8), R0                                ; 2003
          04 AE   04 BC42 9E 002E3           MOVAB   @LBN[FNDCNT], 4(SP)
             51   14   A0 3C 002E9           MOVZWL  20(R0), R1
7E      00   04 AE      01   7A 002ED         EMUL    #1, 4(SP), #0, -(SP)
51      51      8E      51   7B 002F3         EDIV    R1, (SP)+, R1, R1
                        51   D5 002F8         TSTL    R1
                        03   18 002FA         BGEQ    56$
                51      51   CE 002FC         MNEGL   R1, R1
                5B   10 B041 3C 002FF 56$:    MOVZWL  @16(R0)[R1], INDX
                        22   13 00304 57$:    BEQL    59$                                     ; 2005
                51      68   D0 00306         MOVL    (R8), R1                                ; 2011
             50         5B   05 78 00309         ASHL    #5, INDX, R0
                50   18   A1 C0 0030D         ADDL2   24(R1), R0
                50      20   C2 00311         SUBL2   #32, TMPBFRD
                04 AE   08   A0 D1 00314         CMPL    8(TMPBFRD), 4(SP)                       ; 2013
                        07   12 00319         BNEQ    58$
```

```
                        94    AA        0C    A0  D1 0031B           CMPL    12(TMPBFRD), -108(BASE)                         ; 2014
                                        06    13  00320             BEQL    59$
                        5B              1E    A0  3C 00322 58$:      MOVZWL  30(TMPBFRD), INDX                               ; 2018
                                              DC  11 00326           BRB     57$                                             ; 2005
                                              5B  D5 00328 59$:      TSTL    INDX                                            ; 2021
                                              19  12 0032A           BNEQ    60$
                        04                    69  D1 0032C           CMPL    (R9), #4                                        ; 2025
                                              14  15 0032F           BLEQ    60$
                                              69  D7 00331           DECL    (R9)                                            ; 2028
                                        02    A6  B6 00333           INCW    2(R6)                                           ; 2029
                        55                    65  0F 00336           REMQUE  (CUR_BFRD), CUR_BFRD                            ; 2037
                                              52  D6 00339           INCL    FNDCNT                                          ; 2038
                        54                    55  D1 0033B           CMPL    CUR_BFRD, BFRD                                  ; 2040
                                              90  1E 0033E           BGEQU   53$
                        54                    55  D0 00340           MOVL    CUR_BFRD, BFRD                                  ; 2042
                                              8B  11 00343           BRB     53$                                             ; 1966
                        10    BC              52  D0 00345 60$:      MOVL    FNDCNT, @FOUND_COUNT                            ; 2046
                50              52            05  78 00349           ASHL    #5, FNDCNT, R0                                  ; 2051
                                54        E0  A044  9E 0034D         MOVAB   -32(R0)[BFRD], BFRD
                        F4 AA43                 52  A0 00352 61$:      ADDW2   FNDCNT, -12(BASE)[POOL]                         ; 2055
                                                6B  11 00357           BRB     64$                                             ; 2057
                        50                      68  D0 00359 62$:      MOVL    (R8), R0                                        ; 2061
                50              54        18    A0  C3 0035C         SUBL3   24(R0), BFRD, R0
                                50              20  C6 00361         DIVL2   #32, R0
                        5B              01      A0  9E 00364         MOVAB   1(R0), INDX
                                                54  DD 00368         PUSHL   BFRD                                            ; 2063
                                                5B  DD 0036A         PUSHL   INDX
                      0000V     CF              02  FB 0036C         CALLS   #2, UNHOOK_BFRD
                        51                      68  D0 00371         MOVL    (R8), R1                                        ; 2068
                50              52        04    AC  C1 00374         ADDL3   LBN, I, R0
                        55              14      A1  3C 00379         MOVZWL  20(R1), R5
      7E        00      50                      01  7A 0037D         EMUL    #1, R0, #0, -(SP)
      50        50      8E                      55  7B 00382         EDIV    R5, (SP)+, R0, R0
                                                50  D5 00387         TSTL    R0
                                                03  18 00389         BGEQ    63$
                        50                      50  CE 0038B         MNEGL   R0, R0
                        56              10  B140  3E 0038E 63$:      MOVAW   @16(R1)[R0], INDX_ADDR
                  1E    A4                      66  B0 00393         MOVW    (INDX_ADDR), 30(BFRD)                           ; 2069
                        66                      5B  B0 00397         MOVW    INDX, (INDX_ADDR)                               ; 2070
                  08    A4              04  BC42  9E 0039A         MOVAB   @LBN[I], 8(BFRD)                                ; 2076
                  0C    A4              94    AA  D0 003A0         MOVL    -108(BASE), 12(BFRD)                            ; 2077
                  1C    A4              18    AE  B0 003A5         MOVW    PIPINDX, 28(BFRD)                               ; 2078
                  10    A4              10    AE  D0 003AA         MOVL    LOCKBASIS, 16(BFRD)                             ; 2080
                  14    A4                    57  D0 003AF         MOVL    SEQNUM, 20(BFRD)                                ; 2081
                  19    A4              08    AC  90 003B3         MOVB    TYPE, 25(BFRD)                                  ; 2082
                        50              D0  AA43  7E 003B8         MOVAQ   -48(BASE)[POOL], R0                             ; 2084
                  00    B0                      64  0E 003BD         INSQUE  (BFRD), @0(R0)
                        54                      20  C2 003C1         SUBL2   #32, BFRD                                       ; 2086
                        92                      52  F4 003C4 64$:      SOBGEQ  I, 62$                                          ; 2057
                                          0000V 30 003C7             BSBW    RELEASE_CACHE                                   ; 2090
                02              00              53  CF 003CA         CASEL   POOL, #0, #2                                    ; 2095
              0016            000E              0006    003CE 65$:     .WORD   66$-65$,-
                                                                               67$-65$,-
                                                                               68$-65$
                                  00000000G     9F  D6 003D4 66$:     INCL    @#PMS$GL_STORAGMAP_MISS                         ; 2097
                                                0E  11 003DA         BRB     69$
                                  00000000G     9F  D6 003DC 67$:     INCL    @#PMS$GL_DIRDATA_MISS                           ; 2098
```

```
                        06  11 00E2          BRB     69$                                     ;
              00000000G 9F  D6 00E4 68$:      INCL    @#PMS$GL_FILHDR_MISS                    ; 2099
          50        FF  AB  9E 00EA 69$:      MOVAB   -1(R11),R0                              ; 2102
                        04 00EE               RET                                             ; 2104

; Routine Size:  1007 bytes,    Routine Base:  $CODE$ + 02D6
```

```
;  1121          2105  1 ROUTINE WAKE_WAITER (AQB) : NOVALUE =
;  1122          2106  1
;  1123          2107  1 !++
;  1124          2108  1 !
;  1125          2109  1 ! FUNCTIONAL DESCRIPTION:
;  1126          2110  1 !
;  1127          2111  1 ! Remove ourself from cache processing queue.  Wake up
;  1128          2112  1 ! the next process if queue is not empty.
;  1129          2113  1 !
;  1130          2114  1 !--
;  1131          2115  1
;  1132          2116  2 BEGIN
;  1133          2117  2
;  1134          2118  2 MAP
;  1135          2119  2         AQB     : REF BBLOCK;
;  1136          2120  2
;  1137          2121  2 LINKAGE
;  1138          2122  2         L_SCH$QAST = JSB : GLOBAL (PINCL=2, ACB=5) NOPRESERVE (3,4)
;  1139          2123  2                           NOTUSED (6,7,8,9,10,11);
;  1140          2124  2
;  1141          2125  2 GLOBAL REGISTER
;  1142          2126  2         PINCL = 2,
;  1143          2127  2         ACB = 5;
;  1144          2128  2
;  1145          2129  2 EXTERNAL ROUTINE
;  1146          2130  2         SCH$QAST        : L_SCH$QAST ADDRESSING_MODE (ABSOLUTE);
;  1147          2131  2
;  1148          2132  2 ACB = .AQB [AQB$L_ACPQFL] + IRP$C_CDRP; ! first waiter's ACB
;  1149          2133  2 PINCL = PRI$_RESAVL;
;  1150          2134  2 IF NOT SCH$QAST ()
;  1151          2135  2 THEN
;  1152          2136  2     BUG_CHECK (XQPERR, 'Failed to queue ast');
;  1153          2137  2
;  1154          2138  1 END;


                                                        .EXTRN  SCH$QAST

                                        003C 00000 WAKE_WAITER:
                                                        .WORD   Save R2,R3,R4,R5                ;  2105
                     55        04   BC 00000060   8F   C1 00002          ADDL3   #96, @AQB, ACB          ;  2132
                                    52              02   D0 0000B          MOVL    #2, PINCL               ;  2133
                                        00000000G   9F   16 0000E          JSB     @#SCH$QAST              ;  2134
                                    04              50   E8 00014          BLBS    R0, 1$                  ;
                                                      FEFF 00017          BUGW                            ;  2136
                                                     0000* 00019          .WORD   <BUG$_XQPERR!4>         ;
                                                        04 0001B 1$:      RET                             ;  2138

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 06C5
```

```
;  1156          2139  1 GLOBAL ROUTINE RELEASE_CACHE : L_JSB NOVALUE =
;  1157          2140  1
;  1158          2141  1 !++
;  1159          2142  1 !
;  1160          2143  1 ! FUNCTIONAL DESCRIPTION:
;  1161          2144  1 !
;  1162          2145  1 ! Remove ourself from cache processing queue.  Wake up
;  1163          2146  1 ! the next process if queue is not empty.
;  1164          2147  1 !
;  1165          2148  1 !--
;  1166          2149  1
;  1167          2150  2 BEGIN
;  1168          2151  2
;  1169          2152  2 BUILTIN
;  1170          2153  2         REMQUE;
;  1171          2154  2
;  1172          2155  2 BIND_COMMON;
;  1173          2156  2
;  1174          2157  2 LOCAL
;  1175          2158  2         AQB     : REF BBLOCK,
;  1176          2159  2         DUMMY;
;  1177          2160  2
;  1178          2161  2 AQB = .CURRENT_VCB [VCB$L_AQB];
;  1179          2162  2
;  1180          2163  2 IF REMQUE (.AQB [QFLNK], DUMMY) EQL 0   ! queue not empty
;  1181          2164  2 THEN
;  1182          2165  2     WAKE_WAITER (.AQB);
;  1183          2166  2
;  1184          2167  1 END;

                                        52  DD 00000 RELEASE_CACHE::
                                                      PUSHL   R2                              ; 2139
                              50    98    AA  D0 00002          MOVL    -104(BASE), R0          ; 2161
                              52    10    A0  D0 00006          MOVL    16(R0), AQB             ;
                              51    00    B2  0F 0000A          REMQUE  @0(AQB), DUMMY          ; 2163
                                        50  DC 0000E          MOVPSL  R0                      ;
        50          50            02        01  EF 00010          EXTZV   #1, #2, R0, R0          ;
                                        06  12 00015          BNEQ    1$                      ;
                                        52  DD 00017          PUSHL   AQB                     ; 2165
                          C7    AF        01  FB 00019          CALLS   #1, WAKE_WAITER         ;
                                        04  BA 0001D 1$:       POPR    #^M<R2>                 ; 2167
                                            05 0001F          RSB                             ;

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 06E1
```

```
;1186     2168  1 GLOBAL ROUTINE GET_REQD_BFR_CREDITS : L_NORM NOVALUE =
;1187     2169  1
;1188     2170  1 !++
;1189     2171  1 !
;1190     2172  1 ! FUNCTIONAL DESCRIPTION:
;1191     2173  1 !
;1192     2174  1 ! Acquire minimum buffer credits so that this operation may proceed.
;1193     2175  1 ! Wait for their availability if necessary.
;1194     2176  1 ! Initialize the global cell CACHE_HDR, primarily used by the buffer
;1195     2177  1 ! management routines.
;1196     2178  1 !
;1197     2179  1 !--
;1198     2180  1
;1199     2181  2 BEGIN
;1200     2182  2
;1201     2183  2 BIND_COMMON;
;1202     2184  2
;1203     2185  2 LOCAL
;1204     2186  2     AQB     : REF BBLOCK,
;1205     2187  2     REQD;
;1206     2188  2
;1207     2189  2 AQB = .CURRENT_VCB [VCB$L_AQB];
;1208     2190  2
;1209     2191  2 CACHE_HDR = .AQB [AQB$L_BUFCACHE];
;1210     2192  2
;1211     2193  2 SERIAL_CACHE ();
;1212     2194  2
;1213     2195  3 BEGIN
;1214     2196  3 BIND
;1215     2197  3         POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] : VECTOR,
;1216     2198  3         POOL_WAITQ = CACHE_HDR [F11BC$Q_POOL_WAITQ] : BLOCKVECTOR [,8,BYTE];
;1217     2199  3
;1218     2200  3 DECR POOL FROM 3 TO 0
;1219     2201  3 DO
;1220     2202  4     BEGIN
;1221     2203  4
;1222     2204  4 ! The minimal buffer requirements are:
;1223     2205  4 ! 1 for storage bitmap blocks (pool 0)
;1224     2206  4 ! 2 for directory data blocks (pool 1)
;1225     2207  4 ! 3 for file headers (pool 2)
;1226     2208  4 ! 1 for directory index (pool 3)
;1227     2209  4 !
;1228     2210  4
;1229     2211  4
;1230     2212  4     REQD = .POOL + 1;
;1231     2213  4
;1232     2214  4     IF .POOL EQL 3
;1233     2215  4     THEN
;1234     2216  4         REQD = 1;
;1235     2217  4
;1236     2218  4     IF (POOLAVAIL [.POOL] = .POOLAVAIL [.POOL] - .REQD) LSS 0
;1237     2219  4     THEN
;1238     2220  5         BEGIN
;1239     2221  5
;1240     2222  5 ! Insert our ACB temporarily at the head of the queue.  Then when
;1241     2223  5 ! we pull our irp off of it to put on the wait queue, the cache interlock
;1242     2224  5 ! queue will not be left empty until we do our release_cache and
```

```
;  1243          2225  5 ! at that point pull off our temp acb entry.
;  1244          2226  5 !
;  1245          2227  5
;  1246          2228  5         INSQUE (.ACB_ADDR, AQB [AQB$L_ACPQFL]);
;  1247          2229  5         REMQUE (.IO_PACKET, IO_PACKET);             take us out of cache queue
;  1248          2230  5         INSQUE (.IO_PACKET, .POOL_WAITQ [.POOL, QBLNK]); ! into pool queue
;  1249          2231  5         RELEASE_CACHE ();
;  1250          2232  5         WAIT_FOR_AST ();
;  1251          2233  5
;  1252          2234  5         IF .AQB [AQB$L_ACPQFL] NEQ .IO_PACKET
;  1253          2235  5         THEN
;  1254          2236  5             BUG_CHECK (XQPERR, 'messed up cache interlock queues');
;  1255          2237  5
;  1256          2238  4         END;
;  1257          2239  4
;  1258          2240  4     BFR_CREDITS [.POOL] = .REQD;
;  1259          2241  4
;  1260          2242  3     END;
;  1261          2243  2 END;              ! of BIND
;  1262          2244  2
;  1263          2245  2 RELEASE_CACHE ();
;  1264          2246  2
;  1265          2247  1 END;


                                                007C 00000          .ENTRY  GET_REQD_BFR_CREDITS, Save R2,R3,R4,R5,R6    ; 2168
                     50        98    AA   D0 00002          MOVL    -104(BASE), R0                               ; 2189
                     54        10    A0   D0 00006          MOVL    16(R0), AQB                                  ;
                  FC    AA        18    A4   D0 0000A          MOVL    24(AQB), -4(BASE)                            ; 2191
                                    FB73   30 0000F          BSBW    SERIAL_CACHE                                 ; 2193
              55    FC    AA 00000068    8F   C1 00012          ADDL3   #104, -4(BASE), R5                           ; 2197
              56    FC    AA 00000048    8F   C1 0001B          ADDL3   #72, -4(BASE), R6                            ; 2198
                     52                  03   D0 00024          MOVL    #3, POOL                                     ; 2200
                     53        01        A2   9E 00027 1$:      MOVAB   1(R2), REQD                                  ; 2212
                     03                  52   D1 0002B          CMPL    POOL, #3                                     ; 2214
                                         03   12 0002E          BNEQ    2$                                           ;
                     53                  01   D0 00030          MOVL    #1, REQD                                     ; 2216
                   6542                  53   C2 00033 2$:      SUBL2   REQD, (R5)[POOL]                             ; 2218
                                         24   18 00037          BGEQ    3$                                           ;
                     64        C8        BA   0E 00039          INSQUE  @-56(BASE), (AQB)                            ; 2228
                  90    AA        90        BA   0F 0003D          REMQUE  @-112(BASE), -112(BASE)                      ; 2229
                     50        04 A642     7E 00042          MOVAQ   4(R6)[POOL], R0                              ; 2230
                  00    B0        90        BA   0E 00047          INSQUE  @-112(BASE), @0(R0)                          ;
                                         92   10 0004C          BSBB    RELEASE_CACHE                                ; 2231
               0000G    CF                  00   FB 0004E          CALLS   #0, WAIT_FOR_AST                             ; 2232
                  90    AA                  64   D1 00053          CMPL    (AQB), -T12(BASE)                            ; 2234
                                         04   13 00057          BEQL    3$                                           ;
                                            FEFF 00059          BUGW                                                 ; 2236
                                          0000* 0005B          .WORD   <BUG$_XQPERR!4>                              ;
                  EC AA42                 53   B0 0005D 3$:      MOVW    REQD, -20(BASE)[POOL]                        ; 2240
                     C2                   52   F4 00062          SOBGEQ  POOL, 1$                                     ; 2200
                                       FF78   30 00065          BSBW    RELEASE_CACHE                                ; 2245
                                              04 00068          RET                                                  ; 2247
```

; Routine Size: 105 bytes,   Routine Base: $CODE$ + 0701

```
;  1267          2248  1 GLOBAL ROUTINE RETURN_CREDITS : L_NORM NOVALUE =
;  1268          2249  1
;  1269          2250  1 !++
;  1270          2251  1 !
;  1271          2252  1 ! FUNCTIONAL DESCRIPTION:
;  1272          2253  1 !
;  1273          2254  1 ! This routine returns the in-process buffer credits to the
;  1274          2255  1 ! buffer pool.  It moves the next waiter in line to the head
;  1275          2256  1 ! of the cache processing queue, which will cause it to becom
;  1276          2257  1 ! awakened when we release the cache interlock.
;  1277          2258  1 !
;  1278          2259  1 !--
;  1279          2260  1
;  1280          2261  2 BEGIN
;  1281          2262  2
;  1282          2263  2 BIND_COMMON;
;  1283          2264  2
;  1284          2265  2 LOCAL
;  1285          2266  2     POOLAVAIL   : REF VECTOR,
;  1286          2267  2     AQB         : REF BBLOCK;
;  1287          2268  2
;  1288          2269  2 ! This may be called multiple times.   Check if already returned credits.
;  1289          2270  2 !
;  1290          2271  2
;  1291          2272  2 IF .BFR_CREDITS [3] EQL 0
;  1292          2273  2 THEN RETURN;
;  1293          2274  2
;  1294          2275  2 AQB = .CURRENT_VCB [VCB$L_AQB];
;  1295          2276  2 POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL];
;  1296          2277  2
;  1297          2278  2 SERIAL_CACHE ();
;  1298          2279  2
;  1299          2280  2 DECR POOL FROM 3 TO 0
;  1300          2281  2 DO
;  1301          2282  3     BEGIN
;  1302          2283  3
;  1303          2284  3     BIND
;  1304          2285  3         POOL_WAITQ = CACHE_HDR [F11BC$Q_POOL_WAITQ] + .POOL*8 : BBLOCK;
;  1305          2286  3
;  1306          2287  3     IF .BFRS_USED [.POOL] NEQ 0
;  1307          2288  3     THEN
;  1308          2289  3         BUG_CHECK (XQPERR, 'buffer use not to zero yet');
;  1309          2290  3
;  1310          2291  3     POOLAVAIL [.POOL] = .POOLAVAIL [.POOL] + .BFR_CREDITS [.POOL];
;  1311          2292  3     BFR_CREDITS [.POOL] = 0;
;  1312          2293  3
;  1313          2294  3     IF .POOL_WAITQ [QFLNK] NEQ POOL_WAITQ [QFLNK]
;  1314          2295  3     THEN
;  1315          2296  4         BEGIN
;  1316          2297  4         LOCAL
;  1317          2298  4             WAITER;
;  1318          2299  4
;  1319          2300  4         REMQUE (.POOL_WAITQ [QFLNK], WAITER);
;  1320          2301  4         INSQUE (.WAITER, .AQB [AQB$L_ACPQFL]);  ! insert AFTER us.
;  1321          2302  3         END;
;  1322          2303  3
;  1323          2304  2     END;
```

```
;  1324          2305  2
;  1325          2306  2 RELEASE_CACHE ();
;  1326          2307  2
;  1327          2308  1 END;


                                    003C 00000          .ENTRY  RETURN_CREDITS, Save R2,R3,R4,R5        ; 2248
                              54    EC   AA   9E 00002    MOVAB   -20(BASE), R4                          ; 2261
                                    06   A4   B5 00006    TSTW    6(R4)                                  ; 2272
                                         48   13 00009    BEQL    4$                                     ;
                              50         98   AA   D0 0000B    MOVL    -104(BASE), R0                     ; 2275
                              55         10   A0   D0 0000F    MOVL    16(R0), AQB                        ;
                        52    FC   AA 00000068   8F   C1 00013    ADDL3   #104, -4(BASE), POOLAVAIL      ; 2276
                                       FAFD   30 0001C    BSBW    SERIAL_CACHE                           ; 2278
                              50         03   D0 0001F    MOVL    #3, POOL                               ; 2280
                              51         FC BA40   7E 00022 1$:  MOVAQ   @-4(BASE)[POOL], R1           ; 2285
                              51         48   A1   9E 00027    MOVAB   72(R1), R1                     ;
                                    F4   AA40   B5 0002B    TSTW    -12(BASE)[POOL]                    ; 2287
                                         04   13 0002F    BEQL    2$                                     ;
                                         FEFF 00031    BUGW                                           ; 2289
                                        0000* 00033    .WORD   <BUG$_XQPERR!4>                         ;
                              53              6440   3C 00035 2$:  MOVZWL  (R4)[POOL], R3                ; 2291
                        6240                   53   C0 00039    ADDL2   R3, (POOLAVAIL)[POOL]          ;
                                               6440   B4 0003D    CLRW    (R4)[POOL]                     ; 2292
                              51                 61   D1 0C040    CMPL    (R1), R1                       ; 2294
                                                 08   13 00043    BEQL    3$                             ;
                              53         00      B1   0F 00045    REMQUE  @0(R1), WAITER                 ; 2300
                        00    B5                 63   0E 00049    INSQUE  (WAITER), @0(AQB)              ; 2301
                              D2                 50   F4 0004D 3$:  SOBGEQ  POOL, 1$                    ; 2280
                                              FF24   30 00050    BSBW    RELEASE_CACHE                  ; 2306
                                                     04 00053 4$:  RET                                   ; 2308

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 076A
```

```
;  1329          2309  1 GLOBAL ROUTINE WRITE_BLOCK (BUFFER) : L_NORM NOVALUE =
;  1330          2310  1
;  1331          2311  1 !++
;  1332          2312  1 !
;  1333          2313  1 ! FUNCTIONAL DESCRIPTION:
;  1334          2314  1 !
;  1335          2315  1 !       This routine writes the indicated block back to the disk.
;  1336          2316  1 !
;  1337          2317  1 ! CALLING SEQUENCE:
;  1338          2318  1 !       WRITE_BLOCK (ARG1)
;  1339          2319  1 !
;  1340          2320  1 ! INPUT PARAMETERS:
;  1341          2321  1 !       ARG1: address of block buffer
;  1342          2322  1 !
;  1343          2323  1 ! IMPLICIT INPUTS:
;  1344          2324  1 !       BUFFER DESCRIPTOR ARRAYS
;  1345          2325  1 !
;  1346          2326  1 ! OUTPUT PARAMETERS:
;  1347          2327  1 !       NONE
   1348          2328  1 !
;  1349          2329  1 ! IMPLICIT OUTPUTS:
;  1350          2330  1 !       NONE
;  1351          2331  1 !
;  1352          2332  1 ! ROUTINE VALUE:
;  1353          2333  1 !       NONE
;  1354          2334  1 !
;  1355          2335  1 ! SIDE EFFECTS:
;  1356          2336  1 !       block written
;  1357          2337  1 !
;  1358          2338  1 !--
;  1359          2339  1
;  1360          2340  2 BEGIN
;  1361          2341  2
;  1362          2342  2 LOCAL
;  1363          2343  2         BFRD            : REF BBLOCK,
;  1364          2344  2         STATUS,                         ! service status of QIO call
;  1365          2345  2         INDX;                           ! index of buffer
;  1366          2346  2
;  1367          2347  2 EXTERNAL
;  1368          2348  2         CLU$GL_CLUB     : ADDRESSING_MODE (ABSOLUTE),
;  1369          2349  2         ACP$GB_DATACHK  : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
;  1370          2350  2                                         ! ACP datacheck enable flags
;  1371          2351  2
;  1372          2352  2 BIND_COMMON;
;  1373          2353  2
;  1374          2354  2 EXTERNAL LITERAL
;  1375          2355  2         ACP$V_WRITECHK  : UNSIGNED (6); ! write check enable flag
;  1376          2356  2
;  1377          2357  2 INDX = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512 + 1;
;  1378          2358  2
;  1379          2359  2 IF .INDX<0,16> GTRU .CACHE_HDR [F11BC$W_BFRCNT]
;  1380          2360  2 THEN
;  1381          2361  2     BUG_CHECK (BADBUFADR, 'out of this cache, man');
;  1382          2362  2
;  1383          2363  2 BFRD = BFRD_ADDR (.INDX);
;  1384          2364  2
;  1385          2365  2 ! Stuff the UCB of IO_CHANNEL to go where we want.
```

```
;  1386          2366  2 !
;  1387          2367  2
;  1388          2368  2 IO_CCB [CCB$L_UCB] = .BFRD [BFRD$L_UCB];
;  1389          2369  2
;  1390          2370  3 BEGIN
;  1391          2371  3 LOCAL
;  1392          2372  3         PTR             : REF BBLOCK,
;  1393          2373  3         SAVE_PRIV       : VECTOR [4];
;  1394          2374  3
;  1395          2375  3 PTR = .CTL$GL_PCB;
;  1396          2376  3 PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] + 1;
;  1397          2377  3 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] + 1;
;  1398          2378  3 SAVE_PRIV [0] = .(PTR [PCB$Q_PRIV]);
;  1399          2379  3 SAVE_PRIV [1] = .(PTR [PCB$Q_PRIV]+4);
;  1400          2380  3 BBLOCK [PTR [PCB$Q_PRIV], PRV$V_LOG_IO] = 1;
;  1401          2381  3 BBLOCK [PTR [PCB$Q_PRIV], PRV$V_BYPASS] = 1;
;  1402          2382  3 PTR = .CTL$GL_PHD;
;  1403          2383  3 SAVE_PRIV [2] = .(PTR [PHD$Q_PRIVMSK]);
;  1404          2384  3 SAVE_PRIV [3] = .(PTR [PHD$Q_PRIVMSK]+4);
;  1405          2385  3 BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_LOG_IO] = 1;
;  1406          2386  3 BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_BYPASS] = 1;
;  1407          2387  3
;  1408          2388  3 PMS_TOT_WRITE = .PMS_TOT_WRITE + 1;
;  1409        P 2389  3 STATUS = $QIO (
;  1410        P 2390  3                 EFN  = EFN,
;  1411        P 2391  3                 ASTADR = CONTINUE_THREAD,
;  1412        P 2392  3                 ASTPRM = .BASE,
;  1413        P 2393  3                 CHAN = .IO_CHANNEL,
;  1414        P 2394  3                 FUNC = (IO$_WRITELBLK
;  1415        P 2395  3                         OR .ACP$GB_DATACHK[ACP$V_WRITECHK]
;  1416        P 2396  3                         ^ $BITPOSITION (IO$V_DATACHECK)),
;  1417        P 2397  3                 IOSB = IO_STATUS,
;  1418        P 2398  3                 P1   = .BUFFER,
;  1419        P 2399  3                 P2   = 512,
;  1420        P 2400  3                 P3   = .BFRD [BFRD$L_LBN]
;  1421          2401  3                 );
;  1422          2402  3
;  1423          2403  3 (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
;  1424          2404  3 (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
;  1425          2405  3 PTR = .CTL$GL_PCB;
;  1426          2406  3 PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] - 1;
;  1427          2407  3 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
;  1428          2408  3 (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
;  1429          2409  3 (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];
;  1430          2410  3
;  1431          2411  2 END;                ! of block defining PTR, SAVE_PRIV
;  1432          2412  2
;  1433          2413  2 IF NOT .STATUS
;  1434          2414  2 THEN IO_STATUS = .STATUS
;  1435          2415  2 ELSE WAIT_FOR_AST();
;  1436          2416  2
;  1437          2417  2 ! Restore CURRENT_UCB to IO_CHANNEL.
;  1438          2418  2 !
;  1439          2419  2
;  1440          2420  2 IO_CCB [CCB$L_UCB] = .CURRENT_UCB;
;  1441          2421  2
;  1442          2422  2 IF .IO_STATUS
```

```
;  1443          2423  2 THEN
;  1444          2424  2     BEGIN
;  1445          2425  2     LOCAL
;  1446          2426  2         CLUSTER,
;  1447          2427  2         LCKINDX;
;  1448          2428  2
;  1449          2429  2     BFRD [BFRD$V_DIRTY] = 0;
;  1450          2430  2
;  1451          2431  2 ! Now let's find the lock that backs this buffer and bump it's
;  1452          2432  2 ! sequence number so the rest of the world will know we've modified
;  1453          2433  2 ! this buffer.
;  1454          2434  2 !
;  1455          2435  2
;  1456          2436  2     LCKINDX = 0;
;  1457          2437  2
;  1458          2438  2     DO
;  1459          2439  3         BEGIN
;  1460          2440  3
;  1461          2441  3         IF .BFRD [BFRD$L_LOCKBASIS] EQL .LB_BASIS [.LCKINDX]
;  1462          2442  3         THEN
;  1463          2443  3             EXITLOOP;
;  1464          2444  3
;  1465          2445  3         LCKINDX = .LCKINDX + 1;
;  1466          2446  3         END
;  1467          2447  2     UNTIL .LCKINDX EQL LB_NUM;
;  1468          2448  2
;  1469          2449  2     IF .LCKINDX EQL LB_NUM
;  1470          2450  2     THEN
;  1471          2451  2         BUG_CHECK (XQPERR, 'no backing lock for dirty buffer');
;  1472          2452  2
;  1473          2453  2     CLUSTER = 0;
;  1474          2454  2
;  1475          2455  2     IF .BBLOCK [CURRENT_UCB [UCB$L_DEVCHAR2], DEV$V_CLU]
;  1476          2456  2         AND .CLU$GL_CLUB NEQ 0
;  1477          2457  2     THEN
;  1478          2458  2         CLUSTER = 1;
;  1479          2459  2
;  1480          2460  2     CASE .BFRD [BFRD$B_BTYPE] FROM 0 TO 5 OF
;  1481          2461  2     SET
;  1482          2462  2     [HEADER_TYPE]:
;  1483          2463  2         IF .CLUSTER
;  1484          2464  2         THEN
;  1485          2465  2             LB_HDRSEQ [.LCKINDX] = .LB_HDRSEQ [.LCKINDX] + 1;
;  1486          2466  2
;  1487          2467  2     [DIRECTORY_TYPE, DATA_TYPE]:
;  1488          2468  2         IF .CLUSTER
;  1489          2469  2         THEN
;  1490          2470  2             LB_DATASEQ [.LCKINDX] = .LB_DATASEQ [.LCKINDX] + 1;
;  1491          2471  2
;  1492          2472  2     [BITMAP_TYPE]:
;  1493          2473  2         (LB_DATASEQ [0])<0,16,0> = .(LB_DATASEQ [0])<0,16,0> + 1;
;  1494          2474  2
;  1495          2475  2     [INDEX_TYPE]:
;  1496          2476  2         (LB_DATASEQ [0])<16,16,0> = .(LB_DATASEQ [0])<16,16,0> + 1;
;  1497          2477  2
;  1498          2478  2     [QUOTA_TYPE]:
;  1499          2479  2         SAVE_VC_FLAGS<1,15,0> = .SAVE_VC_FLAGS<1,15,0> + 1;
```

```
;  1500          2480  3
;  1501          2481  3      TES;
;  1502          2482  3
;  1503          2483  3      END
;  1504          2484  2  ELSE
;  1505          2485  3      BEGIN
;  1506          2486  3
;  1507          2487  3  ! clean up will iterate on write failures if the volume is write locked until
;  1508          2488  3  ! the best effort clean up has been done. However, if the write error was on the
;  1509          2489  3  ! storage bitmap, we allocation lock the volume to prevent further damage.
;  1510          2490  3  !
;  1511          2491  3
;  1512          2492  3      IF .BFRD [BFRD$V_POOL] EQL 0        ! this is storage bitmap pool
;  1513          2493  3          AND SURFACE_ERROR (.IO_STATUS [0])
;  1514          2494  3      THEN
;  1515          2495  3          CURRENT_VCB [VCB$V_NOALLOC] = 1;
;  1516          2496  3
;  1517          2497  3      INVALIDATE (.BUFFER);
;  1518          2498  3      ERR_EXIT (.IO_STATUS[0]);
;  1519          2499  2      END;
;  1520          2500  2
;  1521          2501  1  END;                                   ! end of routine WRITE_BLOCK


                                                        .EXTRN  CLU$GL_CLUB, ACP$V_WRITECHK
                                                        .EXTRN  BUG$_BADBUFADR

                                                003C 00000      .ENTRY  WRITE_BLOCK, Save R2,R3,R4,R5                   ; 2309
                     55 00000000G   00    9E 00002      MOVAB   CTL$GL_PCB, R5
                     5E             10    C2 00009      SUBL2   #16, SP
                     54       00A8  CA    9E 0000C      MOVAB   168(BASE), R4                                   ; 2349
                52   04       AC    FC    BA C3 00011      SUBL3   @-4(BASE), BUFFER, R2                           ; 2357
                     52 00000200    8F    C6 00017      DIVL2   #512, R2
                                    52    D6 0001E      INCL    INDX
                     50         FC  AA    D0 00020      MOVL    -4(BASE), R0                                    ; 2359
                   16   A0          52    B1 00024      CMPW    INDX, 22(R0)
                                    04    1B 00028      BLEQU   1$
                                  FEFF    0002A         BUGW                                                    ; 2361
                                 0000*    0002C         .WORD   <BUG$_BADBUFADR!4>
                     50         FC  AA    D0 0002E 1$:  MOVL    -4(BASE), R0                                    ; 2363
                     52             20    C4 00032      MULL2   #32, R2
                     52         18  A0    C0 00035      ADDL2   24(R0), R2
                     52             20    C2 00039      SUBL2   #32, BFRD
                 FF74   DA         0C  A2    D0 0003C      MOVL    12(BFRD), @-140(BASE)                           ; 2368
                     53             65    D0 00042      MOVL    CTL$GL_PCB, PTR                                 ; 2375
                                 3E  A3    B6 00045      INCW    62(PTR)                                         ; 2376
                                 38  A3    B6 00048      INCW    56(PTR)                                         ; 2377
                     6E       0084  C3    7D 0004B      MOVQ    132(PTR), SAVE_PRIV                             ; 2378
                 0084   C3 20000080  8F    C8 00050      BISL2   #536871040, 132(PTR)                            ; 2381
                     53 00000000G   00    D0 00059      MOVL    CTL$GL_PHD, PTR                                 ; 2382
                   08   AE          63    7D 00060      MOVQ    (PTR), SAVE_PRIV+8                              ; 2383
                     63 20000080    8F    C8 00064      BISL2   #536871040, (PTR)                               ; 2386
                           08EC     CA    D6 0006B      INCL    2284(BASE)                                      ; 2388
                                    7E    7C 0006F      CLRQ    -(SP)                                           ; 2401
                                    7E    D4 00071      CLRL    -(SP)
                                 08  A2    DD 00073      PUSHL   8(BFRD)
```

```
                                7E      0200    8F  3C 00076          MOVZWL  #512, -(SP)                                                     ;
                                  04    AC  DD 0007B                  PUSHL   BUFFER                                                          ;
                                        5A  DD 0007E                  PUSHL   BASE                                                            ;
                                  0000G CF  9F 00080                  PUSHAB  CONTINUE_THREAD                                                 ;
                                    88  AA  9F 00084                  PUSHAB  -120(BASE)                                                      ;
        50 00000000G    9F              01      00G EF 00087          EXTZV   S^ACP$V_WRITECHK, #1, @#ACP$GB_DATACHK, R0                      ;
                        50              50      0E  78 00090          ASHL    #14, R0, R0                                                     ;
                        7E              50      20  C9 00094          BISL3   #32, R0, -(SP)                                                  ;
                                      FF78  CA  DD 00098              PUSHL   -136(BASE)                                                      ;
                                            1E  DD 0009C              PUSHL   #30                                                             ;
                             00000000G  00      0C  FB 0009E          CALLS   #12, SYS$QIO                                                    ;
                                        63        08  AE 7D 000A5      MOVQ    SAVE_PRIV+8, (PTR)                                              ;2403
                                        53        65  D0 000A9      MOVL    CTL$GL_PCB, PTR                                                 ;2405
                                                  3E  A3 B7 000AC      DECW    62(PTR)                                                         ;2406
                                                  38  A3 B7 000AF      DECW    56(PTR)                                                         ;2407
                                 0084   C3              6E  7D 000B2      MOVQ    SAVE_PRIV, 132(PTR)                                             ;2408
                                        06              50  E8 000B7      BLBS    STATUS, 2$                                                      ;2413
                                   88   AA              50  D0 000BA      MOVL    STATUS, -120(BASE)                                              ;2414
                                                        05  11 000BE      BRB     3$                                                              ;
                                 0000G  CF              00  FB 000C0  2$:  CALLS   #0, WAIT_FOR_AST                                                ;2415
                                 FF74   DA        94    AA  D0 000C5  3$:  MOVL    -108(BASE), @-140(BASE)                                         ;2420
                                        51        88    AA  D0 000CB      MOVL    -120(BASE), R1                                                  ;2422
                                        6B              51  E9 000CF      BLBC    R1, 14$                                                         ;
                                   18   A2              04  8A 000D2      BICB2   #4, 24(BFRD)                                                    ;2429
                                                        50  D4 000D6      CLRL    LCKINDX                                                         ;2436
                                 0080 CA40        10    A2  D1 000D8  4$:  CMPL    16(BFRD), 128(BASE)[LCKINDX]                                    ;2441
                                                        07  13 000DF      BEQL    5$                                                              ;
                                                        50  D6 000E1      INCL    LCKINDX                                                         ;2445
                                        05              50  D1 000E3      CMPL    LCKINDX, #5                                                     ;2447
                                                        F0  12 000E6      BNEQ    4$                                                              ;
                                        05              50  D1 000E8  5$:  CMPL    LCKINDX, #5                                                     ;2449
                                                        04  12 000EB      BNEQ    6$                                                              ;
                                                          FEFF 000ED      BUGW                                                                    ;2451
                                                         0000* 000EF      .WORD   <BUG$_XQPERR!4>                                                 ;
                                                        53  D4 000F1  6$:  CLRL    CLUSTER                                                         ;2453
                                        51        94    AA  D0 000F3      MOVL    -108(BASE), R1                                                  ;2455
                                        0B        3C    A1  E9 000F7      BLBC    60(R1), 7$                                                      ;
                                             00000000G  9F  D5 000FB      TSTL    @#CLU$GL_CLUB                                                   ;2456
                                                        03  13 00101      BEQL    7$                                                              ;
                                        53              01  D0 00103      MOVL    #1, CLUSTER                                                     ;2458
                             05         00        19    A2  8F 00106  7$:  CASEB   25(BFRD), #0, #5                                               ;2460
        001F                 0015       001C            000C      0010B  8$:  .WORD   9$-8$,-                                                     ;
                                        0023            0015      00113          11$-8$,-                                                        ;
                                                                                 10$-8$,-                                                        ;
                                                                                 12$-8$,-                                                        ;
                                                                                 10$-8$,-                                                        ;
                                                                                 13$-8$                                                          ;
                                        60              53  E9 00117  9$:  BLBC    CLUSTER, 18$                                                    ;2463
                                              0094 CA40 D6 0011A          INCL    148(BASE)[LCKINDX]                                              ;2465
                                                        04  0011F       RET                                                                     ;2463
                                        57              53  E9 00120  10$:  BLBC    CLUSTER, 18$                                                    ;2468
                                                     6440 D6 00123          INCL    (R4)[LCKINDX]                                                   ;2470
                                                        04  00126       RET                                                                     ;2468
                                                     64 B6 00127  11$:  INCW    (R4)                                                            ;2473
                                                        04  00129       RET                                                                     ;
                                              02    A4  B6 0012A  12$:  INCW    2(R4)                                                           ;2476
                                                        04  0012D       RET                                                                     ;
```

```
                50        A4    AA              0F              01  EF 0012E 13$:    EXTZV   #1, #15, -92(BASE), R0                  ; 2479
                                                                50  D6 00134         INCL    R0
            A4  AA              0F              01              50  F0 00136         INSV    R0, #1, #15, -92(BASE)
                                                                04  0013C         RET                                              ; 2422
                                                03          18  A2  93 0013D 14$:    BITB    24(BFRD), #3                         ; 2492
                                                                09  12 00141         BNEQ    15$
                                  000001F4      8F              51  D1 00143         CMPL    R1, #500                             ; 2493
                                                                1B  13 0014A         BEQL    16$
                                  0000005C      8F              51  D1 0014C 15$:    CMPL    R1, #92
                                                                12  13 00153         BEQL    16$
                                  000000BC      8F              51  D1 00155         CMPL    R1, #188
                                                                09  13 0015C         BEQL    16$
                                  00002144      8F              51  D1 0015E         CMPL    R1, #8516
                                                                08  12 00165         BNEQ    17$
                                                50          98  AA  D0 00167 16$:    MOVL    -104(BASE), R0                       ; 2495
                                       0B       A0              10  88 0016B         BISB2   #16, 11(R0)
                                                            04  AC  DD 0016F 17$:    PUSHL   BUFFER                               ; 2497
                                     0000V      CF              01  FB 00172         CALLS   #1, INVALIDATE
                                                            88  AA  BF 00177         CHMU    -120(BASE)                           ; 2498
                                                                04  0017A 18$:    RET                                              ; 2501

; Routine Size:  379 bytes,    Routine Base:  $CODE$ + 07BE
```

```
;  1523          2502  1 GLOBAL ROUTINE CREATE_BLOCK (LBN, COUNT, TYPE) : L_NORM =
;  1524          2503  1
;  1525          2504  1 !++
;  1526          2505  1 !
;  1527          2506  1 ! FUNCTIONAL DESCRIPTION:
;  1528          2507  1 !
;  1529          2508  1 !       This routine fabricates block buffer(s) containing the designated
;  1530          2509  1 !       block(s). The type code is as for READ_BLOCK and determines the buffer
;  1531          2510  1 !       pool to be used.
;  1532          2511  1 !
;  1533          2512  1 ! CALLING SEQUENCE:
;  1534          2513  1 !
;  1535          2514  1 ! INPUT PARAMETERS:
;  1536          2515  1 !
;  1537          2516  1 ! IMPLICIT INPUTS:
;  1538          2517  1 !
;  1539          2518  1 ! OUTPUT PARAMETERS:
;  1540          2519  1 !
;  1541          2520  1 ! IMPLICIT OUTPUTS:
;  1542          2521  1 !       NONE
;  1543          2522  1 !
;  1544          2523  1 ! ROUTINE VALUE:
;  1545          2524  1 !       address of buffer
;  1546          2525  1 !
;  1547          2526  1 ! SIDE EFFECTS:
;  1548          2527  1 !       buffer zeroed and recorded as a block read
;  1549          2528  1 !
;  1550          2529  1 !--
;  1551          2530  1
;  1552          2531  2 BEGIN
;  1553          2532  2
;  1554          2533  2 BIND_COMMON;
;  1555          2534  2
;  1556          2535  2 LOCAL
;  1557          2536  2         BUFFER,
;  1558          2537  2         FOUND_COUNT,
;  1559          2538  2         BFRD            : REF BBLOCK,
;  1560          2539  2         INDX0;
;  1561          2540  2
;  1562          2541  2 INDX0 = FIND_BUFFER (.LBN, .TYPE, 1, FOUND_COUNT);
;  1563          2542  2
;  1564          2543  2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .INDX0*BFRD$S_BFRDDEF;
;  1565          2544  2
;  1566          2545  2 BUFFER = .CACHE_HDR [F11BC$L_BUFBASE] + .INDX0 ^12;
;  1567          2546  2 CH$FILL (0, 512, .BUFFER);
;  1568          2547  2
;  1569          2548  2 IF (.LBN + 1) NEQ 0
;  1570          2549  2 THEN
;  1571          2550  3     BEGIN
;  1572          2551  3     BFRD [BFRD$V_DIRTY] = 1;
;  1573          2552  3     BFRD [BFRD$V_VALID] = 1;
;  1574          2553  2     END;
;  1575          2554  2
;  1576          2555  2 RETURN .BUFFER;
;  1577          2556  2
;  1578          2557  1 END;                                    ! end of routine CREATE_BLOCK
```

```
                                                00FC 00000          .ENTRY  CREATE_BLOCK, Save R2,R3,R4,R5,R6,R7        ; 2502
                                          5E            04   C2 00002  SUBL2   #4, SP                                  ;
                                                        5E   DD 00005  PUSHL   SP                                      ; 2541
                                                        01   DD 00007  PUSHL   #1                                      ;
                                                   0C   AC   DD 00009  PUSHL   TYPE                                    ;
                                                   04   AC   DD 0000C  PUSHL   LBN                                     ;
                                              F989 CF        04   FB 0000F  CALLS   #4, FIND_BUFFER                     ;
                                                   52        50   D0 00014  MOVL    R0, INDX0                         ;
                                                   51        FC   AA D0 00017  MOVL    -4(BASE), R1                      ; 2543
                                         50        52        05   78 0001B  ASHL    #5, INDX0, R0                     ;
                                         56        50        18   A1 C1 0001F  ADDL3   24(R1), R0, BFRD               ;
                                         50        52        09   78 00024  ASHL    #9, INDX0, R0                     ; 2545
                                         57        50        FC   BA C1 00028  ADDL3   @-4(BASE), R0, BUFFER         ;
                0200    8F       00        6E             00   2C 0002D  MOVC5   #0, (SP), #0, #512, (BUFFER)       ; 2546
                                                             67      00034  ;
                                         50   04   AC        01   C1 00035  ADDL3   #1, LBN, R0                      ; 2548
                                                             04   13 0003A  BEQL    1$                                ;
                                              18   A6        0C   88 0003C  BISB2   #12, 24(BFRD)                    ; 2552
                                                   50        57   D0 00040 1$:  MOVL    BUFFER, R0                    ; 2555
                                                             04 00043  RET                                           ; 2557

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 0939
```

```
;1580     2558  1 GLOBAL ROUTINE UNHOOK_BFRD (INDXARG, BFRDARG) : L_NORM NOVALUE =
;1581     2559  1
;1582     2560  1 !++
;1583     2561  1 !
;1584     2562  1 ! FUNCTIONAL DESCRIPTION:
;1585     2563  1 !
;1586     2564  1 ! This routine extricates a BFRD and its associated BFRL from
;1587     2565  1 ! their hash lookup lists.  It also takes care of the BFRL if
;1588     2566  1 ! this was the last reference to it.
;1589     2567  1 ! The caller of this routine is responsible for clearing or
;1590     2568  1 ! rewriting other relevant fields in the BFRD.
;1591     2569  1 !
;1592     2570  1 !--
;1593     2571  1
;1594     2572  2 BEGIN
;1595     2573  2
;1596     2574  2 BIND_COMMON;
;1597     2575  2
;1598     2576  2 EXTERNAL ROUTINE
;1599     2577  2         DEL_EXTFCB      : L_NORM,
;1600     2578  2         NUKE_HEAD_FCB   : L_NORM NOVALUE;
;1601     2579  2
;1602     2580  2 LOCAL
;1603     2581  2         BFRD            : REF BBLOCK,
;1604     2582  2         INDX            : WORD,
;1605     2583  2         INDX_ADDR,
;1606     2584  2         BFRL            : REF BBLOCK;
;1607     2585  2
;1608     2586  2 BFRD = .BFRDARG;
;1609     2587  2 INDX = .INDXARG;
;1610     2588  2
;1611     2589  2 IF .BFRD [BFRD$V_DIRTY]
;1612     2590  2 THEN
;1613     2591  2     BUG_CHECK (XQPERR, 'Should not find dirty buffer');
;1614     2592  2
;1615     2593  2 BFRD [BFRD$V_VALID] = 0;
;1616     2594  2
;1617     2595  2 IF .BFRD [BFRD$B_BTYPE] EQL DIRINDX_TYPE
;1618     2596  2 THEN
;1619     2597  3     BEGIN
;1620     2598  3     LOCAL
;1621     2599  3         BUFR    : REF BBLOCK,
;1622     2600  3         DIRFCB  : REF BBLOCK;
;1623     2601  3
;1624     2602  3 ! DIRINDX_TYPE blocks are not hashed by lbn, and the field is used
;1625     2603  3 ! to point to the directory fcb.
;1626     2604  3
;1627     2605  3     IF (DIRFCB = .BFRD [BFRD$L_LBN]) NEQ 0
;1628     2606  3     THEN
;1629     2607  4         BEGIN
;1630     2608  4
;1631     2609  4         IF .DIRFCB [FCB$B_TYPE] NEQ DYN$C_FCB
;1632     2610  4             OR .DIRFCB [FCB$L_DIRINDX] EQL 0
;1633     2611  4         THEN
;1634     2612  4             BUG_CHECK (XQPERR, 'not legit dirfcb');
;1635     2613  4
;1636     2614  4 ! This will break the association of this directory index buffer
```

```
;  1637          2615  4 ! from the directory fcb.
;  1638          2616  4 !
;  1639          2617  4
;  1640          2618  4         DIRFCB [FCB$L_DIRINDX] = 0;
;  1641          2619  4
;  1642          2620  4         IF .DIRFCB [FCB$W_REFCNT] EQL 0
;  1643          2621  4         THEN
;  1644          2622  4             IF TESTBITSC (DIRFCB [FCB$V_DIR])
;  1645          2623  4
;  1646          2624  4 ! from this point on, because the refcnt was zero, and the DIR
;  1647          2625  4 ! flag was set and is now clear, the search_fcb routine will no
;  1648          2626  4 ! longer find this fcb, so we are free to deallocate it.
;  1649          2627  4 ! Furthermore, because the DIR was set, we know that no search_fcb
;  1650          2628  4 ! had found it at the time we cleared it because search_fcb will
;  1651          2629  4 ! unconditionally clear it.
;  1652          2630  4 !
;  1653          2631  4                 THEN
;  1654          2632  5                     BEGIN
;  1655          2633  5
;  1656          2634  5 ! Note that we are under the cache interlock here, and assume that
;  1657          2635  5 ! the these routines will do nothing that could cause a stall.
;  1658          2636  5 !
;  1659          2637  5
;  1660          2638  5                     DEL_EXTFCB (.DIRFCB);
;  1661          2639  5                     NUKE_HEAD_FCB (.DIRFCB);
;  1662          2640  4                     END;
;  1663          2641  4
;  1664          2642  4         BFRD [BFRD$L_LBN] = 0;
;  1665          2643  4
;  1666          2644  3         END;              ! of bfrd$l_lbn (dirfcb pointer) neq zero
;  1667          2645  3
;  1668          2646  2     END;
;  1669          2647  2
;  1670          2648  2 IF .BFRD [BFRD$L_LBN] NEQ 0
;  1671          2649  2 THEN
;  1672          2650  2
;  1673          2651  2 ! BFRD$L_LBN neq 0 means that this buffer is in the hash lookup list.
;  1674          2652  2 ! We need to remove it from the list it is currently
;  1675          2653  2 ! in as we are about to use the same buffer for some other LBN.
;  1676          2654  2 !
;  1677          2655  2
;  1678          2656  3     BEGIN
;  1679          2657  3
;  1680          2658  3     INDX_ADDR = LOOKUP_LBN (.BFRD [BFRD$L_LBN]);
;  1681          2659  3
;  1682          2660  3     IF .(.INDX_ADDR)<0,16> EQL .INDX
;  1683          2661  3     THEN
;  1684          2662  3         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD]
;  1685          2663  3     ELSE
;  1686          2664  4         BEGIN
;  1687          2665  4         DO
;  1688          2666  4             INDX_ADDR = BBLOCK [ BFRD_ADDR (.(.INDX_ADDR)<0,16>), BFRD$W_NXTBFRD]
;  1689          2667  4         UNTIL .(.INDX_ADDR)<0,16> EQL .INDX;
;  1690          2668  4
;  1691          2669  4         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD];
;  1692          2670  3         END;
;  1693          2671  2     END;
```

```
;  1694          2672  2
;  1695          2673  2 ! If it had a buffer lock hooked up with it, adjust the reference count
;  1696          2674  2 ! on the lock descriptor, and remove the lock descriptor from its hash
;  1697          2675  2 ! lookup list if this is the last reference to it.
;  1698          2676  2 !
;  1699          2677  2
;  1700          2678  2 IF .BFRD [BFRD$W_BFRL] NEQ 0
;  1701          2679  2 THEN
;  1702          2680  3     BEGIN
;  1703          2681  3
;  1704          2682  3     BFRL = BFRLD_ADDR (.BFRD [BFRD$W_BFRL]);
;  1705          2683  3
;  1706          2684  3     BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] - 1;
;  1707          2685  3     IF .BFRL [BFRL$W_REFCNT] EQL 0
;  1708          2686  3     THEN
;  1709          2687  4         BEGIN
;  1710          2688  4
;  1711          2689  4         IF .BFRL [BFRL$L_LKID] NEQ 0
;  1712          2690  4         THEN
;  1713          2691  5             BEGIN
;  1714          2692  6             IF NOT $DEQ (LKID = .BFRL [BFRL$L_LKID])
;  1715          2693  5             THEN
;  1716          2694  5                 BUG_CHECK (XQPERR, 'unexpected lock manager reaction');
;  1717          2695  4             END;
;  1718          2696  4
;  1719          2697  4 ! Hash on the lockbasis,parlkid pair and scan the list to find the
;  1720          2698  4 ! desired bfrl, if necessary.  Pull it from the list when located.
;  1721          2699  4 !
;  1722          2700  4
;  1723        P 2701  4         INDX_ADDR = LOOKUP_LOCK (.BFRL [BFRL$L_LCKBASIS],
;  1724          2702  4                                  .BFRL [BFRL$L_PARLKID]);
;  1725          2703  4
;  1726          2704  4         IF .(.INDX_ADDR)<0,16> EQL .BFRD [BFRD$W_BFRL]
;  1727          2705  4         THEN
;  1728          2706  4             (.INDX_ADDR)<0,16> = .BFRL [BFRL$W_NXTBFRL]
;  1729          2707  4         ELSE
;  1730          2708  5             BEGIN
;  1731          2709  5
;  1732          2710  5             DO
;  1733          2711  5                 INDX_ADDR = BBLOCK [ BFRLD_ADDR (.(.INDX_ADDR)<0,16>), BFRL$W_NXTBFRL]
;  1734          2712  5             UNTIL .(.INDX_ADDR)<0,16> EQL .BFRD [BFRD$W_BFRL];
;  1735          2713  5
;  1736          2714  5             (.INDX_ADDR)<0,16> = .BFRL [BFRL$W_NXTBFRL];
;  1737          2715  4             END;
;  1738          2716  4
;  1739          2717  4         BFRL [BFRL$L_LKID] = 0;
;  1740          2718  4         BFRL [BFRL$L_LCKBASIS] = 0;
;  1741          2719  4         BFRL [BFRL$L_PARLKID] = 0;
;  1742          2720  4
;  1743          2721  4         BFRL [BFRL$W_NXTBFRL] = .CACHE_HDR [F11BC$W_FREEBFRL];
;  1744          2722  4         CACHE_HDR [F11BC$W_FREEBFRL] = .BFRD [BFRD$W_BFRL];
;  1745          2723  3         END;
;  1746          2724  3
;  1747          2725  3 ! There is no longer a BFRL associated with this BFRD.
;  1748          2726  3 !
;  1749          2727  3
;  1750          2728  3     BFRD [BFRD$W_BFRL] = 0;
```

```
;  1751          2729  3
;  1752          2730  2      END;
;  1753          2731  2
;  1754          2732  1 END;


                                                        .EXTRN  DEL_EXTFCB, NUKE_HEAD_FCB
                                                        .EXTRN  SYS$DEQ

                                                        .ENTRY  UNHOOK_BFRD, Save R2,R3,R4,R5,R6                ; 2558
                                        55      FC      MOVAB   -4(BASE), R5                                    ; 2572
                                        53      08      MOVL    BFRDARG, BFRD                                   ; 2586
                                        52      04      MOVW    INDXARG, INDX                                   ; 2587
                                04      18      A3      BBC     #2, 24(BFRD), 1$                                ; 2589
                                                        BUGW                                                    ; 2591
                                                        .WORD   <BUG$_XQPERR!4>
                                        18      A3      BICB2   #8, 24(BFRD)                                    ; 2593
                                                06      CMPB    25(BFRD), #6                                    ; 2595
                                                        BNEQ    5$
                                        54      08      MOVL    8(BFRD), DIRFCB                                 ; 2605
                                                        BEQL    5$
                                        07      0A      CMPB    10(DIRFCB), #7                                  ; 2609
                                                        BNEQ    2$
                                        00B0    C4      TSTL    176(DIRFCB)                                     ; 2610
                                                        BNEQ    3$
                                                        BUGW                                                    ; 2612
                                                        .WORD   <BUG$_XQPERR!4>
                                        00B0    C4      CLRL    176(DIRFCB)                                     ; 2618
                                          18    A4      TSTW    24(DIRFCB)                                      ; 2620
                                                        BNEQ    4$
                                0E      22      A4      BBCC    #0, 34(DIRFCB), 4$                              ; 2622
                                                        PUSHL   DIRFCB                                          ; 2638
                                        0000G   CF      CALLS   #1, DEL_EXTFCB
                                                        PUSHL   DIRFCB                                          ; 2639
                                        0000G   CF      CALLS   #1, NUKE_HEAD_FCB
                                            08  A3      CLRL    8(BFRD)                                         ; 2642
                                            08  A3      TSTL    8(BFRD)                                         ; 2648
                                                        BEQL    9$
                                        51              MOVL    (R5), R1                                        ; 2658
                                        50      14      MOVZWL  20(R1), R0
7E              00      08      A3                      EMUL    #1, 8(BFRD), #0, -(SP)
50              50              8E                      EDIV    R0, (SP)+, R0, R0
                                                        TSTL    R0
                                                        BGEQ    6$
                                        50              MNEGL   R0, R0
                                        54      10 B140 MOVAW   @16(R1)[R0], INDX_ADDR
                                        52              CMPW    (INDX_ADDR), INDX                               ; 2660
                                                        BEQL    8$
                                        50              MOVL    (R5), R0                                        ; 2666
                                        51              MOVZWL  (INDX_ADDR), R1
                                        51              MULL2   #32, R1
                                        51      18      ADDL2   24(R0), R1
                                        54      FE      MOVAB   -2(R1), INDX_ADDR
                                        52              CMPW    (INDX_ADDR), INDX                               ; 2667
                                                        BNEQ    7$
                                        64      1E      MOVW    30(BFRD), (INDX_ADDR)                           ; 2669
```

```
                                1A   A3  B5 00098 9$:     TSTW    26(BFRD)                                ; 2678
                                     01  12 0009B        BNEQ    10$
                                         04 0009D        RET
                                     51  65 D0 0009E 10$: MOVL    (R5), R1                                ; 2682
                                     50  1A  A3 3C 000A1 MOVZWL  26(BFRD), R0
                                     50      10 C4 000A5 MULL2   #16, R0
                                     50  1C  A1 C0 000A8 ADDL2   28(R1), R0
                                     52  F0  A0 9E 000AC MOVAB   -16(R0), BFRL
                                     02  A2  B7 000B0    DECW    2(BFRL)                                 ; 2684
                                         72  12 000B3    BNEQ    15$                                     ; 2685
                                     04  A2  D5 000B5    TSTL    4(BFRL)                                 ; 2689
                                         15  13 000B8    BEQL    11$
                                         7E  7C 000BA    CLRQ    -(SP)                                   ; 2692
                                         7E  D4 000BC    CLRL    -(SP)
                                     04  A2  DD 000BE    PUSHL   4(BFRL)
                          00000000G  00      04  FB 000C1 CALLS   #4, SYS$DEQ
                                     04      50  E8 000C8 BLBS    R0, 11$
                                               FEFF 000CB BUGW                                           ; 2694
                                              0000* 000CD .WORD   <BUG$_XQPERR!4>
                                     51      65  D0 000CF 11$:    MOVL    (R5), R1                        ; 2702
                    50       08      A2      0C  A2  C1 000D2    ADDL3   12(BFRL), 8(BFRL), R0
                                     56      24  A1  3C 000D8    MOVZWL  36(R1), R6
                7E       00          50          01  7A 000DC    EMUL    #1, R0, #0, -(SP)
                50       50          8E          56  7B 000E1    EDIV    R6, (SP)+, R0, R0
                                     50          D5 000E6        TSTL    R0
                                     03          18 000E8        BGEQ    12$
                                     50          CE 000EA        MNEGL   R0, R0
                                     54      20 B140 3E 000ED 12$:    MOVAW   @32(R1)[R0], INDX_ADDR          ;
                            1A       A3          64  B1 000F2        CMPW    (INDX_ADDR), 26(BFRD)           ; 2704
                                                17  13 000F6        BEQL    14$
                                     50          65  D0 000F8 13$:    MOVL    (R5), R0                        ; 2711
                                     51          64  3C 000FB        MOVZWL  (INDX_ADDR), R1
                                     51          10  C4 000FE        MULL2   #16, R1
                                     51      1C  A0  C0 00101        ADDL2   28(R0), R1
                                     54      F0  A1  9E 00105        MOVAB   -16(R1), INDX_ADDR
                                1A   A3          64  B1 00109        CMPW    (INDX_ADDR), 26(BFRD)           ; 2712
                                                E9  12 0010D        BNEQ    13$
                                     64          62  B0 0010F 14$:    MOVW    (BFRL), (INDX_ADDR)             ; 2714
                                         04      A2  7C 00112        CLRQ    4(BFRL)                         ; 2717
                                         0C      A2  D4 00115        CLRL    12(BFRL)                        ; 2719
                                     50          65  D0 00118        MOVL    (R5), R0                        ; 2721
                                     62      26  A0  B0 0011B        MOVW    38(R0), (BFRL)
                                     50          65  D0 0011F        MOVL    (R5), R0                        ; 2722
                                26   A0      1A  A3  B0 00122        MOVW    26(BFRD), 38(R0)
                                             1A  A?  B4 00127 15$:    CLRW    26(BFRD)                        ; 2728
                                                 04 0012A        RET                                       ; 2732

; Routine Size:  299 bytes,    Routine Base:  $CODE$ + 097D
```

```
;  1756          2733  1 ROUTINE FREE_ONE (POOL) : L_JSB NOVALUE =
;  1757          2734  1
;  1758          2735  1 !++
;  1759          2736  1 !
;  1760          2737  1 ! FUNCTIONAL DESCRIPTION:
;  1761          2738  1 !
;  1762          2739  1 ! This routine tosses a buffer off the in-process list to make room
;  1763          2740  1 ! for a new one.  This only happens when we already have used up
;  1764          2741  1 ! the minimum number of buffers for this pool, and there are not
;  1765          2742  1 ! enough available in the cache to steal more from it.
;  1766          2743  1 !
;  1767          2744  1 !--
;  1768          2745  1
;  1769          2746  2 BEGIN
;  1770          2747  2
;  1771          2748  2 BIND_COMMON;
;  1772          2749  2
;  1773          2750  2 LOCAL
;  1774          2751  2         INDX,
;  1775          2752  2         POOL_LRU        : REF BBLOCK,
;  1776          2753  2         FREED           : REF BBLOCK;
;  1777          2754  2
;  1778          2755  2 BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
;  1779          2756  2
;  1780          2757  2 REMQUE (.BFR_LIST [.POOL, QFLNK], FREED);
;  1781          2758  2
;  1782          2759  2 INDX = (.FREED - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
;  1783          2760  2
;  1784          2761  2 IF .FREED [BFRD$V_DIRTY]
;  1785          2762  2 THEN
;  1786          2763  3     BEGIN
;  1787          2764  3     RELEASE_CACHE ();
;  1788          2765  3     WRITE_BLOCK (.INDX*512 + .CACHE_HDR [F11BC$L_BUFBASE]);
;  1789          2766  3     SERIAL_CACHE ();
;  1790          2767  2     END;
;  1791          2768  2
;  1792          2769  2 ! Insert at front for now to recycle immediately.
;  1793          2770  2 !
;  1794          2771  2
;  1795          2772  2 POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
;  1796          2773  2 INSQUE (.FREED, POOL_LRU [QFLNK]);
;  1797          2774  2 FREED [BFRD$W_CURPID] = 0;
;  1798          2775  2
;  1799          2776  1 END;


                                0C  BB 00000 FREE_ONE:
                                                 PUSHR   #^M<R2,R3>                      ;  2733
                50      0C   AE  D0 00002        MOVL    POOL, R0                        ;  2755
                        F4 AA40  B7 00006        DECW    -12(BASE)[R0]
                50      0C   AE  D0 0000A        MOVL    POOL, R0                        ;  2757
                51      CC AA40  7E 0000E        MOVAQ   -52(BASE)[R0], R1
                52      00   B1  0F 00013        REMQUE  @0(R1), FREED
                50      FC   AA  D0 00017        MOVL    -4(BASE), R0                    ;  2759
```

```
                50              52      18   A0  C3 0001B          SUBL3   24(R0), FREED, R0                       ;
                53              50           20  C7 00020          DIVL3   #32, R0, INDX                          ;
                16      18      A2           02  E1 00024          BBC     #2, 24(FREED), 1$                      ; 2761
                                          FC0D  30 00029          BSBW    RELEASE_CACHE                          ; 2764
                53              53           09  78 0002C          ASHL    #9, R3, R3                             ; 2765
                                50       FC  BA  D0 00030          MOVL    @-4(BASE), R0                          ;
                                          6043  9F 00034          PUSHAB  (R0)[R3]                               ;
                        FCDA    CF           01  FB 00037          CALLS   #1, WRITE_BLOCK                        ;
                                          F79F  30 0003C          BSBW    SERIAL_CACHE                           ; 2766
                                50       0C  AE  D0 0003F 1$:     MOVL    POOL, R0                               ; 2772
                                50       FC BA40 7E 00043          MOVAQ   @-4(BASE)[R0], POOL_LRU                ;
                                50           28  C0 00048          ADDL2   #40, POOL_LRU                          ;
                                60           62  0E 0004B          INSQUE  (FREED), (POOL_LRU)                    ; 2773
                                          1C  A2  B4 0004E          CLRW    28(FREED)                              ; 2774
                                              0C  BA 00051          POPR    #^M<R2,R3>                             ; 2776
                                              05  00053          RSB                                            ;

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 0AA8
```

```
;  1801          2777  1 GLOBAL ROUTINE RELEASE_LOCKBASIS (LCKINDX) : L_NORM =
;  1802          2778  1
;  1803          2779  1 !++
;  1804          2780  1 !
;  1805          2781  1 ! FUNCTIONAL DESCRIPTION:
;  1806          2782  1 !
;  1807          2783  1 ! This routine scans in-process buffers and prepares them for lowering
;  1808          2784  1 ! the synchronization lock associated with LOCKBASIS.
;  1809          2785  1 ! All dirty buffers associated with the given LOCKBASIS must have been
;  1810          2786  1 ! written to disk previously.  This routine runs under the cache serialization
;  1811          2787  1 ! interlock, and if a conversion is required, will exit without releasing
;  1812          2788  1 ! the cache serialization interlock.
;  1813          2789  1 !
;  1814          2790  1 !--
;  1815          2791  1
;  1816          2792  2 BEGIN
;  1817          2793  2
;  1818          2794  2 BIND_COMMON;
;  1819          2795  2
;  1820          2796  2 EXTERNAL
;  1821          2797  2         CLU$GL_CLUB     : ADDRESSING_MODE (GENERAL);
;  1822          2798  2
;  1823          2799  2 LOCAL
;  1824          2800  2         LOCKBASIS,
;  1825          2801  2         LKID_ADDR,
;  1826          2802  2         INDX;
;  1827          2803  2
;  1828          2804  2 LABEL
;  1829          2805  2         FIXBFRL,
;  1830          2806  2         LBMATCH;
;  1831          2807  2
;  1832          2808  2 LKID_ADDR = 0;
;  1833          2809  2 INDX = 0;
;  1834          2810  2 LOCKBASIS = .LB_BASIS [.LCKINDX];
;  1835          2811  2
;  1836          2812  2 SERIAL_CACHE ();
;  1837          2813  2
;  1838          2814  2 INCR POOL FROM 0 TO 3
;  1839          2815  2 DO
;  1840          2816  3     BEGIN
;  1841          2817  3     LOCAL
;  1842          2818  3         I,
;  1843          2819  3         QHEAD   : REF BBLOCK,
;  1844          2820  3         BFRL    : REF BBLOCK,
;  1845          2821  3         BFRD    : REF BBLOCK;
;  1846          2822  3
;  1847          2823  3     QHEAD = BFR_LIST [.POOL, QFLNK];
;  1848          2824  3     BFRD = .QHEAD [QFLNK];
;  1849          2825  3
;  1850          2826  3     I = .BFRS_USED [.POOL];
;  1851          2827  3
;  1852          2828  3     UNTIL .I EQL 0
;  1853          2829  3     DO
;  1854          2830  4         BEGIN
;  1855          2831  4         LOCAL
;  1856          2832  4             NXTBFRD;
;  1857          2833  4
```

```
;1858     2834  4          I = .I - 1;
;1859     2835  4          NXTBFRD = .BFRD [BFRD$L_QFL];
;1860     2836  4
;1861     2837  4 ! Check this BFRD for LOCKBASIS match.  Note that we are scanning an
;1862     2838  4 ! in-process list which can only contain buffers from a given volume
;1863     2839  4 ! or volume set.  Because the LOCKBASIS contains the RVN, it is not
;1864     2840  4 ! necessary to make a separate check for a volume set match.
;1865     2841  4 ! When the buffer was initially brought into the in-process list, it
;1866     2842  4 ! matched correctly on the desired UCB.
;1867     2843  4 !
;1868     2844  4
;1869     2845  4          IF .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS
;1870     2846  4          THEN
;1871     2847  5 LBMATCH:     BEGIN
;1872     2848  5
;1873     2849  5              LOCAL
;1874     2850  5                  POOL_LRU : REF BBLOCK;
;1875     2851  5
;1876     2852  5              IF NOT .BFRD [BFRD$V_VALID]
;1877     2853  5              THEN
;1878     2854  6                  BEGIN
;1879     2855  6                  LOCAL
;1880     2856  6                      INDX0;
;1881     2857  6
;1882     2858  6                  INDX0 = (.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
;1883     2859  6
;1884     2860  6                  UNHOOK_BFRD (.INDX0 + 1, .BFRD);
;1885     2861  6
;1886     2862  6                  RETURN_BFRD (.BFRD);
;1887     2863  6
;1888     2864  6                  LEAVE LBMATCH;
;1889     2865  5                  END;
;1890     2866  5
;1891     2867  5 ! This is a valid buffer to be returned to the cache.
;1892     2868  5 !
;1893     2869  5
;1894     2870  6 FIXBFRL:         BEGIN
;1895     2871  6
;1896     2872  6                  LOCAL
;1897     2873  6                      LBINDX_ADDR,
;1898     2874  6                      PARLKID;
;1899     2875  6
;1900     2876  6                  IF .BFRD [BFRD$V_DIRTY]
;1901     2877  6                  THEN
;1902     2878  6                      BUG_CHECK (XQPERR, 'should not have dirty buffers now');
;1903     2879  6
;1904     2880  7                  PARLKID = .(if .CURRENT_VCB [VCB$W_RVN] EQL 0
;1905     2881  7                              THEN CURRENT_VCB [VCB$L_VOLLKID]
;1906     2882  6                              ELSE CURRENT_RVT [RVT$L_STRUCLKID]);
;1907     2883  6
;1908     2884  6                  IF .BFRD [BFRD$W_BFRL] NEQ 0
;1909     2885  6                  THEN
;1910     2886  6
;1911     2887  6 ! We're already hooked up to a BFRL.  This means we found it already
;1912     2888  6 ! in the cache in the first place.  Nothing needs to be done with
;1913     2889  6 ! the BFRL.  However, let's be paranoid and make sure that it is
;1914     2890  6 ! hooked up to the correct lock.
```

```
;  1915          2891  6 !
;  1916          2892  6
;  1917          2893  7                     BEGIN
;  1918          2894  7                     BFRL = BFRLD_ADDR (.BFRD [BFRD$W_BFRL]);
;  1919          2895  7
;  1920          2896  7                     IF .BFRL [BFRL$L_LCKBASIS] NEQ .LOCKBASIS
;  1921          2897  7                         OR .BFRL [BFRL$L_PARLKID] NEQ .PARLKID
;  1922          2898  7                     THEN
;  1923          2899  7                         BUG_CHECK (XQPERR, 'wrong lockbasis');
;  1924          2900  7
;  1925          2901  7                     LEAVE FIXBFRL;
;  1926          2902  7                     END
;  1927          2903  6                 ELSE IF .INDX NEQ 0
;  1928          2904  6                 THEN
;  1929          2905  6
;  1930          2906  6 ! This BFRD is not hooked up with a BFRL yet, but we've already
;  1931          2907  6 ! got a BFRL in hand (from an earlier hit) so just hook it up
;  1932          2908  6 ! and bump the REFCNT in the BFRL.
;  1933          2909  6 !
;  1934          2910  7                     BEGIN
;  1935          2911  7
;  1936          2912  7                     BFRD [BFRD$W_BFRL] = .INDX;
;  1937          2913  7                     BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] + 1;
;  1938          2914  7
;  1939          2915  7                     LEAVE FIXBFRL;
;  1940          2916  6                     END;
;  1941          2917  6
;  1942          2918  6 ! There is no BFRL for this BFRD yet, so let's first see if a BFRL
;  1943          2919  6 ! for this LOCKBASIS is already in the cache.
;  1944          2920  6 !
;  1945          2921  6
;  1946          2922  6                 LBINDX_ADDR = LOOKUP_LOCK (.LOCKBASIS, .PARLKID);
;  1947          2923  6                 INDX = .(.LBINDX_ADDR)<0,16>;
;  1948          2924  6
;  1949          2925  6                 WHILE .INDX NEQ 0
;  1950          2926  6                 DO
;  1951          2927  7                     BEGIN
;  1952          2928  7
;  1953          2929  7                     BFRL = BFRLD_ADDR (.INDX);
;  1954          2930  7
;  1955          2931  7                     IF .BFRL [BFRL$L_LCKBASIS] EQL .LOCKBASIS
;  1956          2932  7                         AND .BFRL [BFRL$L_PARLKID] EQL .PARLKID
;  1957          2933  7                     THEN
;  1958          2934  7                         EXITLOOP;
;  1959          2935  7
;  1960          2936  7                     INDX = .BFRL [BFRL$W_NXTBFRL];
;  1961          2937  6                     END;
;  1962          2938  6
;  1963          2939  6                 IF .INDX EQL 0
;  1964          2940  6                 THEN
;  1965          2941  6
;  1966          2942  6 ! We did not find a BFRL in the cache already  so take one off
;  1967          2943  6 ! the free list and use it.
;  1968          2944  6 !
;  1969          2945  7                     BEGIN
;  1970          2946  7
;  1971          2947  7                     IF (INDX = .CACHE_HDR [F11BC$W_FREEBFRL]) EQL 0
```

```
;1972     2948  7                       THEN
;1973     2949  7                           BUG_CHECK (XQPERR, 'no free lock block');
;1974     2950  7
;1975     2951  7 ! Take our new BFRL out of the free BFRL list.
;1976     2952  7 ! Link it into the front of the hash chain.
;1977     2953  7 ! Fill in the lockbasis and parent lock ID fields.
;1978     2954  7 !
;1979     2955  7
;1980     2956  7                       BFRL = BFRLD_ADDR (.INDX);
;1981     2957  7                       CACHE_HDR [FT1BC$W_FREEBFRL] = .BFRL [BFRL$W_NXTBFRL];
;1982     2958  7                       BFRL [BFRL$W_NXTBFRL] = .(.LBINDX_ADDR)<0,16>;
;1983     2959  7                       (.LBINDX_ADDR)<0,16> = .INDX;
;1984     2960  7
;1985     2961  7                       BFRL [BFRL$L_LCKBASIS] = .LOCKBASIS;
;1986     2962  7                       BFRL [BFRL$L_PARLKID] = .PARLKID;
;1987     2963  7
;1988     2964  7 ! Test for LOCKBASIS of -1.  -1 is used for all buffers backed by the
;1989     2965  7 ! volume allocation lock.  Because there is a volume allocation lock
;1990     2966  7 ! associated with the VCB at all times, there is no need to separately
;1991     2967  7 ! keep a lock on those buffers in the cache.  However, we will keep
;1992     2968  7 ! a BFRL around to avoid a lot of special casing, but it's LKID field
;1993     2969  7 ! will be zero.
;1994     2970  7 !
;1995     2971  7
;1996     2972  7                       IF (.LOCKBASIS + 1) NEQ 0
;1997     2973  7                           AND .BBLOCK [CURRENT_UCB [UCB$L_DEVCHAR2], DEV$V_CLU]
;1998     2974  7                           AND .CLU$GL_CLUB NEQ 0
;1999     2975  7                       THEN
;2000     2976  7                           LKID_ADDR = BFRL [BFRL$L_LKID];
;2001     2977  7
;2002     2978  6                       END;
;2003     2979  6
;2004     2980  6 ! Fill in the BFRD field that points to the BFRL.  Bump the REFCNT
;2005     2981  6 ! in the BFRL.
;2006     2982  6 !
;2007     2983  6
;2008     2984  6                   BFRD [BFRD$W_BFRL] = .INDX;
;2009     2985  6                   BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] + 1;
;2010     2986  6
;2011     2987  5                   END;                        ! of FIXBFRL block
;2012     2988  5
;2013     2989  5 ! Store the current sequence number from the backing lock for this buffer.
;2014     2990  5 ! This must be done here because multiple buffers may be backed by a
;2015     2991  5 ! single lock.  If one of them has been modified, the sequence number
;2016     2992  5 ! for that lock will have changed, yet unmodified buffers backed by
;2017     2993  5 ! the same do not have the same sequence number, even though they
;2018     2994  5 ! are still valid.  Hence, stamp all buffers with the current sequence
;2019     2995  5 ! number.
;2020     2996  5
;2021     2997  5               CASE .BFRD [BFRD$B_BTYPE] FROM 0 TO 6 OF
;2022     2998  5               SET
;2023     2999  5               [HEADER_TYPE]:
;2024     3000  5                   BFRD [BFRD$L_SEQNUM] = .LB_HDRSEQ [.LCKINDX];
;2025     3001  5
;2026     3002  5               [DIRECTORY_TYPE, DATA_TYPE]:
;2027     3003  5                   BFRD [BFRD$L_SEQNUM] = .LB_DATASEQ [.LCKINDX];
;2028     3004  5
```

```
;  2029     3005  5              [DIRINDX_TYPE]:
;  2030     3006  6                  BEGIN
;  2031     3007  6                  LOCAL
;  2032     3008  6                      DIRINDX     : REF BBLOCK FIELD (DIRC);
;  2033     3009  6
;  2034     3010  6                  BFRD [BFRD$L_SEQNUM] = .LB_HDRSEQ [.LCKINDX];
;  2035     3011  6                  DIRINDX = .CACHE_HDR [F11BC$L_BUFBASE]
;  2036     3012  6                          + 512*(.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
;  2037     3013  6                  DIRINDX [DIRC$L_DATASEQ] = .LB_DATASEQ [.LCKINDX];
;  2038     3014  5                  END;
;  2039     3015  5
;  2040     3016  5              [BITMAP_TYPE]:
;  2041     3017  5                  BFRD [BFRD$L_SEQNUM] = .(LB_DATASEQ [0])<0,16,0>;
;  2042     3018  5
;  2043     3019  5              [INDEX_TYPE]:
;  2044     3020  5                  BFRD [BFRD$L_SEQNUM] = .(LB_DATASEQ [0])<16,16,0>;
;  2045     3021  5
;  2046     3022  5              [QUOTA_TYPE]:
;  2047     3023  5                  BFRD [BFRD$L_SEQNUM] = .SAVE_VC_FLAGS<1,15,0>;
;  2048     3024  5
;  2049     3025  5              TES;
;  2050     3026  5
;  2051     3027  5
;  2052     3028  5  ! Take the buffer off of the in-process queue and put it back
;  2053     3029  5  ! on the appropriate pool LRU list in the cache.
;  2054     3030  5  ! Clear out CURPID in the BFRD to note we aren't using it anymore.
;  2055     3031  5  !
;  2056     3032  5
;  2057     3033  5              REMQUE (.BFRD, BFRD);
;  2058     3034  5              BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
;  2059     3035  5              BFRD [BFRD$W_CURPID] = 0;
;  2060     3036  5              POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
;  2061     3037  5              INSQUE (.BFRD, .POOL_LRU [QBLNK]);
;  2062     3038  4              END;                    ! of LOCKBASIS match.
;  2063     3039  4
;  2064     3040  4          BFRD = .NXTBFRD;
;  2065     3041  4
;  2066     3042  3          END;              ! of loop through pool bfrds.
;  2067     3043  3
;  2068     3044  3      IF .QHEAD NEQA .BFRD
;  2069     3045  3      THEN
;  2070     3046  3          BUG_CHECK (XQPERR, 'screwed up in-process list');
;  2071     3047  3
;  2072     3048  2      END;         ! of loop through all pools
;  2073     3049  2
;  2074     3050  2  IF .LKID_ADDR EQL 0
;  2075     3051  2  THEN
;  2076     3052  2      RELEASE_CACHE ();
;  2077     3053  2
;  2078     3054  2  RETURN .LKID_ADDR
;  2079     3055  2
;  2080     3056  1  END;
```

```
                              0BFC 00000          .ENTRY  RELEASE_LOCKBASIS, Save R2,R3,R4,R5,R6,R7,- ; 2777
                                                          R8,R9,R11                                   ;
                5E         10   C2 00002          SUBL2   #16, SP                                     ;
                57    FC   AA   9E 00005          MOVAB   -4(BASE), R7                                ; 2792
                5B  00A8   CA   9E 00009          MOVAB   168(BASE), R11                              ;
                      0C   AE   D4 0000E          CLRL    LKID_ADDR                                   ; 2808
                           56   D4 00011          CLRL    INDX                                        ; 2809
                50    04   AC   D0 00013          MOVL    LCKINDX, R0                                 ; 2810
                59  0080 CA40   D0 00017          MOVL    128(BASE)[R0], LOCKBASIS                    ;
                         F76A   30 0001D          BSBW    SERIAL_CACHE                                ; 2812
                           53   D4 00020          CLRL    POOL                                        ; 2814
          08    AE    CC AA43   7E 00022 1$:      MOVAQ   -52(BASE)[POOL], QHEAD                      ; 2823
                54    08   BE   D0 00028          MOVL    @QHEAD, BFRD                                ; 2824
          04    AE    F4 AA43   3C 0002C          MOVZWL  -12(BASE)[POOL], I                          ; 2826
                      04   AE   D5 00032 2$:      TSTL    I                                           ; 2828
                           03   12 00035          BNEQ    3$                                          ;
                         019A   31 00037          BRW     29$                                         ;
                      04   AE   D7 0003A 3$:      DECL    I                                           ; 2834
                6E         64   D0 0003D          MOVL    (BFRD), NXTBFRD                             ; 2835
                59    10   A4   D1 00040          CMPL    16(BFRD), LOCKBASIS                         ; 2845
                           20   12 00044          BNEQ    4$                                          ;
    1E    18    A4         03   E0 00046          BBS     #3, 24(BFRD), 5$                            ; 2852
                50         67   D0 0004B          MOVL    (R7), R0                                    ; 2858
    50          54    18   A0   C3 0004E          SUBL3   24(R0), BFRD, R0                            ;
                50         20   C6 00053          DIVL2   #32, INDX0                                  ;
                           54   DD 00056          PUSHL   BFRD                                        ; 2860
                      01   A0   9F 00058          PUSHAB  1(INDX0)                                    ;
        FE21    CF         02   FB 0005B          CALLS   #2, UNHOOK_BFRD                             ;
                50         54   D0 00060          MOVL    BFRD, R0                                    ; 2862
                        0000V   30 00063          BSBW    RETURN_BFRD                                 ;
                         0165   31 00066 4$:      BRW     28$                                         ; 2864
    04    18    A4         02   E1 00069 5$:      BBC     #2, 24(BFRD), 6$                            ; 2876
                              FEFF 0006E          BUGW                                                ; 2878
                             0000* 00070          .WORD   <BUG$_XQPERR!4>                             ;
                50    98   AA   D0 00072 6$:      MOVL    -104(BASE), R0                              ; 2880
                      0E   A0   B5 00076          TSTW    14(R0)                                      ;
                           06   12 00079          BNEQ    7$                                          ;
                50    7C   A0   9E 0007B          MOVAB   124(R0), R0                                 ; 2881
                           04   11 0007F          BRB     8$                                          ;
                50    9C   AA   D0 00081 7$:      MOVL    -100(BASE), R0                              ; 2882
                55         60   D0 00085 8$:      MOVL    (R0), PARLKID                               ; 2880
                      1A   A4   B5 00088          TSTW    26(BFRD)                                    ; 2884
                           25   13 0008B          BEQL    11$                                         ;
                50         67   D0 0008D          MOVL    (R7), R0                                    ; 2894
                51    1A   A4   3C 00090          MOVZWL  26(BFRD), R1                                ;
                51         10   C4 00094          MULL2   #16, R1                                     ;
                51    1C   A0   C0 00097          ADDL2   28(R0), R1                                  ;
                52    F0   A1   9E 0009B          MOVAB   -16(R1), BFRL                               ;
                59    08   A2   D1 0009F          CMPL    8(BFRL), LOCKBASIS                          ; 2896
                           06   12 000A3          BNEQ    9$                                          ;
                55    0C   A2   D1 000A5          CMPL    12(BFRL), PARLKID                           ; 2897
                           04   13 000A9          BEQL    10$                                         ;
                              FEFF 000AB 9$:      BUGW                                                ; 2899
                             0000* 000AD          .WORD   <BUG$_XQPERR!4>                             ;
                         00A1   31 000AF 10$:     BRW     19$                                         ; 2901
                           56   D5 000B2 11$:     TSTL    INDX                                        ; 2903
                           48   12 000B4          BNEQ    16$                                         ;
```

```
                                 51            67  D0 000B6          MOVL    (R7), R1                                        ; 2922
                              50 59            55  C1 000B9          ADDL3   PARLKID, LOCKBASIS, R0                          ;
                                 58        24  A1  3C 000BD          MOVZWL  36(R1), R8                                      ;
                           7E    50            01  7A 000C1          EMUL    #1, R0, #0, -(SP)                               ;
                           50    50            8E  7B 000C6          EDIV    R8, (SP)+, R0, R0                               ;
                                               50  D5 000CB          TSTL    R0                                              ;
                                               03  18 000CD          BGEQ    12$                                             ;
                                 50            50  CE 000CF          MNEGL   R0, R0                                          ;
                                 58    20 B140  3E  00D2 12$:        MOVAW   @32(R1)[R0], LBINDX_ADDR                        ;
                                 56            68  3C 000D7          MOVZWL  (LBINDX_ADDR), INDX                             ; 2923
                                               20  13 000DA 13$:      BEQL    15$                                             ; 2925
                                 50            67  D0 000DC          MOVL    (R7), R0                                        ; 2929
                              51 56            04  78 000DF          ASHL    #4, INDX, R1                                    ;
                                 51        1C  A0  C0 000E3          ADDL2   28(R0), R1                                      ;
                                 52        F0  A1  9E 000E7          MOVAB   -16(R1), BFRL                                   ;
                                 59        08  A2  D1 000EB          CMPL    8(BFRL), LOCKBASIS                              ; 2931
                                               06  12 000EF          BNEQ    14$                                             ;
                                 55        0C  A2  D1 000F1          CMPL    12(BFRL), PARLKID                               ; 2932
                                               05  13 000F5          BEQL    15$                                             ;
                                 56            62  3C 000F7 14$:      MOVZWL  (BFRL), INDX                                    ; 2936
                                               DE  11 000FA          BRB     13$                                             ; 2925
                                               56  D5 000FC 15$:      TSTL    INDX                                            ; 2939
                                               4C  12 000FE 16$:      BNEQ    18$                                             ;
                                 50            67  D0 00100          MOVL    (R7), R0                                        ; 2947
                                 56        26  A0  3C 00103          MOVZWL  38(R0), INDX                                    ;
                                               04  12 00107          BNEQ    17$                                             ;
                                                 FEFF 00109          BUGW                                                    ; 2949
                                                0000* 0010B          .WORD   <BUG$_XQPERR!4>                                 ;
                                 50            67  D0 0010D 17$:      MOVL    (R7), R0                                        ; 2956
                              51 56            04  78 00110          ASHL    #4, INDX, R1                                    ;
                                 51        1C  A0  C0 00114          ADDL2   28(R0), R1                                      ;
                                 52        F0  A1  9E 00118          MOVAB   -16(R1), BFRL                                   ;
                                 51            67  D0 0011C          MOVL    (R7), R1                                        ; 2957
                              26 A1            62  B0 0011F          MOVW    (BFRL), 38(R1)                                  ;
                                 62            68  B0 00123          MOVW    (LBINDX_ADDR), (BFRL)                           ; 2958
                                 68            56  B0 00126          MOVW    INDX, (LBINDX_ADDR)                             ; 2959
                              08 A2            59  D0 00129          MOVL    LOCKBASIS, 8(BFRL)                              ; 2961
                              0C A2            55  D0 0012D          MOVL    PARLKID, 12(BFRL)                               ; 2962
                                 50        01  A9  9E 00131          MOVAB   1(R9), R0                                       ; 2972
                                               15  13 00135          BEQL    18$                                             ;
                                 50        94  AA  D0 00137          MOVL    -108(BASE), R0                                  ; 2973
                                 0D        3C  A0  E9 0013B          BLBC    60(R0), 18$                                     ;
                                    00000000G  00  D5 0013F          TSTL    CLU$GL_CLUB                                     ; 2974
                                               05  13 00145          BEQL    18$                                             ;
                              0C AE            04  A2 9E 00147          MOVAB   4(R2), LKID_ADDR                                ; 2976
                              1A A4                56  B0 0014C 18$:      MOVW    INDX, 26(BFRD)                              ; 2984
                                               02  A2  B6 00150          INCW    2(BFRL)                                         ; 2985
                                 51            14  A4 9E 00153 19$:      MOVAB   20(BFRD), R1                                    ; 3000
                        06                     00  19  A4 8F 00157          CASEB   25(BFRD), #0, #6                                ; 2997
                0050          001A                 004B          000E      0015C 20$:      .WORD   21$-20$,-                       ;
                              0024                 0056          001A      00164           24$-20$,-                       ;
                                                                                           22$-20$,-                       ;
                                                                                           25$-20$,-                       ;
                                                                                           22$-20$,-                       ;
                                                                                           26$-20$,-                       ;
                                                                                           23$-20$                         ;
                                 50            04  AC  D0 0016A 21$:      MOVL    LCKINDX, R0                                     ; 3000
```

```
                                61      0094 CA40   D0 0016E          MOVL    148(BASE)[R0], (R1)                                     ;
                                          42   11 00174             BRB     27$                                                     ;
                                50        04   AC   D0 00176 22$:    MOVL    LCKINDX, R0                                             ; 3003
                                61           6B40 D0 0017A          MOVL    (R11)[R0], (R1)                                         ;
                                             38   11 0017E          BRB     27$                                                     ;
                                50        04   AC   D0 00180 23$:    MOVL    LCKINDX, R0                                             ; 3010
                                61      0094 CA40   D0 00184          MOVL    148(BASE)[R0], (R1)                                     ;
                                50           67   D0 0018A          MOVL    (R7), R0                                                ; 3011
                          51                 54     18 A0 C3 0018D    SUBL3   24(R0), BFRD, R1                                        ; 3012
                          51                 51        09 78 00192    ASHL    #9, R1, R1                                              ;
                                             51        20 C6 00196    DIVL2   #32, R1                                                 ;
                                             51        60 C0 00199    ADDL2   (R0), DIRINDX                                           ;
                                50        04   AC   D0 0019C          MOVL    LCKINDX, R0                                             ; 3013
                             08 A1           6B40 D0 001A0          MOVL    (R11)[R0], 8(DIRINDX)                                   ;
                                             11   11 001A5          BRB     27$                                                     ; 2997
                                61           6B   3C 001A7 24$:    MOVZWL  (R11), (R1)                                             ; 3017
                                             0C   11 001AA          BRB     27$                                                     ;
                                61        02   AB   3C 001AC 25$:    MOVZWL  2(R11), (R1)                                            ; 3020
                                             06   11 001B0          BRB     27$                                                     ;
                61        A4 AA              0F     01 EF 001B2 26$:    EXTZV   #1, #15, -92(BASE), (R1)                                ; 3023
                                54           64   0F 001B8 27$:    REMQUE  (BFRD), BFRD                                            ; 3033
                                          F4 AA43   B7 001BB          DECW    -12(BASE)[POOL]                                         ; 3034
                                          1C   A4   B4 001BF          CLRW    28(BFRD)                                                ; 3035
                                50        00 B743   7E 001C2          MOVAQ   @0(R7)[POOL], POOL_LRU                                  ; 3036
                                50           28   C0 001C7          ADDL2   #40, POOL_LRU                                           ;
                             04 B0           64   0E 001CA          INSQUE  (BFRD), @4(POOL_LRU)                                    ; 3037
                                54           6E   D0 001CE 28$:    MOVL    NXTBFRD, BFRD                                           ; 3040
                                          FE5E   31 001D1          BRW     2$                                                      ; 2828
                                54        08   AE   D1 001D4 29$:    CMPL    QHEAD, BFRD                                             ; 3044
                                             04   13 001D8          BEQL    30$                                                     ;
                                               FEFF 001DA          BUGW                                                            ; 3046
                                              0000* 001DC          .WORD   <BUG$_XQPERR!4>                                         ;
          FE3E                  53        01     03   F1 001DE 30$:    ACBL    #3, #T, POOL, 1$                                        ; 2814
                                          0C   AE   D5 001E4          TSTL    LKID_ADDR                                               ; 3050
                                             03   12 001E7          BNEQ    31$                                                     ;
                                          F9F9   30 001E9          BSBW    RELEASE_CACHE                                           ; 3052
                                50        0C   AE   D0 001EC 31$:    MOVL    LKID_ADDR, R0                                           ; 3054
                                             04   00 001F0          RET                                                             ; 3056

; Routine Size:  497 bytes,    Routine Base:  $CODE$ + 0AFC
```

```
;2082     3057  1 GLOBAL ROUTINE RESET_LBN (BUFFER, LBN) . L_NORM NOVALUE =
;2083     3058  1
;2084     3059  1 !++
;2085     3060  1 !
;2086     3061  1 ! FUNCTIONAL DESCRIPTION:
;2087     3062  1 !
;2088     3063  1 !       This routine changes the resident LBN of the indicated block
;2089     3064  1 !       and marks it as dirty and valid.
;2090     3065  1 !       It is assumed this will be followed soon by a buffer write
;2091     3066  1 !       operation for the new lbn.
;2092     3067  1 !
;2093     3068  1 ! CALLING SEQUENCE:
;2094     3069  1 !       RESET_LBN (ARG1, ARG2)
;2095     3070  1 !
;2096     3071  1 ! INPUT PARAMETERS:
;2097     3072  1 !       ARG1: address of block buffer
;2098     3073  1 !       ARG2: new LBN
;2099     3074  1 !
;2100     3075  1 ! IMPLICIT INPUTS:
;2101     3076  1 !       buffer descriptor arrays
;2102     3077  1 !
;2103     3078  1 ! OUTPUT PARAMETERS:
;2104     3079  1 !       NONE
;2105     3080  1 !
;2106     3081  1 ! IMPLICIT OUTPUTS:
;2107     3082  1 !       NONE
;2108     3083  1 !
;2109     3084  1 ! ROUTINE VALUE:
;2110     3085  1 !       NONE
;2111     3086  1 !
;2112     3087  1 ! SIDE EFFECTS:
;2113     3088  1 !       backing LBN for buffer altered
;2114     3089  1 !       buffer marked as dirty and valid.
;2115     3090  1 !
;2116     3091  1 !--
;2117     3092  1
;2118     3093  2 BEGIN
;2119     3094  2
;2120     3095  2 BIND_COMMON;
;2121     3096  2
;2122     3097  2 LOCAL
;2123     3098  2     INDX,
;2124     3099  2     BFRD        : REF BBLOCK,
;2125     3100  2     INDX_ADDR;
;2126     3101  2
;2127     3102  2 INDX = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512 + 1;
;2128     3103  2
;2129     3104  2 BFRD = BFRD_ADDR (.INDX);
;2130     3105  2
;2131     3106  2 SERIAL_CACHE ();
;2132     3107  2
;2133     3108  2 IF .BFRD [BFRD$L_LBN] NEQ 0
;2134     3109  2 THEN
;2135     3110  2
;2136     3111  2 ! BFRD$L_LBN neq 0 means that this buffer is in the hash lookup list.
;2137     3112  2 ! We need to remove it from the list it is currently
;2138     3113  2 ! in prior to resetting the LBN field.
```

```
;  2139          3114  2 !
;  2140          3115  2
;  2141          3116  2     BEGIN
;  2142          3117  2
;  2143          3118  2     INDX_ADDR = LOOKUP_LBN (.BFRD [BFRD$L_LBN]);
;  2144          3119  2
;  2145          3120  2     IF .(.INDX_ADDR)<0,16> EQL .INDX
;  2146          3121  2     THEN
;  2147          3122  2         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD]
;  2148          3123  3     ELSE
;  2149          3124  4         BEGIN
;  2150          3125  4         DO
;  2151          3126  4             INDX_ADDR = BBLOCK [ BFRD_ADDR (.(.INDX_ADDR)<0,16>), BFRD$W_NXTBFRD]
;  2152          3127  4         UNTIL .(.INDX_ADDR)<0,16> EQL .INDX;
;  2153          3128  4
;  2154          3129  4         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD];
;  2155          3130  3         END;
;  2156          3131  2     END;
;  2157          3132  2
;  2158          3133  2 ! Store the desired LBN and hook it up in the LBN hash list.
;  2159          3134  2 !
;  2160          3135  2
;  2161          3136  2 BFRD [BFRD$L_LBN] = .LBN;
;  2162          3137  2
;  2163          3138  2 INDX_ADDR = LOOKUP_LBN (.LBN);
;  2164          3139  2 BFRD [BFRD$W_NXTBFRD] = .(.INDX_ADDR)<0,16>,
;  2165          3140  2 (.INDX_ADDR)<0,16> = .INDX;
;  2166          3141  2
;  2167          3142  2 BFRD [BFRD$V_VALID] = 1;
;  2168          3143  2 BFRD [BFRD$V_DIRTY] = 1;
;  2169          3144  2
;  2170          3145  2 RELEASE_CACHE ();
;  2171          3146  2
;  2172          3147  1 END;                                    ! end of routine RESET_LBN


                                  003C 00000     .ENTRY  RESET_LBN, Save R2,R3,R4,R5                     ; 3057
                        55        FC   AA  9E 00002 MOVAB   -4(BASE), R5                                  ; 3093
                52      04        AC   00  B5  C3 00006 SUBL3   @0(R5), BUFFER, R2                        ; 3102
                        52 00000200 8F   C6  0000C DIVL2   #512, R2                                     ;
                        53        01   A2  9E 00013 MOVAB   1(R2), INDX                                 ;
                        50              65  D0 00017 MOVL    (R5), R0                                  ; 3104
                52      53              05  78 0001A ASHL    #5, INDX, R2                              ;
                        52        18    A0  C0 0001E ADDL2   24(R0), R2                                ;
                        52              20  C2 00022 SUBL2   #32, BFRD                                 ;
                                  F571  30  00025 BSBW    SERIAL_CACHE                                  ; 3106
                                  08    A2  D5 00028 TSTL    8(BFRD)                                   ; 3108
                                        41  13 0002B BEQL    4$                                        ;
                        51              65  D0 0002D MOVL    (R5), R1                                  ; 3118
                        50        14    A1  3C 00030 MOVZWL  20(R1), R0                                ;
      7E        00      08        A2      01  7A 00034 EMUL    #1, 8(BFRD), #0, -(SP)                    ;
      50        50              8E      50  7B 0003A EDIV    R0, (SP)+, R0, R0                         ;
                                        50  D5 0003F TSTL    R0                                        ;
                                        03  18 00041 BGEQ    1$                                        ;
```

```
                                50          50  CE 00043          MNEGL   R0, R0
                                50      10 B140 3E 00046 1$:      MOVAW   @16(R1)[R0], INDX_ADDR
            53          60      10          00 ED 0004B           CMPZV   #0, #16, (INDX_ADDR), INDX                      3120
                                            18 13 00050           BEQL    3$
                                51          65 D0 00052 2$:       MOVL    (R5), R1                                        3126
                                54          60 3C 00055           MOVZWL  (INDX_ADDR), R4
                                54          20 C4 00058           MULL2   #32, R4
                                54      18  A1 C0 0005B           ADDL2   24(R1), R4
                                50      FE  A4 9E 0005F           MOVAB   -2(R4), INDX_ADDR
            53          60      10          00 ED 00063           CMPZV   #0, #16, (INDX_ADDR), INDX                      3127
                                            E8 12 00068           BNEQ    2$
                                60      1E  A2 B0 0006A 3$:       MOVW    30(BFRD), (INDX_ADDR)                           3129
                        08      A2      08  AC D0 0006E 4$:       MOVL    LBN, 8(BFRD)                                    3135
                                54          65 D0 00073           MOVL    (R5), R4                                        3138
                                51      14  A4 3C 00076           MOVZWL  20(R4), R1
            7E          00      08      AC  01 7A 0007A           EMUL    #1, LBN, #0, -(SP)
            51          51              8E  51 7B 00080           EDIV    R1, (SP)+, R1, R1
                                            51 D5 00085           TSTL    R1
                                            03 18 00087           BGEQ    5$
                                51          51 CE 00089           MNEGL   R1, R1
                                50      10 B441 3E 0008C 5$:      MOVAW   @16(R4)[R1], INDX_ADDR
                        1E      A2          60 B0 00091           MOVW    (INDX_ADDR), 30(BFRD)                           3139
                                60          53 B0 00095           MOVW    INDX, (INDX_ADDR)                               3140
                        18      A2          0C 88 00098           BISB2   #12, 24(BFRD)                                   3143
                                        F955    30 0009C          BSBW    RELEASE_CACHE                                   3145
                                                04 0009F          RET                                                     3147

; Routine Size:  160 bytes,    Routine Base:  $CODE$ + 0CED
```

```
;  2174          3148  1 GLOBAL ROUTINE MARK_DIRTY (BUFFER) : L_NORM NOVALUE =
;  2175          3149  1
;  2176          3150  1 !++
;  2177          3151  1 !
;  2178          3152  1 ! FUNCTIONAL DESCRIPTION:
;  2179          3153  1 !
;  2180          3154  1 !       This routine marks the indicated buffer for write-back.
;  2181          3155  1 !
;  2182          3156  1 ! CALLING SEQUENCE:
;  2183          3157  1 !       MARK_DIRTY (ARG1)
;  2184          3158  1 !
;  2185          3159  1 ! INPUT PARAMETERS:
;  2186          3160  1 !       ARG1: address of block buffer
;  2187          3161  1 !
;  2188          3162  1 ! IMPLICIT INPUTS:
;  2189          3163  1 !       CURRENT_VCB: VCB of volume
;  2190          3164  1 !
;  2191          3165  1 ! OUTPUT PARAMETERS:
;  2192          3166  1 !       NONE
;  2193          3167  1 !
;  2194          3168  1 ! IMPLICIT OUTPUTS:
;  2195          3169  1 !       NONE
;  2196          3170  1 !
;  2197          3171  1 ! ROUTINE VALUE:
;  2198          3172  1 !       NONE
;  2199          3173  1 !
;  2200          3174  1 ! SIDE EFFECTS:
;  2201          3175  1 !       buffer marked for write-back                                    ; F
;  2202          3176  1 !
;  2203          3177  1 !--
;  2204          3178  1
;  2205          3179  2 BEGIN
;  2206          3180  2
;  2207          3181  2 BIND_COMMON;
;  2208          3182  2
;  2209          3183  2 LOCAL
;  2210          3184  2         BFRD    : REF BBLOCK,
;  2211          3185  2         INDX0;
;  2212          3186  2
;  2213          3187  2 INDX0 = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512;
;  2214          3188  2
;  2215          3189  2 IF .INDX0<0,16> GEQU .CACHE_HDR [F11BC$W_BFRCNT]
;  2216          3190  2 THEN
;  2217          3191  2     BUG_CHECK (BADBUFADR, 'buffer addr out of range');
;  2218          3192  2
;  2219          3193  2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .INDX0*BFRD$S_BFRDDEF;
;  2220          3194  2
;  2221          3195  2 IF NOT .BFRD [BFRD$V_VALID] OR .BFRD [BFRD$W_CURPID] EQL 0
;  2222          3196  2 THEN
;  2223          3197  2     BUG_CHECK (XQPERR, 'expect it to be valid and in-process');
;  2224          3198  2
;  2225          3199  2 BFRD [BFRD$V_DIRTY] = 1;
;  2226          3200  2
;  2227          3201  1 END;                                     ! end of routine MARK_DIRTY
```

```
                                                0000 00000          .ENTRY  MARK_DIRTY, Save nothing                         ; 3148
                50      04      AC      FC      BA   C3 00002          SUBL3   @-4(BASE), BUFFER, R0                       ; 3187
                                50 00000200     8F   C6 00008          DIVL2   #512, INDX0
                                51      FC      AA   D0 0000F          MOVL    -4(BASE), R1                              ; 3189
                        16      A1              50   B1 00013          CMPW    INDX0, 22(R1)
                                                04   1F 00017          BLSSU   1$
                                                  FEFF 00019          BUGW                                              ; 3191
                                                 0000* 0001B          .WORD   <BUG$_BADBUFADR!4>
                                51      FC      AA   D0 0001D 1$:      MOVL    -4(BASE), R1                              ; 3193
                                50              20   C4 00021          MULL2   #32, R0
                                50      18      A1   C0 00024          ADDL2   24(R1), BFRD
                05      18      A0              03   E1 00028          BBC     #3, 24(BFRD), 2$                          ; 3195
                                        1C      A0   B5 0002D          TSTW    28(BFRD)
                                                04   12 00030          BNEQ    3$
                                                  FEFF 00032 2$:      BUGW                                              ; 3197
                                                 0000* 00034          .WORD   <BUG$_XQPERR!4>
                        18      A0              04   88 00036 3$:      BISB2   #4, 24(BFRD)                              ; 3199
                                                    04 0003A          RET                                               ; 3201

; Routine Size:  59 bytes,    Routine Base:  $CODE$ + 0D8D
```

```
;  2229     3202  1 ROUTINE RETURN_BFRD (BFRD) : L_JSB_1ARG NOVALUE =
;  2230     3203  1
;  2231     3204  1 !++
;  2232     3205  1 !
;  2233     3206  1 ! This routine completes the return of the given BFRD to the appropriate
;  2234     3207  1 ! pool in the cache.  It expects that it has already been unhooked from
;  2235     3208  1 ! the hash lists and the BFRL taken care of.
;  2236     3209  1 !
;  2237     3210  1 !--
;  2238     3211  1
;  2239     3212  2 BEGIN                                                                    ; F
;  2240     3213  2
;  2241     3214  2 MAP
;  2242     3215  2     BFRD : REF BBLOCK;
;  2243     3216  2
;  2244     3217  2 BIND_COMMON;
;  2245     3218  2
;  2246     3219  2 LOCAL
;  2247     3220  2     POOL,
;  2248     3221  2     POOL_LRU    : REF BBLOCK;
;  2249     3222  2
;  2250     3223  2
;  2251     3224  2 POOL = .BFRD [BFRD$V_POOL];
;  2252     3225  2 POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
;  2253     3226  2
;  2254     3227  2 ! Credit our buffers in use count if this was on our in-process list.
;  2255     3228  2 !
;  2256     3229  2
;  2257     3230  2 IF .BFRD [BFRD$W_CURPID] NEQ 0
;  2258     3231  2 THEN
;  2259     3232  2     BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
;  2260     3233  2
;  2261     3234  2 BFRD [BFRD$L_LBN] = 0;
;  2262     3235  2 BFRD [BFRD$L_UCB] = 0;
;  2263     3236  2 BFRD [BFRD$L_LOCKBASIS] = 0;
;  2264     3237  2 BFRD [BFRD$L_SEQNUM] = 0;
;  2265     3238  2 BFRD [BFRD$W_CURPID] = 0;
;  2266     3239  2 BFRD [BFRD$W_NXTBFRD] = 0;
;  2267     3240  2
;  2268     3241  2 ! Remove from either the in-process list or it's current position in
;  2269     3242  2 ! the cache LRU (whether curpid was zero or not) and move to the head
;  2270     3243  2 ! of the cache LRU list.
;  2271     3244  2 !
;  2272     3245  2
;  2273     3246  2 REMQUE (.BFRD, BFRD);
;  2274     3247  2 INSQUE (.BFRD, POOL_LRU [QFLNK]);
;  2275     3248  2
;  2276     3249  1 END;


                                        52  DD 00000 RETURN_BFRD:
                                                      PUSHL   R2                                    ; 3202
        51      18   A0          02            00  EF 00002                EXTZV   #0, #2, 24(BFRD), POOL                ; 3224
                             52       FC BA41  7E 00008                MOVAQ   a-4(BASE)[POOL], POOL_LRU             ; 3225
```

```
                          52              28  C0 0000D            ADDL2   #40, POOL_LRU                                   ;
                                      1C  A0  B5 00010            TSTW    28(BFRD)                                        ; 3230
                                          04  13 00013            BEQL    1$                                              ;
                                      F4 AA41 B7 00015            DECW    -12(BASE)[POOL]                                 ; 3232
                                      08  A0  7C 00019 1$:         CLRQ    8(BFRD)                                         ; 3234
                                      10  A0  7C 0001C            CLRQ    16(BFRD)                                        ; 3236
                                      1C  A0  D4 0001F            CLRL    28(BFRD)                                        ; 3238
                          50              60  0F 00022            REMQUE  (BFRD), BFRD                                    ; 3246
                          62              60  0E 00025            INSQUE  (BFRD), (POOL_LRU)                              ; 3247
                                          04  BA 00028            POPR    #^M<R2>                                         ; 3249
                                              05 0002A            RSB                                                     ;

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0DC8
```

```
;  2278     3250  1 GLOBAL ROUTINE INVALIDATE (BUFFER) : L_NORM NOVALUE =
;  2279     3251  1
;  2280     3252  1 !++
;  2281     3253  1 !
;  2282     3254  1 ! FUNCTIONAL DESCRIPTION:
;  2283     3255  1 !
;  2284     3256  1 !       This routine invalidates the indicated buffer.
;  2285     3257  1 !
;  228      3258  1 ! CALLING SEQUENCE:
;  228      3259  1 !       INVALIDATE (ARG1)
;  2288     3260  1 !
;  2289     3261  1 ! INPUT PARAMETERS:
;  2290     3262  1 !       ARG1: address of block buffer
;  229      3263  1 !
;  2292     3264  1 ! IMPLICIT INPUTS:
;  2293     3265  1 !       NONE
;  2294     3266  1 !
;  2295     3267  1 ! OUTPUT PARAMETERS:
;  2296     3268  1 !       NONE
;  2297     3269  1 !
;  2298     3270  1 ! IMPLICIT OUTPUTS:
;  2299     3271  1 !       NONE
;  2300     3272  1 !
;  2301     3273  1 ! ROUTINE VALUE:
;  2302     3274  1 !       NONE
;  2303     3275  1 !
;  2304     3276  1 ! SIDE EFFECTS:
;  2305     3277  1 !       buffer contents forgotten
;  2306     3278  1 !
;  2307     3279  1 !--
;  2308     3280  1
;  2309     3281  2 BEGIN
;  2310     3282  2
;  2311     3283  2 BIND_COMMON;
;  2312     3284  2
;  2313     3285  2 LOCAL
;  2314     3286  2         POOL,
;  2315     3287  2         BFRD    : REF BBLOCK;
;  2316     3288  2
;  2317     3289  2 BFRD = ((.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF
;  2318     3290  2         + .CACHE_HDR [F11BC$L_BFRDBAS];
;  2319     3291  2
;  2320     3292  2 BFRD [BFRD$V_DIRTY] = 0;
;  2321     3293  2 BFRD [BFRD$V_VALID] = 0;
;  2322     3294  2
;  2323     3295  2 IF .BFRD [BFRD$W_CURPID] NEQ .(CTL$GL_PCB [PCB$L_PID])<0,16>
;  2324     3296  2 THEN
;  2325     3297  2     BUG_CHECK (XQPERR, 'expected this to be in-process');
;  2326     3298  2
;  2327     3299  2 ! Pull this buffer out of it's current position and place it at the
;  2328     3300  2 ! front of the list so it will be the next buffer tossed off of our
;  2329     3301  2 ! in-process list.
;  2330     3302  2 !
;  2331     3303  2
;  2332     3304  2 POOL = .BFRD [BFRD$V_POOL];
;  2333     3305  2 REMQUE (.BFRD, BFRD);
;  2334     3306  2 INSQUE (.BFRD, BFR_LIST [.POOL, QFLNK]);
```

```
;  2335          3307  2
;  2336          3308  2 IF .BUFFER EQL .BITMAP_BUFFER
;  2337          3309  2 THEN
;  2338          3310  2     BITMAP_VBN = 0;
;  2339          3311  2
;  2340          3312  1 END;                                    ! end of routine INVALIDATE


                                        0000 00000          .ENTRY  INVALIDATE, Save nothing                ; 3250
                              50    FC  AA  D0 00002          MOVL    -4(BASE), R0                            ; 3289
                        51    04  AC      60  C3 00006          SUBL3   (R0), BUFFER, R1                        ;
                                  51 00000200   8F  C6 0000B          DIVL2   #512, R1                        ;
                                  51          20  C4 00012          MULL2   #32, R1                         ;
                                  51    18    A0  C0 00015          ADDL2   24(R0), BFRD                    ; 3290
                              18  A1          0C  8A 00019          BICB2   #12, 24(BFRD)                   ; 3293
                                  50 00000000G  00  D0 0001D          MOVL    CTL$GL_PCB, R0                  ; 3295
                              60  A0    1C    A1  B1 00024          CMPW    28(BFRD), 96(R0)                ;
                                            04  13 00029          BEQL    1$                              ;
                                              FEFF 0002B          BUGW                                    ; 3297
                                             0000* 0002D          .WORD   <BUG$_XQPERR!4>                 ;
        50        18  A1          02          00  EF 0002F 1$:     EXTZV   #0, #2, 24(BFRD), POOL          ; 3304
                                  51          61  0F 00035          REMQUE  (BFRD), BFRD                    ; 3305
                                  50    CC AA40   7E 00038          MOVAQ   -52(BASE)[POOL], R0             ; 3306
                                  60          61  0E 0003D          INSQUE  (BFRD), (R0)                    ;
                              BC  AA    04    AC  D1 00040          CMPL    BUFFER, -68(BASE)               ; 3308
                                            03  12 00045          BNEQ    2$                              ;
                                        B4  AA  D4 00047          CLRL    -76(BASE)                       ; 3310
                                            04 0004A 2$:          RET                                     ; 3312

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 0DF3
```

```
;  2342          3313  1 GLOBAL ROUTINE WRITE_HEADER : L_NORM NOVALUE  =
;  2343          3314  1
;  2344          3315  1 !++
;  2345          3316  1 !
;  2346          3317  1 ! FUNCTIONAL DESCRIPTION:
;  2347          3318  1 !
;  2348          3319  1 !       This routine writes out the currently resident file header.
;  2349          3320  1 !
;  2350          3321  1 ! CALLING SEQUENCE:
;  2351          3322  1 !       WRITE_HEADER ()
;  2352          3323  1 !
;  2353          3324  1 ! INPUT PARAMETERS:
;  2354          3325  1 !       NONE
;  2355          3326  1 !
;  2356          3327  1 ! IMPLICIT INPUTS:
;  2357          3328  1 !       FILE_HEADER: address of current file header
;  2358          3329  1 !
;  2359          3330  1 ! OUTPUT PARAMETERS:
;  2360          3331  1 !       NONE
;  2361          3332  1 !
;  2362          3333  1 ! IMPLICIT OUTPUTS:
;  2363          3334  1 !       IO_STATUS: status of I/O transfer
;  2364          3335  1 !
;  2365          3336  1 ! ROUTINE VALUE:
;  2366          3337  1 !       NONE
;  2367          3338  1 !
;  2368          3339  1 ! SIDE EFFECTS:
;  2369          3340  1 !       checksum checked, header written
;  2370          3341  1 !
;  2371          3342  1 !--
;  2372          3343  1
;  2373          3344  2 BEGIN
;  2374          3345  2
;  2375          3346  2 BIND_COMMON;
;  2376          3347  2
;  2377          3348  2 EXTERNAL ROUTINE
;  2378          3349  2         CHECKSUM        : L_NORM;       ! compute file header checksum
;  2379          3350  2
;  2380          3351  2
;  2381          3352  2 ! The checksum of the header should be good, since all routines that modify
;  2382          3353  2 ! the header bless it with a new checksum when they are finished. Check the
;  2383          3354  2 ! checksum and write the header.
;  2384          3355  2 !
;  2385          3356  2
;  2386          3357  2 IF NOT CHECKSUM (.FILE_HEADER)
;  2387          3358  2 THEN BUG_CHECK (WRTINVHDR, FATAL, 'ACP attempted to write an invalid file header');
;  2388          3359  2
;  2389          3360  2 WRITE_BLOCK (.FILE_HEADER);
;  2390          3361  2
;  2391          3362  1 END;                                  ! end of routine WRITE_HEADER


                                                .EXTRN  CHECKSUM, BUG$_WRTINVHDR

                                0000 00000      .ENTRY  WRITE_HEADER, Save nothing              ; 3313
                          04    AA  DD 00002    PUSHL   4(BASE)                                 ; 3357
```

```
                0000G  CF          01  FB 00005          CALLS   #1, CHECKSUM                          ;
                       04          50  E8 0000A          BLBS    R0, 1$                               ;
                                     FEFF 0000D          BUGW                                         ; 3358
                                    0000* 0000F          .WORD   <BUG$_WRTINVHDR!4>                   ;
                           04       AA  DD 00011 1$:     PUSHL   4(BASE)                              ; 3360
                F967       CF          01  FB 00014          CALLS   #1, WRITE_BLOCK                  ;
                                           04 00019          RET                                      ; 3362

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 0E3E
```

```
;2393     3363  1 GLOBAL ROUTINE WRITE_DIRTY (LOCKBASIS) : L_NORM NOVALUE =
;2394     3364  1
;2395     3365  1 !++
;2396     3366  1 !
;2397     3367  1 ! FUNCTIONAL DESCRIPTION:
;2398     3368  1 !
;2399     3369  1 !       This routine writes all buffers which were modified back to the
;2400     3370  1 !       disk from whe ce they came.
;2401     3371  1 !
;2402     3372  1 ! INPUT PARAMETERS:
;2403     3373  1 !
;2404     3374  1 ! SIDE EFFECTS:
;2405     3375  1 !       dirty buffers written.
;2406     3376  1 !
;2407     3377  1 !--
;2408     3378  1
;2409     3379  2 BEGIN
;2410     3380  2
;2411     3381  2 BIND_COMMON;
;2412     3382  2
;2413     3383  2 INCR POOL FROM 0 TO 2
;2414     3384  2 DO
;2415     3385  3     BEGIN
;2416     3386  3     LOCAL
;2417     3387  3         I,
;2418     3388  3         QHEAD    : REF BBLOCK,
;2419     3389  3         BFRD     : REF BBLOCK;
;2420     3390  3
;2421     3391  3     QHEAD = BFR_LIST [.POOL, QFLNK];
;2422     3392  3     BFRD = .QHEAD [QFLNK];
;2423     3393  3
;2424     3394  3     I = .BFRS_USED [.POOL];
;2425     3395  3
;2426     3396  3     UNTIL .I EQL 0
;2427     3397  3     DO
;2428     3398  4         BEGIN
;2429     3399  4         I = .I - 1;
;2430     3400  4
;2431     3401  5         IF (.LOCKBASIS EQL 0
;2432     3402  5             OR .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS)
;2433     3403  4             AND .BFRD [BFRD$V_DIRTY]
;2434     3404  4         THEN
;2435     3405  4             WRITE_BLOCK (((.BFRD-.CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF)*512
;2436     3406  4                          + .CACHE_HDR [F11BC$L_BUFBASE]);
;2437     3407  4
;2438     3408  4         BFRD = .BFRD [BFRD$L_QFL];
;2439     3409  3         END;
;2440     3410  3
;2441     3411  3     IF .QHEAD NEQA .BFRD
;2442     3412  3     THEN
;2443     3413  3         BUG_CHECK (XQPERR, 'in-process queue screwed up');
;2444     3414  3
;2445     3415  2     END;
;2446     3416  2
;2447     3417  1 END;                        ! of routine WRITE_DIRTY
```

```
                                003C 00000          .ENTRY  WRITE_DIRTY, Save R2,R3,R4,R5                   ; 3363
                             53 D4 00002            CLRL    POOL                                            ; 3383
                          54 CC AA43 7E 00004 1$:   MOVAQ   -52(BASE)[POOL], QHEAD                          ; 3391
                          52    64 D0 00009         MOVL    (QHEAD), BFRD                                   ; 3392
                          55 F4 AA43 3C 0000C       MOVZWL  -12(BASE)[POOL], I                              ; 3394
                             55 D5 00011 2$:        TSTL    I                                               ; 3396
                             33 13 00013            BEQL    5$
                             55 D7 00015            DECL    I                                               ; 3399
                          04 AC D5 00017            TSTL    LOCKBASIS                                       ; 3401
                             07 13 0001A            BEQL    3$
                    04    AC 10 A2 D1 0001C         CMPL    16(BFRD), LOCKBASIS                             ; 3402
                             20 12 00021            BNEQ    4$
                1B  18    A2    02 E1 00023 3$:     BBC     #2, 24(BFRD), 4$                                ; 3403
                          51 FC AA D0 00028         MOVL    -4(BASE), R1                                    ; 3405
                50        52 18 A1 C3 0002C         SUBL3   24(R1), BFRD, R0
                          50    20 C6 00031         DIVL2   #32, R0
                50        50    09 78 00034         ASHL    #9, R0, R0
                          51    61 D0 00038         MOVL    (R1), R1                                        ; 3406
                             6140 9F 0003B          PUSHAB  (R1)[R0]
                      F923  CF    01 FB 0003E       CALLS   #1, WRITE_BLOCK
                          52    62 D0 00043 4$:     MOVL    (BFRD), BFRD                                    ; 3408
                             C9 11 00046            BRB     2$                                              ; 3396
                          52    54 D1 00048 5$:     CMPL    QHEAD, BFRD                                     ; 3411
                             04 13 0004B            BEQL    6$
                               FEFF 0004D           BUGW                                                    ; 3413
                                0000* 0004F         .WORD   <BUG$_XQPERR!4>
                AF        53    02 F3 00051 6$:     AOBLEQ  #2, POOL, 1$                                    ; 3383
                                   04 00055         RET                                                     ; 3417

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 0E58
```

```
;2449     3418  1 GLOBAL ROUTINE TOSS_CACHE_DATA (LCKINDX) : L_NORM NOVALUE =
;2450     3419  1
;2451     3420  1 !++
;2452     3421  1 !
;2453     3422  1 ! FUNCTIONAL DESCRIPTION:
;2454     3423  1 !
;2455     3424  1 !       This routine invalidates all data blocks associated with
;2456     3425  1 !       the given lock basis, after first writing dirty blocks.
;2457     3426  1 !       The data sequence number is incremented, which invalidates
;2458     3427  1 !       copies of these blocks on other nodes, and blocks found in
;2459     3428  1 !       our cache will be marked invalid, causing them to be tossed
;2460     3429  1 !       from the cache when the serialization lock is lowered.
;2461     3430  1 !       Only the data block pool is scanned.
;2462     3431  1 !
;2463     3432  1 ! SIDE EFFECTS:
;2464     3433  1 !       dirty buffers written, appropriate buffers invalidated
;2465     3434  1 !
;2466     3435  1 !--
;2467     3436  1
;2468     3437  2 BEGIN
;2469     3438  2
;2470     3439  2 BIND_COMMON;
;2471     3440  2
;2472     3441  2 LOCAL
;2473     3442  2     I,
;2474     3443  2     LOCKBASIS,
;2475     3444  2     QHEAD       : REF BBLOCK,
;2476     3445  2     BFRD        : REF BBLOCK;
;2477     3446  2
;2478     3447  2 LOCKBASIS = .LB_BASIS [.LCKINDX];
;2479     3448  2
;2480     3449  2 QHEAD = BFR_LIST [1, QFLNK];
;2481     3450  2 BFRD = .QHEAD [QFLNK];
;2482     3451  2
;2483     3452  2 I = .BFRS_USED [1];
;2484     3453  2
;2485     3454  2 UNTIL .I EQL 0
;2486     3455  2 DO
;2487     3456  3     BEGIN
;2488     3457  3     I = .I - 1;
;2489     3458  3
;2490     3459  3     IF .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS
;2491     3460  3     THEN
;2492     3461  4         BEGIN
;2493     3462  4
;2494     3463  4         IF .BFRD [BFRD$V_DIRTY]
;2495     3464  4         THEN
;2496     3465  4             WRITE_BLOCK (((.BFRD-.CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF)*512
;2497     3466  4                         + .CACHE_HDR [F11BC$L_BUFBASE]);
;2498     3467  4
;2499     3468  4         BFRD [BFRD$V_VALID] = 0;
;2500     3469  3         END;
;2501     3470  3
;2502     3471  3     BFRD = .BFRD [BFRD$L_QFL];
;2503     3472  2     END;                    ! of loop through in-process data block pool
;2504     3473  2
;2505     3474  2 IF .QHEAD NEQA .BFRD
```

```
;  2506          3475  2 THEN
;  2507          3476  2     BUG_CHECK (XQPERR, 'in-process data block queue screwed up');
;  2508          3477  2
;  2509          3478  2 LB_DATASEQ [.LCKINDX] = .LB_DATASEQ [.LCKINDX] + 1;
;  2510          3479  2
;  2511          3480  1 END;                        ! of routine TOSS_CACHE_DATA


                                    003C 00000          .ENTRY  TOSS_CACHE_DATA, Save R2,R3,R4,R5       ; 3418
                              51    F4   AA   9E 00002  MOVAB   -12(BASE), R1                           ; 3437
                              50    04   AC   D0 00006  MOVL    LCKINDX, R0                             ; 3447
                              55  0080 CA40   D0 0000A  MOVL    128(BASE)[R0], LOCKBASIS
                              53    D4   AA   9E 00010  MOVAB   -44(BASE), QHEAD                        ; 3449
                              52         63   D0 00014  MOVL    (QHEAD), BFRD                           ; 3450
                              54    02   A1   3C 00017  MOVZWL  2(R1), I                                ; 3452
                                         54   D5 0001B 1$:    TSTL    I                               ; 3454
                                         31   13 0001D  BEQL    4$
                                         54   D7 0001F  DECL    I                                       ; 3457
                              55    10   A2   D1 00021  CMPL    16(BFRD), LOCKBASIS                     ; 3459
                                         24   12 00025  BNEQ    3$
                        1B    18   A2        02   E1 00027  BBC     #2, 24(BFRD), 2$                    ; 3463
                              51    FC   AA   D0 0002C  MOVL    -4(BASE), R1                            ; 3465
                        50    52    18   A1   C3 00030  SUBL3   24(R1), BFRD, R0
                              50         20   C6 00035  DIVL2   #32, R0
                        50    50         09   78 00038  ASHL    #9, R0, R0
                              51         61   D0 0003C  MOVL    (R1), R1                                ; 3466
                                       6140   9F 0003F  PUSHAB  (R1)[R0]
                           F8C9   CF        01   FB 00042  CALLS   #1, WRITE_BLOCK
                             18   A2        08   8A 00047 2$:    BICB2   #8, 24(BFRD)                   ; 3468
                              52         62   D0 0004B 3$:    MOVL    (BFRD), BFRD                    ; 3471
                                         CB   11 0004E  BRB     1$                                      ; 3454
                              52         53   D1 00050 4$:    CMPL    QHEAD, BFRD                     ; 3474
                                         04   13 00053  BEQL    5$
                                        FEFF  00055  BUGW                                              ; 3476
                                       0000* 00057  .WORD   <BUG$_XQPERR!4>
                              50    04   AC   D0 00059 5$:    MOVL    LCKINDX, R0                     ; 3478
                                 00A8 CA40   D6 0005D  INCL    168(BASE)[R0]
                                         04   00062  RET                                               ; 3480

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 0EAE

;  2512          3481  1
```

```
;  2514          3482  1 GLOBAL ROUTINE KILL_CACHE (UCB) : L_NORM NOVALUE =
;  2515          3483  1
;  2516          3484  1 !++
;  2517          3485  1 !
;  2518          3486  1 ! This routine scans through all BFRD's in the cache tossing out
;  2519          3487  1 ! all those that match the specified UCB.
;  2520          3488  1 !
;  2521          3489  1 !--
;  2522          3490  1
;  2523          3491  2 BEGIN
;  2524          3492  2
;  2525          3493  2 BIND_COMMON;
;  2526          3494  2
;  2527          3495  2 LOCAL
;  2528          3496  2     BFRD : REF BBLOCK,
;  2529          3497  2     PIDINDX : WORD;
;  2530          3498  2
;  2531          3499  2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS];
;  2532          3500  2 PIDINDX = .CT[$GL_PCB [PCB$L_PID];
;  2533          3501  2
;  2534          3502  2 SERIAL_CACHE ();
;  2535          3503  2
;  2536          3504  3 INCR INDX0 FROM 0 TO (.CACHE_HDR [F11BC$W_BFRCNT] - 1)
;  2537          3505  2 DO
;  2538          3506  3     BEGIN
;  2539          3507  3
;  2540          3508  3     IF .BFRD [BFRD$L_UCB] EQLA .UCB
;  2541          3509  3     THEN
;  2542          3510  4         BEGIN
;  2543          3511  4
;  2544          3512  4         IF .BFRD [BFRD$W_CURPID] NEQ 0
;  2545          3513  4         THEN
;  2546          3514  5             BEGIN
;  2547          3515  5             IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
;  2548          3516  5             THEN
;  2549          3517  5                 BFRD [BFRD$V_VALID] = 0;        ! it will get tossed on unlock
;  2550          3518  5             END
;  2551          3519  4         ELSE
;  2552          3520  5             BEGIN
;  2553          3521  5             UNHOOK_BFRD (.INDX0+1, .BFRD);
;  2554          3522  5             RETURN_BFRD (.BFRD);
;  2555          3523  4             END;
;  2556          3524  3         END;
;  2557          3525  3
;  2558          3526  3     BFRD = .BFRD + BFRD$S_BFRDDEF;
;  2559          3527  3
;  2560          3528  2     END;
;  2561          3529  2
;  2562          3530  2 RELEASE_CACHE ();
;  2563          3531  2
;  2564          3532  1 END;


                                003C 00000         .ENTRY  KILL_CACHE, Save R2,R3,R4,R5                    ; 3482
```

```
                        50        FC    AA   D0 00002          MOVL    -4(BASE), R0                            ; 3499
                        52        18    A0   D0 00006          MOVL    24(R0), BFRD                            ;
                        50 00000000G    00   D0 0000A          MOVL    CTL$GL_PCB, R0                          ; 3500
                        55        60    A0   B0 0000E          MOVW    96(R0), PIDINDX                         ;
                                  F35D  30   00011             BSBW    SERIAL_CACHE                            ; 3502
                        50        FC    AA   D0 00014          MOVL    -4(BASE), R0                            ; 3504
                        54        16    A0   3C 00018          MOVZWL  22(R0), R4                              ;
                        53              01   CE 0001C          MNEGL   #1, INDX0                               ;
                                        2B   11 0001F          BRB     4$                                      ;
                     04 AC        0C    A2   D1 00021 1$:      CMPL    12(BFRD), UCB                           ; 3508
                                        21   12 00026          BNEQ    3$                                      ;
                                  1C    A2   B5 00028          TSTW    28(BFRD)                                ; 3512
                                        0C   13 0002B          BEQL    2$                                      ;
                        55        1C    A2   B1 0002D          CMPW    28(BFRD), PIDINDX                       ; 3515
                                        16   12 00031          BNEQ    3$                                      ;
                     18 A2              08   8A 00033          BICB2   #8, 24(BFRD)                            ; 3517
                                        10   11 00037          BRB     3$                                      ; 3512
                                        52   DD 00039 2$:      PUSHL   BFRD                                    ; 3521
                                  01    A3   9F 0003B          PUSHAB  1(INDX0)                                ;
                           FA25 CF            02   FB 0003E          CALLS   #2, UNHOOK_BFRD                         ;
                        50              52   D0 00043          MOVL    BFRD, R0                                ; 3522
                                     FE6A  30   00046          BSBW    RETURN_BFRD                             ;
                        52              20   C0 00049 3$:      ADDL2   #32, BFRD                               ; 3526
                     D1 53              54   F2 0004C 4$:      AOBLSS  R4, INDX0, 1$                           ; 3504
                                     F779  30   00050          BSBW    RELEASE_CACHE                           ; 3530
                                           04   00053          RET                                             ; 3532

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 0F11
```

```
;  2566          3533  1 GLOBAL ROUTINE WRONG_LOCKBASIS (HEADER) : L_NORM NOVALUE =
;  2567          3534  1
;  2568          3535  1 !++
;  2569          3536  1 !
;  2570          3537  1 ! This routine returns the given buffer as a result of discovering
;  2571          3538  1 ! we had the wrong lockbasis when finding it.
;  2572          3539  1 ! If it was read from disk, we must invalidate it, as we stored the
;  2573          3540  1 ! wrong lockbasis in its bfrd.  If it was found in the cache,
;  2574          3541  1 ! simply return it and credit us for returning it.
;  2575          3542  1 !
;  2576          3543  1 !--
;  2577          3544  1
;  2578          3545  2 BEGIN
;  2579          3546  2
;  2580          3547  2 BIND_COMMON;
;  2581          3548  2
;  2582          3549  2 LOCAL
;  2583          3550  2     POOL,
;  2584          3551  2     BFRD        : REF BBLOCK,
;  2585          3552  2     POOL_LRU    : REF BBLOCK;
;  2586          3553  2
;  2587          3554  2 SERIAL_CACHE ();
;  2588          3555  2
;  2589          3556  2 BFRD = ((.HEADER - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF
;  2590          3557  2          + .CACHE_HDR [F11BC$L_BFRDBAS];
;  2591          3558  2
;  2592          3559  2 IF .BFRD [BFRD$W_BFRL] EQL 0
;  2593          3560  2 THEN
;  2594          3561  2     BFRD [BFRD$V_VALID] = 0
;  2595          3562  2 ELSE
;  2596          3563  3     BEGIN
;  2597          3564  3     POOL = .BFRD [BFRD$V_POOL];
;  2598          3565  3     REMQUE (.BFRD, BFRD);
;  2599          3566  3     BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
;  2600          3567  3     BFRD [BFRD$W_CURPID] = 0;
;  2601          3568  3     POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
;  2602          3569  3     INSQUE (.BFRD, .POOL_LRU [QBLNK]);
;  2603          3570  2     END;
;  2604          3571  2
;  2605          3572  2 RELEASE_CACHE ();
;  2606          3573  1 END;


                                        000C 00000          .ENTRY  WRONG_LOCKBASIS, Save R2,R3                     ; 3533
                                    F318  30 00002          BSBW    SERIAL_CACHE                                    ; 3554
                          50        FC  AA  D0 00005          MOVL    -4(BASE), R0                                    ; 3556
                51        04  AC        60  C3 00009          SUBL3   (R0), HEADER, R1                                ;
                              51 00000200 8F  C6 0000E          DIVL2   #512, R1                                        ;
                              51          20  C4 00015          MULL2   #32, R1                                         ;
                              51        18  A0  C0 00018          ADDL2   24(R0), BFRD                                    ; 3557
                                        1A  A1  B5 0001C          TSTW    26(BFRD)                                        ; 3559
                                            06  12 0001F          BNEQ    1$                                              ;
                                    18  A1      08  8A 00021          BICB2   #8, 24(BFRD)                            ; 3561
                                            1C  11 00025          BRB     2$                                              ;
```

```
                50      18   A1          02          00  EF 00027 1$:    EXTZV   #0, #2, 24(BFRD), POOL                  ; 3564
                                        51          61  0F 0002D        REMQUE  (BFRD), BFRD                        ; 3565
                                                F4 AA40  B7 00030        DECW    -12(BASE)[POOL]                     ; 3566
                                                1C   A1  B4 00034        CLRW    28(BFRD)                            ; 3567
                                        50      FC BA40  7E 00037        MOVAQ   @-4(BASE)[POOL], POOL_LRU           ; 3568
                                        50          28  C0 0003C        ADDL2   #40, POOL_LRU                       ;
                                    04  B0          61  0E 0003F        INSQUE  (BFRD), @4(POOL_LRU)                ; 3569
                                            F732        30 00043 2$:    BSBW    RELEASE_CACHE                       ; 3572
                                                        04 00046        RET                                         ; 3573

; Routine Size:  71 bytes,    Routine Base:  $CODE$ + 0F69
```

```
;2608     3574  1 GLOBAL ROUTINE KILL_BUFFERS (POOL, LOCKBASIS) : L_NORM NOVALUE =
;2609     3575  1
;2610     3576  1 !++
;2611     3577  1 !
;2612     3578  1 ! This routine scans through all BFRD's in the cache tossing out
;2613     3579  1 ! all those in the specified POOL for the CURRENT_UCB.
;2614     3580  1 ! If the directory data pool is specified, LOCKBASIS must match also.
;2615     3581  1 !
;2616     3582  1 !--
;2617     3583  1
;2618     3584  2 BEGIN
;2619     3585  2
;2620     3586  2 BIND_COMMON;
;2621     3587  2
;2622     3588  2 LOCAL
;2623     3589  2     BFRD : REF BBLOCK,
;2624     3590  2     BFRD_CNT : WORD,
;2625     3591  2     START_BFRD : WORD,
;2626     3592  2     PIDINDX : WORD;
;2627     3593  2
;2628     3594  3 BEGIN
;2629     3595  3
;2630     3596  3 BIND
;2631     3597  3     POOLCNT = CACHE_HDR [F11BC$W_POOLCNT] : VECTOR [,WORD];
;2632     3598  3
;2633     3599  3 START_BFRD = 0;
;2634     3600  3
;2635     3601  3 IF .POOL NEQ 0
;2636     3602  3 THEN
;2637     3603  4     BEGIN
;2638     3604  4     START_BFRD = .POOLCNT [0];
;2639     3605  4     IF .POOL EQL 2
;2640     3606  4     THEN START_BFRD = .START_BFRD + .POOLCNT [1];
;2641     3607  3     END;
;2642     3608  3
;2643     3609  3 BFRD_CNT = .POOLCNT [.POOL];
;2644     3610  2 END;
;2645     3611  2
;2646     3612  2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .START_BFRD*BFRD$S_BFRDDEF;
;2647     3613  2 PIDINDX = .CT[$GL_PCB [PCB$L_PID];
;2648     3614  2
;2649     3615  2 SERIAL_CACHE ();
;2650     3616  2
;2651     3617  3 INCR INDX0 FROM .START_BFRD TO (.START_BFRD + .BFRD_CNT - 1)
;2652     3618  2 DO
;2653     3619  3     BEGIN
;2654     3620  3
;2655     3621  3     IF .CURRENT_UCB EQL .BFRD [BFRD$L_UCB]
;2656     3622  4         AND (.POOL NEQ 1 OR
;2657     3623  4             .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS)
;2658     3624  3     THEN
;2659     3625  4         BEGIN
;2660     3626  4
;2661     3627  4         IF .BFRD [BFRD$W_CURPID] NEQ 0
;2662     3628  4         THEN
;2663     3629  5             BEGIN
;2664     3630  5             IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
```

```
;  2665          3631  5              THEN
;  2666          3632  5                  BFRD [BFRD$V_VALID] = 0;        ! it will get tossed on unlock
;  2667          3633  5              END
;  2668          3634  4          ELSE
;  2669          3635  5              BEGIN
;  2670          3636  5              UNHOOK_BFRD (.INDX0+1, .BFRD);
;  2671          3637  5              RETURN_BFRD (.BFRD);
;  2672          3638  4              END;
;  2673          3639  3          END;
;  2674          3640  3
;  2675          3641  3      BFRD = .BFRD + BFRD$S_BFRDDEF;
;  2676          3642  3
;  2677          3643  2      END;
;  2678          3644  2
;  2679          3645  2 RELEASE_CACHE ();
;  2680          3646  2
;  2681          3647  1 END;
```

```
                                    003C 00000          .ENTRY  KILL_BUFFERS, Save R2,R3,R4,R5                  ; 3574
          51    FC    AA 00000078   8F   C1 00002       ADDL3   #120, -4(BASE), R1                              ; 3597
                                    53   B4 0000B       CLRW    START_BFRD                                      ; 3599
                      50        04  AC   D0 0000D       MOVL    POOL, R0                                        ; 3601
                                    0C   13 00011       BEQL    1$                                              ;
                      53            61   B0 00013       MOVW    (R1), START_BFRD                                ; 3604
                      02            50   D1 00016       CMPL    R0, #2                                          ; 3605
                                    04   12 00019       BNEQ    1$                                              ;
                      53        02  A1   A0 0001B       ADDW2   2(R1), START_BFRD                               ; 3606
                      54          6140   B0 0001F 1$:    MOVW    (R1)[R0], BFRD_CNT                              ; 3609
                      50        FC  AA   D0 00023       MOVL    -4(BASE), R0                                    ; 3612
                      51            53   3C 00027       MOVZWL  START_BFRD, R1                                  ;
                      51            20   C4 0002A       MULL2   #32, R1                                         ;
          52          51        18  A0   C1 0002D       ADDL3   24(R0), R1, BFRD                                ;
                      50 00000000G  00   D0 00032       MOVL    CTL$GL_PCB, R0                                  ; 3613
                      55        60  A0   B0 00039       MOVW    96(R0), PIDINDX                                 ;
                              F296  30   3D               BSBW    SERIAL_CACHE                                    ; 3615
                      50            53   3C 00040       MOVZWL  START_BFRD, R0                                  ; 3617
                      54            54   3C 00043       MOVZWL  BFRD_CNT, R4                                    ;
                      54            50   C0 00046       ADDL2   R0, R4                                          ;
                      53            53   3C 00049       MOVZWL  START_BFRD, INDX0                               ;
                                    53   D7 0004C       DECL    INDX0                                           ;
                                    38   11 0004E       BRB     6$                                              ;
                0C    A2        94  AA   D1 00050 2$:    CMPL    -108(BASE), 12(BFRD)                            ; 3621
                                    2E   12 00055       BNEQ    5$                                              ;
                      01        04  AC   D1 00057       CMPL    POOL, #1                                        ; 3622
                                    07   12 0005B       BNEQ    3$                                              ;
                08    AC        10  A2   D1 0005D       CMPL    16(BFRD), LOCKBASIS                             ; 3623
                                    21   12 00062       BNEQ    5$                                              ;
                              1C  A2   B5 00064 3$:    TSTW    28(BFRD)                                        ; 3627
                                    0C   13 00067       BEQL    4$                                              ;
                      55        1C  A2   B1 00069       CMPW    28(BFRD), PIDINDX                               ; 3630
                                    16   12 0006D       BNEQ    5$                                              ;
                18    A2            08   8A 0006F       BICB2   #8, 24(BFRD)                                    ; 3632
                                    10   11 00073       BRB     5$                                              ; 3627
```

```
                                        52  DD 00075 4$:    PUSHL   BFRD                            ; 3636
                                    0'  A3  9F 00077        PUSHAB  1(INDX0)                        ;
                                F94E    CF  02  FB 0007A        CALLS   #2, UNHOOK_BFRD             ;
                                        50  52  D0 0007F        MOVL    BFRD, R0                    ; 3637
                                    FD93    52  30 00082        BSBW    RETURN_BFRD                 ;
                                        52  20  C0 00085 5$:    ADDL2   #32, BFRD                   ; 3641
                                C4      53  54  F2 00088 6$:    AOBLSS  R4, INDX0, 2$               ; 3617
                                    F6A2    30  30 0008C        BSBW    RELEASE_CACHE               ; 3645
                                            04  0008F           RET                                 ; 3647

; Routine Size:  144 bytes,    Routine Base:  $CODE$ + 0FB0

; 2682          3648  1
; 2683          3649  1 END
; 2684          3650  0 ELUDOM



;                                       PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         4160  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics

;                                       --------- Symbols ---------       Pages       Processing
;       File                            Total     Loaded   Percent       Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1        18619         90         0       1000          00:01.9



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RDBLOK/OBJ=OBJ$:RDBLOK MSRC$:RDBLOK/UPDATE=(ENH$:RDBLOK)

; Size:         4153 code + 7 data bytes
; Run Time:        04:03.1
; Elapsed Time:    07:51.7
; Lines/CPU Min:       900
; Lexemes/CPU-Min: 64276
; Memory Used:  476 pages
; Compilation Complete
```

RWVB
LIS

RDBLOK
LIS

REQUEU
LIS

RWATTR
LIS

REMOVE
LIS

RDHEDR
LIS

RETDIR
LIS