

HIGH SPEED TEXT SEARCH SYSTEM

HSTS SOFTWARE
LISTINGS

VOL. 2 OF 5

Control
Computer
Support

Data Base
Update

STAT

HSTS MASTER COMPUTER SOFTWARE LISTINGS

SL120100

VOLUME 2 of 5

Prepared for:

Central Intelligence Agency
Washington, DC 20505

R80-016

March 1980

STAT

STAT



CONTROL COMPUTER
SUPPORT

INITCC - MACRO M1110 27-MAR-80 14:28
TABLE OF CONTENTS

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	2-	MACRO'S AND CONSTANTS
9-	18	ASSEMBLY-TIME DATA DEFINITION
10-	55	INITIALIZE CONTROL COMMON

```

1          .TITLE- INITCC-...
2          .SBTTL- MACRO'S AND CONSTANTS-
3          ;
4          .MCALL- NMBLK$,FDOF$,FCSBT$,FINIT$
5          .MCALL- FDBDF$,FDRC$,FDBK$,FDOP$,FSRSZ$
6          .MCALL- EXIT$S-
7          ;
8          .GLOBL- DIRP74,DIRP75
9          .GLOBL- .GTDIR-
10         ;
11         ;LUNS-
12         000001 DPLUN=1
13         ;
14         ;MISC EQUATES-
15         000001 EF,IO=1 ;I/O EVENT FLAG-
16         ;

```

```
18          .SBTTL· ASSEMBLY-TIME· DATA· DEFINITION·
19          ;
20          ;DUMMY· FDB
21          FDOF$·
22          FCSBT$
23          ;
24          FIB:  FDBDF$
25          FDRC$·  FD·RUM·
26          FDBK$·  BUFFER·N·BUFB··EF·IO·IOSTAT·
27          FDOP$·  DPLUN·
28          ;
29          FRSZ$  0
30          ;
31          ;DUMMY· NAME· BLOCK· TO· GET· DIRECTORY· FID'S
32          DUMNBK: NMBLK$  ··SY·0
33          ;
34          ;DIRECTORY· NAME· DESCRIPTORS·
35          DIRDS1:  .WORD· 5
36          .WORD· DIRDT1
37          DIRDT1: .ASCII· /L7,4J/
38          .EVEN·
39          DIRDS2:  .WORD· 5
40          .WORD· DIRDT2·
41          DIRDT2: .ASCII· /L7,5J/
42          .EVEN·
43          ;
44          ;DIRECTORY· ADDRESS· TABLE·
45          DIRTBL: .WORD· DIRDS1
46          .WORD· DIRP74
47          .WORD· DIRDS2·
48          .WORD· DIRP75
49          ;
50          ;MISC· LOCATIONS·
51          BUFFER:          ;DUMMY· BUFFER·
52          IOSTAT:  .BLKW· 2·  ;I/O· STATUS· BLOCK·
53          ;
```

```
55          .SBTTL INITIALIZE CONTROL COMMON
56          ;
57 000236      INITCC:
58 000236      FINIT$
59          ;
60          ; (INIT FID'S OF SYSTEM DIRECTORIES)
61 000242 012700 000000*      MOV.      #FDB,R0          ; FDB ADDRESS
62 000246 012701 000140*      MOV.      #DUMNBK,R1      ; DUMMY NAME BLOCK
63          ;
64 000252 012703 000222*      MOV.      #DIRTAB,R3      ; DIRECTORY ADDRESS TABLE
65 000256 012704 000002      MOV.      #2,R4          ; LOOP COUNT
66          ;
67 000262 012302      DIRLOP: MOV.      (R3)+,R2      ; DIRECTORY DESCRIPTOR ADDRESS
68 000264          .GTDIR      ; GET DIRECTORY FID
69 000270 012302      MOV.      (R3)+,R2      ; COMMON MEMORY ADDRESS
70 000272 016712 177666      MOV.      DUMNBK+R1,DIR,(R2) ; COPY FID TO COMMON
71 000276 016762 177664 000002*      MOV.      DUMNBK+R1,DIR+2,2(R2)
72 000304 016762 177660 000004      MOV.      DUMNBK+R1,DIR+4,4(R2)
73 000312 005304      DEC.      R4
74 000314 001362      BNE.      DIRLOP
75          ;
```

INITCC-MACRO-M1110 27-MAR-80 14:28 PAGE 11
INITIALIZE CONTROL-COMMON

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

77 000316 EXIT\$S
78 ;
79 000236' .END INITCC

BITVAL = 000000	FA.EXT = 000004	FO.WRT = 000016	NB.SD2 = 001000	S.FNBW = 000017
BIT0 = 000001	FA.NSP = 000100	F.ACTL = 000076	NB.SNM = 000040	S.FNTY = 000004
BIT1 = 000002	FA.POS = 010000	F.ALDC = 000040	NB.STP = 000020	S.FTYP = 000002
BIT10 = 002000	FA.RD = 000001	F.BBFS = 000062	NB.SVR = 000010	S.NFEN = 000020
BIT11 = 004000	FA.RWD = 004000	F.BDB = 000070	NB.TYP = 000002	TD#CTR = 176370
BIT12 = 010000	FA.SEQ = 000000	F.BGBC = 000057	NB.VER = 000001	TD#CTW = 176360
BIT13 = 020000	FA.SHR = 000040	F.BKDN = 000026	N.BFAC = 000004	TD#INL = 004000
BIT14 = 040000	FA.TMP = 000020	F.BKDS = 000020	N.BHGH = 000006	TD#MEM = 000270
BIT15 = 100000	FA.WCK = 020000	F.BKES = 000050	N.BTCH = 000004	TD#OAR = 176344
BIT2 = 000004	FA.WRT = 000002	F.BKPI = 000051	N.BUFB = 004000	TD#OTR = 176346
BIT3 = 000010	FDB = 000000R	F.BKST = 000024	N.BUFW = 002000	TD#QRD = 000274
BIT4 = 000020	FD.BLK = 000010	F.BKVB = 000064	N.DID = 000024	TD#RST = 176366
BIT5 = 000040	FD.CCL = 000002	F.CHR = 000075	N.DVNM = 000032	TD#SW = 176376
BIT6 = 000100	FD.COM = 020000	F.CNTG = 000034	N.FID = 000000	TD#TAR = 176372
BIT7 = 000200	FD.CR = 000002	F.DFNB = 000046	N.FNAM = 000006	TD#TAW = 176362
BIT8 = 000400	FD.DIR = 000010	F.DSPT = 000044	N.FOS = 000764	TD#TDR = 176374
BIT9 = 001000	FD.FID = 000000	003 F.DVNM = 000134	N.FTYP = 000014	TD#TDW = 176364
BUFFER = 000232R	FD.FNB = 000006	003 F.EFBK = 000010	N.FVER = 000016	T#AD = 000020
BYTE0 = 000000	FD.FTN = 000001	F.EFN = 000050	N.NEXT = 000022	T#BA = 000002
BYTE1 = 000001	FD.FVR = 000004	003 F.EOB = 000032	N.QUERY = 000031	T#BD = 000010
BYTE2 = 000002	FD.F11 = 040000	F.ERR = 000052	N.STAT = 000020	T#BSO = 100000
BYTE3 = 000003	FD.INS = 000010	F.FACC = 000043	N.SUNT = 000002	T#BT = 000020
BYTE4 = 000004	FD.ISP = 002000	F.FFBY = 000014	N.UNIT = 000034	T#BTAR = 000030
BYTE5 = 000005	FD.LEN = 000010	003 F.FNAM = 000110	R.FIX = 000001	T#BDT = 002000
BYTE6 = 000006	FD.MNT = 100000	F.FNB = 000102	R.SEQ = 000003	T#CD = 000100
BYTE7 = 000007	FD.OSP = 004000	F.FTYP = 000116	R.VAR = 000002	T#CLK = 002000
BYTE8 = 000010	FD.PLC = 000004	F.FVER = 000120	SR.ARE = 000114	002 T#DISK = 000200
BYTE9 = 000011	FD.PRN = 000004	F.HIBK = 000004	SR.ARS = 000106	002 T#DRD = 000004
BYTVAL = 000012	FD.PSE = 010000	F.LUN = 000042	SR.DAY = 000010	002 T#MEM = 010000
CF.COT = 000041	FD.RAH = 000001	F.MBCT = 000054	SR.DLT = 000014	002 T#FSA = 000000
CF.DGN = 000046	FD.RAN = 000002	F.MBC1 = 000055	SR.ECB = 000047	002 T#FSAB = 000004
CF.DHR = 000042	FD.REC = 000001	F.MBFG = 000056	SR.ECH = 000046	002 T#FSAC = 000014
CF.DMC = 000047	FD.RWM = 000001	F.NRBD = 000024	SR.ECL = 000050	002 T#FSB = 000010
CF.HBR = 000045	FD.SDI = 000020	F.NREC = 000030	SR.FIB = 000012	002 T#IB = 000026
CF.HRL = 000044	FD.SQD = 000040	F.OVBS = 000030	SR.GRE = 000100	002 T#IBAR = 000024
CF.UPD = 000043	FD.TTY = 000004	F.RACC = 000016	SR.GRS = 000072	002 T#IBE = 020000
CH.AND = 000001	FD.UBH = 000002	F.RATT = 000001	SR.LEN = 000122	002 T#IBF = 040000
DBSLEN = 000116	FF.CHR = 000005	F.RCNM = 000034	SR.LIN = 000066	002 T#ICD = 000040
DG.ERR = 001000	FF.NV = 000003	F.RCTL = 000017	SR.LIP = 000062	002 T#MODE = 004000
DG.SDF = 002000	FF.POE = 000002	F.RSIZ = 000002	SR.MON = 000006	002 T#OB = 000036
DG.TDF = 004000	FF.RWD = 000001	F.RTYP = 000000	SR.NDC = 000042	002 T#OBE = 004000
DIRDS1 = 000176R	FF.RUF = 000006	F.SEQN = 000100	SR.NDS = 000036	002 T#OBF = 010000
DIRDS2 = 000210R	FF.SPC = 000004	F.SPDV = 000072	SR.NIN = 000030	002 T#OBRA = 000034
DIRDT1 = 000202R	FN.ACK = 000016	004 F.SPUN = 000074	SR.NIP = 000022	002 T#OBWA = 000032
DIRDT2 = 000214R	FN.FSA = 000000	004 F.STBK = 000036	SR.SDB = 000032	002 T#OUTA = 100000
DIRLOP = 000262R	FN.FSB = 000002	004 F.UNIT = 000136	SR.SRC = 000002	002 T#RBD0 = 000200
DIRP74 = *****	FN.FSC = 000004	004 F.URBD = 000020	SR.SUN = 000000	002 T#RNB = 000040
DIRP75 = ***** G.	FN.FSD = 000020	004 F.VBN = 000064	SR.TWS = 000056	002 T#RSET = 040000
DIRTBL = 000222R	FN.NHR = 000010	004 F.VBSZ = 000060	SR.WSL = 000052	002 T#SC = 000022
DPLUN = 000001	FN.NMB = 000022	004 INITCC = 000236R	SR.YR = 000004	002 T#SCLK = 020000
DUNHKB = 000140R	FN.OLS = 000006	004 IOSTAT = 000232R	SR.1IN = 000024	002 T#SEG1 = 000000
EF.IO = 000001	FN.RDC = 000014	004 MS.DGN = 010000	SR.1IP = 000016	002 T#SEG2 = 000001
FA.APD = 000100	FN.RPD = 000012	004 N = 000012	S.BFHD = 000020	T#SEG3 = 000002
FA.CRE = 000010	FO.APD = 000106	NB.DEV = 000200	S.FATT = 000016	T#SO = 000001
FA.DLK = 001000	FO.MFY = 000002	NB.DIR = 000100	S.FDB = 0000140	T#UBUS = 100000
FA.ENB = 100000	FO.RD = 000001	NB.NAM = 000004	S.FNAM = 000006	T#1CLK = 000400
FA.EXC = 002000	FO.UPD = 000006	NB.SD1 = 000400	S.FNB = 000036	T#BEN = 000020

INITCC·MACRO·M1110 27-MAR-80 14:28 PAGE 11-3
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

WORD0 = 000000	WORD4 = 000010	WORD8 = 000020	.FSRCB = ***** G.	...PC2 = 000174R.
WORD1 = 000002	WORD5 = 000012	WORD9 = 000022	.GTDIR = ***** G.	...PC3 = 000000R.
WORD2 = 000004	WORD6 = 000014	WORDVAL = 000024	...GBL = 000000	...TPC = 000020
WORD3 = 000006	WORD7 = 000016	.FINIT = ***** G.	...PC1 = 000000R.	
. ABS. 000000 000	000324 001			
SRCOFF. 000122 002				
FDSCDF. 000010 003				
FNOFFS. 000022 004				
\$\$\$FSR1 000000 005				
ERRORS DETECTED: 0				

VIRTUAL MEMORY USED: 4873 WORDS (20 PAGES)
DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
ELAPSED TIME: 00:00:29
INITCC·INITCC /-SP=C20,1JP,C,INITCC·

.MAIN. MACRO M1110 27-MAR-80 13:54 PAGE 1

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

2.

.NLIST ME.

```

3          ;
4          ; P:MAC -- SYSTEM-WIDE PREFIX FILE.
5          ;
6          ;
7          000012.          .RADIX 10.
8          ;
9          ; BIT DEFINITIONS.
10         ;
11         .MACRO BITDEF X.
12         BIT X = BITVAL          ; SYMBOL SUBSTITUTE = BIT POSITION.
13         .NLIST
14         N = N + 1
15         BITVAL = BITVAL * 2.    ; SINGLE BIT LEFT SHIFT ONE.
16         .LIST
17         .ENDM BITDEF
18         000000          N = 0          ; BIT '0'
19         000001          BITVAL = 1    ; BIT 0 IS ONE.
20         000020          .REPT 16.
21         BITDEF \N.        ; MACRO CALL, PASS CHARACTER
22         .ENDM
23         ;
24         ; WORD ADDRESS OFFSET DEFINITIONS.
25         ;
26         .MACRO WORDOFF X.
27         .NLIST
28         WORD X = WRDVAL          ; SYMBOL SUBSTITUTE = OCTAL VALUE.
29         N = N + 1                ; INCREMENT 'X'
30         WRDVAL = WRDVAL + 2.
31         .LIST
32         .ENDM WORDOFF
33         000000          N = 0
34         000000          WRDVAL = 0
35         000012.          .REPT 10.
36         .NLIST
37         WORDOFF \N.          ; MACRO CALL, PASS CHARACTER
38         .LIST
39         .ENDM
40         ;
41         ; BYTE ADDRESS OFFSET DEFINITIONS.
42         ;
43         .MACRO BYTEOFF X.
44         .NLIST
45         BYTE X = BYTVAL          ; SYMBOL SUBSTITUTE = OCTAL VALUE.
46         N = N + 1                ; INCREMENT 'X'
47         BYTVAL = BYTVAL + 1
48         .LIST
49         .ENDM BYTEOFF
50         000000          N = 0
51         000000          BYTVAL = 0
52         000012.          .REPT 10.
53         .NLIST
54         BYTEOFF \N.         ; MACRO CALL, PASS CHARACTER
55         .LIST
56         .ENDM
57         000010          .RADIX 8.

```

```

59      .MCALL CALL
60      ;
61      ; RETURN FROM SUBROUTINE -- PC LINKAGE
62      ;
63      .MACRO RTN
64      RTS      PC
65      .ENDM RTN
66      ;
67      ; EXIT A SUBROUTINE
68      .MACRO EXIT SUBR
69      RTS      PC
70      .ENDM EXIT
71      ;
72      ; BRANCH ON ANY TESTED BIT ON -- USED AFTER BIT TEST (BIT OR BITB)
73      .MACRO BON LOC
74      .NLIST
75      BNE      LOC          ; BRANCH IF BIT(S) SET
76      .LIST
77      .ENDM
78      ;
79      ; BRANCH ON ALL TESTED BIT(S) OFF -- USED AFTER BIT TEST (BIT OR BITB)
80      .MACRO BOFF LOC
81      .NLIST
82      BEQ      LOC          ; BRANCH IF BIT(S) NOT SET
83      .LIST
84      .ENDM
85      ;
86      .MACRO SAVE A1,A2,A3,A4,A5,A6
87      .IF NB <A1>
88      MOV      A1,-(SP)
89      .ENDC
90      .IF NB <A2>
91      MOV      A2,-(SP)
92      .ENDC
93      .IF NB <A3>
94      MOV      A3,-(SP)
95      .ENDC
96      .IF NB <A4>
97      MOV      A4,-(SP)
98      .ENDC
99      .IF NB <A5>
100     MOV      A5,-(SP)
101     .ENDC
102     .IF NB <A6>
103     MOV      A6,-(SP)
104     .ENDC
105     .ENDM SAVE
106     .MACRO RESTOR A1,A2,A3,A4,A5,A6
107     .IF NB <A6>
108     MOV      (SP)+,A6
109     .ENDC
110     .IF NB <A5>
111     MOV      (SP)+,A5
112     .ENDC
113     .IF NB <A4>
114     MOV      (SP)+,A4
115     .ENDC

```

116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141

```

      .IF . . . NB      <A3>
      MOV      (SP)+,A3
    .ENDC
      .IF . . . NB      <A2>
      MOV      (SP)+,A2
    .ENDC
      .IF . . . NB      <A1>
      MOV      (SP)+,A1
    .ENDC
    .ENDM RESTOR
:
:
: MESSAGE OUTPUT MACRO
:
.MACRO MOUT$S, MSG,PAR
  MOV      MSG,-(SP)
  .IF . . . NB      <PAR>
  MOV      PAR,-(SP)
  .ENDC
  .IF . . . B      <PAR>
  CLR      -(SP)
  .ENDC
  JSR      PC,MSGOUT
  ADD      #4,SP
:EDIT OUTPUT MSG STRING AND PRINT IT
:RESTORE STACK POINTER
    .ENDM

```

```

143 ;
144 ; SYSTEM EQUATES.
145 ;
146 000004 N.BTCH=4 ;MAX NUMBER OF ACTIVE BATCHES.
147 000006 N.BHGH=<N.BTCH-1>*2 ;HIGH BATCH NUMBER.
148 000031 N.QUERY=25 ;MAX # OF QUERIES IN A BATCH.
149 000764 N.FOS=500 ;MAX # OF FOS ENTRIES (DOUBLE WORDS)
150 002000 N.BUFW=1024 ;SIZE OF SYSTEM BUFFERS (WORDS)
151 004000 N.BUFB=N.BUFW*2 ;SIZE OF SYSTEM BUFFERS (BYTES)
152 000004 N.BFAC=N.BUFW/256 ;# OF SECTORS IN SYSTEM BLOCK
153 000002 N.SUHT=2 ;NUMBER OF SEARCH UNITS.
154 ;
155 ;
156 ; STATUS RECORD OFFSETS -SREC: SREC0: SREC1: ETC.
157 ;
158 000000 .PSECT SRCOFF,ABS.
159 000000 SR.SUN: .BLKW 1 ;SEARCH UNIT NUMBER.
160 000002 SR.SRC: .BLKW 1 ;SEARCH TIME OF LAST SEARCH (SEC)
161 ; BEGINNING OF DATA BASE STATUS AREA.
162 000004 SR.YR: .BLKW 1 ;YEAR
163 000006 SR.MON: .BLKW 1 ;MONTH
164 000010 SR.DAY: .BLKW 1 ;DAY OF DISK INIT.
165 000012 SR.FIB: .BLKW 1 ;FILE IDENTIFICATION BLOCK
166 000014 SR.DLT: .BLKW 1 ;DELTA FOR INDEX SECTORS REPRESENTED
167 000016 SR.IIP: .BLKW 2 ;ADDRESS OF FIRST IPR.
168 000022 SR.NIP: .BLKW 1 ;NUMBER OF IPR SECTORS ON DISK.
169 000024 SR.IIN: .BLKW 2 ;ADDRESS OF FIRST INDEX RECORD.
170 000030 SR.NIN: .BLKW 1 ;NUMBER OF INDEX RECORDS ALLOCATED.
171 000032 SR.SDB: .BLKW 2 ;ADDRESS OF START OF DATA BASE.
172 000036 SR.NDS: .BLKW 2 ;NUMBER OF DOCUMENTS AT INIT.
173 000042 SR.HDC: .BLKW 2 ;CURRENT NUMBER OF DOCUMENTS.
174 000046 SR.ECH: .BLKB 1 ;HIGH ORDER ADDRESS OF EOC
175 000047 SR.ECB: .BLKB 1 ;BYTE INDEX OF EOC.
176 000050 SR.ECL: .BLKW 1 ;LOW ORDER ADDRESS OF EOC.
177 000052 SR.WSL: .BLKW 2 ;WHITE SPACE AFTER EOC.
178 000056 SR.TWS: .BLKW 2 ;TOTAL WHITE SPACE.
179 000062 SR.LIP: .BLKW 2 ;ADDRESS OF LATEST IPR.
180 000066 SR.LIN: .BLKW 2 ;ADDRESS OF LATEST IR.
181 000072 SR.GRS: .BLKW 3 ;GIVEN START DOC ID.
182 000100 SR.GRE: .BLKW 3 ;GIVEN END DOC ID.
183 000106 SR.ARS: .BLKW 3 ;CURRENT START DOC ID.
184 000114 SR.ARE: .BLKW 3 ;CURRENT END DOC ID.
185 ;
186 000116 DBSLEN=-SR.YR ;LENGTH OF DB STATUS AREA.
187 000122 SR.LEN: ;LENGTH OF STATUS RECORD.
188 ;
189 ;
190 ; FILE DESCRIPTOR OFFSETS.
191 ;
192 000000 .PSECT FDSCOF,ABS.
193 000000 FD.FID: .BLKW 2 ;FILE ID.
194 000004 FD.FVR: .BLKW 1 ;VERSION NUMBER.
195 000006 FD.FNB: .BLKW 1 ;FILE NUMBER.
196 000010 FD.LEN: ;LENGTH OF FDSC.
197 ;

```

```
2.
3.
4.
5.
6.
7.
8.
9.      000041      ;DATA IN MSG FOR CCOUT.
10     000042      ;DHR BUFFER NOW EMPTY FOR SQR.
11     000043      CF,UPD=35.      ;SEND DIRECTIVE QUEUED TO DBUPD.
12     000044      CF,HRL=36.      ;HRL DATA AVAILABLE FOR SQR.
13     000045      CF,HBR=37.      ;REQUEST FROM SQR TO HRLGET.
14     000046      CF,DGN=38.      ;DHR OR FOS DATA AVAILABLE FOR CCDIAG.
15     000047      CF,DMC=39.      ;DHR OR FOS BUFFER AVAILABLE TO CCOUT.
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
```

```

;
; G:MAC -- CONTROL COMPUTER PREFIX FILE.
;
;
; GLOBAL EVENT FLAG DEFINITIONS.
;
CF,COT=33.      ;DATA IN MSG FOR CCOUT.
CF,DHR=34.      ;DHR BUFFER NOW EMPTY FOR SQR.
CF,UPD=35.      ;SEND DIRECTIVE QUEUED TO DBUPD.
CF,HRL=36.      ;HRL DATA AVAILABLE FOR SQR.
CF,HBR=37.      ;REQUEST FROM SQR TO HRLGET.
CF,DGN=38.      ;DHR OR FOS DATA AVAILABLE FOR CCDIAG.
CF,DMC=39.      ;DHR OR FOS BUFFER AVAILABLE TO CCOUT.
;
;
; DIAGNOSTIC BIT DEFINITIONS.
;
DG,ERR=BIT9     ;DIAGNOSTIC MODE ERROR.
DG,SDF=BIT10    ;START DIAGNOSTIC CYCLE.
DG,TDF=BIT11    ;TERMINATE DIAGNOSTIC CYCLE.
MS,DGN=BIT12    ;SEND DIAGNOSTIC STATUS.
;
; SOB MACRO.
;
.MACRO SSOB, R,A.
    DEC R.
    BNE A.
.ENDM.
;
; FILE NUMBERS.
;
.PSECT FNOFFS,ABS.
FN,FSA: .BLKW 1 ;DP0:[7.5]TDCTA,FSA.
FN,FSB: .BLKW 1 ;DP0:[7.5]TDCTB,FSA.
FN,FSC: .BLKW 1 ;DP0:[7.5]TDCTC,FSA.
FN,QLS: .BLKW 1 ;DP0:[7.5]JEMATRIX,QLS.
FN,MHR: .BLKW 1 ;DP0:[7.5]HRL,MRG.
FN,UPD: .BLKW 1 ;DP0:[7.4]DBUPD,UPD.
FN,RDC: .BLKW 1 ;DP0:[7.4]RETRIV,DOC.
FN,ACK: .BLKW 1 ;DP0:[7.4]DBUPDT,ACK.
FN,FSD: .BLKW 1 ;DP0:[7.4]FOS,DAT.
FN,NMB:      ;LENGTH OF TABLE.
;
;
```



```

48      ;
49      ;
50      ;
51      ;
52      004000      TD#INL  - 2048      ;SIZE OF TD INPUT BUFFER
53      176344      TD#OAR  - 176344      ;TD OUTPUT DATA AVAILABLE REGISTER
54      176346      TD#OTR  - 176346      ;TD OUTPUT REGISTER
55      100000      T#OUTA  - 100000      ;OUTPUT AVAILABLE BIT
56      ;
57      ;
58      176366      TD#RST  - 176366      ;TD MASTER RESET REGISTER
59      ;
60      ;
61      176360      TD#CTW  - 176360      ;TD CONTROL REGISTER - WRITE
62      176370      TD#CTR  - 176370      ;TD CONTROL REGISTER - READ
63      100000      T#UBUS  - BIT15      ;UNIBUS SWITCH OPEN
64      040000      T#RSET  - BIT14      ;RESET TD
65      020000      T#SCLK  - BIT13      ;START FSA CLOCK
66      010000      T#EMEM  - BIT12      ;ENABLE MEMORY ERROR INTERRUPT
67      004000      T#MODE  - BIT11      ;LOAD/SEARCH MODE (1 = SEARCH)
68      002000      T#BTD   - BIT10      ;BINARY TEXT SEARCH DISABLE
69      000400      T#ICLK  - BIT9       ;SINGLE CYCLE FSA CLOCK
70      000200      T#DISK  - BIT7       ;SIMULATE DISK INPUT
71      000040      T#ICD   - BITS        ;ENABLE INTERFACE AND CONTROL DIAGNOSTICS
72      000020      T#BBEN  - BIT4       ;ENABLE 8-BIT BINARY OPERATION
73      000000      T#FSAA  - 000000      ;SELECT FSA GROUP - FSA A
74      000004      T#FSAB  - 000004      ;FSA B
75      000010      T#FSB2  - 000010      ;FSA B2
76      000014      T#FSAC  - 000014      ;FSA C
77      000000      T#SEG1  - 000000      ;STATE WORD SEGMENT - 1
78      000001      T#SEG2  - 000001      ;2
79      000002      T#SEG3  - 000002      ;3
80      ;
81      176372      TD#TAR  - 176372      ;TD TRANSFER ADDRESS - READ
82      176362      TD#TAW  - 176362      ;TD TRANSFER ADDRESS - WRITE
83      ;
84      ;
85      ;
86      000020      T#BT   - 20        ;BYTE TRANSLATOR
87      000022      T#SC   - 22        ;SECTOR COUNTER
88      000024      T#IBAR - 24        ;INPUT BUFFER READ ADDRESS REG
89      000026      T#IB   - 26        ;INPUT BUFFER
90      000030      T#BTAR - 30        ;BYTE TRANSLATOR WRITE ADDRESS REG
91      000032      T#OBWA - 32        ;OUTPUT BUFFER WRITE ADDRESS REG
92      000034      T#OBRA - 34        ;OUTPUT BUFFER READ ADDRESS REG
93      000036      T#OB   - 36        ;OUTPUT BUFFER
94      ;
95      176376      TD#SW   - 176376      ;STATUS WORD
96      100000      T#BSO  - BIT15      ;BUS SWITCH OPEN
97      040000      T#IBF  - BIT14      ;INPUT BUFFER FULL
98      020000      T#IBE  - BIT13      ;INPUT BUFFER EMPTY
99      010000      T#OBF  - BIT12      ;OUTPUT BUFFER FULL
100     004000      T#OBE  - BIT11      ;OUTPUT BUFFER EMPTY
101     002000      T#CLK  - BIT10      ;FSA CLOCK ON
102     000200      T#RBDO - BIT7       ;REPORT BUSY DELAY OFF
103     000100      T#CD   - BIT6       ;FSA C DONE
104     000040      T#RNB  - BITS        ;REPORT NOT BUSY

```

105	000020	T#AD	=	BIT4	:FSA-A-DONE
106	000010	T#BD	=	BIT3	:FSA-B-DONE
107	000004	T#DRD	=	BIT2	:DOC-REPORT-DONE
108	000002	T#BA	=	BIT1	:BYTE-AVAILABLE
109	000001	T#SO	=	BIT0	:SEARCH-ON
110		:			
111	176374	TD#TDR	=	176374	:TD-TRANSFER-DATA-READ
112	176364	TD#TDW	=	176364	:TD-TRANSFER-DATA-WRITE
113		:			
114	000270	TD#MEMI	=	270	:TD-MEMORY-ERROR-INT-VECTOR
115	000274	TD#QRDI	=	274	:TD-QR-DATA-AVAILABLE-INT-VECTOR
116		:			

.MAIN.. MACRO M1110 27-MAR-80 13:54 PAGE 8

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

118 000000

.PSECT.....

.MAIN. MAC M1110 27-MAR-80 13:54 PAGE 9

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1

000001

.END....

BITVAL = 000000	CF.DMC = 000047	N.SUNT = 000002	TD#CAR = 176344	T#MODE = 004000
BIT0 = 000001	CF.HBR = 000045	SR.ARE = 000114	002.TD#QTR = 176346	T#OB = 000036
BIT1 = 000002	CF.HRL = 000044	SR.ARS = 000106	002.TD#QRD = 000274	T#OBE = 004000
BIT10 = 002000	CF.UPD = 000043	SR.DAY = 000010	002.TD#RST = 176366	T#OBF = 010000
BIT11 = 004000	DBSLEN = 000116	SR.DLT = 000014	002.TD#SJW = 176376	T#OBRA = 000034
BIT12 = 010000	DG.ERR = 001000	SR.ECB = 000047	002.TD#STAR = 176372	T#OBWA = 000032
BIT13 = 020000	DG.SDF = 002000	SR.ECH = 000046	002.TD#TAM = 176362	T#OUTA = 100000
BIT14 = 040000	DG.TDF = 004000	SR.ECL = 000050	002.TD#TDR = 176374	T#RBD0 = 000200
BIT15 = 100000	FD.FID = 000000	003.SR.FIB = 000012	002.TD#TDW = 176364	T#RNB = 000040
BIT2 = 000004	FD.FNB = 000006	003.SR.GRE = 000100	002.T#AD = 000020	T#RSET = 040000
BIT3 = 000010	FD.FYR = 000004	003.SR.GRS = 000072	002.T#BA = 000002	T#SC = 000022
BIT4 = 000020	FD.LEN = 000010	003.SR.LEN = 000122	002.T#BD = 000010	T#SCLK = 020000
BIT5 = 000040	FN.ACK = 000016	004.SR.LIN = 000066	002.T#BS0 = 100000	T#SEG1 = 000000
BIT6 = 000100	FN.FSA = 000000	004.SR.LIP = 000062	002.T#BT = 000020	T#SEG2 = 000001
BIT7 = 000200	FN.FSB = 000002	004.SR.MON = 000006	002.T#BTAR = 000030	T#SEG3 = 000002
BIT8 = 000400	FN.FSD = 000004	004.SR.NDC = 000042	002.T#BTD = 002000	T#S0 = 000001
BIT9 = 001000	FN.FSD = 000020	004.SR.NDS = 000036	002.T#CD = 000100	T#SUBS = 100000
BYTE0 = 000000	FN.MHR = 000010	004.SR.NIN = 000030	002.T#CLK = 002000	T#ICLK = 000400
BYTE1 = 000001	FN.NMB = 000022	004.SR.NIP = 000022	002.T#DISK = 000200	T#IBEN = 000020
BYTE2 = 000002	FN.QLS = 000006	004.SR.SDB = 000032	002.T#DRD = 000004	WORD0 = 000000
BYTE3 = 000003	FN.RDC = 000014	004.SR.SRC = 000002	002.T#EMEM = 010000	WORD1 = 000002
BYTE4 = 000004	FN.UPD = 000012	004.SR.SUN = 000000	002.T#FSA = 000000	WORD2 = 000004
BYTE5 = 000005	MS.DGN = 010000	SR.TWS = 000056	002.T#FSAB = 000004	WORD3 = 000006
BYTE6 = 000006	N = 000012	SR.WSL = 000052	002.T#FSAC = 000014	WORD4 = 000010
BYTE7 = 000007	N.BFAC = 000004	SR.YR = 000004	002.T#FSB2 = 000010	WORD5 = 000012
BYTE8 = 000010	N.BHGH = 000006	SR.1IN = 000024	002.T#IB = 000026	WORD6 = 000014
BYTE9 = 000011	N.BTCH = 000004	SR.1IP = 000016	002.T#IBAR = 000024	WORD7 = 000016
BYTVAL = 000012	N.BUFB = 004000	TD#CTR = 176370	T#IBE = 020000	WORD8 = 000020
CF.COT = 000041	N.BUFW = 002000	TD#CTW = 176360	T#IBF = 040000	WORD9 = 000022
CF.DGN = 000046	N.FOS = 000764	TD#INL = 004000	T#ICD = 000040	WRDVAL = 000024
CF.DHR = 000042	N.QUERY = 000031	TD#MEM = 000270		

. ABS. 000000 000
000000 001
SRCOFF 000122 002
FDSCOF 000010 003
FNOFFS 000022 004
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 1735 WORDS (7 PAGES)
DYNAMIC MEMORY: 3068 WORDS (11 PAGES)
ELAPSED TIME: 00:00:12
;B~>=LIST.P.C.END

CCOM-...MACRO-M1110 27-MAR-80 14:22
TABLE OF CONTENTS

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8- 2- CONTROL-COMPUTER-COMMON-

```

1          .TITLE- GCOM
2          .SBTTL- CONTROL COMPUTER COMMON
3 000000   .PSECT- CCOM
4          ;
5          ; STATUS RECORD
6          ;
7 000000   122   103   .ASCII- /RC/ ; IDENTIFICATION WORD OF DMC EXCHANGE
8 000002
9 000002 000000
10 000004 000000
11 000006
12          ;
13          ;
14          ; SEARCH START/END TIME
15 000124   SHSTIM::.BLKW 8. ; START
16 000144   SHETIM::.BLKW 8. ; END
17          ;
18          ; FOS - FLU OCCURRENCE SUMMARY ACCUMULATION AREA
19          ;
20 000164 000000   FOSCTL::.WORD 0 ; NUMBER OF WORDS OF FOS USED
21 000166 000000   .WORD 0 ; NUMBER OF QUERIES IN BATCH
22 000170
23          FOS:: .BLKW N.BUFW
24          ;
25          ; MSQ - MEMORY SEND QUEUE OF CCOUT
26 004170 000000   MSQ:: .WORD 0
27          ;
28          ; DHR - DOCUMENT HIT REPORT BUFFER
29          ;
30 004172   DHR0:: .BLKW N.BUFW
31          ;
32          ; HRL BUFFER INTERFACE - HRLGET TO SQR
33          ;
34 010172 000001   .WORD 1 ; # OF HRL BUFFERS AVAILABLE FOR ROTATION
35 010174 002000   .WORD N.BUFW ; HRL BUFFER SIZE - WORDS
36 010176 010200   HRLBUF::.WORD HRL0 ; HRL BUFFER ADDRESSES
37          ;
38 010200   HRL0:: .BLKW N.BUFW
39 014200 040000   .WORD BIT14 ; TABLE TERMINATOR
40          ;
41          ;
42          ;

```

```
44 ;
45 ;
46 ; FILE DESCRIPTORS OF CONTROL TABLES
47 ;
48 014202 . FDSFSA::BLKW 3 ;TDCTA.FSA
49 014210 000000 .WORD FN.FSA
50 014212 . FDSFSB::BLKW 3 ;TDCTB.FSA
51 014220 000002 .WORD FN.FSB
52 014222 . FDSFSC::BLKW 3 ;TDCTC.FSA
53 014230 000004 .WORD FN.FSC
54 014232 . FDSQLS::BLKW 3 ;EMATRIX.QLS
55 014240 000006 .WORD FN.QLS
56 014242 . FDSMHR::BLKW 3 ;HRL.MRG
57 014250 000010 .WORD FN.MHR
58 ;
59 ;
60 ; BLDNFL.-BLDEFL DATA STRUCTURES
61 ;
62 ; FILE NAME/TYPE INDEX
63 ;
64 014252 014274' FNINDEX: .WORD FSNAM: ;FN.FSA
65 014254 014304' .WORD FSBNAM: ;FN.FSB
66 014256 014314' .WORD FSCNAM: ;FN.FSC
67 014260 014324' .WORD QLSNAM: ;FN.QLS
68 014262 014334' .WORD MHRNAM: ;FN.MHR
69 014264 014344' .WORD UPDNAM: ;FN.UPD
70 014266 014354' .WORD RDCNAM: ;FN.RDC
71 014270 014364' .WORD ACKNAM: ;FN.ACK
72 014272 014374' .WORD FSDNAM: ;FN.FSD
73 ;
74 ; FILE NAME/TYPE
75 ;
76 014274 076643 076450 000000 FSNAM: .RAD50 /TDCTA FSA/ ;FN.FSA
77 014302 024171 014304 076643 076520 000000 FSBNAM: .RAD50 /TDCTB FSA/ ;FN.FSB
78 014312 024171 014314 076643 076570 000000 FSCNAM: .RAD50 /TDCTC FSA/ ;FN.FSC
79 014322 024171 014324 020511 077731 113000 QLSNAM: .RAD50 /EMATRIX.QLS/ ;FN.QLS
80 014332 066063 014334 032334 000000 000000 MHRNAM: .RAD50 /HRL MRG/ ;FN.MHR
81 014342 052027 014344 014545 062240 000000 UPDNAM: .RAD50 /DBUPD UPD/ ;FN.UPD
82 014352 102704 014354 070534 070776 000000 RDCNAM: .RAD50 /RETRIV DOC/ ;FN.RDC
83 014362 015533 014364 014545 062264 000000 ACKNAM: .RAD50 /DBUPDT ACK/ ;FN.ACK
84 014372 003303 014374 023753 000000 000000 FSDNAM: .RAD50 /FOS DAT/ ;FN.FSD
85 014402 014474
86 ;
87 ; FILE DIRECTORY INDEX
88 ;
89 014404 014434' FDINDEX: .WORD DIRP75 ;FN.FSA
90 014406 014434' .WORD DIRP75 ;FN.FSB
91 014410 014434' .WORD DIRP75 ;FN.FSC
014412 014434' .WORD DIRP75 ;FN.QLS
```


92	014414	014434'		.WORD	DIRP75	:FN.MHR.
93	014416	014426'		.WORD	DIRP74	:FN.UPD.
94	014420	014426'		.WORD	DIRP74	:FN.RDC.
95	014422	014426'		.WORD	DIRP74	:FN.ACK.
96	014424	014426'		.WORD	DIRP74	:FN.FSD.
97						
98						
99						
100	014426					
101	014434					
102	014442					
103						
104						
105						
106	014450	104	120			
107	014452	104	120			
108	014454	104	120			
109	014456	104	120			
110	014460	104	120			
111	014462	104	120			
112	014464	104	120			
113	014466	104	120			
114	014470	104	120			
115						
116						
117						
118	014472	000000				
119	014474	000000				
120	014476	000000				
121	014500	000000				
122	014502	000000				
123	014504	000000				
124	014506	000000				
125	014510	000000				
126	014512	000000				
127						

 : FILE DIRECTORY FID'S
 :
DIRP74:::BLKW 3 :DP0:[7,4]
DIRP75:::BLKW 3 :DP0:[7,5]
DIRK75:::BLKW 3 :DK0:[7,5]
 :
 : DEVICE NAME INDEX
 :
DVINDEX: .ASCII /DP/ :FN.FSA
 :FN.FSB
 :FN.FSC
 :FN.QLS
 :FN.MHR
 :FN.UPD
 :FN.RDC
 :FN.ACK
 :FN.FSD
 :
 : DEVICE UNIT INDEX
 :
UNINDEX: .WORD 0 :FN.FSA
 :FN.FSB
 :FN.FSC
 :FN.QLS
 :FN.MHR
 :FN.UPD
 :FN.RDC
 :FN.ACK
 :FN.FSD
 :
:

```
129. ;
130. ;
131. ;
132. ; BUILD FILE NAME BLOCKS IN FDB
133. ;
134. ; .MCALL FDOF$L,FCSBT$
135 014514 FDOF$L
136 014514 FCSBT$
137. ;
138. ; BLDNFL-- BUILD FILE NAME BLOCK FOR NEW FILE
139. ;
140. ; INPUT: R0 - FDB ADDRESS
141. ; R1 - FILE NUMBER (FN,XXX)
142. ; OUTPUT: ALL REGISTERS PRESERVED
143. ;
144 014514 062700 000102 BLDNFL::ADD #F,FNB,R0 ;POINT TO FILE NAME BLOCK
145 014520 005060 000000 CLR N,FID(R0) ;CLEAR FID
146 014524 005060 000002 CLR N,FID+2(R0)
147 014530 005060 000004 CLR N,FID+4(R0)
148 014534 010246 MOV R2,-(SP) ;SAVE R2
149 014536 016102 014252* MOV FNINDX(R1),R2 ;FILE NAME/TYPE ADDRESS
150 014542 011260 000006 MOV (R2),N,FNAM(R0) ;FILE NAME/TYPE IN FDB
151 014546 016260 000002 000010 MOV 2(R2),N,FNAM+2(R0)
152 014554 016260 000004 000012 MOV 4(R2),N,FNAM+4(R0)
153 014562 016260 000006 000014 MOV 6(R2),N,FTYP(R0)
154 014570 005060 000016 CLR N,FVER(R0) ;VERSION IS ZERO
155 014574 005060 000020 CLR N,STAT(R0) ;CLEAR STATUS
156 014600 005060 000022 CLR N,NEXT(R0) ;CLEAR WILD CARD WORD
157 014604 016102 014404* MOV FDINDX(R1),R2 ;DIRECTORY FID ADDRESS
158 014610 011260 000024 MOV (R2),N,DID(R0) ;DIRECTORY FID IN FDB
159 014614 016260 000002 000026 MOV 2(R2),N,DID+2(R0)
160 014622 016260 000004 000030 MOV 4(R2),N,DID+4(R0)
161 014630 016160 014450* 000032 MOV DVINDX(R1),N,DVNM(R0) ;DEVICE NAME
162 014636 016160 014472* 000034 MOV UNINDX(R1),N,UNIT(R0) ;DEVICE UNIT
163 014644 012602 MOV (SP)+,R2 ;RESTORE R2
164 014646 162700 000102 SUB #F,FNB,R0 ;RESTORE R0
165 014652 000207 RTS PC ;RETURN
166. ;
167. ;
168. ; BLDEFL-- BUILD FILE NAME BLOCK FOR EXISTING FILE
169. ;
170. ; INPUT: R0 - FDB ADDRESS
171. ; R1 - FILE DESCRIPTOR ADDRESS
172. ; OUTPUT: ALL REGISTERS PRESERVED
173. ;
174 014654 010146 BLDEFL::MOV R1,-(SP) ;SAVE FDSC ADDRESS
175 014656 016101 000006 MOV FD,FNB(R1),R1 ;GET FILE # FROM FDSC
176 014662 CALL BLDNFL ;BUILD SKELETON FILE NAME BLOCK
177 014666 012601 MOV (SP)+,R1 ;RESTORE FDSC ADDRESS
178 014670 016160 000000 000102 MOV FD,FID(R1),F,FNB+N,FID(R0) ;ADD FID TO FILE NAME BLOCK
179 014676 016160 000002 000104 MOV FD,FID+2(R1),F,FNB+N,FID+2(R0)
180 014704 016160 000004 000120 MOV FD,FVR(R1),F,FNB+N,FVER(R0) ;ADD VERSION
181 014712 000207 RTS PC ;RETURN
182. ;
183. ;
184 014714 CCOMS2:
185. ;
```

CCOM: ... MAIL MI110 27-MAR-80 14:22 PAGE 10-1
CONTROL COMPUTER COMMON

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

186

000001

.END.

ACKNAM = 014364R	005	FA.POS = 010000	FOS = 000170RG	005	F.VBN = 000064	SR.MDN = 000006	002	
BITVAL = 000000		FA.RD = 000001	FOSCTL = 000164RG	005	F.VBSZ = 000060	SR.NDC = 000042	002	
BIT0 = 000001		FA.RWD = 004000	FO.APD = 000106		HRLBUF = 010176RG	005	SR.NDS = 000036	002
BIT1 = 000002		FA.SEQ = 040000	FO.MFY = 000002		HRL0 = 010200RG	005	SR.NIN = 000030	002
BIT10 = 002000		FA.SHR = 000040	FO.RD = 000001		MHRNAM = 014334R	005	SR.NIP = 000022	002
BIT11 = 004000		FA.TMP = 000020	FO.UPD = 000006		MSQ = 004170RG	005	SR.SDB = 000032	002
BIT12 = 010000		FA.UCK = 020000	FO.WRT = 000016		MS.DGN = 010000		SR.SRC = 000002	002
BIT13 = 020000		FA.WRT = 000002	FSANAM = 014274R	005	N = 000012		SR.SUN = 000000	002
BIT14 = 040000		FDINDX = 014404R	005	F.SBNAM = 014304R	005	NB.DEV = 000200	SR.TWS = 000055	002
BIT15 = 100000		FDSFSA = 014202RG	005	F.SCNAM = 014314R	005	NB.DIR = 000100	SR.WSL = 000052	002
BIT2 = 000004		FDSFSB = 014212RG	005	F.SDNAM = 014374R	005	NB.NAM = 000004	SR.YR = 000004	002
BIT3 = 000010		FDSFSC = 014222RG	005	F.ACTL = 000076		NB.SD1 = 000400	SR.IIN = 000024	002
BIT4 = 000020		FDSMHR = 014242RG	005	F.ALOC = 000040		NB.SD2 = 001000	SR.IIP = 000016	002
BIT5 = 000040		FDSOLS = 014232RG	005	F.BBFS = 000062		NB.SNM = 000040	SUNMR = 00002RG	005
BIT6 = 000100		FD.BLK = 000010		F.BDB = 000070		NB.STP = 000020	S.FATT = 000016	
BIT7 = 000200		FD.CCL = 000002		F.BGBC = 000057		NB.SVR = 000010	S.FDB = 000140	
BIT8 = 000400		FD.COM = 020000		F.BKDN = 000026		NB.TYP = 000002	S.FNAM = 000006	
BIT9 = 001000		FD.CR = 000002		F.BKDS = 000020		NB.VER = 000001	S.FNB = 000036	
BLDEFL = 014654RG	005	FD.DIR = 000010		F.BKEF = 000050		N.BFAC = 000004	S.FNBW = 000017	
BLDNFL = 014514RG	005	FD.FID = 000000	003	F.BKPI = 000051		N.BHGH = 000006	S.FNTY = 000004	
BYTE0 = 000000		FD.FNB = 000006	003	F.BKST = 000024		N.BTCH = 000004	S.FTYP = 000002	
BYTE1 = 000001		FD.FTN = 000001		F.BKVB = 000064		N.BUFB = 000400	S.FTEN = 000020	
BYTE2 = 000002		FD.FVR = 000004	003	F.CH = 000075		N.BUFW = 002000	TD*CTR = 176370	
BYTE3 = 000003		FD.F11 = 040000		F.CHNG = 000034		N.DID = 000024	TD*TLW = 176360	
BYTE4 = 000004		FD.INS = 000010		F.DFNB = 000046		N.DVNM = 000032	TD*INL = 004000	
BYTE5 = 000005		FD.ISP = 002000		F.DSPT = 000044		N.FID = 000000	TD*NEH = 000270	
BYTE6 = 000006		FD.LEN = 000010	003	F.DVNM = 000134		N.FNAM = 000006	TD*OAR = 176344	
BYTE7 = 000007		FD.MNT = 100000		F.EFBK = 000010		N.FOS = 000764	TD*OTR = 176346	
BYTE8 = 000010		FD.OSP = 004000		F.EFN = 000050		N.FVER = 000014	TD*QRD = 000274	
BYTE9 = 000011		FD.PLC = 000004		F.EOBB = 000032		N.NEXT = 000022	TD*IRST = 176366	
BYTVAL = 000012		FD.PRN = 000004		F.ERR = 000052		N.NURY = 000031	TD*SWJ = 176376	
CCOMSZ = 014714R	005	FD.PSE = 010000		F.FACC = 000043		N.STAT = 000020	TD*STAR = 176372	
CF.COT = 000041		FD.RAH = 000001		F.FBY = 000014		N.SUNT = 000002	TD*TAU = 176362	
CF.DGN = 000046		FD.RAN = 000002		F.FNAM = 000110		N.UNIT = 000034	TD*TDW = 176374	
CF.DHR = 000042		FD.REC = 000001		F.FNB = 000102		QLSNAM = 014324R	005	T*AD = 000020
CF.DMC = 000047		FD.RWM = 000001		F.FTYP = 000116		RDCNAM = 014354R	005	T*BA = 000002
CF.HBR = 000045		FD.SDI = 000020		F.FVER = 000120		R.FIX = 000001		T*BD = 000010
CF.HRL = 000044		FD.SOD = 000040		F.HIBK = 000004		R.SEQ = 000003		T*BSO = 100000
CF.UPD = 000043		FD.TTY = 000004		F.LUN = 000042		R.VAR = 000002		T*BT = 000020
CH.AND = 000001		FD.WBH = 000002		F.MBCT = 000054		SHTIM = 000144RG	005	T*BTAR = 000030
DBLEH = 000116		FF.CHR = 000005		F.MBC1 = 000055		SHTIM = 000124RG	005	T*BTDD = 002000
DBSTST = 000006RG	005	FF.NV = 000003		F.MBFG = 000056		SHTIME = 000004RG	005	T*CD = 000100
DG.ERR = 001000		FF.PDE = 000002		F.NRBD = 000024		SREC = 000002RG	005	T*CLK = 002000
DG.SDF = 002000		FF.RWD = 000001		F.NREC = 000030		SR.ARE = 000114	002	T*DISK = 000200
DG.TDF = 004000		FF.RWF = 000006		F.OVBS = 000030		SR.ARS = 000106	002	T*DRD = 000004
DHR0 = 004172RG	005	FF.SPC = 000004		F.RACC = 000016		SR.DAY = 000010	002	T*EMEM = 010000
DIRK75 = 014442RG	005	FNINDX = 014252R	005	F.RATT = 000001		SR.DLT = 000014	002	T*FSAA = 000000
DIRP74 = 014426RG	005	FN.ACK = 000016	004	F.RCNM = 000034		SR.ECB = 000047	002	T*FSAB = 000004
DIRP75 = 014434RG	005	FN.FSA = 000000	004	F.RCTL = 000017		SR.ECH = 000046	002	T*FSAC = 000014
DVINDX = 014450R	005	FN.FSB = 000002	004	F.RSIZ = 000002		SR.ECL = 000050	002	T*FSB2 = 000010
FA.APD = 000100		FN.FSC = 000004	004	F.RTYP = 000000		SR.FIB = 000012	002	T*IB = 000026
FA.CRE = 000010		FN.FSD = 000020	004	F.SDON = 000100		SR.GRE = 000100	002	T*IBAR = 000024
FA.DLK = 001000		FN.MHR = 000010	004	F.SPDV = 000072		SR.GRS = 000072	002	T*IBF = 000000
FA.ENB = 100000		FN.NMB = 000022	004	F.SPUN = 000074		SR.LEN = 000122	002	T*ICD = 000040
FA.EXC = 002000		FN.OLS = 000006	004	F.STBK = 000036		SR.LIN = 000066	002	T*MODE = 004000
FA.EXT = 000004		FN.RDC = 000014	004	F.UINT = 000136		SR.LTR = 000062		
FA.NSP = 000100		FN.UPD = 000012	004	F.URPD = 000020				

T#OB.. = 000036	T#RNB = 000040	T#SD. = 000001	WORD1 = 000002	WORD7 = 000016
T#OBE = 004000	T#RSET = 040000	T#UBUS = 100000	WORD2 = 000004	WORD8 = 000020
T#OBF = 010000	T#RSC = 000022	T#ICLK = 000400	WORD3 = 000006	WORD9 = 000022
T#OBRA = 000034	T#SCLK = 020000	T#BBEN = 000020	WORD4 = 000010	WRDVAL = 000024
T#OBWA = 000032	T#SEG1 = 000000	UNINDX = 014472R	005 WORD5 = 000012	...GEL = 000000
T#OUTA = 100000	T#SEG2 = 000001	UPDNAM = 014344R	005 WORD6 = 000014	...TPC = 000140
T#RBD0 = 000200	T#SEG3 = 000002	WORD0 = 000000		

. ABS. 000000 000
000000 001
SRCOFF 000122 002
FDSCOF 000010 003
FNDFFS 000022 004
CCOM 014714 005
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4151 WORDS (17 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:21
CCOM, 660M/SP, C, CCOM

CCIN...MAC...M1110 27-MAR-80 14:24
TABLE OF CONTENTS.

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	2	MACRO'S AND CONSTANTS.
9-	24	ASSEMBLY-TIME DATA DEFINITIONS.
10-	109	INIT THE DMC - READ HANG LOOP.
11-	140	STATUS RECORD REQUEST.
12-	149	CONTROL TABLES.
13-	182	DATA BASE UPDATES.
15-	315	SUBROUTINES.

```

1      .TITLE  GCIN
2      .SBTTL  MACRO'S AND CONSTANTS
3      ;
4      .MCALL  QIOW#C,QIOW$$,QIOW$,DIR$,EXIT$$
5      .MCALL  SETF$$,DECL$$,RGST#C,SDAT#C,MRKT#C,WTSE#C
6      .MCALL  FINIT$,FDAT#R,OFNB#W,WRITE$,WAIT$,CLOSE$
7      .MCALL  FDOF#L,FCSBT$,FDBDF$,FDRC#A,FDOP#A,FSRSZ$
8      .MCALL  FDBK#A,SPND$$
9      ;
10     .GLOBL  HRL0,MS0
11     .GLOBL  FDSFSA,FDSFSB,FDSFSC,FDSQLS,FDSMHR
12     .GLOBL  BLDNFL,.DLFNB
13     ;
14     ;LUNS
15     DPLUN=1
16     XNLUN=2
17     COLUN=6
18     ;
19     ;MISC EQUATES
20     EF:IO=1      ; I/O EVENT FLAG
21     DBSTAL=30   ; TIME TO WAIT BETWEEN CALLS TO DBUPD (TICKS)
22     ;

```

24
25 000000
26
27
28 000000
29 000004
30
31
32 000010
33
34
35 000040
36 000040
37
38
39 000040
40 000200
41 000200
42 000200
43
44 000200
45
46
47 000200 122 123
48 000202 000644'
49 000204 061 103
50 000206 000674'
51 000210 062 103
52 000212 000732'
53 000214 063 103
54 000216 000744'
55 000220 064 103
56 000222 000756'
57 000224 065 103
58 000226 000770'
59 000230 104 102
60 000232 001026'
61 000234 105 104
62 000236 001142'
63 000040
64
65
66 000240 000032
67 000242 103 103 111
000245 116 072 040
000250 104 115 103
000253 040 123 124
000256 101 122 124
000261 055 125 120
000264 040 106 101
000267 111 114 125
000272 122 105
68 000032
69
70
71 000274 000026
72 000276 103 103

```
.SBTTL ASSEMBLY-TIME DATA DEFINITIONS
.PSECT
;
; I/O STATUS BLOCKS
DMCIOS: .BLKW 2
SYIOS: .BLKW 2
;
; DMC QIO DPB
DMCRED: QIOW$ IO,RLB,XMLUN,EF,IO,DMCIOS,<HRL0,N,BUFB,BITTST>
;
;
; FDOF$L
; FCSBT$
;
; OUTPUT FILE FDB
OF FILE: FDBDF$
; FDBR$A: FD,RUM
; FDBK$A: HRL0,N,BUFB,EF,IO,SYIOS
; FDBP$A: DPLUN
;
; FRSZ$ 0
;
; EXCHANGE ID PARSE TABLE
EXCHID: .ASCII /RS/
; .WORD SRRCVD
; .ASCII /1C/
; .WORD C1RCVD
; .ASCII /2C/
; .WORD C2RCVD
; .ASCII /3C/
; .WORD C3RCVD
; .ASCII /4C/
; .WORD C4RCVD
; .ASCII /5C/
; .WORD C5RCVD
; .ASCII /DB/
; .WORD DBRCVD
; .ASCII /ED/
; .WORD DERCVD
EXCHLG=-EXCHID ; TABLE LENGTH
;
; ERROR MESSAGES
ERR1: .WORD ERR1L
ERR1T: .ASCII /CCIN: DMC START-UP FAILURE/
;
;
; ERR2: .WORD ERR2L
ERR2T: .ASCII /CCIN: DMC READ FAILURE/
```


000301	116	072	040	
000304	104	115	103	
000307	040	122	105	
000312	101	104	040	
000315	106	101	111	
000320	114	125	122	
000323	105			
73	000026			ERR2L=-ERR2T
74				.EVEN
75				:
76	000324	000037		ERR3: .WORD: ERR3L
77	000326	103	111	ERR3T: .ASCII: /CCIN: UNKNOWN EXCHANGE RECEIVED/
	000331	116	072	040
	000334	125	116	113
	000337	116	117	127
	000342	116	040	105
	000345	130	103	110
	000350	101	116	107
	000353	105	040	122
	000356	105	103	105
	000361	111	126	105
	000364	104		
78	000037			ERR3L=-ERR3T
79				.EVEN
80				:
81	000366	000020		ERR4: .WORD: ERR4L
82	000370	103	103	111
	000373	116	072	040
	000376	104	111	123
	000401	113	040	105
	000404	122	122	117
	000407	122		
83	000020			ERR4L=-ERR4T
84				.EVEN
85				:
86	000410	000026		ERR5: .WORD: ERR5L
87	000412	103	103	111
	000415	116	072	011
	000420	114	111	116
	000423	113	040	111
	000426	116	114	040
	000431	040	040	040
	000434	040	040	040
	000437	040		
88	000026			ERR5L=-ERR5T
89				.EVEN
90				:
91	000440	000024		ERR6: .WORD: ERR6L
92	000442	103	103	111
	000445	116	072	040
	000450	103	101	116
	000453	047	124	040
	000456	122	125	116
	000461	040	124	101
	000464	123	113	
93	000024			ERR6L=-ERR6T
94				.EVEN

```
95  
96  
97  
98 000466 000000  
99 000470 000000  
100 000472  
101  
102 000524 000000G  
103 000526 000002  
104 000530 000532  
105 000532 000000G  
106 000534 000004  
107 000536 000524  
:  
:  
:MISC LOCATIONS:  
BLKCNT: .WORD 0 ;EXCHANGE BLOCK COUNT  
FLAG: .WORD 0 ;BIT0: 1=CONTIG OUTPUT FILE REQUIRED  
SDATA: .BLKW 13 ;SEND DATA BUFFER  
:  
BITTST: .WORD HRL0 ;DOUBLE BUFFER CONTROL BLOCK  
          .WORD BIT1  
          .WORD +2  
          .WORD DHR0 ;SECOND BUFFER  
          .WORD BIT2  
          .WORD BITTST ;POINTER TO FIRST CONTROL BLOCK
```

```
109 .SBTTL INIT THE DMC - READ HANG LOOP
110 ;
111 000540 CCIN: FINIT$
112 000540 BIC #<BIT1!BIT2>,MSQ ;CLEAR DOUBLE BUFFER BUSY FLAGS
113 000544 042767 000006 000000G ;
114 ;
115 ;INIT DMC - WAIT FOR MASTER
116 000552 QIOW$C IO,TRM,XMLUN,EF,IO ;TERMINATE LINK
117 000560 INLDMC: QIOW$C IO,INL,XMLUN,EF,IO ;INIT LINK
118 000566 012700 000410* MOV #ERR5,R0 ;MSG TO CONSOLE -
119 000572 CALL COOUT ; LINK INITIALIZED
120 ;
121 ;
122 ;HANG A READ FOR THE NEXT EXCHANGE
123 000576 HANGRD: CALL READMC ;READ LINK
124 000602 103766 BCS INLDMC ;ERROR, RE-INITIALIZE
125 ;
126 ;PARSE EXCHANGE ID
127 000604 012701 000200* MSTRST: MOV #EXCHID,R1 ;PARSE TABLE ADDRESS
128 000610 012700 000010 MOV #EXCHLG/4,R0 ;TABLE LENGTH
129 000614 022177 177204 1#: CMP (R1)+,@DMCRED+Q,IOPL ;CHECK ID MATCH
130 000620 001001 BNE 2$ ;NO
131 000622 000131 JMP @(R1)+ ;GOT MATCH - GO TO ROUTINE
132 ;
133 000624 005721 2#: TST (R1)+ ;PAST ROUTINE WORD
134 000626 SSOB R0,1$ ;LOOP
135 000632 012700 000324* MOV #ERR3,R0 ;YES - REPORT ERROR
136 000636 CALL COOUT
137 000642 000755 BR HANGRD
138 ;
```

```
140 .SBTTL STATUS RECORD REQUEST  
141 ;  
142 ;STATUS RECORD REQUEST  
143 000644 052767 100000 000000G SRPCVD: BIS #BIT15.MSQ ;REQUEST CCOUT TO -  
144 000652 SETF$S #CF.COT ; SEND STATUS RECORD  
145 000664 DECL$S  
146 000672 000741 BR HANGRD ;GET NEXT EXCHANGE  
147 ;
```

```
.SBTTL - CONTROL TABLES
149
150
151
152 000674 052767 000001 177566 ; FIVE TYPES OF CONTROL TABLES ARE RECEIVED.
153 000702 012704 000000G C1RCVD: BIS #BIT0.FLAG ;NEED CONTIGUOUS FILE
154 000706 012701 000010 MOV #FDSMHR,R4 ;HRL.MRG FILE
155 000712 RCVCWT: CALL WRTFIL ;WRITE EXCHANGE TO FILE
156 000716 042767 000001 177544 BIC #BIT0.FLAG ;RESET CONTIG FILE FLAG
157 000724 103715 BCS INLDMC ;LINK ERROR - INIT
158 000726 000167 177644 JMP HANGRD ;GET NEXT EXCHANGE
159
160 000732 012704 000000G C2RCVD: MOV #FDSQLS,R4 ;EMATRIX.QLS FILE
161 000736 012701 000006 MOV #FN.QLS,R1
162 000742 000763 BR RCVCWT
163
164 000744 012704 000000G C3RCVD: MOV #FDSFSA,R4 ;TDCTA.FSA FILE
165 000750 012701 000000 MOV #FN.FSA,R1
166 000754 000756 BR RCVCWT
167
168 000756 012704 000000G C4RCVD: MOV #FDSFSB,R4 ;TDCTB.FSA FILE
169 000762 012701 000002 MOV #FN.FSB,R1
170 000766 000751 BR RCVCWT
171
172 000770 012704 000000G C5RCVD: MOV #FDSFSC,R4 ;TDCTC.FSA FILE
173 000774 012701 000004 MOV #FN.FSC,R1
174 001000 CALL WRTFIL ;WRITE EXCHANGE TO FILE
175 001004 103665 INLMP1: BCS INLDMC ;IF ERROR, DON'T CALL LOAD
176 001006 ROSTSC LOAD ;RUN LOAD
177 001014 103270 BCC HANGRD ;GET NEXT EXCHANGE
178 001016 012700 000440 TSKERR: MOV #ERRS,R0 ;CAN'T RUN TASK
179 001022 000167 000772 JMP ERROR
180
```

```
182. ; SBTTL DATA BASE UPDATES.
183. ;
184. ; INPUT BUFFERS AND SEND THEM TO DBUPD. IF THERE ARE NO FREE BUFFERS.
185. ; WAIT UNTIL DBUPD FREE'S UP A BUFFER.
186. ;
187. 001026 DBRCVD:
188. 001026 016702 176772 MOV. DMCRED+0,IOPL,R2. ;R2->INPUT BUFFER.
189. 001032 016202 000002 MOV. 2(R2),R2. ;R2=BLOCK INPUT COUNT.
190. 001036 016705 176766 MOV. DMCRED+0,IOPL+4,R5 ;R5->DOUBLE BUFFER CONTROL BLOCK.
191. ;
192. ; SEND BUFFER TO DBUPD.
193. ;
194. 001042 1$:
195. 001042 016567 000002 177422 MOV. 2(R5),SDATA. ;PUT BUFFER ID INTO SEND BLOCK.
196. 001050 056567 000002 000000G BIS. 2(R5),MSQ. ;SHOW BUFFER TO BE BUSY.
197. 001056 CALL. ACTDBU. ;ACTIVATE DATA BASE UPDATE (DBUPD)
198. 001062 005302 DEC. R2. ;DEC INPUT BUFFER COUNT.
199. 001064 003422 BLE. 5$. ;ALL DONE, RETURN TO MAIN.
200. ;
201. ; FIND NEXT EMPTY BUFFER.
202. ;
203. 001066 CALL. GETBUF. ;GET NEXT BUFFER FOR INPUT.
204. ;
205. ; EMPTY BUFFER FOUND. SET UP THE DMC INPUT TO FOR THE BUFFER.
206. ;
207. 001072 4$:
208. 001072 CALL. READMC. ;INPUT THE BUFFER.
209. 001076 103361 BCC. 1$. ;BUFFER INPUT OK, GO SEND IT TO DBUPD.
210. 001100 CALL. ERRSEQ. ;NOTIFY DBUPD THAT CURRENT UPDATE WILL
211. ; BE RESTARTED.
212. 001104 012767 000000G 176712 MOV. #HRL0,DMCRED+0,IOPL. ;PUT DEFAULT BUF ADDR INTO DIRECTIVE.
213. 001112 012767 000524* 176710 MOV. #BITTST,DMCRED+0,IOPL+4. ;PUT DEFAULT DOUB BUF CNTL INTO
214. ; DIRECTIVE
215. 001120 012767 000000G 176734 MOV. #HRL0,OF ILE+F,BKDS+2. ;PUT DEFAULT OUTPUT BUFFER INTO
216. ; FILE CONTROL BLOCK.
217. 001126 000167 177426 JMP. INLDMC. ;GO RE-INITIALIZE THE DMC LINK.
218. ;
219. ; GO RETURN TO THE MAIN CODE AND WAIT FOR THE NEXT DATA BASE UPDATE.
220. ;
221. 001132 5$:
222. 001132 CALL. GETBUF. ;GET NEXT BUFFER FOR INPUT.
223. 001136 000167 177434 JMP. HANGRD.
224. ;
225. ; END OF DATA BASE UPDATE.
226. ;
227. 001142 DBRCVD:
228. 001142 012767 000001 177322 MOV. #1,SDATA. ;SET END OF DATA BASE UPDATE
229. 001150 012767 000524* 176652 MOV. #BITTST,DMCRED+0,IOPL+4. ;PUT DEFAULT DOUBLE BUFFER
230. ; POINTER INTO QIO DIRECTIVE.
231. 001156 012767 000000G 176676 MOV. #HRL0,OF ILE+F,BKDS+2. ;PUT OUTPUT BUFFER POINTER
232. ; INTO FILE CONTROL BLOCK.
233. 001164 012767 000000G 176632 MOV. #HRL0,DMCRED+0,IOPL. ;SET DEFAULT BUFFER INPUT.
234. 001172 CALL. ACTDBU. ;TELL DBUPD ABOUT THE END.
235. 001176 000167 177374 JMP. HANGRD. ;GO WAIT FOR THE NEXT INPUT.
```

```
237 ;  
238 ; THIS SUBROUTINE ACTIVATES THE DBUPD TASK.  
239 ;  
240 ; NO INPUT REGISTERS  
241 ;  
242 001202 ACTDBU:  
243 001202 SDAT#C DBUPD,SDATA ;SEND BLOCK  
244 001210 ROST#C DBUPD  
245 001216 103006 BCC 2$  
246 001220 026727 0000000 0000000 CMP #DSW.#IE,ACT ;DBUPD ALREADY ACTIVE?  
247 001226 001402 BEQ 2$ ;YES, GO ON  
248 001230 000167 177562 JMP TSKERR ;NO, OUTPUT ERROR MESSAGE AND EXIT  
249 001234 2$:  
250 001234 SETF#$ #CF,UPD ;ACTIVATE DBUPD  
251 001246 DECL#$  
252 001254 000207 RTS PC  
253 ;  
254 ; THIS SUBROUTINE IS CALLED WHEN A LINK ERROR IS DETECTED BY CCIN.  
255 ; CCIN NOTIFIES DBUPD TO FORGET ABOUT CURRENT UPDATE EXCHANGE BY  
256 ; SENDING THE FOLLOWING SEQUENCE TO DBUPD IN THE NEXT BUFFER:  
257 ;  
258 ; WORD 0 - 000000  
259 ; 1 - 000000  
260 ; 2 - 000000  
261 ; 3 - 000000  
262 ; 4 - 000002  
263 ; 5 - 000000  
264 ;  
265 ; ON ENTRY, R5->BUFFER'S CONTROL BLOCK ENTRIES.  
266 ;  
267 001256 ERRSEQ: SAVE R4  
268 001260 011504 MOV (R5),R4 ;R4-> BUFFER  
269 001262 005014 CLR (R4)  
270 001264 005064 000002 CLR 2(R4)  
271 001270 005064 000004 CLR 4(R4)  
272 001274 005064 000006 CLR 6(R4)  
273 001300 012764 000002 000010 MOV #2,10(R4)  
274 001306 005064 000012 CLR 12(R4)  
275 ;  
276 ; SEND BUFFER TO DBUPD  
277 ;  
278 001312 016567 000002 177152 MOV 2(R5),SDATA ;PUT BUFFER ID INTO SEND BLOCK  
279 001320 056567 000002 0000000 BIS 2(R5),MSQ ;SHOW BUFFER TO BE BUSY  
280 001326 CALL ACTDBU ;CALL DBUPD  
281 001332 RESTOR R4  
282 001334 000207 RTS PC  
283 ;  
284 ; SUBROUTINE TO FIND AN EMPTY BUFFER IN GLOBAL COMMON. IF A BUFFER  
285 ; CANNOT BE FOUND THE TASK SUSPENDS UNTIL DBUPD RESUMES IT, WHEN  
286 ; A BUFFER IS FOUND R5-> TO ITS DOUBLE BUFFER CONTROL BLOCK,  
287 ;  
288 ; INPUTS:  
289 ; R5->ONE DOUBLE BUFFER CONTROL BLOCK  
290 ;  
291 001336 GETBUF:  
292 001336 6$:  
293 001336 012704 000002 MOV #2,R4 ;R4=LOOP COUNT
```

```
294 001342. 3$:  
295 001342. 036567 000002 000000G. BIT 2(R5).MS0. ;BUFFER-BUSY?  
296 001350 001410 BEQ 4$ ;NOT-BUSY, GO USE IT.  
297 001352 016505 000004 MOV 4(R5).R5 ;R5->NEXT-DOUBLE-BUFFER-CONTROL-BLOCK.  
298 001356 005304 DEC R4 ;GO-LOOK-FOR-THE-  
299 001360 003370 BGT 3$ ;NEXT-BUFFER.  
300 ;  
301 ; NO-BUFFER-WAS-FOUND, WAIT-UNTIL-DBUPD-RESUMES-CCIN-AND-THEN-GO-LOOK-FOR-  
302 ; AN-EMPTY-BUFFER.  
303 ;  
304 001362. SPND$S. ;SUSPEND  
305 001370 000762. BR 6$ ;RESUMES, NOW-GO-FIND-NON-BUSY-BUFFER.  
306 ;  
307 ; RETURN.  
308 ;  
309 001372. 4$:  
310 001372. 011567 176426 MOV (R5).DMCRED+0, IOPL. ;PUT-NEW-BUF-ADDR-INTO-DIRECTIVE.  
311 001376 010567 176426 MOV R5,DMCRED+0, IOPL+4 ;PUT-DOUBLE-BUFFER-CONTROL-POINTER-IP.  
312 001402 011567 176454 MOV (R5).OFILE+F.BKDS+2. ;PUT-NEW-INPUT-BUF-INTO-FILE-CTL-BLK.  
313 001406 000207 RTS PC
```



```
.SBTTL - SUBROUTINES.
:
:READ THE DMC SUBROUTINE.
READMC: DIR$ #DMCRED          :READ LINK
          BEQ  1$              :SUCCESS?
          CMPB #IS,SUC,DMCIOS  :YES
          BEQ  1$              :CONNECTION REJECTED?
          CMPB #IE,CNR,DMCIOS  :YES, RETURN INIT REQUEST
          BEQ  2$              :DEVICE NOT READY?
          CMPB #IE,DNR,DMCIOS  :YES, RETURN INIT REQUEST
          BEQ  2$              :REQUEST ABORTED?
          CMPB #IE,ABO,DMCIOS  :YES, RETURN INIT REQUEST
          BEQ  2$              :TIME OUT?
          CMPB #IE,TMO,DMCIOS  :YES
          BEQ  2$              :NO - ERROR EXIT
          MOV  #ERR2,R0
          BR   ERROR
:
1$:
          CLC
          RTS  PC
:
2$:
          SAVE R0,R1,R2
          MOV  #<ERRST+20>,R0  :R0-> OUTPUT STRING
          MOVB DMCIOS,R1       :R1 = ERROR CODE
          MOV  #1,R2           :DON'T SUPPRESS LEADING ZEROS
          CALL %CBDSG
          RESTOR R0,R1,R2
          SEC                  :REQUEST RE-INITIALIZATION
          RTS  PC              :RETURN
:
:WRITE EXCHANGE TO OUTPUT FILE SUBROUTINE.
:
:R4=ADDRESS OF WHERE TO BUILD FDSC.
:R1=FILE NUMBER.
:
WRTFIL:
          MOV  DMCRED+0,IOP,R2 ;R2-> INPUT BUFFER
          MOV  2(R2),R2        ;R2=BLOCK COUNT OF EXCHANGE
          MOV  R2,BLKCNT       : OF EXCHANGE
          MOV  #N,BUFB,R5      :BYTES PER BLOCK
          CLR  R3              :ACCUMULATOR
          ADD  R2,R3           :CALCULATE NUMBER OF -
          SUB  #512,,R5        : VIRTUAL BLOCKS IN -
          BNE 2$              : THE EXCHANGE
          BIT  #BIT0,FLAG      : CONTIGUOUS OUTPUT -
          BNE 1$              : FILE REQUIRED
          NEG  R3              :NON-CONTIG-OK
          FDAT#R #OFIL,.,,R3  :NUMBER OF BLOCKS TO ALLOCATE
:
          CALL BLDNFL          :BUILD FILE NAME BLOCK
          OFNB#W               :OPEN FILE
          BCS  FILERR          :OPEN FAILURE
:
WRTBLK: WRITE#                :WRITE A BLOCK -
          WAIT#                : AND WAIT
          CMPB #IS,SUC,DMCIOS  :SUCCESSFUL
          BEQ  1$
          BR   ERROR
```


SYMBOL TABLE

ACTDBU = 001202R	DMCRED = 000010R	FD.ISP = 002000	F.ERR = 000052	NB.VER = 000001
BITTST = 000524R	DPLUN = 000001	FD.LEN = 000010	F.FACC = 000043	N.BFAC = 000004
BITVAL = 000000	DROK = 001662R	FD.MNT = 100000	F.FFBY = 000014	N.BHGH = 000006
BIT0 = 000001	EF.ID = 000001	FD.OSP = 004000	F.FNAM = 000110	N.BTCH = 000004
BIT1 = 000002	ERROR = 002020R	FD.PLC = 000004	F.FNB = 000102	N.BUFB = 004000
BIT10 = 002000	ERRSEQ = 001256R	FD.PRN = 000004	F.FTYP = 000116	N.BUFW = 002000
BIT11 = 004000	ERR1 = 000240R	FD.PSE = 010000	F.FVER = 000120	N.DID = 000024
BIT12 = 010000	ERR1L = 000032	FD.RAH = 000001	F.HIBK = 000004	N.DVNM = 000032
BIT13 = 020000	ERR1T = 000242R	FD.RAN = 000002	F.LUN = 000042	N.FID = 000000
BIT14 = 040000	ERR2 = 000274R	FD.REC = 000001	F.MBCT = 000054	N.FNAM = 000006
BIT15 = 100000	ERR2L = 000026	FD.RUM = 000001	F.MBC1 = 000055	N.FOS = 000764
BIT2 = 000004	ERR2T = 000276R	FD.SDI = 000020	F.MBFG = 000056	N.FTYP = 000014
BIT3 = 000010	ERR3 = 000324R	FD.SOD = 000040	F.NRBD = 000024	N.FVER = 000016
BIT4 = 000020	ERR3L = 000037	FD.TTY = 000004	F.NREC = 000030	N.NEXT = 000022
BIT5 = 000040	ERR3T = 000326R	FD.WBH = 000002	F.OVBS = 000030	N.NURY = 000031
BIT6 = 000100	ERR4 = 000366R	FF.CHR = 000005	F.RACC = 000016	N.STAT = 000020
BIT7 = 000200	ERR4L = 000020	FF.NV = 000003	F.RATT = 000001	N.SUNT = 000002
BIT8 = 000400	ERR4T = 000370R	FF.POE = 000002	F.RCNI = 000034	N.SUNIT = 000034
BIT9 = 001000	ERR5 = 000410R	FF.RUD = 000001	F.RCTL = 000017	OFILE = 000040R
BLDNFL = ***** G	ERR5L = 000026	FF.RUF = 000006	F.RSIZ = 000002	PAR*** = 000027
BLKCNT = 000466R	ERRST = 000412R	FF.SPC = 000004	F.RTYP = 000000	Q.IOAE = 000012
BYTE0 = 000000	ERR6 = 000440R	FF.LERR = 001654R	F.SEGN = 000100	Q.IOEF = 000006
BYTE1 = 000001	ERR6L = 000024	FLAG = 000470R	F.SPDV = 000072	Q.IOFN = 000002
BYTE2 = 000002	ERR6T = 000442R	FN.ACK = 000016	F.SPUN = 000074	Q.IOLU = 000004
BYTE3 = 000003	EXCHID = 000200R	FN.FSA = 000000	F.STBK = 000036	Q.IOPL = 000014
BYTE4 = 000004	EXCHLG = 000040	FN.FSB = 000002	F.UNIT = 000136	Q.IOPR = 000007
BYTE5 = 000005	FA.APD = 000100	FN.FSC = 000004	F.URBD = 000020	Q.IOSB = 000010
BYTE6 = 000006	FA.CRE = 000010	FN.FSD = 000020	F.VBN = 000064	RCVCT = 000712R
BYTE7 = 000007	FA.DLK = 001000	FN.MHR = 000010	F.VBSZ = 000060	READMC = 001410R
BYTE8 = 000010	FA.ENB = 100000	FN.NMB = 000022	GETBUF = 001336R	R.FIX = 000001
BYTE9 = 000011	FA.EXC = 002000	FN.OLS = 000006	HANGRD = 000576R	R.QSGC = 000015
BYTVAL = 000012	FA.EXT = 000004	FN.RDC = 000014	HRLO = ***** G	R.QSPC = 000014
CCIN = 000540R	FA.NSP = 000100	FN.UPD = 000012	IE.ABO = ***** GX	R.QSPN = 000006
CF.COT = 000041	FA.POS = 010000	FO.APD = 000106	IE.ACT = ***** GX	R.OSPR = 000012
CF.DGN = 000046	FA.RD = 000001	FO.MFY = 000002	IE.CNR = ***** GX	R.QSTN = 000002
CF.DHR = 000042	FA.RWD = 004000	FO.RD = 000001	IE.DNR = ***** GX	R.SEO = 000003
CF.DMC = 000047	FA.SEQ = 040000	FO.UPD = 000006	IE.TMD = ***** GX	R.VAR = 000002
CF.HBR = 000045	FA.SHR = 000040	FO.WRT = 000016	INLDMC = 000560R	SDATA = 000472R
CF.HRL = 000044	FA.TMP = 000020	F.ACTL = 000076	INLMP1 = 001004R	SRRCVD = 000644R
CF.UPD = 000043	FA.UCK = 020000	F.ALOC = 000040	IO.INL = ***** GX	SR.ARE = 000114
CH.AND = 000001	FA.WRT = 000002	F.BBFS = 000052	IO.RLB = ***** GX	SR.LARS = 000106
COLUN = 000006	FDSFSA = ***** G	F.BDB = 000070	IO.TRM = ***** GX	SR.LDAY = 000010
COOUT = 001746R	FDSFSB = ***** G	F.BGBC = 000057	IO.WLB = ***** GX	SR.DLT = 000014
C1RCVD = 000674R	FDSFSC = ***** G	F.BKDN = 000026	IS.SUC = ***** GX	SR.ECB = 000047
C2RCVD = 000732R	FDSNHR = ***** G	F.BKDS = 000020	MSQ = ***** G	SR.ECH = 000046
C3RCVD = 000744R	FDSOLS = ***** G	F.BKEF = 000050	MSTRST = 000604R	SR.ECL = 000050
C4RCVD = 000756R	FD.BLK = 000010	F.BKFI = 000051	MS.DGN = 010000	SR.FIB = 000012
C5RCVD = 000770R	FD.CCL = 000002	F.BKST = 000024	N = 000012	SR.GRE = 000100
DBRCVD = 001026R	FD.COM = 020000	F.BKVB = 000064	NB.DEV = 000200	SR.GRS = 000072
DBSLEN = 000116	FD.CR = 000002	F.CHR = 000075	NB.DIR = 000100	SR.LEN = 000122
DBSTAL = 000036	FD.DIR = 000010	F.CNTG = 000034	NB.NAM = 000004	SR.LIH = 000066
DERCVD = 001142R	FD.FID = 000000	F.DFNB = 000046	NB.SDI = 000400	SR.LIP = 000062
DG.ERR = 001000	FD.FNB = 000006	F.DSPT = 000044	NB.SDZ = 001000	SR.MON = 000006
DG.SDF = 002000	FD.FTN = 000001	F.DVNM = 000134	NB.SNM = 000040	SR.NDC = 000042
DG.TDF = 004000	FD.FYR = 000004	F.EFBK = 000010	NB.STP = 000020	SR.NDS = 000036
DHR0 = ***** GX	FD.F11 = 040000	F.EFN = 000050	NB.SVR = 000010	SR.NIN = 000030
DMCIOS = 000000R	FD.INS = 000010	F.EOBB = 000032	NB.TYP = 000002	SR.NIP = 000022

SR.SDB. 000032.	002.TD#CTW. 176360	T#CLK. - 002000	T#RNB. - 000040	WRDVAL. 000024
SR.SRC. 000002.	002.TD#INL. 004000	T#DISK. 000200	T#RSET. 040000	WRIBLK. 001634R.
SR.SUN. 000000	002.TD#MEM. 000270	T#DRD. - 000004	T#SC. - 000022.	WRIFIL. 001540R.
SR.TWS. 000056	002.TD#QAR. 176344	T#EMEM. 010000	T#SCLK. 020000	XMLUN. 000002
SR.WSL. 000052.	002.TD#QTR. 176346	T#FSAR. 000000	T#SEG1. 000000	#CDSG. ***** GX.
SR.YR. 000004	002.TD#QRD. 000274	T#FSAB. 000004	T#SEG2. 000001	#DSW. ***** GX.
SR.IIN. 000024	002.TD#RST. 176366	T#FSAC. 000014	T#SEG3. 000002.	###. 000110R. 006
SR.IIP. 000016	002.TD#SU. 176376	T#FSB2. 000010	T#SO. - 000001	###ARG. 000002
SYIOS. 000004R.	TD#TAR. 176372.	T#IB. - 000026	T#UBUS. 100000	###OST. 000012
S.BFHD. 000020	TD#TAW. 176362.	T#IBAR. 000024	T#ICLK. 000400	###T1. 000005
S.DABA. 000006	TD#TDR. 176374	T#IBE. - 020000	T#BBEN. 000020	.CLOSE. ***** G.
S.DAEF. 000010	TD#TDW. 176364	T#IBF. - 040000	WORD0. 000000	.DLFNB. ***** G.
S.DATN. 000002.	TSKERR. 001016R.	T#ICD. - 000040	WORD1. 000002.	.FINIT. ***** G.
S.FATT. 000016	T#AD. - 000020	T#MODE. 000400	WORD2. 000004.	.FSRCB. ***** G.
S.FDB. 000140	T#BA. - 000002.	T#OB. - 000036	WORD3. 000006	.OPFNB. ***** G.
S.FNAM. 000006	T#BD. - 000010	T#OBE. 000400	WORD4. 000010	.WAIT. ***** G.
S.FNB. 000036	T#BSO. - 100000	T#OBF. - 010000	WORDS. 000012.	.WRITE. ***** G.
S.FNBW. 000017	T#BT. - 000020	T#OBRA. 000034	WORD6. 000014	...GBL. 000000
S.FNTY. 000004	T#BTAR. 000030	T#OBUA. 000032.	WORD7. 000016	...PC1. 000040R.
S.FTYP. 000002.	T#BTD. - 002000	T#OUTA. 100000	WORD8. 000020	...PC2. 000200R.
S.NFEN. 000020	T#CD. - 000100	T#RBD0. 000200	WORD9. 000022.	...TPC. 000020
TD#CTR. 176370				

.ABS. 000000 000
002032. 001
SRCOFF. 000122. 002.
FDSCOF. 000010 003
FNDFFS. 000022. 004
###FSR1 000000 005
#DPB## 000126 006
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 7594 WORDS. (30 PAGES)
DYNAMIC MEMORY: 9140 WORDS. (35 PAGES)
ELAPSED TIME: 00:00:54
CCIN,CCIN/-SP=C20.1JP.C.CCIN.

CCOUT... MAC M1110 27-MAR-80 14:25
TABLE OF CONTENTS.

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	2	MACRO'S AND CONSTANTS.
9-	21	DATA AREA.
10-	65	WAIT FOR WORK - SEE WHAT IT IS.
11-	89	SEND MSG WORK.
12-	126	SEND SPOOL FILES.
13-	190	TRANSMIT EXCHANGE SUBROUTINE.

```

1      .TITLE  CCOUT...
2      .SBTTL  MACRO'S AND CONSTANTS
3      ;
4      .MCALL  FDOF$L,FCSBT$,FDBDF$,FDRC$A,FDBK$A,FDOF$A,FSRSZ$
5      .MCALL  WTSE$C,CLEF$C,RCVD$C,SETF$C,DECL$S
6      .MCALL  CLOSE$,OFNB$R,READ$,WAIT$,FINIT$
7      .MCALL  MRKT$C,WTSE$C
8      .MCALL  QIOW$S,EXIT$S
9      ;
10     .GLOBL  .DLFNB,DHR0,BLDEFL,$DDIV,MSQ,SREC
11     ;
12     ;LUNS
13     DPLUN=1
14     X1LUN=2
15     COLUN=6
16     ;
17     ;MISC EQUATES
18     EF:IO=1      ;I/O EVENT FLAG
19     ;

```

```

21                                     .SBTTL DATA AREA
22                                     ;
23                                     ;SPOOL FILE FDB
24 000000                                FDOF$#L
25 000000                                FCSBT$#
26                                     ;
27 000000                                FDB:  FDBDF$#
28 000140                                FDRCS$#  FD,RUM
29 000140                                FDBK$#A  FOS,N,BUFB,,EF,IO,IOSTAT
30 000140                                FDBP$#A  DPLUN
31                                     ;
32 000140                                FRSRZ$#  0
33                                     ;
34                                     ;MISC LOCATIONS
35 000140                                IOSTAT: .BLKW 2:           ;I/O STATUS BLOCK
36 000144                                RDATA: .BLKW 15.        ;RECEIVE DATA AREA
37                                     ;
38                                     ;SU ** DONE EXCHANGE
39 000202 130 123                        SUXDON: .ASCII /XS/    ;IDENTIFICATION WORD
40                                     ;
41                                     ; DBUPD DONNE EXCHANGE
42                                     ;
43 000204 104 125                        DBUDGN: .ASCII /DU/    ;UPDATE DONE ID WORD
44                                     ;
45                                     ;
46                                     ;ERROR MESSAGES
47 000206 000021                          ERR1:  .WORD  ERR1L
48 000210 103 117                          ERR1T: .ASCII /CCOUT: DISK ERROR/
49                                     ;
50                                     ;
51                                     ;
52 000232 000030                          ERR2:  .WORD  ERR2L
53 000234 103 117                          ERR2T: .ASCII /CCOUT: DMC WRITE FAILURE/
54                                     ;
55                                     ;
56                                     ;
57 000264 000031                          ERR3:  .WORD  ERR3L
58 000266 103 117                          ERR3T: .ASCII /CCOUT: LINK INL
59 000271 125 072
60 000274 040 111
61 000277 116 040
62 000302 111 114
63 000305 040 040
64 000310 040 040
65 000313 040 040
```

```
000316 040  
59 000031 ERR3L=-ERR3T  
60 .EVEN  
61 ;  
62 ;  
63 ;
```



```
65 .SBTTL- WAIT FOR WORK - SEE WHAT IT IS
66 ;
67 000320 ; CCOUT:
68 000320 ; FINIT$
69 ;
70 ; WAIT FOR WORK
71 000324 ; WAITHG: WTSE$C CF.COT ; WAIT
72 000332 ; CLEF$C CF.COT ; GOT IT
73 ;
74 ; SEE WHAT WAS QUEUED
75 000340 032767 040000 000000G T$TURK: BIT #BIT14.MSQ ; WORK IN MSQ?
76 000346 ; BON SNDSUX ; SEND SU X$DONE
77 000350 032767 100000 000000G ; BIT #BIT15.MSQ
78 000356 ; BON SNDSRC ; SEND STATUS RECORD
79 000360 032767 020000 000000G ; BIT #BIT13.MSQ ; $BUP DONE?
80 000366 ; BON SNDEND ; BRANCH IF YES
81 000370 032767 000001 000000G ; BIT #BIT0.MSQ
82 000376 ; BON SNDDHR ; SEND DHR0
83 ;
84 000400 ; RCVD$C .RDATA ; PACKET QUEUED?
85 000406 103746 ; BCS WAITHG ; NO WORK NOW
86 000410 000454 ; BR SNDSPL ; SEND SPOOL FILE
87 ;
```

```
.SBTTL SEND MSQ.WORK.
;
90 ;
91 ;SEND.SU.XX.DONE.
92 000412 012705 000202' SNDSUX: MOV. #SUXDON,R5 ;EXCHANGE ADDRESS.
93 000416 CALL. XMTDMC ;SEND IT.
94 000422 103773 BCS. SNDSUX ;SOFT ERROR - RESEND
95 000424 042767 040000 000000G. BIC. #BIT14,MSQ. ;SHOW IT'S SENT.
96 000432 000742 BR. TSTWRK. ;GET NEXT WORK.
97 ;
98 ;
99 ;SEND.STATUS.RECORD.
100 000434 012705 177766G. SNDSRC: MOV. #SREC-2,R5 ;EXCHANGE ADDRESS.
101 000440 CALL. XMTDMC ;SEND IT.
102 000444 103773 BCS. SNDSRC ;SOFT ERROR - RESEND
103 000446 042767 100000 000000G. BIC. #BIT15,MSQ. ;SHOW IT'S SENT.
104 000454 000731 BR. TSTWRK. ;GET NEXT WORK.
105 ;
106 ;
107 ;SEND.DHR0
108 000456 012705 000000G. SNDDHR: MOV. #DHR0,R5 ;EXCHANGE ADDRESS.
109 000462 012715 042110 MOV. #PHD,(R5) ;IDENTIFICATION WORD
110 000466 CALL. XMTDMC ;SEND IT.
111 000472 103771 BCS. SNDDHR ;SOFT ERROR - RESEND
112 000474 042767 000001 000000G. BIC. #BIT0,MSQ. ;SHOW ITS SENT.
113 000502 SETF#C CF,DHR. ;INFORM.SOR.THAT.DHR0 -
114 000510 DECL$S ; IS NOW AVAILABLE.
115 000516 000710 BR. TSTWRK. ;GET NEXT WORK.
116 ;
117 ; NOTIFY MSCHED THAT DBUPD IS DONE.
118 ;
119 000520 012705 000204' SNDEND: MOV. #DBUDON,R5 ;RS->EXCHANGE ID.
120 000524 CALL. XMTDMC ;SEND EXCHANGE.
121 000530 103773 BCS. SNDEND ;SOFT ERROR - RESEND
122 000532 042767 020000 000000G. BIC. #BIT13,MSQ. ;SHOW IT'S SENT.
123 000540 000677 BR. TSTWRK. ;GET NEXT WORK.
124 ;
```

```
126 .SBTTL SEND SPOOL FILES
127 ;
128 ;SEND SPOOL FILE
129 000542 012700 000000' SNDSPL: MOV #FDB,R0 ;SPOOL FILE FDB
130 000546 012701 000152' MOV #RDATA+6,R1 ;FDB ADDRESS
131 000552 CALL BLDEFL ;BUILD FNB
132 000556 OFNB#R ;OPEN SPOOL FILE
133 000570 103442 BCS DSKERR ;ERROR
134 ;
135 000572 016001 000010 MOV F,EFBK(R0),R1 ;# OF VIRTUAL BLOCKS
136 000576 016002 000012 MOV F,EFBK+2(R0),R2
137 000602 162702 000001 SUB #1,R2
138 000606 005601 SBC R1
139 000610 012700 000004 MOV #N,BFAC,R0 ;CALCULATE NUMBER OF
140 000614 CALL $DDIV ; EXCHANGE BLOCKS
141 000620 005700 TST R0 ;REMAINDER?
142 000622 001025 BNE DSKERR ;NOT MULTIPLE OF N,BFAC
143 000624 005701 TST R1 ;HIGH ORDER?
144 000626 001023 BNE DSKERR ;TOO BIG
145 000630 005702 TST R2 ;QUOTIENT
146 000632 001421 BEQ DSKERR ;EMPTY FILE
147 ;
148 000634 READ$ #FDB ;READ FIRST BLOCK
149 000644 WAIT$
150 000650 010267 000002G MOV R2,FOS+2 ;STUFF BLOCK COUNT
151 000654 000404 BR TSTRED ;CONTINUE
152 ;
153 000656 REDFIL: READ$ ;READ NEXT BLOCK
154 000662 WAIT$
155 000666 122767 000000G 177244 TSTRED: CHFB #IS,SUC,I0STAT ;GOOD READ?
156 000674 001403 BEQ DSKOK ;YES
157 000676 012700 000206' DSKERR: MOV #ERR1,R0 ;ERROR
158 000702 000420 BR ERROR
159 ;
160 000704 012705 000000G DSKOK: MOV #FOS,R5 ;EXCHANGE ADDRESS
161 000710 CALL XMTDMC ;SEND IT
162 000714 103005 BCC 1$ ;BUFFER SENT
163 ;
164 000716 CLOSE$ #FDB ;ERROR - CLOSE FILE
165 000726 000705 BR SNDSPL ;RESEND IT
166 ;
167 000730 005302 1$: DEC R2 ;LOOP IF MORE BLOCKS
168 000732 001351 BNE REDFIL ; TO SEND
169 ;
170 000734 CALL .DLFNB ;DELETE SPOOL FILE
171 000740 000167 177374 JMP TSTWRK ;GET NEXT WORK
172 ;
173 ;
174 ;ERROR EXITS
175 ;
176 ;R0=MSG BLOCK ADDRESS
177 000744 ERROR: CALL COOUT ;MSG TO CONSOLE
178 000750 EXIT$
179 ;
180 ;
181 ;SEND MSG TO CONSOLE
182 ;
```

CCOUT: ... MACRO M1110 27-MAR-80 14:25 PAGE 12-1
SEND SPOOL FILES

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
183 ;R0=MSG BLOCK ADDRESS  
184 000756 012001 COOUT: MOV (R0)+,R1 ;MESSAGE ADDRESS AND LENGTH  
185 000760 QIO#S #IO,ULB,#COLUN,#EF,IO,...<R0,R1,#40> ;MSG TO CONSOLE  
186 001026 000207 RTS PC  
187 ;  
188 ;
```

```
.SBTTL TRANSMIT EXCHANGE SUBROUTINE
:
:SEND BUFFER ON THE DMC
:
:INPUT: R5= BUFFER ADDRESS
:
XMTDMC: Q10W$S #IO, WLB, *XMLUN, #EF, IO, #IOSTAT, <R5, #N, BUFB> :SEND
196 001030 CMPB #IS, SUC, IOSTAT :ERROR?
197 001100 122767 000000G 177032 BEQ 1$ :NO
198 001106 001424 BEQ #IE, CHR, IOSTAT :RECOVERABLE ERROR?
199 001110 122767 000000G 177022 CMPB #IE, DNR, IOSTAT
200 001116 001422 BEQ LNKRST
201 001120 122767 000000G 177012 CMPB #IE, TMO, IOSTAT
202 001126 001416 BEQ LNKRST
203 001130 122767 000000G 177002 CMPB #IE, ABO, IOSTAT
204 001136 001412 BEQ LNKRST
205 001140 122767 000000G 176772 CMPB #ERR2, R0
206 001146 001406 BEQ LNKRST :ERROR MSG AND EXIT
207 001150 012700 000232' MOV #ERR2, R0
208 001154 000167 177564 JMP ERROR
209
:
1$: CLC
210 001160 000241 RTS PC :GOOD RETURN
211 001162 000207
:
:RECOVERABLE LINK ERROR
LNKRST: SAVE R0, R1, R2
215 001164 MOV #<ERR3+21>, R0 :R0->OUTPUT STRING
216 001172 012700 000307' MOVB IOSTAT, R1 :R1 = ERROR CODE
217 001176 116701 176736 MOV #1, R2 :DON'T SUPPRESS LEADING ZEROS
218 001202 012702 000001 CALL $CDBDSC
219 001206 RESTOR R0, R1, R2
220 001212 MOV #ERR3, R0 :MSG TO CONSOLE
221 001220 012700 000264' CALL COOUT
222 001224 MRKT$C EF, IO, 60, 1 :WAIT FOR OTHER END TO RECOVER
223 001230 WTSE$C EF, IO
224 001236 SEC :SHOW ERROR
225 001244 000261 RTS PC
226 001246 000207
:
:
:
:END CCOUT
```

BITVAL = 000000	ERR2T = 000234R	FN.MHR = 000010	004 F.VBN = 000064	SREC = ***** G.
BIT0 = 000001	ERR3 = 000264R	FN.NMB = 000022	004 F.VBSZ = 000060	SR.ARE = 000114 002
BIT1 = 000002	ERR3L = 000031	FN.QLS = 000006	004 IE.ABO = ***** GX	SR.ARS = 000106 002
BIT10 = 002000	ERR3T = 000266R	FN.RDC = 000014	004 IE.CNR = ***** GX	SR.DAY = 000010 002
BIT11 = 004000	FA.APD = 000100	FN.UPD = 000012	004 IE.DNR = ***** GX	SR.DLT = 000014 002
BIT12 = 010000	FA.CRE = 000010	FOS = ***** GX	IE.TMO = ***** GX	SR.ECB = 000047 002
BIT13 = 020000	FA.DLK = 001000	FO.APD = 000106	Iostat = 000140R	SR.ECH = 000046 002
BIT14 = 040000	FA.ENB = 100000	FO.MFY = 000002	IO.WLB = ***** GX	SR.ECL = 000050 002
BIT15 = 100000	FA.EXC = 002000	FO.RD = 000001	IS.SUC = ***** GX	SR.FIB = 000012 002
BIT2 = 000004	FA.EXT = 000004	FO.UPD = 000006	LNKRST = 001164R	SR.GRE = 000100 002
BIT3 = 000010	FA.NSP = 000100	FO.WRT = 000016	MSQ = ***** G.	SR.GRS = 000072 002
BIT4 = 000020	FA.POS = 010000	F.ACTL = 000076	MS.DGN = 010000	SR.LEN = 000122 002
BIT5 = 000040	FA.RD = 000001	F.ALCC = 000040	M.KTAE = 000010	SR.LIN = 000066 002
BIT6 = 000100	FA.RWD = 004000	F.BBFS = 000062	M.KTEF = 000002	SR.LIP = 000062 002
BIT7 = 000200	FA.SEQ = 000000	F.BDB = 000070	M.KTMG = 000004	SR.MON = 000006 002
BIT8 = 000400	FA.SHR = 000040	F.BGBC = 000057	M.KTUN = 000006	SR.NDC = 000042 002
BIT9 = 001000	FA.TMP = 000020	F.BKDN = 000026	N = 000012	SR.NDS = 000036 002
BLDEFL = ***** G.	FA.UCK = 020000	F.BKDS = 000020	NB.DEV = 000200	SR.NIN = 000030 002
BYTE0 = 000000	FA.WRT = 000002	F.BKEF = 000050	NB.DIR = 000100	SR.NIP = 000022 002
BYTE1 = 000001	FDB = 000000R	F.BKPI = 000051	NB.NAM = 000004	SR.SDB = 000032 002
BYTE2 = 000002	FD.BLK = 000010	F.BKST = 000024	NB.SD1 = 000400	SR.SRC = 000002 002
BYTE3 = 000003	FD.CCL = 000002	F.BKVB = 000054	NB.SD2 = 001000	SR.SUN = 000000 002
BYTE4 = 000004	FD.COM = 020000	F.CHRC = 000075	NB.SNI = 000040	SR.TWS = 000056 002
BYTE5 = 000005	FD.CR = 000002	F.CNTG = 000034	NB.STP = 000020	SR.UJL = 000052 002
BYTE6 = 000006	FD.DIR = 000010	F.DFNB = 000046	NB.SVR = 000010	SR.YR = 000004 002
BYTE7 = 000007	FD.FID = 000000	003 F.DSPT = 000044	NB.TYP = 000002	SR.IIN = 000024 002
BYTE8 = 000010	FD.FNB = 000006	003 F.DVNM = 000134	NB.VER = 000001	SR.IIP = 000016 002
BYTE9 = 000011	FD.FTN = 000001	003 F.EFBK = 000010	N.BFAC = 000004	SUXDON = 000202R
BYTVAL = 000012	FD.FVR = 000004	003 F.EOBB = 000032	N.BHGH = 000006	S.BFHD = 000020
CCOUT = 000320R	FD.F11 = 040000	F.ERR = 000052	N.BTCH = 000004	S.ETEF = 000002
CF.COT = 000041	FD.INS = 000010	F.FACC = 000043	N.BUFB = 004000	S.FATT = 000016
CF.DGN = 000046	FD.ISP = 002000	003 F.FBFC = 000014	N.BUFW = 002000	S.FDB = 000140
CF.DHR = 000042	FD.LEN = 000010	F.FNAM = 000110	N.DID = 000024	S.FNAM = 000006
CF.DMC = 000047	FD.MNT = 100000	F.FNB = 000102	N.DVNM = 000032	S.FNB = 000036
CF.HBR = 000045	FD.OSP = 004000	F.FNFB = 000116	N.FID = 000000	S.FNBW = 000017
CF.HRL = 000044	FD.PLC = 000004	F.FTYP = 000116	N.FNAM = 000006	S.FNTY = 000004
CF.UPD = 000043	FD.PRN = 000004	F.FVER = 000120	N.FOS = 000764	S.FTYP = 000002
CH.AND = 000001	FD.PSE = 010000	F.HIBK = 000004	N.FTYP = 000014	S.NFEN = 000020
COLUN = 000006	FD.RAH = 000001	F.LUN = 000042	N.FVER = 000016	TD#CTR = 176370
CCOUT = 000756R	FD.RAN = 000002	F.MBCT = 000054	N.NEXT = 000022	TD#CTW = 176360
C.LEEF = 000002	FD.REC = 000001	F.MBCI = 000055	N.QUERY = 000031	TD#INL = 004000
DBSLEN = 000116	FD.RWM = 000001	F.MBFG = 000056	N.STAT = 000020	TD#MEM = 000270
DBUDDN = 000204R	FD.SDI = 000020	F.NRBD = 000024	N.SUNT = 000002	TD#OAR = 176344
DG.ERR = 001000	FD.SDI = 000040	F.OVBS = 000030	N.UNIT = 000034	TD#QTR = 176346
DG.SDF = 002000	FD.TTY = 000004	F.OVBS = 000030	PAR\$\$\$ = 000027	TD#ORD = 000274
DG.TDF = 004000	FD.WBH = 000002	F.RABS = 000016	RDATA = 000144R	TD#RST = 176366
DHR0 = ***** G.	FF.CHR = 000005	F.RATT = 000001	REDFIL = 000656R	TD\$SU = 176376
DPLUN = 000001	FF.NV = 000003	F.RCNM = 000034	R.FIX = 000001	TD#STAR = 176372
DSKERR = 000676R	FF.POE = 000002	F.RCTL = 000017	R.SEQ = 000003	TD#TAW = 176362
DSKOK = 000704R	FF.RWD = 000001	F.RSTZ = 000002	R.VAR = 000002	TD#TDR = 176374
EF.ID = 000001	FF.RWF = 000006	F.RTYP = 000000	R.VDBA = 000006	TD#TDW = 176364
ERROR = 000744R	FF.SPC = 000004	F.SEON = 000100	R.VDTN = 000002	TSTRED = 000666R
ERR1 = 000206R	FN.ACK = 000016	004 F.SP0V = 000072	SNDHDR = 000456R	TSTURK = 000340R
ERR1L = 000021	FN.FSA = 000000	004 F.SPUN = 000074	SNDEND = 000520R	T\$AD = 000000
ERR1T = 000210R	FN.FSB = 000002	004 F.STBK = 000036	SNDSP = 000542R	T\$BR = 000000
ERR2 = 000232R	FN.FSC = 000004	004 F.UNIT = 000136	SNDSRC = 000434R	T\$BD = 000010
ERR2L = 000000	FN.FSD = 000020	004 F.URPD = 000020	SND\$UX = 000412R	T\$SO = 100000

T\$BT = .000020	T\$IIBE = .020000	T\$SCLK = .020000	WORD5 = .000012	\$\$OST = .000012
T\$BTAR = .000030	T\$IIBF = .040000	T\$SEG1 = .000000	WORD6 = .000014	\$\$T1 = .000000
T\$BTD = .002000	T\$IICD = .000040	T\$SEG2 = .000001	WORD7 = .000016	.CLOSE = ***** G
T\$CD = .000100	T\$MODE = .004000	T\$SEG3 = .000002	WORD8 = .000020	.DLFNB = ***** G
T\$CLK = .002000	T\$OB = .000036	T\$SO = .000001	WORD9 = .000022	.FINIT = ***** G
T\$DISK = .000200	T\$OBE = .004000	T\$UBUS = .100000	WRDVAL = .000024	.FSRCB = ***** G
T\$DRD = .000004	T\$OBF = .010000	T\$ICLK = .000400	W.TSEF = .000002	.OPFNB = ***** G
T\$EMEM = .010000	T\$OBRA = .000034	T\$BEN = .000020	XMLUN = .000002	.READ = ***** G
T\$FSAA = .000000	T\$OBWA = .000032	WAITHG = .000324R	XMTDMC = .001030R	.WAIT = ***** G
T\$FSAB = .000004	T\$OUTA = .100000	WORD0 = .000000	\$CBDSG = ***** GX	...GBL = .000000
T\$FSAC = .000014	T\$RBD0 = .000200	WORD1 = .000002	\$\$DIV = ***** G	...PC1 = .000000R
T\$FSB2 = .000010	T\$RNB = .000040	WORD2 = .000004	\$\$\$ = .000036R	...PC2 = .000140R
T\$IB = .000026	T\$RSET = .040000	WORD3 = .000006	\$\$\$ARG = .000002	...TPC = .000020
T\$IBAR = .000024	T\$SC = .000022	WORD4 = .000010		

. ABS. 000000 000
001250 001
SRCOFF 000122 002
FDSCOF 000010 003
FNDFFS 000022 004
\$\$FSR1 000000 005
\$DPB\$\$ 000042 006
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 6936 WORDS (28 PAGES)
DYNAMIC MEMORY: 8084 WORDS (31 PAGES)
ELAPSED TIME: 00:00:49
CCOUT, CCOUT, -SP=C20, 1JP, C, CCOUT

SEARCH: MAC M1110 27-MAR-80 14:28
TABLE OF CONTENTS:

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	2	DATA SECTION
10-	97	SEARCH INITIALIZATION
11-	135	READ SEARCH DATA BASE


```
1 .TITLE SEARCH
2 .SBTTL DATA SECTION
3 ;
4 .MCALL QIDW%C,QIDW%$EXIT%$
5 .MCALL MKT%$C,WTSE%$
6 .MCALL SETF%$,DECL%$,GTIM%$
7 ;
8 .GLOBL DD2CT,$CBTA
9 .GLOBL MSQ,SHSTIM
10 ;
11 ;
12 000175 TRFSEC=125. ;MAX TRANSFER SIZE
13 175000 TRFSIZ=TRFSEC*512.
14 047040 NEMTHR=20000. ;TOTAL NEM THRESHOLD
15 001750 NOTRTH=1000. ;NEM IN A ROW THRESHOLD
16 ;
17 000005 TILUN=5
18 000006 COLUN=6
19 000001 DPLUN=1
20 000001 EF,IO=1
21 ;
```

```
23 000000          .PSECT-
24                ;
25 000000          TIIN:  .BLKB  8.      ; OPERATOR RESPONSE
26 000010          DBIN:  .BLKW  256.    ; DATA BASE IPR
27                ;
28 001010          IOSTAT: .BLKW  2.     ; IO STATUS
29                ;
30 001014          HLBN:  .BLKW  1       ; DATA BASE START
31 001016          LLBN:  .BLKW  1       ;
32 001020          HSIZE: .BLKW  1       ; DATA BASE SIZE
33 001022          LSIZE: .BLKW  1       ;
34                ;
35 001024 000000    DBSADD: .WORD  0      ; IO START ADDRESS
36 001026          DBBCNT: .BLKW  1      ; IO BYTE COUNT
37                ;
38 001030 000000    NUMNOT: .WORD  0      ; NEM IN A ROW
39 001032 000000    NUMNEM: .WORD  0      ; NEM ERRORS
40 001034 000000    HTRSF:  .WORD  0      ; GOOD BLOCKS XFERED
41 001036 000000    LTRSF:  .WORD  0      ;
42                ;
43 001040 000001    TDFUCT: .WORD  1      ; TD SECTOR XFER COUNT BIAS
44                ;
45 001042 000000    IPRSTR: .WORD  0      ; DEFAULT IPR START
46 001044 165140          .WORD  60000.  ;
47                ;
48                ;
49                ;
50 001046 123      105    101    ; OPERATOR PROMPT
   001051 122     103    110    PRMT:  .ASCII  /SEARCH - DB START>/
   001054 040     055    040
   001057 104     102    040
   001062 123     124    101
   001065 122     124    076
51                ;
52 000022          PRMTS=.-PRMT
53                ;
54                ;
55                ;
56 001070 000026          ; ERROR MESSAGES
57 001072 123      105    101    ERR1:  .WORD  ERR1L
   001075 122     103    110    ERR1T: .ASCII  /SEARCH: TERMINAL ERROR/
   001100 072     040    124
   001103 105     122    115
   001106 111     116    101
   001111 114     040    105
   001114 122     122    117
   001117 122
58                ;
59 000026          ERR1L=.-ERR1T
60                ;
61                ;
62 001120 000022          ERR2:  .WORD  ERR2L
   001122 123      105    101    ERR2T: .ASCII  /SEARCH: DISK ERROR/
   001125 122     103    110
   001130 072     040    104
   001133 111     123    113
   001136 040     105    122
   001141 122     117    122
```

63	000022			ERR2L = -ERR2T	
64				.EVEN	
65				:	
66	001144	000027		ERR3: .WORD	ERR3L:
67	001146	123	105	ERR3T: .ASCII	/SEARCH: TD MEMORY ERROR/
	001151	122	103		
	001154	072	040		
	001157	104	040		
	001162	105	115		
	001165	122	131		
	001170	105	122		
	001173	117	122		
68	000027			ERR3L = -ERR3T	
69				.EVEN	
70				:	
71	001176	000025		ERR4: .WORD	ERR4L:
72	001200	123	105	ERR4T: .ASCII	/SEARCH: MEM THRESHOLD/
	001203	122	103		
	001206	072	040		
	001211	105	115		
	001214	124	110		
	001217	105	123		
	001222	117	114		
73	000025			ERR4L = -ERR4T	
74				.EVEN	
75				:	
76	001226	000036		ERR5: .WORD	ERR5L:
77	001230	123	105	ERR5T: .ASCII	/SEARCH: MEM IN A ROW THRESHOLD/
	001233	122	103		
	001236	072	040		
	001241	105	115		
	001244	111	116		
	001247	101	040		
	001252	117	127		
	001255	124	110		
	001260	105	123		
	001263	117	114		
78	000036			ERR5L = -ERR5T	
79				.EVEN	
80				:	
81	001266	000037		ERR6: .WORD	ERR6L:
82	001270	123	105	ERR6T: .ASCII	/SEARCH: BAD SECTOR XFERED COUNT/
	001273	122	103		
	001276	072	040		
	001301	101	104		
	001304	123	105		
	001307	124	117		
	001312	040	130		
	001315	105	122		
	001320	104	040		
	001323	117	125		
	001326	124	116		
83	000037			ERR6L = -ERR6T	
84				.EVEN	
85				:	
86				:	
87				TERMINATION MESSAGE	

88	001330	123	105	101	DONEM:	.ASCII	/	SEARCH:	DONE	.NEM-	/
	001333	122	103	110							
	001336	072	040	104							
	001341	117	116	105							
	001344	056	040	116							
	001347	105	115	055							
	001352	040									
89	001353	040	040	040	D1:	.ASCII	/		/		
	001356	040	040	040							
90	001361	040	073	040		.ASCII	/	:	TRAN-	/	
	001364	124	122	101							
	001367	116	055	040							
91	001372	040	040	040	D2:	.ASCII	/		/		
	001375	040	040	040							
	001400	040	040								
92	001402	040	040	040	D3:	.ASCII	/		/		
	001405	040	040	040							
93		000060			DONEL=.	-DONEM					
94						.EVEN					
95					:						

```
          .SBTTL SEARCH: INITIALIZATION.
SEARCH:
;
;GET DATA BASE ID FROM OPERATOR.
;
102 001410 000424          BR      DEFLT5          ;BYPASS OPERATOR INPUT.
103 001412          QIOW#C  IO,RPR,TILUN,EF,IO,,IOSTAT,<#TIIN,8,,,PRMT,PRMTS,44>
104 001420 122767 000000G 177362  CMPB   #19,SUC,IOSTAT      ;GOOD TRANSFER.
105 001426 001404          BEQ     NUMBIN
106 001430 012700 001070*   MOV    #ERR1,R0          ;ERROR.
107 001434 000167 000616   JMP    ERROR.
;
109 001440 016704 177346   NUMBIN: MOV    IOSTAT+2,R4      ;# OF CHARACTERS.
110 001444 001406          BEQ    DEFLT5          ;NONE.
111 001446 012703 001042*   MOV    #IPRSTR,R3      ;CONVERT TO BINARY.
112 001452 012705 000000*   MOV    #TIIN,R5
113 001456          CALL   .DD2CT.
114          ;READ IPR.
115 001462          DEFLT5: QIOW#S  #10,RLB,#DPLUN,#EF,IO,,#IOSTAT,<#DBIN,#512,,,IPRSTR,IPRSTR+2>
116 001540 122767 000000G 177242  CMPB   #19,SUC,IOSTAT      ;NO ERROR.
117 001546 001404          BEQ    GOTIPR.
118 001550 012700 001120*   DSKERR: MOV    #ERR2,R0      ;ERROR.
119 001554 000167 000476   JMP    ERROR.
;
121 001560 016767 000000C 177226  GOTIPR: MOV    DBIN+SR,SDB,HLBN. ;START OF DATA.
122 001566 016767 000000C 177222  MOV    DBIN+SR,SDB+2,LLBN.
123 001574 116767 000000C 177216  MOV    DBIN+SR,ECH,HSIZE. ;END OF DATA.
124 001602 016767 000000C 177212  MOV    DBIN+SR,ECL,LSIZE.
125 001610 166767 177202 177204  SUB    LLBN,LSIZE. ;SIZE OF DATA.
126 001616 005667 177176   SBC    HSIZE.
127 001622 166767 177166 177170  SUB    HLBIN,HSIZE.
128 001630 062767 000002 177164  ADD    #2,LSIZE. ;READ EOC SECTOR AND NEXT ONE.
129 001636 005567 177156   ADC    HSIZE.
;
; SAVE SEARCH START TIME.
131          GTIM#C  SHSTIM.
132 001642          ;
133          ;
```

```
135 .SBTTL READ SEARCH DATA BASE
136 ;
137 ;START NEXT I/O IF NECESSARY
138 001650 005767 177144 READSG: TST HSIZE ;DONE IF # BLOCKS -
139 001654 002404 BLT 1$ ; LEFT IS NEGATIVE
140 001656 003040 BGT MMORDT ; MUCH MORE DATA LEFT
141 001660 005767 177136 TST LSIZE ; MORE IF LOW ORDER PART -
142 001664 001020 BNE MOREDT ; NOT ZERO
143 ;ALL BLOCKS TRANSFERED
144 001666 1$: MRKT$C EF 10,3,2 ;WAIT 3 SEC FOR TD -
145 001674 WTSE$C EF 10 ; INPUT BUFFER TO DRAIN
146 001702 032737 002000 176376 BIT #T$CLK,@#TD$SW ;BR IF TD MEMORY -
147 001710 001402 BEQ TDMEME ; ERROR
148 001712 000167 000410 JMP EXIT ;NORMAL EXIT
149 ;TD MEMORY ERROR
150 001716 012700 001144 TDMEME: MOV #ERR3,R0 ;ERROR MESSAGE
151 001722 000167 000330 JMP ERROR ;EXIT
152 ;
153 ;MORE BLOCKS TO TRANSFER
154 001726 026727 177070 000175 MDREDT: CMP LSIZE,#TRFSEC ;BR IF MORE DATA LEFT -
155 001734 103011 BHIS MMORDT ; THAN MAX TRANSFER SIZE
156 001736 016767 177060 177062 MOV LSIZE,DBBCNT ;# OF SEC ACTUALLY LEFT
157 001744 000367 177056 SWAB DBBCNT ;MAKE IT A -
158 001750 066767 177052 177050 ADD DBBCNT,DBBCNT ; BYTE COUNT
159 001756 000403 BR DDBIO ;SEND IT
160 ;
161 001760 012767 175000 177040 MMORDT: MOV #TRFSIZ,DBBCNT ;MAX TRANSFER SIZE
162 ;
163 ;SPECIAL DB READ TO TD INPUT BUFFER
164 001766 DDBIO: QIOW$S #IO,RLB!100,#DPLUN,#EF,IO,#IOSTAT,<DBSADD,DBBCNT,HLBN,LLBN>
165 002044 105767 176740 TSTB IOSTAT ;BRANCH IF I/O ERROR
166 002050 100432 DBERR1: BMI DBERR0
167 ;SUCCESSFUL DB I/O
168 002052 016701 176750 MOV DBBCNT,R1 ;NUMBER OF BYTES TRANSFERED
169 002056 005067 176756 CLR TDFUCT ;BIAS =0 AFTER FIRST XFER
170 002062 005067 176742 CLR NUMNOT ;CLEAR NEM IN A ROW COUNT
171 002066 060167 176732 UPDTSC: ADD R1,DBSADD ;NEXT START ADDRESS
172 002072 042767 174000 176724 BIC #-TD#INL,DBSADD ;MODULO BUFFER SIZE
173 002100 000301 SWAB R1 ;GET NUMBER -
174 002102 006201 ASR R1 ; OF SECTORS
175 002104 060167 176706 ADD R1,LLBN ;ADD TO SECTOR TRANSFER -
176 002110 005567 176700 ADC HLBH ; START ADDRESS
177 002114 160167 176702 SUB R1,LSIZE ;SUBTRACT FROM NUMBER OF -
178 002120 005667 176674 SBC HSIZE ; SECTORS LEFT
179 002124 060167 176706 ADD R1,LTRFSR ;ADD TO NUMBER OF SECTORS -
180 002130 005567 176700 ADC HTRFSR ; SUCCESSFULLY TRANSFERED
181 002134 000645 BR READSG ;READ NEXT SEGMENT
182 ;
```

```
184 ;
185 ;DB: ERROR OCCURED.
186 002136 022767 000000C 176644 DBEROR: CMP #IE,VER&777,IOSTAT ;BRANCH IF -
187 002144 001201 BNE DSKERR ; NORMAL DISK ERROR
188 ;NEM: ERROR
189 002146 032737 002000 176376 BIT #T#CLK,@#TD#SW ;ERROR IS TD -
190 002154 001660 BEQ TDMEME ; MEMORY FAILURE
191 ;NOT A TD MEMORY ERROR
192 002156 005267 176650 INC NUMNEM ;BUMP NEM ERROR COUNT
193 ; CMP NUMNEM,#NEMTHR ;BRANCH IF NOT -
194 ; BLO 3$ ; NEM THRESHOLD
195 002162 000403 BR 3$
196 002164 012700 001176* MOV #ERR4,R0 ;NEM THRESHOLD
197 002170 000432 BR ERROR
198 ;NOT NEM THRESHOLD - READ # SECTORS XFERED
199 002172 012737 000022 176362 3$: MOV #22,@#TD#TAW
200 002200 013701 176374 MOV @#TD#TDR,R1 ;NUMBER OF SECTORS
201 002204 066701 176630 ADD TDFUCT,R1 ;ADD OFFSET FOR FIRST XFER
202 002210 003010 BGT 5$ ;GOT SOME
203 002212 002417 BLT MINMSG ;NEGATIVE COUNT
204 ; INC NUMNOT ;NEM WITH NO SECTORS XFERED
205 002214 026727 176510 001750 CMP NUMNOT,#NOTRTH ;BRANCH IF NOT -
206 002222 103405 BLO 4$ ; THRESHOLD
207 002224 012700 001226* MOV #ERR5,R0 ;NO SECTORS TRANSFERED THRESHOLD
208 002230 000412 BR ERROR
209 ;SOME SECTORS WERE TRANSFERED BEFORE ERROR
210 002232 005067 176572 5$: CLR NUMNOT ;CLEAR COUNT OF NEM NO XFER
211 002236 000301 4$: SWAB R1 ;GET # OF -
212 002240 060101 ADD R1,R1 ; BYTES TRANSFERED
213 002242 005067 CLR TDFUCT ;ZERO BIAS AFTER FIRST XFER
214 002246 000167 177614 JMP UPDTSC ;SET UP NEXT TRANSFER
215 ;
216 002252 012700 001266* MINMSG: MOV #ERR6,R0 ;BAD COUNT IN HARDWARE
217 ;
218 ;EXITS:
219 002256 012001 ERROR: MOV (R0)+,R1 ;ERROR MESSAGE POINTER AND LENGTH
220 002260 QIOW$S #IO,WLB,#COLUN,#EF,IO,,,<R0,R1,#40>
221 ;
222 002326 016701 176500 EXIT: MOV NUMNEM,R1 ;PUT DATA IN OUTPUT STRING
223 002332 012700 001353* MOV #D1,R0
224 002336 012700 030010 MOV #30010,R2
225 002342 CALL #CBTA
226 002346 016701 176462 MOV HTRSF,R1
227 002352 012700 001372* MOV #D2,R0
228 002356 012700 030010 MOV #30010,R2
229 002362 CALL #CBTA
230 002366 016701 176444 MOV LTRSF,R1
231 002372 012700 001402* MOV #D3,R0
232 002376 012700 030010 MOV #30010,R2
233 002402 CALL #CBTA
234 ;
235 002406 QIOW$C IO,WLB,COLUN,EF,IO,,,<DONEM,DONEL,40> ;TERM MESSAGE
236 ;
237 ;
238 002414 052767 040000 000000G SEND: SUXX DONE TO MASTER COMPUTER.
239 002422 BIS #BIT14,MSG ; TELL CCOUT TO SEND EXCHANGE
240 002424 SETF$S #CF,COT
241 002426 BECL $S
```

SEARCH: MA M1110 27-MAR-88 14:28 PAGE 12-1
READ SEARCH DATA BASE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

241 :
242 002442 : EXIT\$S :EXIT
243 :
244 :
245 001410* .END SEARCH

BITVAL = 000000	DSKERR 001550R	HSIZE = 001020R	SR.DLT = 000014	002.T\$CLK = 002000
BIT0 = 000001	D1 001353R	HTRSFR 001034R	SR.ECB = 000047	002.T\$DISK = 000200
BIT1 = 000002	D2 001372R	IE.VER = ***** GX	SR.ECH = 000046	002.T\$DRD = 000004
BIT10 = 002000	D3 001402R	IOSTAT 001010R	SR.ECL = 000050	002.T\$ENEM = 010000
BIT11 = 004000	EF.10 = 000001	ID.RLB = ***** GX	SR.FIB = 000012	002.T\$FSA = 000000
BIT12 = 010000	ERROR 002256R	ID.RPR = ***** GX	SR.GRE = 000100	002.T\$FSAB = 000004
BIT13 = 020000	ERR1 001070R	ID.WLB = ***** GX	SR.GRS = 000072	002.T\$FSAC = 000014
BIT14 = 040000	ERR1L = 000026	IPRSTR 001042R	SR.LEN = 000122	002.T\$FSB2 = 000010
BIT15 = 100000	ERR1T = 001072R	IS.SUC = ***** GX	SR.LIN = 000066	002.T\$IB = 000026
BIT2 = 000004	ERR2 = 001120R	LLBN = 001016R	SR.LIP = 000062	002.T\$IBAR = 000024
BIT3 = 000010	ERR2L = 000022	LSIZE = 001022R	SR.MDN = 000006	002.T\$IBE = 020000
BIT4 = 000020	ERR2T = 001122R	LTRSFR 001036R	SR.NDC = 000042	002.T\$IBF = 040000
BIT5 = 000040	ERR3 001144R	MINMSG 002252R	SR.NDS = 000036	002.T\$ICD = 000040
BIT6 = 000100	ERR3L = 000027	MMORDT 001760R	SR.NIN = 000030	002.T\$MODE = 004000
BIT7 = 000200	ERR3T = 001146R	MOREDTR 001726R	SR.NIP = 000022	002.T\$OB = 000036
BIT8 = 000400	ERR4 001176R	MSQ = ***** G	SR.SDB = 000032	002.T\$OBE = 004000
BIT9 = 001000	ERR4L = 000025	MS.DGN = 010000	SR.SRC = 000002	002.T\$OBF = 010000
BYTE0 = 000000	ERR4T = 001200R	M.KTAE = 000010	SR.SUN = 000000	002.T\$OBRA = 000034
BYTE1 = 000001	ERR5 001226R	M.KTEF = 000002	SR.TWS = 000056	002.T\$OBWA = 000032
BYTE2 = 000002	ERR5L = 000036	M.KTHG = 000004	SR.WSL = 000052	002.T\$OUTA = 100000
BYTE3 = 000003	ERR5T = 001230R	M.KTUN = 000006	SR.YR = 000004	002.T\$RBD = 000200
BYTE4 = 000004	ERR6 001266R	N = 000012	SR.11N = 000024	002.T\$RNB = 000040
BYTE5 = 000005	ERR6L = 000037	NEMTHR = 047040	SR.11P = 000016	002.T\$RSET = 040000
BYTE6 = 000006	ERR6T = 001270R	NOTRTH = 001750	TD.FUCT = 001040R	T\$SC = 000022
BYTE7 = 000007	EXIT 002326R	NUMBIN = 001440R	TD.MEM = 001716R	T\$SCLK = 020000
BYTE8 = 000010	FD.FID 000000	003 NUMMEM = 001032R	TD\$CTR = 176370	T\$SEG1 = 000000
BYTE9 = 000011	FD.FNB 000006	003 NUMNOT = 001030R	TD\$CTW = 176360	T\$SEG2 = 000001
BYTVAL = 000012	FD.FYR 000004	003 N.BFAC = 000004	TD\$INL = 004000	T\$SEG3 = 000002
CF.COT = 000041	FD.LEN 000010	004 N.BHGH = 000006	TD\$MEM = 000270	T\$SO = 000001
CF.DGN = 000046	FN.ACK 000016	004 N.BTCH = 000004	TD\$OAR = 176344	T\$UBUS = 100000
CF.DHR = 000042	FN.FSA 000000	004 N.BUFB = 004000	TD\$OTR = 176346	T\$ICLN = 000400
CF.DMC = 000047	FN.FSB 000002	004 N.BUFW = 002000	TD\$ORD = 000274	T\$GBEN = 000020
CF.HBR = 000045	FN.FSC 000004	004 N.FOS = 000764	TD\$RST = 176366	UPDTSC = 002066R
CF.HRL = 000044	FN.FSD 000020	004 N.QURY = 000031	TD\$SW = 176376	WORD0 = 000000
CF.UPD = 000043	FN.MHR 000010	004 N.SUNT = 000002	TD\$TAR = 176372	WORD1 = 000002
COLUN = 000006	FN.NMB 000022	004 PRMT = 001046R	TD\$TAW = 176362	WORD2 = 000004
DBBCNT = 001026R	FN.OLS 000006	004 PRMTS = 000022	TD\$TDR = 176374	WORD3 = 000006
DBEROR = 002136R	FN.RDC 000014	004 Q.IDAE = 000012	TD\$TDW = 176364	WORD4 = 000010
DBERR1 = 002050R	FN.UPD 000012	004 Q.IDEF = 000006	TIUN = 000000R	WORD5 = 000012
DBIN = 000010R	GOTIPR 001560R	004 Q.IOFN = 000002	TILUN = 000005	WORD6 = 000014
DBSADD = 001024R	G.TIBA = 000002	Q.IOLU = 000004	TRFSEC = 000175	WORD7 = 000016
DBSLEN = 000116	G.TICP = 000016	Q.IOPL = 000014	TRFSIZ = 175000	WORD8 = 000020
DEFLT3 = 001462R	G.TICT = 000014	Q.IOPR = 000007	T\$AD = 000020	WORD9 = 000022
DG.ERR = 001000	G.TIDA = 000004	Q.IOSB = 000010	T\$BA = 000002	WRDVAL = 000024
DG.SDF = 002000	G.TIHR = 000006	READSG 001650R	T\$BD = 000010	W.TSEF = 000002
DG.HBR = 004000	G.TIMI = 000010	SEARCH 001410R	T\$BSO = 100000	\$CBTA = ***** G
DDBD10 = 001766R	G.TIMO = 000002	SHSTIM = ***** G	T\$BT = 000020	\$\$\$ = 000052R
DONEL = 000060	G.TISC = 000012	SR.ARE = 000114	002.T\$BTAR = 000030	\$\$\$ARG = 000003
DONEM = 001330R	G.TIYR = 000000	SR.ARS = 000106	002.T\$BTD = 002000	\$\$\$OST = 000004
DPLUN = 000001	HLBN 001014R	SR.DAY = 000010	002.T\$CD = 000100	.DD2CT = ***** G

. ABS. 000000 000
002450 001
SRCOFF 000122 002
FDSCOF 000010 003
FNOFFS 000022 004
\$DPB\$\$ 000102 005

SEARCH: M1110 27-MAR-80 14:28 PAGES 13-7
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4477 WORDS (18 PAGES)
DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
ELAPSED TIME: 00:00:35
SEARCH SEARCH/SP=C20.IJP.C. SEARCH

```
1 .TITLE QR-DRIVER
2 .MCALL ABODF$,HWDDF$,PKTDF$,TCBDF$
3 000000 ABODF$
4 000000 HWDDF$
5 000000 PKTDF$
6 000000 TCBDF$
7 ;
8 ; LOCAL DATA FOR INTSV$ MACRO
9 000000 CNTBL: .BLKW 1
10 000002 TEMP: .BLKW 1
11 ;
12 ; TO MAKE INTSV$ HAPPY
13 000000 LD$QR=0
14 000001 Q$R11=1
15 ;
16 ; DRIVER DISPATCH TABLE
17 000004 $QRTBL:
18 000004 000030' QRINIT
19 000006 000022' QRCANC
20 000010 000020' QRTOUT
21 000012 000014' QRPWRF
22 ;
```

```

24          ; TIMEOUT AND POWER FAIL ENTRY
25 000014 010567 177760  ORPWF: MOV.  R5,CNTBL.      ;SAVE UCB PTR FOR INRPT.
26 000020          ORTOUT:          ;NO T/O REQUESTED.
27 000020  EXECRT: RETURN.
28          ;
29          ; CANCEL I/O.
30 000022 012700 000000C  ORCANC: MOV.  #IE,AB0&377,R0  ;ABORT STATUS CODE.
31 000026 000422          BR      DONEIO.      ;RETURN TO CALLER.
32          ;
33          ; I/O INITIATE.
34 000030  ORINIT: CALL.  $GTPKT.      ;GET THE NEXT PKT.
35 000034 000771          BR      EXECRT.      ;GOT IT OR NOT.
36          ;
37          ; I/O INTERRUPT.
38 000036  $ORINT:
39 000036          INTSV$ QR,PRS,0#$R11      ;SAVE INTRPT STATUS.
40 000042 016504 000000G  MOV.  U,SCB(R5),R4      ;GET SCB ADDRESS.
41 000046 105764 000000G  TSTB. S,STS(R4)        ;GOT A PACKET?
42 000052 001004          BNE.  1$              ;YES.
43 000054 005267 000024  INC.  SPURIT.           ;*** BUMP SPURIOUS COUNT.
44 000060 000167 000000G  JMP.  $INTXT.         ;NO - DISMISS SPUR INT.
45          ;
46 000064 1$: CALL.  $FORK.          ;CREATE FORK PROCESS
47 000070 012700 000000C  MOV.  #IS,SUC&377,R0    ;SUCCESS STATUS.
48 000074 005001  DONEIO: CLR.  R1          ;NO SECOND STATUS WORD.
49 000076          CALL.  $I0DON.        ;I/O COMPLETE.
50 000102 000752          BR      ORINIT.        ;GET NEXT PKT.
51          ;
52 000104 000000  SPURIT: .WORD. 0          ;***** SPURIOUS INT COUNT
53          ;
54          .END.

```

A\$\$CHK = 000000	DBSLEN = 000116	I.PRM = 000024	P\$\$SRF = 000000	S.CSST = 000020
A\$\$CPS = 000000	DG.ERR = 001000	I.TCB = 000004	P\$\$WRD = 000000	S.CTRP = 000014
A\$\$NSI = 000000	DG.SDF = 002000	I.UCB = 000010	QR.CANC = 000022R	S.IOMG = 000032
A\$\$PRI = 000000	DG.TDF = 004000	KDSAR0 = 172360	QR.INIT = 000030R	S.PRTY = 000034
A\$\$TRP = 000000	DONEIO = 000074R	KDSDR0 = 172320	QR.PURF = 000014R	S.STS = *****GX
A.AST = 000006	D\$\$ISK = 000000	KISAR0 = 172340	QR.OUT = 000020R	TD#CTR = 176370
A.BYT = 000004	D\$\$L11 = 000004	KISAR5 = 172352	Q\$\$OPT = 000007	TD#CTW = 176360
A.CBL = 000002	D\$\$SHF = 000000	KISAR6 = 172354	Q\$\$R11 = 000001	TD#INL = 004000
A.DQSR = 177775	D\$\$YNC = 000000	KISAR7 = 172356	R\$\$DER = 000000	TD#MEM = 000270
A.KSR5 = 177774	D\$\$YNM = 000000	KISDR0 = 172300	R\$\$EXV = 000000	TD#OAR = 176344
A.NPR = 000010	EXECRT = 000020R	KISDR6 = 172314	R\$\$K11 = 000001	TD#OTR = 176346
A.PRM = 000012	E\$\$XPR = 000000	KISDR7 = 172316	R\$\$P11 = 000001	TD#ORD = 000274
BITVAL = 000000	FD.FID = 000000	003 K\$\$CNT = 177546	R\$\$SND = 000000	TD#RST = 176366
BIT0 = 000001	FD.FNB = 000006	003 K\$\$CSR = 177546	R\$\$11M = 000000	TD#SU = 176376
BIT1 = 000002	FD.FVR = 000004	003 K\$\$IEN = 000115	SISDR0 = 172200	TD#STAR = 176372
BIT10 = 002000	FD.LEN = 000010	003 K\$\$LDC = 000001	SPURIT = 000104R	TD#TAU = 176362
BIT11 = 004000	FE.CAL = 000040	K\$\$TPS = 000074	SR.ARE = 000114	002 TD#TDR = 176374
BIT12 = 010000	FE.CEX = 020000	LD#QR = 000000	SR.ARS = 000106	002 TD#TDW = 176364
BIT13 = 020000	FE.DRV = 000010	LD#X1 = 000000	SR.DAY = 000010	002 TEMP = 000002R
BIT14 = 040000	FE.EXP = 000200	L\$\$ASG = 000000	SR.DLT = 000014	002 TPS = 177564
BIT15 = 100000	FE.EXT = 000001	L\$\$DRV = 000000	SR.ECB = 000047	002 TS.BLK = 170700
BIT2 = 000004	FE.EXV = 000004	L\$\$PTO = 000144	SR.ECH = 000046	002 TS.CKP = 000200
BIT3 = 000010	FE.LSI = 000400	L\$\$P11 = 000001	SR.ECL = 000050	002 TS.CKR = 000100
BIT4 = 000020	FE.MUP = 000002	MPAR = 172100	SR.FIB = 000012	002 TS.EXE = 100000
BIT5 = 000040	FE.MKT = 040000	MPCSR = 177746	SR.GRE = 000100	002 TS.MSG = 020000
BIT6 = 000100	FE.NLG = 100000	MS.DGN = 010000	SR.GRS = 000072	002 TS.NRP = 010000
BIT7 = 000200	FE.PKT = 000100	M\$\$CRB = 000124	SR.LEN = 000122	002 TS.OUT = 000400
BIT8 = 000400	FE.PLA = 000020	M\$\$CRX = 000000	SR.LIN = 000066	002 TS.RDN = 040000
BIT9 = 001000	FN.ACK = 000016	004 M\$\$FCS = 000000	SR.LIP = 000062	002 TS.RUN = 004000
BYTE0 = 000000	FN.FSA = 000000	004 M\$\$MGE = 000000	SR.MDN = 000006	002 T#AD = 000020
BYTE1 = 000001	FN.FSB = 000002	004 M\$\$MUP = 000000	SR.NDC = 000042	002 T#BA = 000002
BYTE2 = 000002	FN.FSC = 000004	004 M\$\$MVR = 000000	SR.NDS = 000036	002 T#BD = 000010
BYTE3 = 000003	FN.FSD = 000020	004 N = 000012	SR.NIN = 000030	002 T#BS0 = 100000
BYTE4 = 000004	FN.MHR = 000010	004 N\$\$LDV = 000001	SR.NIP = 000022	002 T#BT = 000020
BYTE5 = 000005	FN.NMB = 000022	004 N\$\$MOV = 000041	SR.SDB = 000032	002 T#BTAR = 000030
BYTE6 = 000006	FN.OLS = 000006	004 N.BFAC = 000004	SR.SRC = 000002	002 T#BD = 002000
BYTE7 = 000007	FN.RDC = 000014	004 N.BHGH = 000006	SR.SUN = 000000	002 T#CD = 000100
BYTE8 = 000010	FN.UPD = 000012	004 N.BTCH = 000004	SR.TWS = 000056	002 T#CLK = 002000
BYTE9 = 000011	F\$\$LVL = 000001	N.BUFB = 004000	SR.WSL = 000052	002 T#DISK = 000200
BYTVAL = 000012	G\$\$TFP = 000000	N.BUFW = 002000	SR.YR = 000004	002 T#DRD = 000004
CF.COT = 000041	G\$\$TSS = 000000	N.FUS = 000764	SR.11N = 000024	002 T#MEM = 010000
CF.DGN = 000046	G\$\$TTK = 000000	N.QURY = 000031	SR.11P = 000016	002 T#FSA = 000000
CF.DHR = 000042	G\$\$URD = 000000	N.SUNT = 000002	SR0 = 177572	T#FSAB = 000004
CF.DMC = 000047	H\$\$RTZ = 000074	PIRQ = 177772	SR3 = 172516	T#FSAC = 000014
CF.HBR = 000045	IE.ABO = *****GX	PNODE = 030000	SWR = 177570	T#FSB2 = 000010
CF.HRL = 000044	IS.SUC = *****GX	PR0 = 000000	S\$\$YSZ = 007600	T#IB = 000026
CF.UPD = 000043	I\$\$RAR = 000000	PR1 = 000040	S.CAB0 = 000024	T#IBAR = 000024
CMODE = 140000	I\$\$RDN = 000000	PR4 = 000200	S.CAST = 000022	T#IBE = 020000
CNTBL = 000000R	I.AST = 000022	PR5 = 000240	S.CBPT = 000004	T#IBF = 040000
C\$\$CDA = 000002	I.ATTL = 000044	PR6 = 000300	S.CCRF = 000030	T#ICD = 000040
C\$\$CKP = 000000	I.EFN = 000003	PR7 = 000340	S.CENT = 000012	T#HODE = 004000
C\$\$CSR = 177404	I.FCH = 000012	PS = 177776	S.CFLT = 000016	T#OB = 000036
C\$\$INT = 000000	I.IOSB = 000014	P\$\$GMX = 000000	S.CILI = 000010	T#OBE = 004000
C\$\$ORE = 002000	I.LGTH = 000044	P\$\$LAS = 000000	S.CIOT = 000006	T#OBF = 010000
C\$\$RSH = 177564	I.LNK = 000000	P\$\$P45 = 000000	S.CLRF = 000026	T#OBRA = 000034
C\$\$RUN = 000000	I.LN2 = 000006	P\$\$RFL = 000000	S.COAD = 000000	T#OBWA = 000032
C\$\$TTY = 177564	I.PRI = 000002	P\$\$RTY = 000000	S.CSGE = 000002	T#OUTA = 100000

QR-DRIVER
SYMBOL TABLE

MACRO-M1110 27-MAR-66 14:27 PAGE 11-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

T\$RBD0 = 000200	T\$SYN = 000000	T.NLUP = 000016	T2.WFR = 000001	UISDR6 = 177614
T\$RNB = 000040	T\$TRW = 000000	T.OFF = 000060	T3.ACP = 100000	UISDR7 = 177616
T\$RSET = 040000	T\$UTB = 000000	T.PCB = 000046	T3.CAL = 000100	U.SCB = ***** GX
T\$SC = 000022	T\$VBF = 000000	T.PRI = 000002	T3.CLI = 001000	V\$CTR = 000404
T\$SCLK = 020000	T\$30P = 000000	T.RCFL = 000012	T3.MCR = 004000	V\$RSN = 000031
T\$SEG1 = 000000	T\$1CLK = 000400	T.RRFL = 000064	T3.NET = 000020	WORD0 = 000000
T\$SEG2 = 000001	T\$BEN = 000020	T.SRCT = 000063	T3.NSD = 000200	WORD1 = 000002
T\$SEG3 = 000002	T.ACTL = 000052	T.STAT = 000032	T3.PMD = 040000	WORD2 = 000004
T\$SO = 000001	T.ASTL = 000016	T.ST2 = 000034	T3.PRV = 010000	WORD3 = 000006
T\$UBUS = 100000	T.ATT = 000054	T.ST3 = 000036	T3.REM = 020000	WORD4 = 000010
T\$ACR = 000000	T.CPCB = 000004	T.TCBL = 000030	T3.RDV = 000040	WORD5 = 000012
T\$BTW = 000000	T.DPRI = 000040	T.UCB = 000026	T3.RST = 000400	WORD6 = 000014
T\$BUF = 000000	T.EFLG = 000022	T2.AB0 = 000100	T3.SLV = 000200	WORD7 = 000016
T\$CCA = 000000	T.IOC = 000003	T2.AST = 100000	UBMPR = 170200	WORD8 = 000020
T\$CCO = 000000	T.LBN = 000041	T2.BFX = 004000	UDSAR0 = 177600	WORD9 = 000022
T\$CTR = 000000	T.LDV = 000044	T2.CAF = 000400	UDSDR0 = 177620	WRDVAL = 000024
T\$GHC = 000000	T.LNK = 000000	T2.CHK = 020000	UISAR0 = 177640	X\$DBT = 000000
T\$GTS = 000000	T.MXSZ = 000050	T2.CKD = 010000	UISAR4 = 177650	X\$M11 = 000002
T\$KMG = 000000	T.NAM = 000006	T2.DST = 040000	UISAR5 = 177652	\$FORK = ***** GX
T\$LWC = 000000	T.NCRE = 000006	T2.FXD = 002000	UISAR6 = 177654	\$GTPKT = ***** GX
T\$M11 = 000001	T.NCWF = 000004	T2.HLT = 000200	UISAR7 = 177656	\$INTXT = ***** GX
T\$RNE = 000000	T.NDMF = 000010	T2.SPN = 000004	UISDR0 = 177600	\$IODON = ***** GX
T\$RPR = 000000	T.NDNR = 000000	T2.STP = 000020	UISDR4 = 177610	\$ORINT = 000036RG
T\$RST = 000000	T.NDSE = 000002	T2.TIO = 001000	UISDR5 = 177612	\$ORTBL = 00004RG
T\$RUB = 000000	T.NLDN = 000014			

. ABS = 177776 000
000106 001
SRCOFF = 000122 002
FDSCOF = 000010 003
FNOFFS = 000022 004
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 5441 WORDS (22 PAGES)
DYNAMIC MEMORY: 7028 WORDS (27 PAGES)
ELAPSED TIME: 00:00:21
QRDRV,QRDRV/-SP=C 1,1JEXMC/HL,C 200,200JRSXMC/PA:1,C 20,1JP,C,QRDRV

LOAD TERM DETECTOR
TABLE OF CONTENTS

MACRO M1110 27-MAR-90 14:26

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	2	
8-	3	
8-	4	MODULE NAME: LOAD TERM DETECTOR (LOAD)
8-	5	
8-	6	
8-	7	
8-	8	
8-	9	
8-	10	
8-	11	
8-	12	*****
8-	13	LOAD TERM DETECTOR (LOAD)
8-	14	*****
8-	16	PURPOSE:
9-	43	FILE INITIALIZATION:
10-	64	DATA AREAS AND EQUATES:
11-	98	START LTD:
12-	148	LOAD BYTE CODE TRANSLATOR:
13-	191	INTERNAL CHARACTER CODE TABLE:
14-	262	LDBUF - LOAD TDCT MEMORY:
15-	367	ERROR HANDLING ROUTINE:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

.FILE LOAD TERM DETECTOR
.SBTL
.SBTL MODULE NAME: LOAD TERM DETECTOR (LOAD)
.SBTL AUTHOR:
.SBTL
.SBTL
.SBTL
.SBTL
.SBTL RELEASE DATE: 15 DEC 1978
.SBTL RELEASE NUMBER: SL120024
.SBTL
.SBTL *****
.SBTL LOAD TERM DETECTOR (LOAD)
.SBTL *****
.NLIST MEB
.SBTL PURPOSE

```

```

;
; THE PURPOSE OF THIS PROGRAM IS TO LOAD TERM DETECTOR CONTROL TABLES
; FOR FSA-A, B, AND C.
; THIS PROGRAM IS ACTIVATED BY CCIN. IT PERFORMS THE FOLLOWING
; PROCESSING STEPS:
;

```

1. LOADS THE TERM DETECTOR'S BYTE CODE TRANSLATOR.
2. LOADS SY: [7,5]TDC TA.FSA INTO FSA-A AND DELETES THE FILE.
3. LOADS SY: [7,5]TDC TB.FSA INTO FSA-B AND DELETES THE FILE.
4. LOADS SY: [7,5]TDC TC.FSA INTO FSA-C AND DELETES THE FILE.
5. ACTIVATES HOROLS (OR HRLGET) PASSING IT NO DATA.
6. EXITS.

```

;
; ASSEMBLER COMMAND FILE (LOADMAC.CMD):
; LOAD,LOAD/-SP=P,C,LOAD
;
; TASK BUILD COMMAND FILE (LOADTKB.CMD) TO BE USED IN PDP-11/45:
; LOAD/-MM,LOAD/-SP=LOAD.MOUTCC
;
;
; COMMON=CCOM:RW
; RESLIB=C2,2JFCSRES/RO
; PAR=GEN:40000:40000
;

```


LOAD TERM DETECTOR
FILE INITIALIZATION

MACRO M1110 27-MAR-88 14:26 PAGE 8

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
43          .SBTTL FILE INITIALIZATION
44          ;
45          ; MACRO CALLS
46          ;
47          .MCALL FDBDF$,FDRC$,FDBK$,FDOP$,FSRSZ$
48          .MCALL FINIT$,RST$,READ$,WAIT$,EXIT$
49          .MCALL OFNB$R
50          .PSECT
51          ;
52          ; TDCTA FILE
53          ;
54          EFN,3 = 3
55          TDCLUN = 3
56          TDCTDB: FDBDF$
57          FDRC$: FD,RWM
58          FDBK$: RDBUF,N:BUFB,..EFN,3,I0ST
59          FDOP$: TDCLUN
60          ;
61          FSRSZ$ 0
62          ;
```

LOAD TERM DETECTOR
DATA AREAS AND EQUATES

MACRO M1110 27-MAR-88 14:26 PAGE 18

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
64 .SBTTL DATA AREAS AND EQUATES
65 .EVEN
66 ;
67 ; BUFFER AREAS
68 000140 063 103 RDBUF: .ASCII /3C/ ; READ BUFFER
69 000142 000000 .WORD 0
70 000144 000000 .WORD 0
71 000146 000000 .WORD 0
72 000150 TDCBUF: .BLKB <N,BUFB+10>
73 004140 TDCEND: = ; END OF READ BUFFER
74 ;
75 004140 000000 WRDLGT: .WORD 0 ; LENGTH OF TDCT DATA IN ONE 2048
76 ; ; BYTE BLOCK (WORDS)
77 ;
78 004142 000000 B.I: .WORD 0 ; POINTER TO START OF DATA IN FIRST BLOCK
79 004144 000000 B.X: .WORD 0 ; " " " OTHER BLOCKS
80 ;
81 004146 IOST: .BLKW 2 ; IO STATUS BLOCK FOR TDCT FILE
82 ;
83 ; TERM DETECTOR REGISTERS
84 ;
85 ;
86 004152 000000 CBUF: .WORD 0 ; CONTROL REGISTER BUFFER
87 ;
88 004154 000000 FSAA: .WORD 0 ; CONTROL REGISTER VALUE FOR FSA A
89 004156 000004 FSAB: .WORD 4 ; CONTROL REGISTER VALUE FOR FSA B
90 004160 000014 FSAC: .WORD 14 ; CONTROL REGISTER VALUE FOR FSA C
91 ;
92 004162 000000 STRTAD: .WORD 0 ; STARTING ADDRESS FOR TERM DETECTOR MEMORY
93 004164 000000 CURAD: .WORD 0 ; CURRENT TDCT ADDRESS FOR LOADING
94 004166 030000 MAXMEM: .WORD <3*4*N,BUFW> ; MAX MEMORY SIZE FOR TD
95 ;
96 004170 020000 CLKWD: .WORD T$SCLK ; START CLOCK WORD
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

LOAD TERM DETECTOR
START LTD

MACRO M1110 27-MAR-68 14:26 PAGE 11
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
          .SBTTL START LTD
START:
          FINIT$
          :
          :   LOAD BYTE CODE TRANSLATOR
          :
          :   JSR PC,LDBYTE
          :
          :   LOAD TD CONTROL TABLES
          :
          :   OPEN FILE, READ TDCT INTO BUFFER, LOAD TDCT INTO HARDWARE
          :   AND CLOSE FILE, REPEAT PROCESS FOR EACH FSA
          :
111 004202 012767 000000 177742 MOV #T$FSA,CBUF ;SET UP TO LOAD FSA A
112 004210 012701 000000G MOV #FDSFSA,R1 ;R1-> FDSC OF TDCTA
113 004214 012767 000150* 177720 MOV #TDCBUF,B.1 ;POINTER TO DATA IN FIRST BLOCK
114 : ;OF TDCT
115 004222 012767 000150* 177714 MOV #TDCBUF,B.X ;POINTER TO TDCT DATA IN OTHER BLOCKS
116 : ;OF FILE
117 004230 CALL LDBUF ;LOAD BUFFER
118 :
119 004234 012767 000004 177710 MOV #T$FSAB,CBUF ;SET UP TO LOAD FSA B
120 004242 012701 000000G MOV #FDSFSB,R1 ;R1-> FDSC OF TDCTB
121 004246 012767 000150* 177666 MOV #TDCBUF,B.1 ;POINTER TO DATA IN FIRST BLOCK
122 : ;OF TDCT
123 004254 012767 000140* 177662 MOV #RDBUF,B.X ;POINTER TO TDCT DATA IN OTHER BLOCKS
124 : ;OF FILE
125 004262 CALL LDBUF ;LOAD BUFFER
126 :
127 004266 012767 000014 177656 MOV #T$FSAC,CBUF ;SET UP TO LOAD FSA C
128 004274 012701 000000G MOV #FDSFSC,R1 ;R1-> FDSC OF TDCTC
129 004300 012767 000144* 177634 MOV *RDBUF+4>.B.1 ;POINTER TO DATA IN FIRST BLOCK
130 : ;OF TDCT
131 004306 012767 000140* 177630 MOV #RDBUF,B.X ;POINTER TO TDCT DATA IN OTHER BLOCKS
132 : ;OF FILE
133 004314 CALL LDBUF ;LOAD BUFFER
134 :
135 :   HERE IF DONE
136 :
137 004320 ROST$C HQRQLS ;ACTIVATE HQRQLS
138 004326 103007 BCC EXIT
139 004330 026727 000000G-000000G CMP #DSW,#LE,INS ;IS HQRQLS INSTALLED?
140 004336 001003 BNE EXIT ;BRANCH IF YES (OTHER ERROR)
141 :
142 :   HQRQLS NOT INSTALLED, ACTIVATE HRLGET WHICH, IN TURN, WILL
143 :   ACTIVATE SQR
144 :
145 004340 ROST$C HRLGET ;ACTIVATE SQR
146 004346 EXIT: EXIT$S
```

```

148 .SBTTL LOAD BYTE CODE TRANSLATOR
149 ;
150 ; SUBROUTINE TO LOAD INTERNAL CHARACTER CODE TO THE BYTE CODE
151 ; TRANSLATOR OF THE TERM DETECTOR.
152 ;
153 004354 010146 LDBYTE MOV R1, -(SP)
154 004356 010346 MOV R3, -(SP)
155 004360 010546 MOV R5, -(SP)
156 ;
157 ; DO MASTER RESET
158 ;
159 004362 012767 000000 176366 MOV #0, TD#RST
160 ;
161 ; LOAD 20040 INTO CONTROL REG.
162 ;
163 004370 012767 020040 176360 MOV #20040, TD#CTW
164 ;
165 ; LOAD 20 INTO TRANSFER ADDRESS REGISTER
166 ;
167 004376 012767 000020 176362 MOV #20, TD#TAW
168 ;
169 ; SUCCESSIVELY LOAD 64 WORDS TO TRANSFER DATA REGISTER.
170 ; REPEAT FOUR TIMES TO LOAD A TOTAL OF 256 WORDS.
171 ;
172 004404 012703 000004 MOV #4, R3 ;R3 = LOAD CYCLE COUNTER
173 004410 012701 000100 MOV #64, R1 ;R1 = WORD COUNTER
174 004414 012705 004456 MOV #NIBTAB, R5 ;R5 -> INTERNAL CHARACTER CODE TABLE
175 004420 012567 176364 1$: MOV (R5)+, TD#TDW ;LOAD WORD INTO TRANSFER DATA REG.
176 004424 005301 DEC R1
177 004426 003374 BGT 1$ ;LOOP
178 ;
179 ; GROUP OF 64 DONE
180 ;
181 004430 005303 DEC R3
182 004432 003405 BLE 2$ ;BRANCH IF ALL 256 WORDS ARE LOADED
183 004434 012701 000100 MOV #64, R1 ;SET UP TO LOAD 64 MORE WORDS
184 004440 012705 004456 MOV #NIBTAB, R5 ;START AT TOP OF TABLE AGAIN
185 004444 000765 BR 1$ ;LOOP
186 004446 012605 2$: MOV (SP)+, R5
187 004450 012603 MOV (SP)+, R3
188 004452 012601 MOV (SP)+, R1
189 004454 000207 RTS PC ;RETURN
    
```

```

191
192
193
194
195
196
197 004456 002077
198 004460 010001
199 004462 040002
200 004464 100003
201 004466 004004
202 004470 000400
203 004472 000400
204 004474 000400
205 004476 000301
206 004500 000302
207 004502 000313
208 004504 000307
209 004506 000400
210 004510 000400
211 004512 000305
212 004514 000306
213 004516 000400
214 004520 000304
215 004522 000324
216 004524 000310
217 004526 000257
218 004530 000277
219 004532 000057
220 004534 000077
221 004536 000117
222 004540 000137
223 004542 000157
224 004544 000177
225 004546 000217
226 004550 000237
227 004552 000311
228 004554 000001
229 004556 000002
230 004560 000003
231 004562 000004
232 004564 000005
233 004566 000006
234 004570 000007
235 004572 000010
236 004574 000011
237 004576 000012
238 004600 000013
239 004602 000014
240 004604 000015
241 004606 000016
242 004610 000033
243 004612 000034
244 004614 000021
245 004616 000022
246 004620 000023
247 004622 000024
    
```

```

      .SBTTL  INTERNAL CHARACTER CODE TABLE
      :
      : TABLE TO DEFINE INTERNAL CHARACTER CODES TO BE LOADED INTO TERM
      : DETECTOR'S BYTE CODE TRANSLATOR. TABLE IS INDEXED BY 6-BIT
      : DATA BASE CHARACTER CODE.
      :
      NIBTAB: .WORD 2077      :DOC ID
              .WORD 10001    :PARAGRAPH MK
              .WORD 40002    :ZONE MARK
              .WORD 100003   :SUBZONE MARK
              .WORD 4004     :SENTENCE MK
              .WORD 400      :SPACE
              .WORD 400      :!
              .WORD 400      :"
              .WORD 301      :#
              .WORD 302      :$
              .WORD 313      :%
              .WORD 307      :^
              .WORD 400      :_
              .WORD 400      :.
              .WORD 400      :/
              .WORD 304      :+
              .WORD 304      :-
              .WORD 324      :.
              .WORD 310      :/
              .WORD 257      :0
              .WORD 277      :1
              .WORD 57       :2
              .WORD 77       :3
              .WORD 117      :4
              .WORD 137      :5
              .WORD 157      :6
              .WORD 177      :7
              .WORD 217      :8
              .WORD 237      :9
              .WORD 311      :- (SIG. SPACE)
              .WORD 1       :A
              .WORD 2       :B
              .WORD 3       :C
              .WORD 4       :D
              .WORD 5       :E
              .WORD 6       :F
              .WORD 7       :G
              .WORD 10      :H
              .WORD 11      :I
              .WORD 12      :J
              .WORD 13      :K
              .WORD 14      :L
              .WORD 15      :M
              .WORD 16      :N
              .WORD 33      :O
              .WORD 34      :P
              .WORD 21      :Q
              .WORD 22      :R
              .WORD 23      :S
              .WORD 24      :T
    
```

LOAD TERM: VECTOR
INTERNAL CHARACTER CODE TABLE

MACRO: M1110

27-MAR-88 14:36 PAGE 17

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

248	004624	000025	.WORD	25	:U			
249	004626	000026	.WORD	26	:V			
250	004630	000027	.WORD	27	:W			
251	004632	000030	.WORD	30	:X			
252	004634	000031	.WORD	31	:Y			
253	004636	000032	.WORD	32	:Z			
254	004640	000312	.WORD	312	:E			
255	004642	000303	.WORD	303	:DEC	POINT		
256	004644	000314	.WORD	314	:J			
257	004646	000315	.WORD	315	:+			
258	004650	000400	.WORD	400	:		(WT)	(325)
259	004652	000400	.WORD	400	:		(WT)	(326)
260	004654	000400	.WORD	400	:?		(WT)	(327)

```

262 .SBTTL LDBUF - LOAD TDCT MEMORY.
263 ;
264 ; THIS ROUTINE OPENS THE SPECIFIED TDCT FILE, READS IT ONE BLOCK
265 ; (2048 BYTES) AT A TIME, LOADS THE CORRESPONDING FSA MEMORY,
266 ; CLOSING AND DELETES FILE WHEN DONE AND RETURNS.
267 ;
268 ; ON ENTRY,
269 ; R1 -> FILE DESCRIPTOR BLOCK OF GIVEN TDCT FILE.
270 ; B.1 -> START OF TDCT DATA IN FIRST BLOCK READ.
271 ; B.X -> START OF TDCT DATA IN OTHER BLOCKS.
272 ; CBUF -> CONTROL REGISTER.
273 ;
274 004656 LDBUF:
275 ;
276 ; SET UP FILENAME BLOCK OF FILE.
277 ; R1 -> FDCB.
278 ;
279 004656 012700 000000 MOV #TDCFDB,R0 ;R0-> FDB.
280 004662 CALL BLDFL ;BUILD FNB
281 ;
282 ; OPEN.
283 ;
284 004666 OFNB#R R0
285 004700 103006 BCC 1#
286 004702 116001 000052 MOV# F,ERR(R0),R1
287 004706 010167 000274 MOV R1,PAR1
288 004712 CALL ERROR
289 ;
290 ; DETERMINE SECTOR LENGTH OF FILE.
291 ;
292 004716 016002 000012 1#: MOV F,EFBK+2(R0),R2 ;R2 = END-OF-FILE SECTOR NUMBER.
293 004722 005302 DEC R2 ;R2-> LAST SECTOR OF DATA.
294 004724 006202 ASR R2
295 004726 006202 ASR R2 ;R2=LENGTH OF DATA IN 2048 BYTE BLOCKS.
296 ;
297 ; READ FIRST BLOCK AND LOAD IT INTO FSA MEMORY.
298 ;
299 004730 READ# R0
300 004734 WAIT# R0
301 004740 103006 BCC 2#
302 004742 116001 000052 MOV# F,ERR(R0),R1
303 004746 010167 000234 MOV R1,PAR1
304 004752 CALL ERROR
305 ;
306 ; CLEAR STATUS REGISTER.
307 ;
308 004756 005067 176366 2#: CLR TD#RST ;CLEAR IT.
309 ;
310 ; START CLOCK.
311 ;
312 004762 016767 177202 176360 MOV CLKWD,TD#CTW ;CLOCK START WORD TO CONTROL REG.
313 ;
314 ; NOW SET CONTROL REGISTER AND START ADDRESS OF MEMORY.
315 ;
316 004770 016767 177156 176360 MOV CBUF,TD#CTW ;SET CONTROL REGISTER FOR WRITES.
317 ;
318 004776 016767 177160 177160 MOV STRAD,CHRAD ;START ADDRESS FOR MEMORY
  
```

```

319 ;
320 ; SET UP R5 TO POINT TO FIRST WORD OF DATA TO BE TRANSFERED, AND
321 ; SET UP R1 TO CONTAIN THE NUMBER OF WORDS TO BE TRANSFERED TO
322 ; THE TD MEMORY.
323 ;
324 005004 016705 177132      MOV.   B.1,R5           ;R5-> START OF DATA IN FIRST BLOCK.
325 005010 012701 004140      MOV.   #TDCEND,R1      ;R1-> END OF BUFFER.
326 005014 160501             SUB.   R5,R1           ;R1 = BYTE LENGTH OF DATA IN BUFFER.
327 005016 006201             ASR.   R1              ;R1 = WORD LENGTH OF DATA IN BUFFER.
328 ;
329 ; SET UP TO TRANSFER WORDS 3 AT A TIME
330 ;
331 005020 016767 177140 176362 MOV.   CURAD,TD$TAW.    ;SET UP LOAD ADDRESS.
332 ;
333 ; TRANSFER WORDS TO MEMORY.
334 ;
335 005026      30$:
336 005026 012567 176364      MOV.   (R5)+,TD$TDW.   ;LOAD WORD.
337 005032 005301             DEC.   R1              ;DECREMENT WORD COUNT.
338 005034 003374             BGT.   30$           ;CONTINUE.
339 ;
340 ; HERE FOR TESTING OF # OF BLOCKS -- CONTINUE IF NOT 0
341 ;
342 005036 005302      32$: DEC.   R2              ;DECR CURRENT COUNT.
343 005040 003422             BLE.   4$            ;GO IF DONE.
344 ;
345 ; READ IN ANOTHER BLOCK OF TDCT DATA.
346 ;
347 005042      READ$ R0
348 005046      WAIT$ R0
349 005052 103006      BCC.   3$
350 005054 116001      MOVB.  F,ERR(R0),R1
351 005060 010167 000122      MOV.   R1,PAR1
352 005064      CALL.  ERROR.
353 ;
354 ; SET UP R5 AND R1 AGAIN.
355 ;
356 005070 016705 177050      3$: MOV.   B.X,R5           ;R5->FIRST WORD OF DATA IN BUFFER.
357 005074 012701 004140      MOV.   #TDCEND,R1      ;R1-> END OF BUFFER.
358 005100 160501             SUB.   R5,R1           ;R1 = BYTE LENGTH OF CURRENT BUFFER.
359 005102 006201             ASR.   R1              ;R1 = WORD "
360 005104 000750             BR.   30$             ;CONTINUE LOADING.
361 ;
362 ; CLOSE AND DELETE FILE.
363 ;
364 005106      4$: CALL.  .DLFNB.        ;CLOSE AND DELETE FILE.
365 005112      EXIT.  LDTDC.        ;RETURN.
  
```


LOAD TERM DETECTOR.
ERROR HANDLING ROUTINE.

MACRO M1110 27-MAR-88 14:26 PAGE 15

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
367                                     .SBTTL .ERROR HANDLING ROUTINE.
368                                     ;
369                                     ;
370 005114                               ERROR:
371 005114 011667 000070                MOV. (SP),PAR2. ;SP.
372 005120 010167 000062                MOV. R1,PAR1 ;ERROR CODE.
373 005124                                MOUT$. #MSG1,#PAR1
374 005144 000167 177176                JMP. EXIT
375                                     ;
376                                     ;
377 005150 000032.                       MSG1: .WORD LN1E-LN1
378 005152. 005154.                       .WORD LN1
379 005154 105 122 122. LN1: .ASCIZ /ERROR NO. =%X1D, PC =%X10/
    005157 117 122 040
    005162. 116 117 056
    005165 040 075 040
    005170 045 061 104
    005173 054 040 120
    005176 103 040 075
    005201 040 045 061
    005204 117 000
380 005206                               LN1E:
381                                     .EVEN.
382                                     ;
383 005206 000000                       PAR1: .WORD 0
384 005210 000000                       PAR2: .WORD 0
385                                     ;
386                                     ;
387 004172.                               .END. START.
```

LOAD TERM. SECTOR.
SYMBOL TABLE

MACRO M1110 27-MAR-88 14:26 PAGE 15-1

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

BITVAL = 000000	FD.RWM = ***** GX.	F.SEQN = 000100	SR.GRS = 000072.	002.T\$DRD = 000004
BIT0 = 000001	FN.ACK 000016	004 F.SP DV = 000072.	SR.LEN = 000122.	002.T\$EMEM = 010000
BIT1 = 000002	FN.FSA 000000	004 F.SPUN = 000074.	SR.LIN = 000066	002.T\$FSA = 000000
BIT10 = 002000	FN.FSB 000002	004 F.STBK = 000036	SR.LIP = 000062	002.T\$FSAB = 000004
BIT11 = 004000	FN.FSC 000004	004 F.UNIT = 000136	SR.MON = 000006	002.T\$FSAC = 000014
BIT12 = 010000	FN.FSD 000020	004 F.URBD = 000020	SR.NDC = 000042	002.T\$FSB2 = 000010
BIT13 = 020000	FN.MMR 000010	004 F.VBN = 000064	SR.NDS = 000036	002.T\$IB = 000026
BIT14 = 040000	FN.NMB 000022	004 F.VBSZ = 000060	SR.NIN = 000030	002.T\$IBAR = 000024
BIT15 = 100000	FN.QLS 000006	004 IE.INS = ***** GX.	SR.NIP = 000022	002.T\$IBE = 020000
BIT2 = 000004	FN.RDC 000014	004 IOST = 004146R	SR.SDB = 000032	002.T\$IBF = 040000
BIT3 = 000010	FN.UPD 000012	004 LDBUF = 004656R	SR.SRC = 000002	002.T\$ICD = 000040
BIT4 = 000020	FO.RD = ***** GX.	LDBYTE = 004354R	SR.SUN = 000000	002.T\$10DE = 004000
BIT5 = 000040	FSAA = 004154R	LN1 = 005154R	SR.TWS = 000056	002.T\$OB = 000036
BIT6 = 000100	FSAB = 004156R	LN1E = 005206R	SR.WSL = 000052	002.T\$OBE = 004000
BIT7 = 000200	FSAC = 004160R	MAXMEM = 004166R	SR.YR = 000004	002.T\$OBF = 010000
BIT8 = 000400	F.ACTL = 000076	MSGOUT = ***** GX.	SR.1IN = 000024	002.T\$OBRA = 000034
BIT9 = 001000	F.ALDC = 000040	MSG1 = 005150R	SR.1IP = 000016	002.T\$OBWA = 000032
BLDEFL = ***** GX.	F.BBFS = 000062	MS.DGN = 010000	START = 004172R	T\$OUTA = 100000
BYTE0 = 000000	F.BDB = 000070	N = 000012	STRAD = 004162R	T\$RBD0 = 000200
BYTE1 = 000001	F.BGBC = 000057	NIBTAB = 004456R	S.BFHD = 000020	T\$RNB = 000040
BYTE2 = 000002	F.BKDN = 000026	N.BFAC = 000004	S.FATT = 000016	T\$RSET = 040000
BYTE3 = 000003	F.BKDS = 000020	N.BHGH = 000006	S.FDB = 000140	T\$SC = 000022
BYTE4 = 000004	F.BKEF = 000050	N.BTCH = 000004	S.FNAM = 000006	T\$SCLK = 020000
BYTE5 = 000005	F.BKP1 = 000051	N.BUFB = 004000	S.FNB = 000036	T\$SEG1 = 000000
BYTE6 = 000006	F.BKST = 000024	N.BUFUW = 002000	S.FNBW = 000017	T\$SEG2 = 000001
BYTE7 = 000007	F.BKVB = 000064	N.DID = 000024	S.FNTY = 000004	T\$SEG3 = 000002
BYTE8 = 000010	F.CHR = 000075	N.DVNM = 000032	S.FTYP = 000002	T\$SO = 000001
BYTE9 = 000011	F.CNTG = 000034	N.FID = 000000	S.NFEN = 000020	T\$SUBS = 100000
BYTVAL = 000012	F.DFNB = 000046	N.FNAM = 000006	TDCBUF = 000150R	T\$1CLK = 000400
B.X = 004144R	F.DSPT = 000044	N.FOS = 000764	TDCEND = 004140R	T\$SBEH = 000020
B.1 = 004142R	F.DVNM = 000134	N.FO = 000014	TDCFDB = 000000R	WORD0 = 000000
CBUF = 004152R	F.EFBK = 000010	N.FVER = 000016	TDCUN = 000003	WORD1 = 000002
CF.COT = 000041	F.EFN = 000050	N.NEXT = 000022	TDCCTR = 176370	WORD2 = 000004
CF.DGN = 000046	F.EQBB = 000032	N.QUERY = 000031	TDCCTW = 176360	WORD3 = 000006
CF.DHR = 000042	F.ERR = 000052	N.STAT = 000020	TDCINL = 004000	WORD4 = 000010
CF.DMC = 000047	F.FACC = 000043	N.SUNT = 000002	TDCMEM = 000270	WORD5 = 000012
CF.HBR = 000045	F.FBY = 000014	N.UNIT = 000034	TDCOAR = 176344	WORD6 = 000014
CF.HRL = 000044	F.FNAM = 000110	PAR\$\$\$ = 000000	TDCOTR = 176346	WORD7 = 000016
CF.UPD = 000043	F.FNB = 000102	PAR1 = 005206R	TDCORD = 000274	WORD8 = 000020
CLKUD = 004170R	F.FTYP = 000116	PAR2 = 005210R	TDCRST = 176366	WORD9 = 000022
CURAD = 004164R	F.FVER = 000120	RDBUF = 000140R	TDCSW = 176376	WRDLGT = 004140R
DBSLEN = 000116	F.HIBK = 000004	R.OSGC = 000015	TDCSTAR = 176372	WRDVAL = 000024
DG.ERR = 001000	F.LUN = 000042	R.QSPC = 000014	TDCSTAW = 176362	\$DSW = ***** GX.
DG.SDF = 002000	F.MBCT = 000054	R.QSPN = 000006	TDCSTR = 176374	\$\$\$ = 000016R
DG.TDF = 004000	F.MBC1 = 000055	R.QSPR = 000012	TDCTDW = 176364	\$\$\$OST = 000016
EFN.3 = 000003	F.MBFG = 000056	R.QSTN = 000002	T\$AD = 000020	\$\$\$T1 = 000006
ERROR = 005114R	F.NRBD = 000024	SR.ARE = 000114	002.T\$BA = 000002	.DLFNB = ***** GX.
EXIT = 004346R	F.NREC = 000030	SR.ARS = 000106	002.T\$BD = 000010	.FINIT = ***** G.
FDSFSA = ***** GX.	F.OVBS = 000030	SR.DAY = 000010	002.T\$BSO = 100000	.FSRCB = ***** G.
FDSFSB = ***** GX.	F.RACC = 000016	SR.DLT = 000014	002.T\$BT = 000020	.OPFNB = ***** G.
FDSFSC = ***** GX.	F.RATT = 000001	SR.ECB = 000047	002.T\$BTAR = 000030	.READ = ***** G.
FD.FID = 000000	003 F.RCHM = 000034	SR.ECH = 000046	002.T\$BTD = 002000	.WAIT = ***** G.
FD.FNB = 000006	003 F.RCTL = 000017	SR.ECL = 000050	002.T\$CD = 000100	...PC1 = 000000R
FD.FVR = 000004	003 F.RSIZ = 000002	SR.FIB = 000012	002.T\$CLK = 002000	...PC2 = 000140R
FD.LEN = 000010	003 F.RTYP = 000000	SR.GRE = 000100	002.T\$DISK = 000200	...TPC = 000020

. ABS = 000000 000

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

LOAD TERM DETECTOR
SYMBOL TABLE

MACRO M1110 27-MAR-88 14:26 PAGE 15-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

	005212	001
SRCOFF	000122	002
FDSCOF	000010	003
FNOFFS	000022	004
\$\$FSR1	000000	005
\$DPB\$\$	000034	006
ERRORS DETECTED:		0

VIRTUAL MEMORY USED: 5034 WORDS (20 PAGES)
DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
ELAPSED TIME: 00:00:45
LOAD,LOAD/-SP=C20,1JP,C,LOAD

HRLGET: MACRO: M1110 27-MAR-80 14:33
TABLE OF CONTENTS:

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

9- 19 FILE INITIALIZATION
12- 121 ERROR HANDLING ROUTINE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

```

; .TITLE .HRLGET.
;
; GET HRL BUFFER TASK.
;
; HRLGET IS RESPONSIBLE FOR ACTIVATING THE SQR TASK.
; BEFORE DOING SO, IT PRIMES THE HRL0 BUFFER (IN COMMON) WITH THE
; FIRST BUFFER'S WORTH OF HRL DATA. AFTER ACTIVATING SQR, IT WAITS
; FOR A REQUEST ON THE NEXT BUFFER USING EVENT FLAG CF.HBR.
; AFTER SQR PROCESSES THE END OF DATA SENTINEL IN THE HRL BUFFER,
; SQR POSTS CF.HBR ONE MORE TIME WITH HRLBUF NOT SET TO ZERO.
; THIS IS INTERPRETED AS A REQUEST BY SQR FOR HRLGET TO SHUT DOWN,
; NOT TO FETCH THE NEXT HRL BUFFER. HRLGET, UPON RECEIVING THIS
; SPECIAL SIGNAL, THEN DELETES THE HRL FILE AND EXITS.
;
; .MCALL .EXIT$S,ROST$C,WTSE$C,SETF$C,CLEF$C,DECL$S,FINIT$
; .MCALL .FDBDF$,FDRC$A,FDBK$A,FDOP$A,FSRSZ$A,OFNB$R,READ$,WAIT$

```

```
19                                     .SBTTL: FILE INITIALIZATION.  
20                                     :  
21                                     : HIT RESOLUTION LIST MERGE FILE.  
22                                     :  
23 000000 FDB: FDBDF$  
24 000140 FDRCSA: FD.RJM.  
25 000140 FDBKSA: HRL0.2*N.BUFW..1.IOST.  
26 000140 FDBPSA: 1  
27 000140 FSRSZ$ 0  
28                                     .EVEN.  
29                                     :  
30 000140 IOST: :  
31 000140 000000 PAR1: .WORD: 0  
32 000142 000000 PAR2: .WORD: 0  
33 000144 000000 UNDEF: .WORD: 0 ;UNDEFINED FILE FLAG.  
34                                     :  
35 140000 EODFLG: -. BIT15|BIT14  
36 040000 EOBFLG: -. BIT14
```

```
38 000146          HRLGET:
39 000146          FINIT$
40 000152          CLEF$C CF,HRL          ;SOR: WAITS FOR HRL DATA USING THIS EVENT FLAG.
41 000160          CLEF$C CF,HBR          ;HRLGET: WAITS FOR SOR REQUEST USING THIS EVENT FLAG.
42 000166 012767 000000G 000000G MOV $HRL0,HRLBUF      ;REINITIALIZE HRL BUFFER POINTER.
43
44
45
46 000174          ; ACTIVATE HOR, IF INSTALLED. OTHERWISE, ACTIVATE SQR.
47 000202 103013          ROST$C HOR          ;ACTIVATE HOR.
48 000204 026727 000000G 000000G BCC HRLG.1
49 000212 001404          CMP $DSW,#IE,INS      ;IS HOR INSTALLED?
50 000214          BEQ 1$
51 000220 000167 000264          CALL DIRERR
52 000224          JMP HRLXIT
53
54
55
56 000232 012700 000000'          1$: ROST$C SQR          ;ACTIVATE SQR TASK
57 000236 012701 000000G          ;
58 000242          ; OPEN OF HRL FILE - OPEN FAILURE TREATED AS NON-EXISTENT FILE.
59 000246          HRLG.1: MOV #FDB,R0          ;FDB ADDRESS
60 000264 103012          MOV #FDSMHR,R1        ;FILE DESCRIPTOR ADDRESS - FROM COMMON
61
62
63
64 000266 116001 000052          CALL BLDFL           ;BUILD FILE NAME BLOCK IN FDB
65 000272          OFNB$R #FDB          ;OPEN FILE FOR READ THRU FNB
66 000276 012767 140000 000000G BCC HRLRD           ;IF OPEN OK, PROCEED TO READ 1ST HRL BLOCK
67 000304 005267 177634          ;
68 000310 000451          ; OPEN ERROR - OUTPUT CONSOLE MESSAGE AND TREAT AS NON-EXISTENT FILE
69
70
71
72 000312          MOVB F,ERR(R0),R1
73 000312          CALL FCERR
74 000322 103005          MOV #EODFLG,HRL0     ;MOVE END OF FILE SENTINEL INTO HRL BUFFER
75 000324 116001 000052          INC UNDEF           ;SET UNDEFINED FILE FLAG
76 000330          BR SQRKIK          ;NOTIFY SQR THAT HRL BUFFER IS READY
77 000334 000465          ;
78 000336          ; READ NEXT BLOCK FROM HRL FILE
79 000342 005767 000000G          HRLRD:
80 000346 001415          READ$ #FDB          ;READ A BLOCK
81 000350 027727 000000G 041461 BCC 10$
82 000356 001405          MOVB F,ERR(R0),R1
83 000360 012701 000000G          CALL FCERR
84 000364          BR HRLXIT
85 000370 000447          10$: WAIT$
86 000372 062767 000004 000000G TST HRLBUF          ;IS THIS THE FIRST HRL BLOCK?
87 000400 000415          BEQ 20$            ;IF NOT, SKIP FILE HEADER CHECK
88 000402          CMP @HRLBUF,#"1C    ;IS FILE HEADER OK?
89 000402 012767 000000G 000000G BEQ 15$            ;BRANCH IF OK
90 000410 005777 000000G          MOV #IE,SPC,R1      ;USING THIS CODE FOR IMPROPER FILE HEADER
91 000414 100407          CALL FCERR
92 000416 032777 040000 000000G BR HRLXIT
93 000424          ADD #4,HRLBUF       ;ADJUST BUFFER POINTER
94 000426 005067 000000G          BR SQRKIK          ;NOTIFY SQR
          20$:
          MOV #HRL0,HRLBUF      ;SET LOCATION OF BUFFER FOR SQR
          TST @HRLBUF          ;IS FIRST ENTRY END OF DATA SENTINEL?
          BMI SQRKIK          ;GO DIRECTLY TO WAKE UP SQR TASK
          BIT #EODFLG,@HRLBUF  ;IS FIRST ENTRY END OF HRL BUFFER
          BOFF SQRKIK          ;IF NOT, GO TO WAKE UP SQR
          CLR HRLBUF          ;SUPPRESS FILE HEADER CHECK
```

HRLGET MACRO M1110 27-MAR-80 14:37 PAGE 10-1
FILE INITIALIZATION

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

95 000432 000727

BR

HRLRD

;READ NEXT BLOCK


```
97          ;          SQR TO BE NOTIFIED THAT HRL BUFFER IS AVAILABLE.  
98          ;  
99 000434   ;          SJRKIK:  
100 000434   SETF#C CF.HRL          ;POST HRL BUFFER AVAILABLE EVENT.  
101 000442   DECL$S  
102          ;  
103 000450 005767 177470   TST UNDEF          ;DOES HRL FILE EXIST.  
104 000454 001015   BNE HRLXIT          ;SKIP FILE DELETE ON UNDEFINED FILE.  
105          ;  
106 000456   WTSE#C CF.HBR          ;WAIT FOR NEXT REQUEST FROM SQR.  
107 000464   CLEF#C CF.HBR          ;CLEAR EVENT FLAG.  
108 000472 005767 000000G  TST HRLBUF          ;IS THIS HRL BUFFER OR A SHUTDOWN REQUEST.  
109 000476 001705   BEQ HRLRD          ;IF BUFFER REQUEST, PROCEED TO READ HRL.  
110          ;  
111          ;          TASK SHUTDOWN REQUEST.  
112          ;  
113 000500 012700 000000'   MOV #FDB,R0          ;FDB ADDRESS.  
114 000504   CALL .DLFNB          ;CLOSE AND DELETE FILE.  
115          ;  
116 000510   HRLXIT:  
117 000510 012767 000000G 000000G  MOV #HRL0,HRLBUF     ;REINITIALIZE HRL BUFFER POINTER.  
118          MOUT#$ #MSG0  
119 000516   EXIT#$
```

```
121
122
123
124
125
126 000524 011667 177410
127 000530 116700 000000
128 000534 010067 177402
129 000540
130 000560 000207
131
132
133
134 000562 011667 177352
135 000566 010167 177350
136 000572
137 000612 000207
138
139
140
141 000614 000040
142 000616 000630
143 000620 000032
144 000622 000670
145 000624 000024
146 000626 000722
147
148
149
150 000630 120 103 040
151 000670
152 000670 120 103 040
153 000722
154 000722 110 122 114
155 000746
156
157 000146

.SBTTL: ERROR HANDLING ROUTINE
.NLIST: BEX
:
: DIRECTIVE ERROR
:
DIRERR: MOV: (SP),PAR1
MOV: $DSJ,R0
MOV: R0,PAR2
MOUT$: #MSG1,#PAR1
RETURN
:
: FCS ERROR
:
FCSERR: MOV: (SP),PAR1
MOV: R1,PAR2
MOUT$: #MSG2,#PAR1
RETURN
:
: STRING DESCRIPTORS
:
MSG1: .WORD LN1E-LN1
.WORD LN1
MSG2: .WORD LN2E-LN2
.WORD LN2
MSG0: .WORD LN0E-LN0
.WORD LN0
:
: FORMAT STRINGS
:
LN1: .ASCII: /PC = %10, DIRECTIVE ERROR = %1D/
LN1E:
LN2: .ASCII: /PC = %10, FCS ERROR = %1D/
LN2E:
LN0: .ASCII: /HRLGET PROGRAM EXIT/
LN0E:
:
.END: HRLGET
```

BITYAL = 000000	FN.FSC 000004	004 F.VBSZ = 000060	SR.FIB 000012	002 T#FSAA = 000000
BIT0 = 000001	FN.FSD 000020	004 HRLBUF = ***** GX.	SR.GRE 000100	002 T#FSAB = 000004
BIT1 = 000002	FN.MHR 000010	004 HRLGET 000146R	SR.GRS 000072	002 T#FSAC = 000014
BIT10 = 002000	FN.NMB 000022	004 HRLG.1 000232R	SR.LEN 000122	002 T#FSB2 = 000010
BIT11 = 004000	FN.QLS 000006	004 HRLRD 000312R	SR.LIN 000066	002 T#IB = 000026
BIT12 = 010000	FN.RDC 000014	004 HRLXIT 000510R	SR.LIP 000062	002 T#IBAR = 000024
BIT13 = 020000	FN.UPD 000012	004 HRL0 = ***** GX.	SR.MON 000006	002 T#IBF = 040000
BIT14 = 040000	FO.RD = ***** GX.	IE.INS = ***** GX.	SR.NDC 000042	002 T#IBE = 020000
BIT15 = 100000	F.ACTL = 000076	IE.SPC = ***** GX.	SR.NDS 000036	002 T#IBF = 040000
BIT2 = 000004	F.ALQC = 000040	IOST 000140R	SR.NIN 000030	002 T#ICD = 000040
BIT3 = 000010	F.BBFS = 000062	LN0 000722R	SR.NIP 000022	002 T#MODE = 004000
BIT4 = 000020	F.BDB = 000070	LN0E 000746R	SR.SDB 000032	002 T#OB = 000036
BIT5 = 000040	F.BGBC = 000057	LN1 000630R	SR.SRC 000002	002 T#OBE = 004000
BIT6 = 006100	F.BKDN = 000026	LN1E 000670R	SR.SUN 000000	002 T#OBF = 010000
BIT7 = 000200	F.BKDS = 000020	LN2 000670R	SR.TWS 000056	002 T#OBRA = 000034
BIT8 = 000400	F.BKEF = 000050	LN2E 000722R	SR.WSL 000052	002 T#OBWA = 000032
BIT9 = 001000	F.BKP1 = 000051	MSGOUT = ***** GX.	SR.YR 000004	002 T#OUTA = 100000
BLDEFL = ***** GX.	F.BKST = 000024	MSG0 000624R	SR.11N 000024	002 T#RBD0 = 000200
BYTE0 = 000000	F.BKVB = 000064	MSG1 000614R	SR.1IP 000016	002 T#RNB = 000040
BYTE1 = 000001	F.CHR = 000075	MSG2 000620R	S.BFHD = 000020	002 T#RSET = 040000
BYTE2 = 000002	F.CNTG = 000034	MS.DGN = 010000	S.ETEF = 000002	T#SC = 000022
BYTE3 = 000003	F.DFNB = 000046	N = 000012	S.FATT = 000016	T#SCLK = 020000
BYTE4 = 000004	F.DSPT = 000044	N.BFAC = 000004	S.FDB = 000140	T#SEG1 = 000000
BYTE5 = 000005	F.DVNM = 000134	N.BHGH = 000006	S.FNAM = 000006	T#SEG2 = 000001
BYTE6 = 000006	F.EFBK = 000010	N.BTCH = 000004	S.FNB = 000036	T#SO = 000001
BYTE7 = 000007	F.EFN = 000050	N.BUFB = 004000	S.FNBW = 000017	T#UBUS = 100000
BYTE8 = 000010	F.EOBB = 000032	N.BUFW = 002000	S.FNTY = 000004	T#ICLK = 000400
BYTE9 = 000011	F.ERR = 000052	N.DID = 000024	S.FNEN = 000002	T#BEN = 000020
BYTVAL = 000012	F.FACC = 000043	N.DVNM = 000032	S.FNEN = 000020	UNDEF = 00014R
CF.COT = 000041	F.FFBY = 000014	N.FID = 000000	TD#CTR = 176370	WORD0 = 000000
CF.DGN = 000046	F.FNAM = 000110	N.FNAM = 000006	TD#CTW = 176360	WORD1 = 000002
CF.DHR = 000042	F.FNB = 000102	N.FOS = 000764	TD#IHL = 004000	WORD2 = 000004
CF.DMC = 000047	F.FTYP = 000116	N.FTYP = 000014	TD#MEM = 000270	WORD3 = 000006
CF.HBR = 000045	F.FVER = 000120	N.FVER = 000016	TD#DAR = 176344	WORD4 = 000010
CF.HRL = 000044	F.FVBR = 000120	N.NEXT = 000022	TD#PTR = 176346	WORD5 = 000012
CF.UPD = 000043	F.HIBK = 000004	N.QUERY = 000031	TD#QRD = 000274	WORD6 = 000014
C.LEEF = 000002	F.LUN = 000042	N.STAT = 000020	TD#RST = 176366	WORD7 = 000016
DBSLEN = 000116	F.MBCT = 000054	N.SUNT = 000002	TD#SW = 176376	WORD8 = 000020
DG.ERR = 001000	F.MBC1 = 000055	N.UNIT = 000034	TD#STAR = 176372	WORD9 = 000022
DG.SDF = 002000	F.MBFG = 000056	PAR# = 000027	TD#TAW = 176362	WORDVAL = 000024
DG.TDF = 004000	F.NRBD = 000024	PAR1 000140R	TD#TDR = 176374	W.TSEF = 000002
DIRERR = 000524R	F.NREO = 000030	PAR2 000142R	TD#TDW = 176364	\$DSW = ***** GX.
EOBFLG = 040000	F.OVBS = 000030	R.OSGC = 000015	T#AD = 000020	\$\$\$ = 000054R 006
EODFLG = 140000	F.RACC = 000016	R.OSPC = 000014	T#BA = 000002	\$\$\$OST = 000004
FCSERR = 000562R	F.RATT = 000001	R.OSPN = 000006	T#BD = 000010	\$\$\$T1 = 000003
FDB = 000000R	F.RCNM = 000034	R.OSPR = 000012	T#BS0 = 100000	.DLFNB = ***** GX.
FDSMHR = ***** GX.	F.RCTL = 000017	R.QSTN = 000002	T#BT = 000020	.FINIT = ***** G.
FD.FID 000000	F.RSIZ = 000002	SQRK IK 000434R	T#BTAR = 000030	.FSRCB = ***** G.
FD.FNB 000006	003 F.RTYP = 000000	SR.ARE 000114	002 T#BDT = 002000	.OPFNB = ***** G.
FD.FVR 000004	003 F.SEQN = 000100	SR.ARS 000106	002 T#CD = 000100	.READ = ***** G.
FD.LEN 000010	003 F.SPDI = 000072	SR.DAY 000010	002 T#CLK = 002000	.WAIT = ***** G.
FD.RUM = ***** GX.	003 F.SPUN = 000074	SR.DLT 000014	002 T#DISK = 000200	...PC1 = 000000R
FN.ACK 000016	F.STBK = 000036	SR.ECB 000047	002 T#DRD = 000004	...PC2 = 00014R
FN.FSA 000000	004 F.UNIT = 000136	SR.ECH 000046	002 T#MEM = 010000	...TPC = 000020
FN.FSB 000002	004 F.VBN = 000064	SR.ECL 000050		

. ABS. 000000 000

HRLGET- MACRO-M1110 27-MAR-80 14:37 PAGE 12-2
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

	000746	001
SRCOFF	000122	002
FDSCOF	000010	003
FNOFFS	000022	004
\$\$FSR1	000000	005
\$DPB\$\$	000000	006

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 5598 WORDS (22 PAGES)
DYNAMIC MEMORY: 7028 WORDS (27 PAGES)
ELAPSED TIME: 00:00:40
HRLGET, HRLGET /-SP=C20, 1JP, C, HRLGET

```

1          .TITLE--HQR-
2          ;
3          ;
4          ;   HARDWARE QUERY RESOLVER HANDLER
5          ;
6          ;
7          ;*****
8          ;
9          ;   THIS VERSION REFORMATS DHR'S FROM THE HQR FROM THE
10         ;   NEW LAYOUT (1 QID PER WORD, NO COUNT WORD) TO THE
11         ;   OLD (COUNT BYTE FOLLOWED BY QID BYTES)
12         ;
13         ;*****
14         ;
15         ;
16         ;   THIS PROGRAM IS STARTED UP BY HRLGET.
17         ;
18         ;   HQR PERFORMS THE FOLLOWING TASKS:
19         ;
20         ;   INITIALIZE THE HARDWARE
21         ;   WAIT FOR "BUFFER READY" SIGNAL FROM HRLGET
22         ;   START MRP AND CP MICROCODE
23         ;   RUN "SEARCH"
24         ;   SET UP TIMER FOR CP HIT BUFFER FLUSH
25         ;   ENTER MAIN LOOP TO WAIT FOR INTERRUPTS
26         ;
27         ;   UPON END-OF-COLLECTION START UP HQRFS.
28         ;
29         ;
30         ;
31         ; ASSEMBLY:
32         ; MCR>HQR,LP=[20,1]P,C,[5,3]IM04,HQR2
33         ;
34         ; TASK BUILD:           ON NPIC /04
35         ; HQR/DA,HORQLS=HQR,MRPSUB,CPSUB,PPSUB,[1,50]RSX11M.STB/SS
36         ; /
37         ; TASK=HQR
38         ; PAR=SQR:52000:26000
39         ; ASG=TT0:1
40         ; COMMON=CCOM:RW
41         ; /
42         ; /
43         ;
44         ; .MCALL QIOW#S,EXIT#S,CLEF#S,WTSE#S,RDAF#S,SETF#S,WTLO#S,GTIM#C
45         ; .MCALL MRKT#S,CNKT#S,DECL#S,ABRT#S,RQST#C
46         ; .GLOBL SHTIM,SHSTIM,SHTIME,$MUL
47         ;
48         ;
49         ; LUN.TT = 1 ;LUN FOR TT0
50         ; EFN.1 = 1 ;EVENT FLAG FOR TT0
51         ; EFN.3 = 3 ;EVENT FLAG FOR HARDWARE INTERRUPT
52         ; EFN.4 = 4 ;EVENT FLAG FOR HB FLUSH TIMER
53         ; ECC = 1 ;LOCAL END-OF-COLLECTION FLAG
54         ; BLFACT = 2 ;HIGHEST BCL OFFSET ALLOWED + 2
55         ; SR.NDC = 42 ;OFFSET TO DB POINTERS IN SREC
56         ;
57         ; .NLIST BEX
58         ; MYSEL: RAD50 / HQR

```

```

58 000004 000000 TSKTCB: .WORD 0 ;TCB OF MY TASK
59 000006 000000 OLDVEC: .WORD 0 ;OLD VECTOR AT 274
60 000010 EFBUF: .BLKW 4 ;EVENT FLAG BUFFER
61 000020 STAT: .BLKW 2
62 000024 000000 ERWORD: .WORD 0 ;INDEX VALUE FOR ERROR MESSAGE TABLE
63 000026 000000 BASE: .WORD 0 ;ALL PURPOSE FLAG
64 000030 000000 APLACE: .WORD 0 ;PRELIM BIT SETTINGS FOR CSR 1
65 000032 000000 DATA1: .WORD 0
66 000034 000000 H3OFF: .WORD 0 ;OFFSET FOR HB ADDR IN BCL
67 000036 000000 HROFF: .WORD 0 ;OFFSET FOR HRL ADDR IN BCL
68
69
70 ; LOCAL HIT BUFFER RECEIVES HITS IN NEW FORMAT
71 000040 000000 PCOUNT: .WORD 0 ;POINTER TO QID COUNT WORD
72 000042 000000 CJUNT: .WORD 0 ;QID COUNT
73 000044 LDCAL: .BLKW 1024
74
75 ; REFERENCE BCL
76 ; LOCAL COPY OF BCL NOT CLEARED BY CP VIA DMA
77
78 004044 000006G MYHB: .WORD DHR0+6 ;HIT BUFFER
79 004046 000000 .WORD 0
80 004050 000000 .WORD 0
81 004052 000000 .WORD 0
82 004054 000006G MYHR: .WORD HRL0 ;HRL BUFFER
83 004056 000000 .WORD 0
84 004060 000000 .WORD 0
85 004062 000000 .WORD 0
86
87 ; SHARED BCL
88 ; HIT BUFFER ADDRESS WOULD NORMALLY POINT TO COMMON
89 ; BUT REFORMATTING MUST BE DONE
90
91 004064 001775 BCL: .WORD 1024-3 ;BUFFERS ARE 2048 BYTES - HEADER
92 004066 000006G HBBCL: .WORD DHR0+6 ;HIT BUFFER
93 004070 000000 .WORD 0
94 004072 000000 .WORD 0
95 004074 000000 .WORD 0
96 004076 000006G HRBCL: .WORD HRL0 ;HRL BUFFER
97 004100 000000 .WORD 0
98 004102 000000 .WORD 0
99 004104 000000 .WORD 0
100 004106 E0BBCL: .BLKW 4 ;END OF BUFFER ADDRESSES
101
102 ;
103 ; PRINT LINE
104 ;
105 004116 015 012 PRINT: .BYTE 15,12 ;PRECEDE PRINT LINE WITH CRLF
106 004120
107 000116 .REPT 78
108 .BYTE 40
109 .ENDR
110
111 ;
112 ; TABLE OF MESSAGES
113 ;
114 ;

```

115	004236	000				.BYTE	0
116	004237	015	012	015		.BYTE	15.12.15.12
117	004243	105	116	104		.ASCII	/END OF SEARCH/
118	004260	015	012	000		.BYTE	15.12.0
119	004263	015	012			.BYTE	15.12
120	004265	125	116	111		.ASCII	/UNIDENTIFIED INTERRUPT/
121	004314	015	012			.BYTE	15.12
122	004316	116	117	116		.ASCII	/NON-ZERO BCL ENTRY/
123	004341	377			ASCII:	.BYTE	377
124						.EVEN	
125						.LIST	BEX
126						.NLIST	CND

```

128.      ;
129.      ;
130.      ;      INITIALIZE HQR.
131.      ;      START OFF WITH MASTER RESET, NO-CLOCKS IN CSR #1
132.      ;
133.      ;      START:
134.      004342.      MOV.      #177777,-(SP)      ;CLEAR CSR1
135.      004346.      MOV.      #0#RSET,-(SP)      ;SET RESET
136.      004352.      CALL.     CSR1      ;RESET HQR
137.      004356.      MOV.      #0#RSET,-(SP)      ;CLEAR RESET
138.      004362.      MOV.      *(<0#MNC+0#CNC+0#RNC>,-(SP);SET NO CLKS.
139.      004366.      CALL.     CSR1      ;MOVE TO CSR1
140.      ;
141.      ;
142.      ;      INITIALIZE PPS.
143.      004372.      MOV.      #0#QLA,-(SP)      ;ADDRESS SELECT FOR QLB PAGES.
144.      004376.      CALL.     PPCR
145.      004402.      MOV.      #2000,-(SP)      ;SEND ADDRESS X'400' (ILLEGAL)
146.      004406.      CALL.     LBPP
147.      ;
148.      ;      RESET MRP AND CP.
149.      ;
150.      004412.      CLR.      -(SP)      ;CLEAR NOTHING IN CSR1
151.      004414.      MOV.      #0#MSET,-(SP)      ;SET RESET
152.      004420.      CALL.     CSR1
153.      004424.      MOV.      #0#MSET,-(SP)      ;CLEAR RESET
154.      004430.      CLR.      -(SP)      ;SET NOTHING
155.      004432.      CALL.     CSR1
156.      ;
157.      004436.      CLR.      -(SP)      ;CLEAR NOTHING
158.      004440.      MOV.      #0#CSET,-(SP)      ;SET RESET
159.      004444.      CALL.     CSR1
160.      004450.      MOV.      #0#CSET,-(SP)      ;CLEAR RESET
161.      004454.      CLR.      -(SP)      ;SET NOTHING
162.      004456.      CALL.     CSR1
163.      ;
164.      ;      SET 'QLB ERASE'
165.      ;
166.      004462.      MOV.      #100,-(SP)      ;SEND X'40' TO PPCR
167.      004466.      CALL.     PPCR
168.      ;
169.      ;      MAKE SURE AN HRL BUFFER IS READY
170.      ;
171.      004472.      WTSE$S.  #CF,HRL      ;WAIT UNTIL AN HRL BUFFER IS READY
172.      004504.      CLEF$S.  #CF,HRL
173.      004516.      MOV.      000000G,177352. ;SET START OF HRL BUFFER
174.      ;
175.      ;
176.      ;
177.      ;
178.      ;
179.      ;      PROVIDE FOR INTERRUPTS FROM HQR.
180.      ;
181.      004524.      MOV.      #TKTCB,TSKTCB.      ;SAVE MY TCB
182.      004532.      MOV.      @#274,OLDVEC.      ;SAVE VECTOR AT 274
183.      004540.      MOV.      #BPTISR,@#274      ;MOVE IN MY INTERRUPT HANDLER ADDR
184.      ;

```



```

186 004546 005767 000044G..... TST.....SREC+SR,NDC+2..... ;TEST FOR EMPTY DATA BASE
187 004552 001016 BNE.....QKIK ;BRANCH IF NOT EMPTY
188 004554 005767 000042G. TST.....SREC+SR,NDC.
189 004560 001013 BNE.....QKIK
190
191 004562 052767 000001 173236 ;BIS.....#EOC,BASE ;SET LOCAL EOC FLAG
192 004570 012767 140000 000006G. MOV.....#140000,DHR0+6 ;WRITE EOC WORD TO DHR BUFFER
193 004576 012767 040000 000010G. MOV.....#040000,DHR0+10 ;WRITE EOB WORD TO DHR BUFFER
194 004604 000167 000476 JMP.....HBOUO ;OUTPUT LAST DHR BUFFER
195
196 004610 ; QKIK:
197
198 ;
199 ; SET MRP AND CP TO ZERO
200 004610 005046 CLR.....-(SP) ;START MRP AT ZERO
201 004612 CALL.....SEQMM ;SET START ADDRESS
202 004616 005046 CLR.....-(SP) ;REINHIBIT BR
203 004620 CALL.....MRPCR
204 004624 005046 CLR.....-(SP) ;START CP AT ZERO
205 004626 CALL.....SEQCS ;SEQUENCE ONLY UP TO START ADDRESS
206 004632 005046 CLR.....-(SP) ;REINHIBIT BR
207 004634 CALL.....CPCR
208
209 004640 012767 000600 176422 MOV.....#<Q$CHB+Q$CHRL>,QR$CR2 ;CLEAR OPERATIONAL INTERRUPTS
210 004646 012767 001000 176422 MOV.....#Q$REBK,QR$CR2 ;RE-ARM INTERRUPTS
211 004654 012767 120000 176422 MOV.....#<Q$SM+Q$ENOP>,QR$CR2 ;SET SEARCH MODE
212 004662 012746 000360 MOV.....#Q$CSEL,-(SP) ;CLR SELECT BITS
213 004666 052716 001001 BIS.....#<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE
214 004672 052716 176000 BIS.....#Q$NCLK,(SP) ;CLEAR NO-CLOCKS
215 004676 005046 CLR.....-(SP) ;SET NOTHING
216 004700 CALL.....CSR1 ;RE-WRITE CSR1
217
218 ;
219 ; SET ATTENTION CODE AND SEND BCL ADDRESS
220 004704 012767 000001 176424 MOV.....#Q$BCL,QR$LBR ;SET ATTN CODE = 1
221 004712 012767 120100 176422 MOV.....#<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
222 004720 016701 176422 MOV.....QR$CR2,R1 ;READ CSR #2
223 004724 032701 000100 BIT.....#Q$ATTN,R1 ;ATTN REQUEST RECOGNIZED
224 004730 001373 BNE.....1$
225
226 004732 012767 004064 176424 MOV.....#BCL,QR$LBR ;MOVE BCL ADDR TO LOD BUS REG
227 004740 012767 120040 176422 MOV.....#<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP SLAVE DATA READY
228 004746 016701 176422 MOV.....QR$CR2,R1 ;READ CSR #2
229 004752 032702 000040 BIT.....#Q$CCCP,R2 ;CCCP RECOGNIZED (CLEAR)
230 004756 001373 BNE.....2$ ;NO WAIT
231
232 ;
233 ; NOW RUN 'SEARCH' TO START UP TERM DETECTOR
234
235 ;
236 004760 RQST#C SEARCH
237
238 ;
239 ; SET UP TIMER FOR HIT BUFFER FLUSH
240 INTERVAL = 5 SEC
241
242 004766

```

```

244 ;
245 ;
246 ;
247 ;
248 ;
249 ;
250 ;
251 ;
252 005012. WINT:
253 005012. 1$: RDAF$S #EFBUF :READ EVENT FLAGS
254 ;
255 005024 032767 000014 172756 BIT *(<BIT3+BIT2>),EFBUF :ANY INTERRUPTS?
256 005032 001767 BEQ 1$ :NO, READ AGAIN
257 ;
258 ;
259 ;
260 ;
261 ;
262 ;
263 ;
264 ;
265 ;
266 005034 032767 000010 172746 BIT #BIT3,EFBUF :EF#4 = TIMER EXPIRATION
267 005042 001430 BEQ HBINT :NO, HARDWARE INTERRUPT
268 005044 012767 000002 176424 MOV #2,QR$LBR :MOVE ATTN CODE TO LOD BUS REG
269 005052 012767 120100 176422 MOV *(Q$ATTN+Q$SM+Q$ENOP),QR$CR2 :SET ATTN CODE READY
270 ;
271 005060 GLEF$S #EFN,4 :CLEAR TIMER EVENT FLAG
272 005072 005267 172734 INC DATA1 :COUNT TIMER INTERRUPTS
273 ;
274 ;
275 ;
276 ;
277 005076 MRKT$S #EFN,4,#2,#5 :START TIMER AGAIN
278 005122 000733 BR WINT :WAIT FOR HIT BUFFER INTERRUPT

```

```

280 ;
281 ;
282 ;
283 ;      HARDWARE INTERRUPT.
284 ;      CHECK FOR HIT BUFFER INTERRUPT. IF A HIT BUFFER
285 ;      INTERRUPT HAS COME THROUGH, THE EXPECTED BCL ENTRY
286 ;      SHOULD HAVE BEEN CLEARED. THE FIELD HBOFF CONTAINS
287 ;      THE OFFSET FROM THE TOP OF THE HIT BUFFER SECTION
288 ;      OF THE BCL TO THE ENTRY THAT SHOULD BE CLEAR.
289 ;
290 005124 016767 176422 172676 HBINT: MOV.   QR$CR2,APLACE.      ;READ CSR 2.
291 005132 042767 177770 172670 BIC.   #177770,APLACE.    ;MASK OFF INTERRUPT BITS.
292 005140 122767 000003 172662 CMPB.  #0$IHB,APLACE.     ;HIT LIST INTERRUPT.
293 005146 001402 .        .        BEQ.   10$.              ;YES.
294 005150 000167 000506 .        .        JNP.   HRLINT.          ;TRY HRL INTERRUPT.
295 005154 016700 172654 10$: MOV.   HBOFF,R0          ;LOAD OFFSET INTO BCL.
296 005160 005760 004066 TST.   HBBCL(R0)         ;HAS BCL ENTRY BEEN CLEARED.
297 005164 001404 .        .        BEQ.   1$.              ;YES, CONTINUE.
298 005166 .        .        CALL.  ERR1             ;ENTRY MUST BE CLEAR
299 005172 000167 000702 .        .        JMP.   EXIT
300 ;
301 ;
302 ;      TEST FOR END-OF-COLLECTION.
303 ;      CP MICROCODE MOVES ADDRESS OF END-OF-BUFFER WORD
304 ;      INTO THE BCL. THE END-OF-BUFFER WORD IS AT THE
305 ;      SAME OFFSET FROM THE TOP OF THE END-OF-BUFFER
306 ;      SECTION OF THE BCL AS THE CLEARED HIT BUFFER
307 ;      ADDRESS IS FROM THE TOP OF THE HIT BUFFER ADDRESS.
308 ;      SECTION OF THE BCL. R0 = OFFSET.
309 ;
310 005176 .        .        1$: MOV.   E0BBCL(R0),R1      ;END-OF-COLLECTION (X'C000')
311 005202 005711 .        .        TST.   @R1
312 005204 100040 .        .        BPL.   HBOUT.          ;NO, CONTINUE.
313 005206 052767 000001 172612 .        .        BIS.   #EOC.BASE.      ;SET LOCAL FLAG FOR EOC.
314 ;
315 005214 .        .        GTIM$: SHETIM.          ;EOC TIME.
316 005222 016701 000010G. MOV.   SHETIM+G,TIM1,R1    ;MINUTES.
317 005226 026767 000006G.000006G. CMP.   SHETIM+G,TIHR,SHSTIM+G,TIHR.
318 005234 001402 .        .        BEQ.   2$.              ;BRANCH IF NO HOUR WRAP AROUND
319 005236 062701 000074 .        .        ADD.   #60.,R1          ;HOUR WRAP AROUND.
320 005242 012700 000074 .        .        MOV.   #60.,R0          ;CALCULATE SECONDS.
321 005246 .        .        CALL.  $MUL
322 005252 010102 .        .        MOV.   R1,R2.
323 005254 066702 000012G. .        .        ADD.   SHETIM+G,TISC,R2.    ;SECONDS.
324 ;
325 005260 016701 000010G. MOV.   SHSTIM+G,TIM1,R1    ;START TIME - MINUTES.
326 005264 012700 000074 .        .        MOV.   #60.,R0
327 005270 .        .        CALL.  $MUL
328 005274 066701 000012G. .        .        ADD.   SHSTIM+G,TISC,R1    ;SECONDS.
329 ;
330 ;      ELAPSED TIME.
331 ;
332 005300 160102 .        .        SUB.   R1,R2.          ;SECONDS.
333 005302 010267 000000G. .        .        MOV.   R2,SHTIME.      ;SAVE IN SREC.
334 ;
335 ;
336 ;      RELEASE HIT BUFFER TO CPU.

```

337											
338	005386	012767	042110	000000G	H3OUT:	MOV:	#HD,DHR0				:SET UP HEADER:
339	005314	005067	000002G			CLR:	DHR0+2				
340	005320	005067	000004G			CLR:	DHR0+4				
341	005324	000445				BR	SEND				:SKIP REFORMATTING BYPASS:

```

343 ;
344 ;
345 ; INACTIVE CODE.
346 ;
347 ; REFORMATS DHR BUFFER TO OLD OUTPUT FORMAT.
348 ;
349 ;
350 ;
351 ;
352 ;
353 ;
354 005326 012703 000044* MOV #LOCAL,R3 ;POINT TO LOCAL BUFFER.
355 005332 012704 000006G MOV #DHR0+6,R4 ;POINT TO COMMON.
356 005336 032713 140000 BIT #140000,(R3) ;EOC?.
357 005342 001402 BEQ MOVLOC ;NO, CONTINUE.
358 005344 012324 MOV (R3)+,(R4)+ ;MOVE EOC WORD TO COMMON.
359 005346 000434 BR SEND ;SIGNAL CCOUT.
360 ;
361 ; REFORMAT HIT BUFFER FROM THE NEW FORMAT (ONE HIT
362 ; REPORT PER WORD) TO OLD (HIT COUNT + QID'S IN
363 ; BYTES). SEE INTERFACE SPEC FOR DETAILS.
364 ;
365 ; R3 -> LOCAL BUFFER (NEW FORMAT)
366 ; R4 -> COMMON
367 ;
368 005350 MOVLOC:
369 005350 012324 MOV (R3)+,(R4)+ ;MOVE DOC ID.
370 005352 012324 MOV (R3)+,(R4)+
371 005354 012324 MOV (R3)+,(R4)+
372 005356 010467 172456 MOV R4,PCOUNT ;SAVE POINTER TO COUNT BYTE.
373 005362 005204 INC R4 ;POINT PAST COUNT BYTE.
374 005364 032713 020000 14: BIT #020000,(R3) ;QID?.
375 005370 001406 BEQ 2$ ;NO, TEST EOB.
376 005372 005267 172444 INC COUNT ;COUNT QID'S.
377 005376 111324 MOV (R3),(R4)+ ;MOVE QID TO COMMON.
378 005400 062703 000002 ADD #2,R3 ;BUMP LOCAL POINTER.
379 005404 000767 BR 1$ ;MOVE NEXT.
380 ;
381 005406 032704 000001 2$: BIT #1,R4 ;ODD ADDRESS.
382 005412 001401 BEQ 3$ ;NO.
383 005414 005204 INC R4 ;BUMP TO NEXT WORD.
384 005416 116777 172420 172414 3$: MOV COUNT,@PCOUNT ;MOVE IN QID COUNT.
385 005424 005067 172412 CLR COUNT.
386 005430 032713 040000 BIT #040000,(R3) ;EOB (OR EOC)
387 005434 001745 BEQ MOVLOC ;NO, MOVE NEXT DHR.
388 005436 012324 MOV (R3)+,(R4)+ ;MOVE EOB OR EOC CODE.
389 ;

```

```

391 ; NOW SIGNAL CCOUT.
392 ;
393 005440 ; SEND:
394 005440 052767 000001 000000G BIS #BIT0.MSQ ; SIGNAL CCOUT THAT DHR0 IS READY.
395 ;
396 005446 GLEF$# #CF.DHR ;
397 ;
398 005460 SETF$# #CF.COT ; SET EVENT FLAG FOR CCOUT.
399 ;
400 005472 DECL$# ; MAKE SURE CCOUT WILL LOOK AT MSQ.
401 ;
402 005500 WTSE$# #CF.DHR ; WAIT FOR BUFFER TO BE FREED.
403 ;
404 ;
405 ; CHECK FOR LOCAL END-OF-COLLECTION FLAG SET
406 005512 032767 000001 172306 BIT #E0C.BASE ; WAS EDC INDICATOR ON IN HIT BUFFER.
407 005520 001402 BEQ 3$ ; NO CONTINUE.
408 005522 000167 000344 JMP CALLFOS ; CALL FOS PGM AND EXIT.
409 ;
410 ; PLACE ADDRESS OF READY BUFFER INTO BCL. MOVE HIT
411 ; BUFFER ADDRESS FROM A PRIVATELY HELD COPY OF THE
412 ; COMPLETE BCL (NEVER CLEARED) BACK INTO THE WORKING
413 ; BCL. THAT IS, REPLACE THE BCL ENTRY THAT WAS CLEARED
414 ; BY THE CP MICROCODE. R0 = OFFSET INTO BCL.
415 ;
416 005526 016060 004044 004066 35: MOV MYHB(R0),HBBCL(R0) ; PUT READY BUFFER ADDRESS IN BCL.
417 ;
418 ; POINT TO NEXT HIT BUFFER ENTRY IN BCL. THAT IS, UPDATE
419 ; THE OFFSET. ROLL THE OFFSET OVER IF IT REACHES THE LIMIT.
420 ;
421 005534 062767 000002 172272 ADD #2,HBOFF ; ADVANCE OFFSET.
422 005542 022767 000002 172264 CMP #BUFACT,HSOFF ; REACHED LIMIT?
423 005550 003002 BGT 4$ ; NO RESET.
424 005552 005067 172256 CLR HBOFF ; RESET OFFSET.
425 ;
426 ; CLEAR TIMER, THEN START IT UP AGAIN.
427 ; AFTER ALL, WE DID RECEIVE A HIT BUFFER.
428 ;
429 005556 4$: CMKT$# ;
430 ;
431 005564 GLEF$# #EFN.4 ; CLEAR TIMER EVENT FLAG.
432 ;
433 005576 MRKT$# #EFN.4,#2,#5 ;
434 ;
435 ; CLEAR HARDWARE INTERRUPT EVENT FLAG.
436 ;
437 005622 GLEF$# #EFN.3 ; CLEAR INTERRUPT EVENT FLAG.
438 ;
439 ; RE-ARM CP INTERRUPTS. NOTE THAT CP MICROCODE CAN
440 ; CLEAR BCL ENTRIES ONLY WHEN INTERRUPTS ARE ARMED.
441 ; ONCE AN INTERRUPT IS RECEIVED, INTERRUPTS ARE
442 ; AUTOMATICALLY DISARMED IN THE CP. THIS WAY THE
443 ; CP CANNOT BE CLEARING ONE ENTRY IN THE BCL WHILE
444 ; THIS PROGRAM IS WORKING ON ANOTHER ENTRY.
445 ;
446 005634 012767 100400 176422 MOV #<Q$SM+Q$CHB>,OR$CR2 ; CLEAR HIT BUFFER INTERRUPT.
447 005642 012767 101000 176422 MOV #<Q$SM+Q$REBK>,OR$CR2 ; RE-ARM INTERRUPTS.

```

448	005650	012767	120000	176422	MOV	*(Q\$SM+Q\$ENOP),QR\$CR2	:ENABLE OPERATIONAL INTERRUPTS
449	005656	000167	177130		JMP	WINT	:WAIT FOR INTERRUPTS

```

451 ;
452 ;
453 ; CHECK FOR HRL INTERRUPT.
454 ;
455 ;
456 005662 122767 000002 172140 HRLINT: CMPB #0$IHRL,APLACE ;HRL INTERRUPT.
457 005670 001404 BEQ 1$ ;YES, CONTINUE.
458 005672 CALL ERR2 ;UNIDENTIFIED INTERRUPT.
459 005676 000167 000176 JMP EXIT ;LEAVE.
460 ;
461 ; IF AN HRL INTERRUPT HAS COME THROUGH, THE EXPECTED
462 ; BCL ENTRY SHOULD HAVE BEEN CLEARED. THE FIELD HROFF
463 ; CONTAINS THE OFFSET FROM THE TOP OF THE HRL SECTION
464 ; OF THE BCL TO THE ENTRY THAT SHOULD BE CLEAR.
465 ;
466 005702 016700 172130 1$: MOV HROFF,R0 ;LOAD OFFSET INTO BCL.
467 005706 005760 004076* TST HRBCL(R0) ;HAS BCL ENTRY BEEN CLEARED.
468 005712 001404 BEQ 2$ ;YES, CONTINUE.
469 005714 CALL ERR1 ;ENTRY MUST BE CLEAR
470 005720 000167 000154 JMP EXIT
471 ;
472 ; GET HRL BUFFER ADDRESS FROM HRLGET.
473 ;
474 005724 2$: CLR HRLBUF ;CLEAR BUFFER ADDRESS IN COMMON.
475 005724 005067 000000G SETF#$ #CF,HBR ;SIGNAL HRLGET FOR NEW BUFFER.
476 005730
477 ;
478 005742 DECL$#
479 ;
480 005750 WTSE#$ #CF,HRL ;WAIT FOR BUFFER READY.
481 ;
482 005762 CLEF#$ #CF,HRL
483 ;
484 ; PLACE ADDRESS OF READY BUFFER INTO BCL. MOVE HRL
485 ; ADDRESS FROM A PRIVATELY HELD COPY OF THE COMPLETE
486 ; BCL (NEVER CLEARED) BACK INTO THE WORKING BCL. THAT
487 ; IS, REPLACE THE BCL ENTRY THAT WAS CLEARED BY THE CP
488 ; MICROCODE, R0 = OFFSET INTO BCL.
489 ;
490 005774 016760 000000G 004054* MOV HRLBUF,MYHR(R0) ;GET HRL BUFFER ADDRESS.
491 006002 016060 004054* 004076* MOV MYHR(R0),HRBCL(R0) ;PUT READY BUFFER ADDRESS IN BCL.
492 ;
493 ; POINT TO NEXT HRL ENTRY IN BCL. THAT IS, UPDATE THE OFFSET.
494 ; THE OFFSET, ROLL THE OFFSET OVER IF IT REACHES THE LIMIT.
495 ;
496 006010 062767 000002 172020 ADD #2,HROFF ;ADVANCE OFFSET.
497 006016 022767 000002 172012 CMP #BUFACT,HROFF ;REACHED LIMIT?
498 006024 003002 BGT 3$ ;NO RESET.
499 006026 005067 172004 CLR HROFF ;RESET OFFSET.
500 ;
501 ; CLEAR HARDWARE INTERRUPT EVENT FLAG.
502 ;
503 006032 3$: CLEF#$ #EFN,3 ;CLEAR INTERRUPT EVENT FLAG.
504 ;
505 ;
506 ; RE-ARM CP INTERRUPTS. NOTE THAT CP MICROCODE CAN
507 ; CLEAR BCL ENTRIES ONLY WHEN INTERRUPTS ARE ARMED.
; ONCE AN INTERRUPT IS RECEIVED, INTERRUPTS ARE

```



```
508 ; AUTOMATICALLY DISARMED IN THE CP. THIS WAY THE
509 ; CP CANNOT BE CLEARING ONE ENTRY IN THE BCL WHILE
510 ; THIS PROGRAM IS WORKING ON ANOTHER ENTRY.
511 ;
512 006044 012767 100200 176422 MOV #<0$SM+0$CHRL>,QR$CR2 ;CLEAR HRL INTERRUPT
513 006052 012767 101000 176422 MOV #<0$SM+0$REBK>,QR$CR2 ;RE-ARM INTERRUPTS
514 006060 012767 120000 176422 MOV #<0$SM+0$ENOP>,QR$CR2 ;ENABLE OPERATIONAL INTERRUPTS
515 006066 000167 176720 JMP WINT ;WAIT FOR INTERRUPTS
```

```

517 ;
518 ;
519 ; CALL FDS PGM
520 ; EXIT
521 ;
522 ;
523 006072 CALLFOS:
524 006072 ROST%0 HQRFDS
525 ;
526 006100 ;
527 006100 016737 171702 000274 EXIT:
528 ; MOV OLDVEC,0#274 ;RESTORE SYSTEM VECTOR ADDRESS
529 006106 005046 CLR -(SP) ;CLEAR NOTHING
530 006110 012746 176000 MOV #0#NCLK,-(SP) ;SET NO-CLOCKS
531 006114 CALL CSR1 ;WRITE CSR 1
532 006120 005067 176422 CLR QR%CR2 ;SET LOAD MODE
533 ; ; CAN ONLY CHANGE CP CNTL REG
534 ; ; IN LOAD MODE
535 006124 005046 CLR -(SP) ;SET CP CONTROL REG TO ZERO
536 006126 CALL CPCR ;DO IT
537 006132 012746 000040 MOV #0#CLR,-(SP) ;REINHIBIT FAL PROCESSOR
538 006136 CALL PPCR
539 006142 EX2:
540 006142 CALL ENDTST ;PUT OUT LAST MESSAGE
541 ;
542 ; FORCE HRLGET TERMINATION
543 ;
544 006146 SETF%0 #CF,HBR
545 ;
546 006160 DECL%0
547 ;
548 006166 EXIT%0

```

```

550 ;
551 ;
552 ;
553 ; ROUTINE TO PLACE A VALUE INTO CONTROL/STATUS
554 ; REGISTER 1 (176420).
555 ;
556 ; READ THE CURRENT VALUE OF CSR1. CLEAR THE BITS
557 ; AT 4(SP) IN A CSR1 WORK AREA, SET THE BITS AT 2(SP)
558 ; IN THE SAME WORK AREA. RE-WRITE CSR1.
559 ;
560 ; INPUT:
561 ; 2(SP) - BITS TO BE SET IN CSR1
562 ; 4(SP) - BITS TO BE CLEARED IN CSR1
563 ;
564 ; CSR1::
565 006174 016767 176420 171626 MOV. OR$CR1,APLACE. ;GET THE CURRENT VALUE
566 006202 046667 000004 171620 BIC. 4(SP),APLACE. ;CLEAR FIRST
567 006210 056667 000002 171612 BIS. 2(SP),APLACE. ;THEN SET
568 006216 016767 171606 176420 MOV. APLACE,OR$CR1 ;NOW RETURN IT
569 006224 011666 000004 CMP. (SP),4(SP) ;MOVE RETURN ADDR TO TOP OF STACK
570 006230 022626 RETURN. ;BUMP STACK POINTER PAST ARGS
571 006232 ; LEAVE

```

```

573 ;
574 ;
575 ;
576 ; INTERRUPT SERVICE ROUTINE
577 ; TRAP INTERRUPTS FROM HQR THROUGH VECTOR ADDRESS 274
578 ; SET EVENT FLAG 3
579 ; CP DEBUGGING ROUTINES WILL READ CSR #2 AND DECODE THE INTERRUPT
580 ;
581 ; BPTISR:
582 ; SAVE R0,R1,R2,R3,R4,R5
583 ;
584 ; MOV TSKTCB,R5 ;LOAD MY TCB
585 ; MOV #EFN,3,R0 ;EVENT FLAG TO BE SET
586 ; CALL $CEFI
587 ; BIS R0,(R1) ;SET LOCAL FLAG
588 ; CALL $DRDSE ;DECLARE SIGNIFICANT EVENT
589 ;
590 ; RESTOR R0,R1,R2,R3,R4,R5
591 ; RTI

```

```

593      ;
594      ;
595      ;      WRITE TO TT0 AND PROMPT.
596      ;
597      ;
598 006310 005267 171510      ENDTST: INC      ERWORD.
599 006314 005267 171504      ERR2:  INC      ERWORD.
600 006320 005267 171500      ERR1:  INC      ERWORD.
601      ;
602      ;      USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF THE
603      ;      MESSAGE TABLE. FIND THE END OF THE MESSAGE FIRST, THEN
604      ;      THE BEGINNING, THEN GET THE LENGTH.
605      ;
606 006324 016702 171474      MOV     ERWORD,R2      ;LOAD LOOP COUNT.
607 006330 012701 004341*    MOV     #ASCIZ,R1      ;POINT TO END OF MESSAGE TABLE
608 006334 105741              15$: TSTB   -(R1)           ;LOOK FOR END OF MESSAGE.
609 006336 001376              BNE     1$
610 006340 005302              DEC     R2              ;LOOP COUNT.
611 006342 001374              BNE     1$              ;BACK UP ANOTHER MESSAGE.
612 006344 010100              MOV     R1,R0           ;SAVE POINTER TO END OF MESSAGE.
613 006346 105741              25$: TSTB   -(R1)           ;BACK UP TO BEGINNING OF MESSAGE.
614 006350 001376              BNE     2$
615 006352 005201              INC     R1              ;BUMP TO FIRST CHAR OF MESSAGE
616 006354 160100              SUB     R1,R0           ;R0 NOW = MESSAGE LENGTH.
617      ;
618 006356              QIOW$# #IO,WVB,#LUN,TT,#EFN.1,,#STAT,,<R1,R0>,ABEND
619      ;
620 006432              CLEF$# #EFN.1
621 006444 105767 171350      TSTB   STAT            ;GOOD RETURN.
622 006450 003403              BLE     ABEND           ;NO.
623      ;
624      ;      CLEAR ERROR INDEX AND EXIT.
625      ;
626 006452 005067 171346      CLR     ERWORD.
627 006456              RETURN                ;AND RETURN.
628      ;
629 006460      ABEND: ABRT$# #MYSELF.
630      ;      .END      START.

```

ABEND = 006460R
ALUCKE = 040000
ALUDE = 004000
APLACE = 000030R
ASCIZ = 004341R
A01 = 010000
BASE = 000026R
BCL = 004064R
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BPTISR = 006234R
BUFACT = 000002
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CALLFO = 006072R
CBKALL = 001000
CBKCLK = 000400
CF.COT = 000041
CF.DGN = 000046
CF.DHR = 000042
CF.DMC = 000047
CF.HBR = 000045
CF.HRL = 000044
CF.UPD = 000043
CNOBRE = 100000
COUNT = 000042R
CPCCEN = 010000
CPCR = ***** GX
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSECCI = 100000
CSOE = 000040
CSR1 = 000174RG
CSWRTE = 000100
DATA1 = 000032RG
DBR.RD = 000001
DB#CPP = 001457
DB#SPT = 000026
DB#TPC = 000023
DG.ERR = 001000
DG.SDF = 002000
DG.TDF = 004000
DHR0 = ***** GX
DISPGS = 100000
DMAALR = 000005
DNARLD = 000003
DNARUR = 000004
EFBUF = 000010R
EFN.1 = 000001
EFN.3 = 000003
EFN.4 = 000004
ENBR = 010000
ENDTST = 006310R
EOBCLL = 004105R
EOC = 000001
ERR1 = 006320R
ERR2 = 006314R
ERWORD = 000024R
EXIT = 006100R
EX2 = 006142R
FN.ACK = 000016
FN.FSA = 000000
FN.FSB = 000002
FN.FSC = 000004
FN.FSD = 000020
FN.MHR = 000010
FN.NMB = 000022
FN.QLS = 000006
FN.RDC = 000014
FN.UPD = 000012
G.TIBA = 000002
G.TICP = 000016
G.TICT = 000014
G.TIDA = 000004
G.TIHR = 000006
G.TIMI = 000010
G.TIMO = 000002
G.TISC = 000012
G.TIYR = 000000
HBBCL = 004066R
HBINT = 005124R
HBOFF = 000034R
HBOUT = 005306R
HRBCL = 004076R
HRLBUF = ***** GX
HRLINT = 005662R
HRL0 = ***** GX
HROFF = 000036R
IO.LWB = ***** GX
LBPP = ***** GX
LOCAL = 000044R
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LUN.TT = 000001
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADR0 = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MOVLOC = 005350R
MREN1 = 000001
MREN2 = 020000
MRPCR = ***** GX
MSL = ***** GX
MSYN = 000040
MS.DGN = 010000
MYHB = 004044R
MYHR = 004054R
002.MYSELF = 000000R
002.N = 000144
002.OLDVEC = 000006R
002.PCOUNT = 000040R
002.PLB = 000010
002.PLC = 000020
002.PLD = 000030
002.PLRWR = 000200
002.PLR.EN = 000200
002.PPCR = ***** GX
PRINT = 004120R
QKIK = 004510R
QR#CR1 = 176420
QR#CR2 = 176422
QR#LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHS = 000003
Q\$IHLR = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LCD = 000003
Q\$LDM = 000004
Q\$LDPP = 002000
Q\$LPH = 010000
Q\$PNC = 140000
Q\$PR = 000052
Q\$IRP = 000040
Q\$IRP2 = 000240
Q\$MSC = 040000

Q\$MSET = 000004	S\$QX = 000004	T\$DBRA = 000034	WORD32 = 000100	WORD71 = 000216
Q\$MSP = 100000	S\$SR = 000007	T\$OBWA = 000032	WORD33 = 000102	WORD72 = 000220
Q\$NCLK = 176000	S\$S1 = 000010	T\$OUTA = 100000	WORD34 = 000104	WORD73 = 000222
Q\$PP = 000100	S\$S2 = 000014	T\$RBD0 = 000200	WORD35 = 000106	WORD74 = 000224
Q\$PPSW = 000320	TD\$CTR = 176370	T\$RNB = 000040	WORD36 = 000110	WORD75 = 000226
Q\$PP2 = 000300	TD\$CTW = 176360	T\$RSET = 040000	WORD37 = 000112	WORD76 = 000230
Q\$OHLT = 000013	TD\$INL = 004000	T\$SC = 000022	WORD38 = 000114	WORD77 = 000232
Q\$DL = 000043	TD\$MEM = 000270	T\$SCLK = 020000	WORD39 = 000116	WORD78 = 000234
Q\$QLA = 000053	TD\$OAR = 176344	T\$SEG1 = 000000	WORD40 = 000120	WORD79 = 000236
Q\$QLB = 000054	TD\$OTR = 176346	T\$SEG2 = 000001	WORD41 = 000122	WORD80 = 000240
Q\$QLR = 000001	TD\$ORD = 000274	T\$SEG3 = 000002	WORD42 = 000124	WORD81 = 000242
Q\$QW = 000042	TD\$RST = 176366	T\$SO = 000001	WORD43 = 000126	WORD82 = 000244
Q\$RDCD = 000005	TD\$SW = 176376	T\$UBUS = 100000	WORD44 = 000130	WORD83 = 000246
Q\$RDMD = 000006	TD\$STAR = 176372	T\$ICLK = 000400	WORD45 = 000132	WORD84 = 000250
Q\$REBK = 001000	TD\$TAJW = 176362	T\$BBEN = 000020	WORD46 = 000134	WORD85 = 000252
Q\$RNC = 006000	TD\$TDR = 176374	UBD.IN = 000020	WORD47 = 000136	WORD86 = 000254
Q\$RSC = 004000	TD\$TDU = 176364	WINT = 005012R	WORD48 = 000140	WORD87 = 000256
Q\$RSET = 000010	TSKTCB = 000004R	WORD0 = 000000	WORD49 = 000142	WORD88 = 000260
Q\$SM = 100000	T\$AD = 000020	WORD1 = 000002	WORD50 = 000144	WORD89 = 000262
Q\$SP = 000120	T\$BA = 000002	WORD10 = 000024	WORD51 = 000146	WORD90 = 000264
Q\$SP2 = 000340	T\$BD = 000010	WORD11 = 000026	WORD52 = 000150	WORD91 = 000266
RG0.EN = 000200	T\$BS0 = 100000	WORD12 = 000030	WORD53 = 000152	WORD92 = 000270
RG0.VA = 020000	T\$BT = 000020	WORD13 = 000032	WORD54 = 000154	WORD93 = 000272
R.QSGC = 000015	T\$BTAR = 000030	WORD14 = 000034	WORD55 = 000156	WORD94 = 000274
R.QSPC = 000014	T\$BDT = 002000	WORD15 = 000036	WORD56 = 000160	WORD95 = 000276
R.QSPH = 000006	T\$CD = 000100	WORD16 = 000040	WORD57 = 000162	WORD96 = 000300
R.QSPR = 000012	T\$CLK = 002000	WORD17 = 000042	WORD58 = 000164	WORD97 = 000302
R.QSTN = 000002	T\$DISK = 000200	WORD18 = 000044	WORD59 = 000166	WORD98 = 000304
SEND = 005440R	T\$DRD = 000004	WORD19 = 000046	WORD60 = 000170	WORD99 = 000306
SEDCS = ***** GX	T\$MEM = 010000	WORD20 = 000048	WORD61 = 000172	WORDVAL = 000310
SEQ#1 = ***** GX	T\$FSA = 000000	WORD21 = 000052	WORD62 = 000174	XTREAD = 001000
SEQ.C1 = 000010	T\$FSAB = 000004	WORD22 = 000054	WORD63 = 000176	XTWRITE = 000400
SHETIM = ***** G	T\$FSAC = 000014	WORD23 = 000056	WORD64 = 000200	\$\$\$CEFI = ***** GX
SHSTIM = ***** G	T\$FSB2 = 000010	WORD24 = 000060	WORD65 = 000202	\$\$\$DRDSE = ***** GX
SHTIME = ***** G	T\$IB = 000026	WORD25 = 000062	WORD66 = 000204	\$\$\$MUL = ***** G
SREC = ***** GX	T\$IBAR = 000024	WORD26 = 000064	WORD67 = 000206	\$\$\$STKCB = ***** GX
SR.NDC = 000042	T\$IBE = 020000	WORD27 = 000066	WORD68 = 000210	\$\$\$T = 000022R 003
START = 004342R	T\$IBF = 040000	WORD28 = 000070	WORD69 = 000212	\$\$\$ARG = 000002
STAT = 000020R	T\$ICD = 000040	WORD29 = 000072	WORD70 = 000214	\$\$\$DST = 000020
S\$CLR = 000000	T\$MODE = 004000	WORD30 = 000074		\$\$\$T1 = 000067
S\$LA = 000001	T\$OB = 000036	WORD31 = 000076		\$\$\$T2 = 000027
S\$OB = 000005	T\$OBE = 004000			
S\$OR = 000006	T\$OBF = 010000			

. ABS. 000000 000
006512 001
FNOFFS: 000022 002
\$DPB\$\$ 000040 003
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 6033 WORDS (24 PAGES)
DYNAMIC MEMORY: 7028 WORDS (27 PAGES)
ELAPSED TIME: 00:01:13
HQR, HQR/SP=C20, 1JC, C20, 1JIM, C20, 1JHQR

```

1
2 000000
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 000000
19 000000
20
21
22
23 000000 005046
24 000002 012746 176000
25 000006
26
27
28
29 000012 005067 176422
30
31
32
33 000016 005046
34 000020
35
36
37
38 000024 012746 140000
39 000030 012746 100000
40 000034
41
42
43
44 000040 052766 020000 000002
45 000046 016646 000002
46 000052
47
48 000056 011666 000002
49 000062 005726
50 000064

```

```

;
;
; TITLE MRPSUB
; PSECT MRPSUB
;
; HARDWARE QUERY RESOLVER 'MANUAL' DEBUGGING AIDS
; ***** PROTOTYPE VERSION *****
;
; MRP SUBROUTINES
;
; CONTROL STORE SEQUENCING (MATCH REPORT PROCESSOR)
; JUMP TO START ADDRESS (VIA BRANCH REGISTER)
;
; INPUT:
; 2(SP) START ADDRESS
;
; OUTPUT:
; MRP SEQUENCER SET TO START ADDRESS
;
MRPSUB:
SEQMM:
;
; STOP CP AND MRP CLOCKS
;
; CLR -(SP) ; CLEAR NOTHING
; MOV #0#NCLK,-(SP) ; SET NO CLOCKS
; CALL CSR1
;
; GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
;
; CLR QR#CR2 ; DO IT
;
; SELECT BRANCH REGISTER IN MRP
;
; CLR -(SP) ; DEFINE BR REG SELECT
; CALL MRPCRA ; WRITE CR (WITHOUT AUTOMATIC BR SUPPRESS)
;
; CLOCK MRP ONCE
;
; MOV #0#MNC,-(SP) ; CLEAR MRP NO CLOCKS
; MOV #0#MSP,-(SP) ; SET MRP SINGLE CLOCK
; CALL CSR1
;
; MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
;
; BIS #BIT13,2(SP) ; OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE
; MOV 2(SP),-(SP) ; COPY ARGUMENT INTO STACK FOR LBMRP CALL
; CALL LBMRP ; DO MOVE - FIRST CLOCK
;
; MOV (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
; TST (SP)+ ; POINT TO RETURN ADDRESS
; RETURN

```



```

52.      ;
53.      ; CONTROL STORE SEQUENCING (MATCH REPORT PROCESSOR)
54.      ; SET START ADDRESS FOR WRITE
55.      ;
56.      ; INPUT:
57.      ; 2(SP) START ADDRESS
58.      ;
59.      ; OUTPUT:
60.      ; MRP SEQUENCER SET TO START ADDRESS
61.      ;
62. 000066 WRTMM:
63.      ;
64.      ; STOP CP AND MRP CLOCKS
65.      ;
66. 000066 005046 CLR. -(SP) ; CLEAR NOTHING
67. 000070 012746 176000 MOV. #0$NCLK, -(SP) ; SET NO CLOCKS
68. 000074 CALL. CSR1
69.      ;
70.      ; GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
71.      ;
72. 000100 005067 176422 CLR. OR$CR2. ; DO IT
73.      ;
74.      ; SELECT BRANCH REGISTER IN MRP
75.      ;
76. 000104 005046 CLR. -(SP) ; DEFINE BR REG SELECT
77. 000106 CALL. MRPCRA. ; WRITE CR (WITHOUT AUTOMATIC BR INHIBIT)
78.      ;
79.      ; CLOCK MRP ONCE
80.      ;
81. 000112. 012746 140000 MOV. #0$MNC, -(SP) ; CLEAR MRP NO CLOCKS
82. 000116 012746 100000 MOV. #0$MSP, -(SP) ; SET MRP SINGLE CLOCK
83. 000122. CALL. CSR1
84.      ;
85.      ; MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
86.      ;
87. 000126 052766 020000 000002. BIS. #BIT13, 2(SP) ; OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE
88. 000134 016646 000002 MOV. 2(SP), -(SP) ; COPY ARGUMENT INTO STACK FOR LBMRP CALL
89. 000140 CALL. LBMRP. ; DO MOVE - FIRST CLOCK
90.      ;
91. 000144 011666 000002 MOV. (SP), 2(SP) ; MOVE RETURN ADDRESS DOWN STACK
92. 000150 005726 TST. (SP)+ ; POINT TO RETURN ADDRESS

```

```

94      ;
95      ;      DATA TRANSFER
96      ;      LOD-BUS REGISTER TO A DESTINATION ON THE MRP-BUS
97      ;
98      ;      INPUT:
99      ;      2(SP) DATA FOR PRE-SELECTED MRP DESTINATION
100     ;
101     000152.
102     000152. 016667 000002 176424 LBMRP:: MOV. 2(SP),Q#LBR. ;MOVE DATA TO LOD-BUS REG.
103     000160 012746 001001 MOV. *(Q#LBD+Q#LBP),-(SP) ;CLR DRIVE AND PULSE
104     000164 052716 000360 BIS. *(Q#CSEL), (SP) ;CLR SELECTION BITS
105     000170 012746 176000 MOV. *(Q#NCLK),-(SP) ;SET NO-CLOCKS
106     000174 052716 000240 BIS. #Q#MRP2, (SP) ;SELECT MRP
107     000200
108     ;
109     000204 012746 140000 MOV. #Q#MNC, -(SP) ;CLEAR MRP NO-CLOCK BITS
110     000210 012746 101000 MOV. *(Q#MSP+Q#LBD),-(SP) ;SET MRP CLOCK
111     000214
112     ;
113     ;      DE-SELECTION
114     ;
115     000220 012746 001001 MOV. *(Q#LBD+Q#LBP),-(SP) ;CLEAR DRIVE AND PULSE
116     000224 052716 000360 BIS. *(Q#CSEL), (SP) ;CLR SELECTION BITS
117     000230 012746 176000 MOV. *(Q#NCLK),-(SP) ;SET NO-CLOCKS
118     000234
119     ;
120     000240 011666 000002 MOV. (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
121     000244 005726 TST. (SP)+ ;POINT TO RETURN ADDRESS
122     000246 RETURN

```

```

124      ;
125      ;
126      ; DATA TRANSFER
127      ; LOD BUS REGISTER TO A DESTINATION ON THE MRP BUS
128      ; SINGLE CLOCK SEQUENCER ONLY
129      ;
130      ; INPUT:
131      ; 2(SP) DATA FOR PRE-SELECTED MRP DESTINATION
132      ;
133      ; LBMSC::
134      ; 000250 016667 000002 176424 MOV. 2(SP),DR#LBR. ;MOVE DATA TO LOD BUS REG
135      ; 000256 012746 001001 MOV. #<Q#LBD+Q#LBP>,-(SP) ;CLR DRIVE AND PULSE
136      ; 000262 052716 000360 BIS. #<Q#CSEL>,(SP) ;CLR SELECTION BITS
137      ; 000266 012746 176000 MOV. #<Q#NCLK>,-(SP) ;SET NO-CLOCKS
138      ; 000272 052716 000240 BIS. #Q#MRP2,(SP) ;SELECT MRP
139      ; CALL. CSR1
140      ;
141      ; MOV. #Q#MNC,-(SP) ;CLEAR MRP NO-CLOCK BITS
142      ; 000302 012746 140000 MOV. #<Q#MSC+Q#LBD>,-(SP) ;SET MRP CLOCK
143      ; 000306 012746 041000 CALL. CSR1
144      ;
145      ; DE-SELECTION
146      ;
147      ; MOV. #<Q#LBD+Q#LBP>,-(SP) ;CLR DRIVE AND PULSE
148      ; 000316 052716 000360 BIS. #<Q#CSEL>,(SP) ;CLR SELECTION BITS
149      ; 000322 012746 176000 MOV. #<Q#NCLK>,-(SP) ;SET NO-CLOCKS
150      ; 000326 012746 176000 CALL. CSR1
151      ;
152      ; MOV. (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
153      ; 000336 011666 000002 TST. (SP)+ ;POINT TO RETURN ADDRESS
154      ; 000342 005726 RETURN.

```

```

155      ;
156      ;      DATA TRANSFER TO LOD BUS REG FROM MRP.
157      ;
158      ;      OUTPUT:
159      (SP)      DATA FROM PRE-SELECTED MRP SOURCE.
160      ;
161      000346      MRPLB::
162      000346 012746 001001      MOV.  #<0$LBD+0$LBP>,-(SP)      ;CLR DRIVE AND PULSE
163      000352 052716 000360      BIS.  #<0$CSEL>,(SP)      ;CLR SELECTION BITS.
164      000356 012746 176000      MOV.  #<0$HCLK>,-(SP)      ;SET NO-CLOCKS.
165      000362 052716 000240      BIS.  #0$MRP2,(SP)      ;SOURCE IS MRP.
166      000366      CALL.  CSR1
167      ;
168      000372 011646      MOV.  (SP),-(SP)      ;MOVE RETURN ADDR UP STACK.
169      000374 016766 176424 000002.  MOV.  QR$LBR,2(SP)      ;MOVE DATA ONTO STACK.
170      ;
171      000402 012746 000240      MOV.  #0$MRP2,-(SP)      ;CLEAR MRP SELECT.
172      000406 005046      CLR.  -(SP)      ;SET NOTHING.
173      000410      CALL.  CSR1
174      000414      RETURN.

```

```

176      ;
177      ;
178      ;      MRP CONTROL REGISTER LOADING
179      ;
180      ;      INPUT:
181      ;      2(SP) BIT SETTING FOR MRP CONTROL REGISTER
182      ;
183 000416      MRPCTR:
184 000416 052766 100000 000002      BIS      #MNOBREG,2(SP)      ;INHIBIT BRANCH REGISTER SELECT
185 000424      MRPCTRA:
186 000424 016667 000002 176424      MOV      2(SP),Q#LBR      ;CONTROL BITS DESTINED FOR MRP
187 000432 012746 001001      MOV      #<Q#LBD+Q#LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
188 000436 052716 000360      BIS      #<Q#CSEL>,(SP)      ;CLR SELECTION BITS
189 000442 012746 000040      MOV      #Q#MRP,-(SP)      ;SELECT MRP
190 000446      CALL    CSR1
191      ;
192 000452 005046      CLR      -(SP)      ;CLEAR NOTHING
193 000454 012746 000001      MOV      #Q#LBP,-(SP)      ;SET PULSE
194 000460      CALL    CSR1
195      ;
196 000464 012746 000041      MOV      #<Q#MRP+Q#LBP>,-(SP)      ;CLEAR CR SELECTION AND PULSE
197 000470 005046      CLR      -(SP)      ;SET NOTHING
198 000472      CALL    CSR1
199      ;
200 000476 011666 000002      MOV      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
201 000502 005726      TST     (SP)+
202 000504      RETURN
203      ;
204      000001      .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLR,EN = 000200	Q#RNC = 006000
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR#CR1 = 176420	Q#RSC = 004000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR#CR2 = 176422	Q#RSET = 000010
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	QR#LBR = 176424	Q#SM = 100000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q#ATTN = 000100	Q#SP = 000120
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q#BCL = 000001	Q#SP2 = 000340
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q#CCCP = 000040	RG0,EN = 000200
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#CHB = 000400	RG0,VA = 020000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q#CHRL = 000200	SEMM = 000000RG 002
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	Q#CLR = 000040	SEQ,CI = 000010
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	Q#CNC = 030000	S#CLR = 000000
BIT15 = 100000	BYTE52 = 000064	CPREAD = 040000	Q#CP = 000060	S#LA = 000001
BIT2 = 000004	BYTE53 = 000065	CPURTE = 020000	Q#CPC = 000010	S#OB = 000005
BIT3 = 000010	BYTE54 = 000066	CSADRD = 000004	Q#CP2 = 000260	S#OR = 000006
BIT4 = 000020	BYTE55 = 000067	CSEDCI = 100000	Q#CSC = 010000	S#OX = 000004
BIT5 = 000040	BYTE56 = 000070	CSDOE = 000040	Q#CSEL = 000360	S#SR = 000007
BIT6 = 000100	BYTE57 = 000071	CSR1 = ***** GX	Q#CSET = 000002	S#S1 = 000010
BIT7 = 000200	BYTE58 = 000072	CSURTE = 000100	Q#CSP = 020000	S#S2 = 000014
BIT8 = 000400	BYTE59 = 000073	DBR,RD = 000001	Q#DMA = 000001	TD#CTR = 176370
BIT9 = 001000	BYTE6 = 000006	DB#CPP = 001457	Q#ENBK = 040000	TD#CTW = 176360
BYTE0 = 000000	BYTE60 = 000074	DB#SPT = 000026	Q#ENOP = 020000	TD#INL = 004000
BYTE1 = 000001	BYTE61 = 000075	DB#TPC = 000023	Q#FAL = 004000	TD#MEM = 000270
BYTE10 = 000012	BYTE62 = 000076	DISPGS = 100000	Q#FC = 000045	TD#OAR = 176344
BYTE11 = 000013	BYTE71 = 000077	DNARRD = 000005	Q#FO = 000044	TD#OTR = 176346
BYTE12 = 000014	BYTE64 = 000100	DNARRD = 000003	Q#FP = 000046	TD#ORD = 000274
BYTE13 = 000015	BYTE65 = 000101	DNARRR = 000004	Q#HBF = 000002	TD#SUJ = 176376
BYTE14 = 000016	BYTE66 = 000102	ENBR = 010000	Q#ICP = 000006	TD#TAR = 176372
BYTE15 = 000017	BYTE67 = 000103	LBMRP = 000152RG	002.Q#IHB = 000003	TD#TAW = 176362
BYTE16 = 000020	BYTE68 = 000104	LBMSC = 000250RG	002.Q#IHL = 000002	TD#TDR = 176374
BYTE17 = 000021	BYTE69 = 000105	LOC,EN = 000100	Q#IMRP = 000007	TD#TDW = 176364
BYTE18 = 000022	BYTE7 = 000007	LOC,WA = 040000	Q#LBD = 001000	T#AD = 000020
BYTE19 = 000023	BYTE70 = 000106	LOC,WB = 100000	Q#LBDP = 001001	T#BA = 000002
BYTE2 = 000002	BYTE71 = 000107	MAREN1 = 000001	Q#LBP = 000001	T#BD = 000010
BYTE20 = 000024	BYTE72 = 000110	MAREN2 = 004000	Q#LDCD = 000003	T#BS0 = 100000
BYTE21 = 000025	BYTE73 = 000111	MARL0D = 010000	Q#LDM = 000004	T#BT = 000020
BYTE22 = 000026	BYTE74 = 000112	MAROUT = 000002	Q#LDPP = 002000	T#BTAR = 000030
BYTE23 = 000027	BYTE75 = 000113	MAR,LO = 002000	Q#LHP = 010000	T#BTDR = 002000
BYTE24 = 000030	BYTE76 = 000114	MAR,OU = 000040	Q#MNC = 140000	T#CD = 000100
BYTE25 = 000031	BYTE77 = 000115	MBKALL = 001000	Q#MR = 000052	T#CLK = 002000
BYTE26 = 000032	BYTE78 = 000116	MBKCLK = 000400	Q#MRP = 000040	T#DISK = 000200
BYTE27 = 000033	BYTE79 = 000117	MHADRD = 000100	Q#MRP2 = 000240	T#DRD = 000004
BYTE28 = 000034	BYTE8 = 000010	MLEFT = 000002	Q#MSC = 040000	T#MEM = 010000
BYTE29 = 000035	BYTE80 = 000120	MND = 000004	Q#MSET = 000004	T#FSA = 000000
BYTE3 = 000003	BYTE81 = 000121	MNIURTE = 000010	Q#MSP = 100000	T#FSAB = 000004
BYTE30 = 000036	BYTE82 = 000122	MNOBRE = 100000	Q#NCLK = 176000	T#FSAC = 000014
BYTE31 = 000037	BYTE83 = 000123	MREN1 = 000001	Q#PP = 000100	T#FSB2 = 000010
BYTE32 = 000040	BYTE84 = 000124	MREN2 = 020000	Q#PPSUJ = 000320	T#IB = 000026
BYTE33 = 000041	BYTE85 = 000125	MRPCR = 000416RG	002.Q#PP2 = 000300	T#IBAR = 000024
BYTE34 = 000042	BYTE86 = 000126	MRPCRA = 000424RG	002.Q#QHLT = 000013	T#IBE = 020000
BYTE35 = 000043	BYTE87 = 000127	MRPLB = 000346RG	002.Q#QL = 000043	T#IBF = 040000
BYTE36 = 000044	BYTE88 = 000130	MRPSUB = 000000R	002.Q#QLA = 000053	T#ICD = 000040
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q#QLB = 000054	T#MODE = 004000
BYTE38 = 000046	BYTE9 = 000011	N = 000144	Q#QLR = 000001	T#OB = 000036
BYTE39 = 000047	BYTE90 = 000132	PLB = 000010	Q#QJ = 000042	T#OF = 010000
BYTE4 = 000004	BYTE91 = 000133	PLC = 000020	Q#RDCD = 000005	T#OBRA = 000034
BYTE40 = 000050	BYTE92 = 000134	PLD = 000030	Q#RDMD = 000006	T#OBWA = 000032
BYTE41 = 000051	BYTE93 = 000135	PLWR = 000020	Q#REBK = 001000	

T\$OUTA = 100000	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD82 = 000244
T\$RBDO = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD83 = 000246
T\$RNB = 000040	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD84 = 000250
T\$RSET = 040000	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD85 = 000252
T\$SC = 000022	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD86 = 000254
T\$SCLK = 020000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD87 = 000256
T\$SEG1 = 000000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD88 = 000260
T\$SEG2 = 000001	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD89 = 000262
T\$SEG3 = 000002	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD9 = 000022
T\$SD = 000001	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD90 = 000264
T\$SUBUS = 100000	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD91 = 000266
T\$1CLK = 000400	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD92 = 000270
T\$8BEN = 000020	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD93 = 000272
UBD IN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD94 = 000274
WORD0 = 000000	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD95 = 000276
WORD1 = 000002	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD96 = 000300
WORD10 = 000024	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD97 = 000302
WORD11 = 000026	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD98 = 000304
WORD12 = 000030	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORD99 = 000306
WORD13 = 000032	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WORDVAL = 000310
WORD14 = 000034	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRTM: 000066RG 002
WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240	XTREAD = 001000
WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242	XTWRITE = 000400
WORD17 = 000042	WORD39 = 000116	WORD60 = 000170		

. ABS. 000000 000
000000 001
MRPSUB 000506 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3021 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
MRPSUB.MRPSUB/SP=C20.1JIM,C20.1JMRPSUB

```

1
2 000000 .TITLE-CPSUB
3 .PSECT-CPSUB
4
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 000000 CPSUB:
19 000000 SEQCS:
20 ;
21 ;
22 ;
23 000000 005046 CLR - (SP) ;CLEAR-NOTHING
24 000002 012746 176000 MOV #0#NCLK,-(SP) ;SET-NO-CLOCKS
25 000005 CALL CSRI
26 ;
27 ;
28 ;
29 000012 005067 176422 CLR QR#CR2 ;DO-IT
30 ;
31 ;
32 ;
33 000016 005046 CLR -(SP) ;DEFINE-BR-REG-SELECT
34 000020 CALL CPCRA ;WRITE-CR-(WITHOUT-AUTOMATIC-BR-INHIBIT)
35 ;
36 ;
37 ;
38 000024 012746 030000 MOV #0#CNC,-(SP) ;CLEAR-CP-NO-CLOCKS
39 000030 012746 020000 MOV #0#CSP,-(SP) ;SET-CP-SINGLE-CLOCK
40 000034 CALL CSRI
41 ;
42 ;
43 ;
44 ;
45 000040 052766 020000 000002 BIS #BIT13,2(SP) ;DR-JMP-INSTRUCTION-(02) INTO-HIGH-ORDER-NIBBLE
46 000046 016646 000002 MOV 2(SP),-(SP) ;COPY-ARGUMENT-INTO-STACK-FOR-LBCP-CALL
47 000052 CALL LBCP ;DO-MOVE--ISSUE-CLOCK
48 ;
49 000056 011666 000002 MOV (SP),2(SP) ;MOVE-RETURN-ADDRESS-DOWN-STACK
50 000062 005726 TST (SP)+ ;POINT-TO-RETURN-ADDRESS
51 000064 RETURN

```



```

53      ;
54      ; CONTROL STORE SEQUENCING (CONTROL PROCESSOR)
55      ; SET START ADDRESS FOR WRITE
56      ;
57      ; INPUT:
58      ; 2(SP)  START ADDRESS
59      ;
60      ; OUTPUT:
61      ; CP SEQUENCER SET TO START ADDRESS
62      ;
63 000066      ; WRTCS:
64      ;
65      ; STOP CP AND MRP CLOCKS
66      ;
67 000066 005046      CLR  -(SP)          ; CLEAR NOTHING
68 000070 012746 176000  MOV  #0*INCLK,-(SP) ; SET NO CLOCKS
69 000074      CALL  CSR1
70      ;
71      ; GO TO LOAD MODE (AND CLEAR EVERYTHING ELSE)
72      ;
73 000100 005067 176422  CLR  QR#CR2          ; DO IT
74      ;
75      ; SELECT BRANCH REGISTER IN CP
76      ;
77 000104 005046      CLR  -(SP)          ; DEFINE BR REG SELECT
78 000106      CALL  CPCRA          ; WRITE CR (WITHOUT AUTOMATIC BR INHIBIT)
79      ;
80      ; CLOCK CP ONCE
81      ;
82 000112 012746 030000  MOV  #0*CNC,-(SP)   ; CLEAR CP NO CLOCKS
83 000116 012746 020000  MOV  #0*CSP,-(SP)   ; SET CP SINGLE CLOCK
84 000122      CALL  CSR1
85      ;
86      ; MOVE INSTRUCTION AND ADDRESS TO BRANCH REGISTER VIA LOAD BUS
87      ;
88 000126 052766 020000 000002  BIS  #BIT13,2(SP)   ; OR JMP INSTRUCTION (02) INTO HIGH ORDER NIBBLE
89 000134 016646 000002  MOV  2(SP),-(SP)    ; COPY ARGUMENT INTO STACK FOR LBCP CALL
90 000140      CALL  LBCP          ; DO MOVE -- FIRST CLOCK
91      ;
92 000144      RETURN

```

```

94          ;
95          ;      DATA TRANSFER.
96          ;      LOD BUS REGISTER TO A DESTINATION ON THE CP BUS.
97          ;
98          ;      INPUT:
99          ;      2(SP)  DATA FOR PRE-SELECTED CP DESTINATION.
100         ;
101         LBCP::
102         000146 016667 000002 176424  MOV.    2(SP),0R#LBR.      ; MOVE DATA TO LOD BUS REG.
103         000154 012746 001001        MOV.    #<0#LBD+0#LBP>,-(SP) ; CLR DRIVE AND PULSE
104         000160 052716 000360        BIS.    #<0#CSEL>,(SP)      ; CLR SELECTION BITS
105         000164 012746 176000        MOV.    #<0#NCLK>,-(SP)    ; SET NO-CLOCKS
106         000170 052716 000260        BIS.    #0#CP2,(SP)       ; SELECT CP
107         000174          ;
108         ;
109         000200 012746 030000        MOV.    #0#CNC,-(SP)      ; CLEAR CP NO-CLOCK BITS
110         000204 012746 021000        MOV.    #<0#CSP+0#LBD>,-(SP) ; SET CP CLOCK
111         000210          ;
112         ;
113         ;      DE-SELECTION
114         ;
115         000214 012746 001001        MOV.    #<0#LBD+0#LBP>,-(SP) ; CLEAR DRIVE AND PULSE
116         000220 052716 000360        BIS.    #<0#CSEL>,(SP)      ; CLR SELECTION BITS
117         000224 012746 176000        MOV.    #<0#NCLK>,-(SP)    ; SET NO-CLOCKS
118         000230          ;
119         ;
120         000234 011656 000002        MOV.    (SP),2(SP)        ; MOVE RETURN ADDRESS DOWN STACK
121         000240 005726          TST.   (SP)+              ; POINT TO RETURN ADDRESS
122         000242          RETURN
    
```

```

124      ;
125      ;
126      ; DATA TRANSFER
127      ; LOD-BUS REGISTER TO A DESTINATION ON THE CP-BUS
128      ; SINGLE-CLOCK SEQUENCER ONLY
129      ;
130      ; INPUT:
131      ; 2(SP) DATA FOR PRE-SELECTED CP DESTINATION
132      ;
133      ; LBCSC::
134      ; MOV. 2(SP),DR#LBR ;MOVE DATA TO LOD-BUS-REG
135      ; MOV. #<Q#LBD+Q#LBP>,-(SP) ;CLR DRIVE AND PULSE
136      ; BIS. #<Q#CSEL>,(SP) ;CLR SELECTION BITS
137      ; MOV. #<Q#NCLK>,-(SP) ;SET NO-CLOCKS
138      ; BIS. #Q#CP2,(SP) ;SELECT CP
139      ; CALL. CSR1
140      ;
141      ; MOV. #Q#CNC,-(SP) ;CLEAR CP NO-CLOCK BITS
142      ; MOV. #<Q#CSC+Q#LBD>,-(SP) ;SET CP CLOCK
143      ; CALL. CSR1
144      ;
145      ; DE-SELECTION
146      ; MOV. #<Q#LBD+Q#LBP>,-(SP) ;CLR DRIVE AND PULSE
147      ; BIS. #<Q#CSEL>,(SP) ;CLR SELECTION BITS
148      ; MOV. #<Q#NCLK>,-(SP) ;SET NO-CLOCKS
149      ; CALL. CSR1
150      ;
151      ; MOV. (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
152      ; TST. (SP)+ ;POINT TO RETURN ADDRESS
153      ; RETURN

```

```

155 ;
156 ; DATA TRANSFER TO LOD BUS REG FROM CP
157 ;
158 ; OUTPUT:
159 ; (SP) DATA FROM PRE-SELECTED CP SOURCE
160 ;
161 000342 CPLB::
162 000342 012746 001001 MOV *(<Q#LBD+Q#LBP>,-(SP) ;CLR DRIVE AND PULSE
163 000346 052716 000360 BIS *(<Q#CSEL>,(SP) ;CLR SELECTION BITS
164 000352 012746 176000 MOV *(<Q#NCLK>,-(SP) ;SET NO-CLOCKS
165 000356 052716 000260 BIS #Q#CP2,(SP) ;SOURCE IS CP
166 000362 CALL CSR1
167 000366 011646 MOV (SP),-(SP) ;MOVE RETURN ADDR UP STACK
168 000370 016766 176424 000002 MOV QR#LBR,2(SP) ;MOVE DATA ONTO STACK
169 ;
170 000376 012746 000260 MOV #Q#CP2,-(SP) ;CLEAR CP SELECT
171 000402 005046 CLR -(SP) ;SET NOTHING
172 000404 CALL CSR1
173 000410 RETURN

```

```

175      ;
176      ;
177      ;      CP CONTROL REGISTER LOADING
178      ;
179      ;      INPUT:
180      ;      2(SP)  BIT SETTING FOR CP CONTROL REGISTER
181      ;
182 000412  ;      CPCRA::
183 000412 052766 100000 000002  ;      BIS      #CNOREG,2(SP)      ;AUTOMATIC INHIBIT OF BRANCH REGISTER
184 000420  ;
185 000420 016667 000002 176424  ;      MOV      2(SP),QR$LBR      ;CONTROL BITS DESTINED FOR CP
186 000426 012746 001001  ;      MOV      #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
187 000432 052716 000360  ;      BIS      #<Q$CSEL>,(SP)      ;CLR SELECTION BITS
188 000436 012746 000060  ;      MOV      #Q$CP,-(SP)      ;SELECT CP
189 000442  ;      CALL     CSR1
190      ;
191 000446 005046  ;      CLR      -(SP)      ;CLEAR NOTHING
192 000450 012746 000001  ;      MOV      #Q$LBP,-(SP)      ;SET PULSE
193 000454  ;      CALL     CSR1
194      ;
195 000460 012746 000061  ;      MOV      #<Q$CP+Q$LBP>,-(SP) ;CLEAR CR SELECTION AND PULSE
196 000464 005046  ;      CLR      -(SP)      ;SET NOTHING
197 000466  ;      CALL     CSR1
198      ;
199 000472 011666 000002  ;      MOV      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
200 000476 005726  ;      TST      (SP)+      ;POINT TO RETURN ADDRESS
201 000500  ;      RETURN
202      ;
203      000001  ;      .END

```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
PLY4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CNOBRE = 100000
CPCCEN = 010000
CPCR = 000412RG
CPCRA = 000420RG
CPCLB = 000342RG
CPREAD = 040000
CPSUB = 000000R
CPWRTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSURTE = 000100
DBR_RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DMAAUR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
LBCCP = 000146RG
LBCCSC = 000244RG
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 000200
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADRD = 000100
MMLEFT = 000002
MMOE = 000004
MMURTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGO.EN = 000200
RGO.VA = 020000
SEQCS = 000000RG
SEQ.CI = 000010
S\$CLR = 000000
S\$LA = 000001
S\$QB = 000005
S\$OR = 000006
S\$QX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$OTR = 176346
TD\$ORD = 000274
TD\$SW = 176376
TD\$STAR = 176372
TD\$TAU = 176362
TD\$TDR = 176374
TD\$TDW = 176364
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTB = 000000
T\$CD = 000100
T\$CLK = 002000
T\$ICD = 000040
T\$DRD = 000004
T\$MEM = 010000
T\$FSAA = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBF = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036
T\$OBE = 004000
T\$OBF = 010000
T\$OBRA = 000034
T\$OBWA = 000032

SYMBOL TABLE

T\$GUTA = 100000	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD82 = 000244
T\$RBD0 = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD83 = 000246
T\$RNB = 000040	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD84 = 000250
T\$RSET = 040000	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD85 = 000252
T\$SC = 000022	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD86 = 000254
T\$SCLK = 020000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD87 = 000256
T\$SEG1 = 000000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD88 = 000258
T\$SEG2 = 000001	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD89 = 000262
T\$SEG3 = 000002	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD9 = 000022
T\$SO = 000001	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD90 = 000264
T\$UBUS = 100000	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD91 = 000266
T\$1CLK = 000400	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD92 = 000270
T\$BBEN = 000020	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD93 = 000272
UBD.IN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD94 = 000274
WORD0 = 000000	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD95 = 000276
WORD1 = 000002	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD96 = 000300
WORD10 = 000024	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD97 = 000302
WORD11 = 000026	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD98 = 000304
WORD12 = 000030	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORD99 = 000306
WORD13 = 000032	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WORDVAL = 000310
WORD14 = 000034	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRTCS = 000066RG
WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD00 = 000240	XTREAD = 001000
WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD01 = 000242	XTWRTE = 000400
WORD17 = 000042	WORD39 = 000116	WORD60 = 000170		

. ABS. 000000 000
000000 001
CPSUB: 000502 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3021 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
CPSUB, CPSUB, SP=C 20, 1 JIM, C 20, 1 CPSUB

```

1
2 000000 .TITLE PPSUB
3
4 :
5 .MCALL WTSE$,CLEF$S
6 000003 EFN.3 = 3
7
8
9
10 :
11 :
12 :
13 :
14 :
15 :
16 :
17 :
18 :
19 :
20 000000 LBPP::
21 000000 016667 000002 176424 MOV 2(SP),QR$LBR ;MOVE DATA TO LOD BUS REG
22 000006 012746 001001 MOV *(<Q$LBD+Q$LBP>,-(SP) ;CLR DRIVE AND PULSE
23 000012 052716 000360 BIS *(<Q$CSEL>,(SP) ;CLR SELECTION BITS
24 000016 012746 176000 MOV *(<Q$NCLK>,-(SP) ;SET NO-CLOCKS
25 000022 052716 000300 BIS *Q$PP2,(SP) ;SELECT PP
26 000026
27 :
28 000032 012746 006000 MOV #Q$RNC,-(SP) ;CLEAR CP NO-CLOCK BITS
29 000036 012746 001000 MOV #Q$LBD,-(SP) ;SET LOD BUS DRIVE
30 000042
31 :
32 :
33 :
34 000046 012746 001001 MOV *(<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
35 000052 052716 000360 BIS *(<Q$CSEL>,(SP) ;CLR SELECTION BITS
36 000056 012746 176000 MOV *(<Q$NCLK>,-(SP) ;SET NO-CLOCKS
37 000062
38 :
39 000066 011666 000002 MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
40 000072 005726 TST (SP)+ ;POINT TO RETURN ADDRESS
41 000074 RETURN

```



```

43 ;
44 ;
45 ; DATA TRANSFER
46 ; LOD BUS REGISTER TO A DESTINATION ON THE PPS BUS
47 ; SINGLE CLOCK SEQUENCER ONLY
48 ;
49 ; INPUT:
50 ; 2(SP) DATA FOR PRE-SELECTED PPS DESTINATION
51 ;
52 ;
53 000076 ; LBPSC::
54 000076 016667 000002 176424 MOV 2(SP),Q#LBR ; MOVE DATA TO LOD BUS REG
55 000104 012746 001001 MOV #<Q#LBD+Q#LBP>,-(SP) ; CLEAR DRIVE AND PULSE
56 000110 052716 000360 BIS #Q#CSEL,(SP) ; CLR SELECTION BITS
57 000114 012746 176000 MOV #<Q#NCLK>,-(SP) ; SET NO-CLOCKS
58 000120 052716 000300 BIS #Q#PP2,(SP) ; SELECT PP
59 000124 CALL CSR1 ; WRITE CONTROL REGISTER
60 ;
61 000130 012746 006000 MOV #Q#RNC,-(SP) ; CLEAR PP NO-CLOCK BITS
62 000134 012746 005000 MOV #<Q#RSC+Q#LBD>,-(SP) ; SET PP CLOCK
63 000140 CALL CSR1 ;
64 ;
65 ; DE-SELECTION
66 ;
67 000144 012746 001001 MOV #<Q#LBD+Q#LBP>,-(SP) ; CLEAR DRIVE AND PULSE
68 000150 052716 000360 BIS #Q#CSEL,(SP) ; CLR SELECTION BITS
69 000154 012746 176000 MOV #<Q#NCLK>,-(SP) ; SET NO-CLOCKS
70 000160 CALL CSR1 ;
71 ;
72 000164 011666 000002 MOV (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
73 000170 005726 TST (SP)+ ; POINT TO RETURN ADDRESS
74 000172 RETURN

```

```

76      ;
77      ;
78      ;      DATA TRANSFER TO LOD BUS REG FROM PPS
79      ;
80      ;      OUTPUT:
81      ;      (SP) DATA FROM PRE-SELECTED PPS SOURCE
82      ;
83      ;
84 000174      PPLB::
85 000174 012746 001001      MOV      #<0$LBD+0$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
86 000200 052716 000360      BIS      #0$CSEL,(SP)      ;CLR SELECTION BITS
87 000204 012746 176000      MOV      #<0$NCLK>,-(SP)      ;SET NO-CLOCKS
88 000210 052716 000300      BIS      #0$PP2,(SP)      ;SOURCE IS PP
89 000214      CALL      CSR1      ;
90 000220 011646      MOV      (SP),-(SP)      ;MOVE RETURN ADDR UP STACK
91 000222 016766 176424 000002      MOV      @R$LBR,2(SP)      ;MOVE DATA ONTO STACK
92      ;
93 000230 012746 000300      MOV      #0$PP2,-(SP)      ;CLEAR PP SELECT
94 000234 005046      CLR      -(SP)      ;SET NOTHING
95 000236      CALL      CSR1      ;
96 000242      RETURN

```

```

98      :
99      :
100     :      PPS CONTROL REGISTER LOADING
101     :
102     :      INPUT:
103     :      2(SP) BIT SETTING FOR PPS CONTROL REGISTER
104     :
105     :
106     000244      :      PPCR::
107     000244 016657 000002 176424      MOV      2(SP),QR#LBR      ;CONTROL BITS DESTINED FOR PPS
108     000252 012746 001001      MOV      #<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
109     000256 052716 000360      BIS      #Q#CSEL,(SP)      ;CLR SELECTION BITS
110     000262 012746 000100      MOV      #Q#PP,-(SP)      ;SELECT PP
111     000266      CALL     CSR1              ;
112     :
113     000272 005046      CLR      -(SP)            ;CLEAR NOTHING
114     000274 012746 000001      MOV      #Q#LBP,-(SP)     ;SET PULSE
115     000300      CALL     CSR1              ;
116     :
117     000304 012746 000101      MOV      #<Q#PP+Q#LBP>,-(SP) ;CLEAR CR SELECTION AND PULSE
118     000310 005046      CLR      -(SP)            ;SET NOTHING
119     000312      CALL     CSR1              ;
120     :
121     000316 011666 000002      MOV      (SP),2(SP)       ;MOVE RETURN ADDRESS DOWN STACK
122     000322 005726      TST     (SP)+             ;POINT TO RETURN ADDRESS
123     000324      RETURN

```

```

125 ;
126 ;
127 ;
128 ;
129 ;
130 ;
131 ;
132 ;
133 ;
134 ;
135 ;
136 000326 SELPG::
137 000326 005046 CLR - (SP) ;START-MICROCODE-AT-0
138 000330 CALL SE0CS ;
139 000334 005046 CLR - (SP) ;RESET-BR-INHIBIT
140 000336 CALL CPCR ;
141 000342 012746 000377 MOV #377, -(SP) ;SET-MRP-MICRO-ADDRESS-#X'FF' (JUMP-SELF)
142 000346 CALL SEQMM ;
143 000352 005046 CLR - (SP) ;RESET-BR-INHIBIT
144 000354 CALL MRPCR ;
145 000360 012767 001000 176422 MOV #Q$REBK,QR$CR2 ;RE-ARM-INTERRUPTS
146 000366 012767 120000 176422 MOV #<Q$SM+Q$ENOP>,QR$CR2 ;SET-SEARCH-MODE-+ENABLE-INTERRUPTS
147 000374 012746 000360 MOV #Q$CSEL, -(SP) ;CLEAR-ALL-SELECTIONS
148 000400 052716 001001 BIS #<Q$LBD+Q$LBP>, (SP) ;CLEAR-DRIVE-AND-PULSE
149 000404 052716 030000 BIS #Q$CNC, (SP) ;CLEAR-CP-NO-CLOCK
150 000410 005046 CLR - (SP) ;SET-NOTHING
151 000412 CALL CSR1 ;
152 ;
153 000416 012767 000003 176424 MOV #Q$LCD,QR$LBR ;MOVE-ATTN-CODE-TO-LOD-BUS-REG
154 000424 012767 120100 176422 MOV #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET-ATTN-CODE-READY
155 000432 016701 176422 15: QR$CR2,R1 ;READ-CSR2
156 000436 032701 000100 BIT #Q$ATTN,R1 ;ATTN-CLEAR
157 000442 001373 BNE 1# ;NO-READ-AGAIN
158 ;
159 000444 012767 000041 176424 MOV #041,QR$LBR ;CD-MEMORY-START-ADDRESS-#X'21'
160 000452 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
161 000460 016701 176422 25: MOV QR$CR2,R1 ;READ-CSR2
162 000464 032701 000040 BIT #Q$CCCP,R1 ;IS-CC-TO-CP-CLEAR
163 000470 001373 BNE 2# ;NO-READ-AGAIN
164 ;
165 000472 012767 000001 176424 MOV #1,QR$LBR ;TRANSFER-COUNT-=-1-WORD
166 000500 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
167 000506 016701 176422 35: MOV QR$CR2,R1 ;READ-CSR2
168 000512 032701 000040 BIT #Q$CCCP,R1 ;IS-CC-TO-CP-CLEAR
169 000516 001373 BNE 3# ;NO-READ-AGAIN
170 ;
171 000520 016667 000002 000000G MOV 2(SP),DATA1 ;PUT-PAGE-VALUE-INTO-BUFFER
172 000526 005367 000000G DEC DATA1 ;MICROCODE-REINCREMENTS
173 000532 012767 000000G 176424 MOV #DATA1,QR$LBR ;CC-MEMORY-DATA-BUFFER
174 000540 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET-CC-TO-CP
175 ;
176 ;
177 ;
178 000546 ;
179 ;
180 000560 ;
181 ;

```

```

182:                                     : RE-ARM INTERRUPTS
183:                                     :
184 000572 012767 100400 176422 MOV.    *(<Q$SM+Q$CHB>,QR$CR2) :CLEAR INTERRUPT (USE HIT BUFFER INT)
185 000600 012767 101000 176422 MOV.    *(<Q$SM+Q$REBK>,QR$CR2) :RE-ARM
186 000606 012767 160000 176422 MOV.    *(<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2) :ENABLE
187:                                     :
188 000614 005046 CLR.    -(SP) :CLEAR NOTHING IN CSR1
189 000616 012746 176000 MOV.    #Q$NCLK, -(SP) :SET NO-CLOCKS
190 000622 CALL   CSR1
191 000626 005067 176422 CLR.    QR$CR2 :SET LOAD MODE
192:                                     :
193 000632 012746 001761 MOV.    #1761, -(SP) :START CP AT X '3F1'
194 000636 CALL   SEQCS
195 000642 005046 CLR.    -(SP) :RESET BR INHIBIT
196 000644 CALL   CPCR
197 000650 012746 000377 MOV.    #377, -(SP) :SET MRP MICRO ADDRESS = X'FF' (JUMP SELE)
198 000654 CALL   SEQMM
199 000660 005046 CLR.    -(SP) :RESET BR INHIBIT
200 000662 CALL   MRPCR
201 000666 012767 001000 176422 MOV.    #Q$REBK,QR$CR2 :RE-ARM INTERRUPTS
202 000674 012767 120000 176422 MOV.    *(<Q$SM+Q$ENOP>,QR$CR2) :SET SEARCH MODE + ENABLE INTERRUPTS
203 000702 012746 000360 MOV.    #Q$CSEL, -(SP) :CLEAR ALL SELECTIONS
204 000706 052716 001001 BIS.    *(<Q$LBD+Q$LBP>, (SP) :CLEAR DRIVE AND PULSE
205 000712 052716 036000 BIS.    *(<Q$RNC+Q$CNC>, (SP) :CLEAR CP AND PPS NO-CLOCKS
206 000716 005046 CLR.    -(SP) :SET NOTHING
207 000720 CALL   CSR1
208:                                     :
209:                                     :
210:                                     :
211 000724 012701 000144 MOV.    #100, R1 :LOOP 100 TIMES
212 000730 005002 CLR.    R2 :ADD NOTHING
213 000732 060202 4$: ADD.   R2, R2
214 000734 005301 DEC.    R1
215 000736 001375 BNE.   4$
216:                                     :
217 000740 005046 CLR.    -(SP) :CLEAR NOTHING IN CSR1
218 000742 012746 176000 MOV.    #Q$NCLK, -(SP) :SET NO-CLOCKS
219 000746 CALL   CSR1
220 000752 005067 176422 CLR.    QR$CR2 :SET LOAD MODE
221:                                     :
222 000756 011666 000002 MOV.    (SP), 2(SP) :MOVE RETURN ADDRESS DOWN STACK
223 000762 005726 TST.   (SP)+ :POINT TO RETURN ADDRESS
224 000764 RETURN
225:                                     :
226 000001 .END
    
```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLR,EN = 000200	Q\$RDMD = 000006
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PPCR = 000244RG	002.Q\$REBK = 001000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PPLB = 000174RG	002.Q\$RNC = 006000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$SCR1 = 176420	Q\$RSC = 004000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$SCR2 = 176422	Q\$RSET = 000010
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$SLBR = 176424	Q\$SM = 100000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$ATTN = 000100	Q\$SP = 000120
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CCCP = 000040	RG0,EN = 000200
BIT13 = 020000	BYTE50 = 000062	CNDBRE = 100000	Q\$CHB = 000400	RG0,VA = 020000
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	Q\$CHRL = 000200	SELPG = 000326RG
BIT15 = 100000	BYTE52 = 000064	CPCR = ***** GX	Q\$CLR = 000040	SEQCS = ***** GX
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	Q\$CNC = 030000	SEGM = ***** GX
BIT3 = 000010	BYTE54 = 000066	CPWRTE = 020000	Q\$CP = 000060	SEQ,CI = 000010
BIT4 = 000020	BYTE55 = 000067	CSADRD = 000004	Q\$CPC = 000010	S\$CLR = 000000
BIT5 = 000040	BYTE56 = 000070	CSEQCI = 100000	Q\$CP2 = 000260	S\$LA = 000001
BIT6 = 000100	BYTE57 = 000071	CSOE = 000040	Q\$CSC = 010000	S\$OB = 000005
BIT7 = 000200	BYTE58 = 000072	CSR1 = ***** GX	Q\$CSEL = 000360	S\$OR = 000006
BIT8 = 000400	BYTE59 = 000073	CSWRTE = 000100	Q\$CSET = 000002	S\$OX = 000004
BIT9 = 001000	BYTE6 = 000006	DATA1 = ***** GX	Q\$CSP = 020000	S\$SR = 000007
BYTE0 = 000000	BYTE60 = 000074	DB,RD = 000001	Q\$DMA = 000001	S\$S1 = 000010
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$ENBK = 040000	S\$S2 = 000014
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$ENOP = 020000	TD\$CTR = 176370
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$FAL = 004000	TD\$CTW = 176360
BYTE12 = 000014	BYTE64 = 000100	DISPGS = 100000	Q\$FC = 000045	TD\$INL = 004000
BYTE13 = 000015	BYTE65 = 000101	DMANUR = 000005	Q\$FO = 000044	TD\$INEM = 000270
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$FP = 000046	TD\$OAR = 176344
BYTE15 = 000017	BYTE67 = 000103	DMARUR = 000004	Q\$HBF = 000002	TD\$OTR = 176346
BYTE16 = 000020	BYTE68 = 000104	EFN,3 = 000003	Q\$ICP = 000006	TD\$ORD = 000274
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$IHB = 000003	TD\$SUW = 176376
BYTE18 = 000022	BYTE7 = 000007	LBPB = 000000RG	002.Q\$IHRL = 000002	TD\$STAR = 176372
BYTE19 = 000023	BYTE70 = 000106	LBPSC = 000075RG	002.Q\$IMRP = 000007	TD\$TAJW = 176362
BYTE2 = 000002	BYTE71 = 000107	LOC,EN = 000100	Q\$LBD = 001000	TD\$TAR = 176374
BYTE20 = 000024	BYTE72 = 000110	LOC,UA = 040000	Q\$LBDP = 001001	TD\$TDJW = 176364
BYTE21 = 000025	BYTE73 = 000111	LOC,UB = 000000	Q\$LBP = 000001	T\$AD = 000020
BYTE22 = 000026	BYTE74 = 000112	MAREN1 = 000001	Q\$LCD = 000003	T\$BA = 000002
BYTE23 = 000027	BYTE75 = 000113	MAREN2 = 004000	Q\$LDM = 000004	T\$BD = 000010
BYTE24 = 000030	BYTE76 = 000114	MARLOD = 010000	Q\$LDPP = 002000	T\$BSO = 100000
BYTE25 = 000031	BYTE77 = 000115	MAROUT = 000002	Q\$LHP = 010000	T\$BT = 000020
BYTE26 = 000032	BYTE78 = 000116	MAR,LO = 002000	Q\$MNC = 140000	T\$BTAR = 000030
BYTE27 = 000033	BYTE79 = 000117	MAR,OU = 000040	Q\$MR = 000052	T\$BTD = 002000
BYTE28 = 000034	BYTE8 = 000008	MBKALL = 001000	Q\$MRP = 000040	T\$CD = 000100
BYTE29 = 000035	BYTE80 = 000120	MBKCLK = 000400	Q\$MRP2 = 000240	T\$CLK = 002000
BYTE3 = 000003	BYTE81 = 000121	MNADR1 = 000100	Q\$MSC = 040000	T\$DISK = 000200
BYTE30 = 000036	BYTE82 = 000122	MNLEFT = 000002	Q\$MSET = 000004	T\$DRD = 000004
BYTE31 = 000037	BYTE83 = 000123	MNDE = 000004	Q\$MSP = 100000	T\$MEM = 010000
BYTE32 = 000040	BYTE84 = 000124	MNWRTE = 000010	Q\$NCLK = 176000	T\$FSA = 000000
BYTE33 = 000041	BYTE85 = 000125	MNDBRE = 100000	Q\$PP = 000100	T\$FSAB = 000004
BYTE34 = 000042	BYTE86 = 000126	MREN1 = 000001	Q\$PPSW = 000320	T\$FSAC = 000014
BYTE35 = 000043	BYTE87 = 000127	MREN2 = 020000	Q\$PP2 = 000300	T\$FSB2 = 000010
BYTE36 = 000044	BYTE88 = 000130	MRPCR = ***** GX	Q\$OHLT = 000013	T\$IB = 000026
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q\$OL = 000043	T\$IBAR = 000024
BYTE38 = 000046	BYTE9 = 000009	N = 000144	Q\$OLA = 000053	T\$IBE = 020000
BYTE39 = 000047	BYTE90 = 000132	PLB = 000010	Q\$OLB = 000054	T\$IDF = 000000
BYTE4 = 000004	BYTE91 = 000133	PLC = 000020	Q\$OLR = 000001	T\$ICD = 000040
BYTE40 = 000050	BYTE92 = 000134	PLD = 000030	Q\$OW = 000042	T\$MODE = 004000
BYTE41 = 000051	BYTE93 = 000135	PLWR = 000200	Q\$RDMD = 000005	T\$OB = 000036

T\$0BE = 004000	WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240
T\$0BF = 010000	WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242
T\$0BRA = 000034	WORD17 = 000042	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244
T\$0BWA = 000032	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246
T\$0UTA = 100000	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T\$RBD0 = 000200	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T\$RNB = 000040	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T\$RSET = 040000	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T\$SC = 000022	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T\$SCLK = 020000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T\$SEG1 = 000000	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T\$SEG2 = 000001	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T\$SEG3 = 000002	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T\$SO = 000001	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T\$SUBUS = 100000	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T\$1CLK = 000400	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T\$0BEN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
UBD: IN = 000020	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD0 = 000000	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD1 = 000002	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD10 = 000024	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD11 = 000026	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WRDVAL = 000310
WORD12 = 000030	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	XTREAD = 001000
WORD13 = 000032	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	XTWRTE = 000400
WORD14 = 000034				

. ABS. 000000 000
000000 001
PPSUB 000766 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3288 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:46
PPSUB, PPSUB, SP=C 20.1 JIM, C 20.1 JPPSUB

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

```

;
; MAXIMUM QLS SUB-BUFFER SIZES
;
QRYMSZ==N.QURY.
FALMSZ==160.*2.
TTBMSZ==430.*4
XTBMSZ==516.*2.
QLBMSZ==334.
SLBMSZ==236.
QEXMSZ==192.*2.
;
; QUERY RESOLVER BIT DEFINITIONS
;
B#SUC==BIT7 ;SUCCESS BIT.
B#NOT==BIT6 ;QLB NOT ENTRY
B#MUL==BITS ;QLB MULTI-NOT ENTRY.
B#FMN==BIT4 ;QLB FIRST MULTI-NOT ENTRY.
;
B#FST==BIT13 ;QEX FIRST PROX. ENTRY.
B#PUFH==BIT15 ;QEX FORWARD PROX. UNIT - HIGH ORDER BIT.
B#PUFL==BIT14 ;QEX FORWARD PROX. UNIT - LOW ORDER BIT.
B#PUBH==BIT12 ;QEX BACKWARD PROX. UNIT - HIGH ORDER BIT.
B#PUBL==BIT11 ;QEX BACKWARD PROX. UNIT - LOW ORDER BIT.
;
; QLS BUFFER OFFSETS (CONVERTED TO ADDRESSES)
;
QLSXX1=0 ;OFFSET OF COMM. HEADER.
QLSX17=QLSXX1+4 ;BATCH NUMBER.
QLSXX2=QLSXX1+2. ;BYTE SIZE OF QLB. (USED)
QLSXX3=QLSXX2+2. ;BYTE SIZE OF SDLB. (USED)
QLSXX4=QLSXX3+2. ;BYTE SIZE OF QEX. (USED)
QLSXX5=QLSXX4+2. ;MAXIMUM QUERY ID.
QLSXX6=QLSXX5+2. ;BYTE SIZE OF FAL. (USED)
QLSXX7=QLSXX6+2. ;BYTE SIZE OF TTABLE. (USED)
QLSXX8=QLSXX7+2. ;BYTE SIZE OF XTABLE. (USED)
QLSX10=QLSXX8+2. ;START OF QCL SUB-BUFFER.
QLSX11=QLSX10+(QRYMSZ*2); START OF FAL SUB-BUFFER.
QLSX12=QLSX11+FALMSZ ;START OF TTABLE SUB-BUFFER.
QLSX13=QLSX12+TTBMSZ ;START OF XTABLE SUB-BUFFER.
QLSX14=QLSX13+XTBMSZ ;START OF QLB SUB-BUFFER.
QLSX15=QLSX14+QLBMSZ ;START OF SDLB SUB-BUFFER.
QLSX16=QLSX15+SLBMSZ ;START OF QEX SUB-BUFFER.
QLSZ==QLSX16+QEXMSZ ;SIZE OF QLS IN BYTES.

```



```

1      ; .TITLE LMAIN
2      ;
3      ;
4      ; SOFTWARE QLS CONVERTER AND HQR LOADER
5      ;
6      ; LOAD MICROCODE INTO HQR (MRP AND CP)
7      ; SET MASK WORD, CLEAR QLB PAGES, FAL COUNTER MEMORY
8      ; READ SOFTWARE QLS
9      ; CONVERT SOFTWARE QLB INTO HQR QLB REFERENCE PAGE
10     ; CONVERT SOFTWARE SDLB INTO HQR SDLB REFERENCE PAGE
11     ; CONVERT QEX
12     ; LOAD CP DATA MEMORY
13     ; CONVERT TTABLE AND XTABLE
14     ; CONVERT QCL AND FAL
15     ;
16     ;
17     ; SUB-MODULES OF LMAIN
18     ; MICRO LOAD MICROCODE
19     ; FRAME2 CLEAR FRAME 2 MEMORIES
20     ; QLBLD CONVERT QLB
21     ; SDLBLD CONVERT SDLB
22     ; QEXLD CONVERT QEX
23     ; CD LOAD CP DATA MEMORY
24     ; TTXT CONVERT TTABLE AND XTABLE
25     ; QCLFAL CONVERT QCL AND FAL
26     ;
27     ; ALL CALLS TO SUB-MODULES ARE MADE VIA JSR, SOME OF THE
28     ; SUB-MODULES SET THE CONDITION CODE UPON EXIT
29     ;
30     ; LMAIN ALSO CONTAINS DATA AREAS AND SUBROUTINES
31     ;
32     ;
33     ; ASSEMBLY
34     ; MCR>MAC LMAIN,LP=C20,1JP,C,Q,IM,LMAIN,OG,ENDST
35     ;
36     ; TASK BUILD:
37     ; HQRQLS,HQRQLS=HQRQLS/MP
38     ; /
39     ; TASK=HQRQLS
40     ; PAR=GEN:40000:40000
41     ; ASG=TT0:1
42     ; COMMON=CCOM:RW
43     ; RESLIB=C 1,1JFCSRES/RO
44     ; /
45     ; //
46     ;
47     ; OVERLAY DESCRIPTION HQRQLS,ODL:
48     ; .ROOT LMAIN-MRPSUB-CPSUB-PPSUB-C 1,50JRSX11M,STB/SS-*(A)
49     ; .FCTR (MICRO,FRAME2,QLBLD,QEXLD,CD,TTXT,QCLFAL)
50     ; .END

```

```

52.      ;
53.      ;
54.      ; LOCAL DATA AREAS
55.      ;
56.      ;
57.      ;
58.      ; .MCALL QIDW$, FINIT$, EXIT$, ABRT$, FSRSZ$, CLEF$
59.      ; .MCALL FDBDF$, FDRC$, FDBK$, FDP$, NMBLK$, OFNB$
60.      ; .MCALL CLOSE$, READ$, WTSE$, SETF$, WAIT$, OPEN$
61.      ; .MCALL RDAF$, RQST$C
62.      ;
63.      ; LUN:TT  == 1 ; LUN FOR TT0
64.      ; EFN:1  == 1 ; EVENT FLAG FOR TT0
65.      ; EFN:3  == 3 ; EVENT FLAG FOR INTERRUPTS FROM HQR
66.      ; INLUN: == 3 ; LUN FOR HQR FILES
67.      ; QLSLUN: == 4 ; LUN FOR EMATRIX, QLS
68.      ; FIRST: == 1 ; FIRST READ OF AN HQR DATA FILE
69.      ; LAST:  == 2 ; TRANSFER OF LAST BLOCK OF CP DATA MEMORY
70.      ;
71.      ; LMN:    == BIT0 ; LOCAL LAST MULTI-NOT FLAG
72.      ; B$DEL: == BIT0 ; SOFTWARE FLAG - QLB HAS SUBELEMENT
73.      ; B$EOTR: == BIT1 ; SOFTWARE FLAG - LAST ELEMENT OF TOP OR
74.      ; B$EOD:  == BIT2 ; SOFTWARE FLAG - LAST ELEMENT OF QUERY
75.      ; HLMN:  == BIT1 ; HQR LAST MULTI-NOT (BIT14 = 0, BIT1 = 1)
76.      ; HMN:   == BIT0 ; HQR MULTI-NOT
77.      ; HLNE:  == BIT14+BIT1 ; HQR LAST NON-EOD WORD (BIT14, BIT1 = 1)
78.      ; HEOD:  == BIT15 ; HQR END-OF-QUERY
79.      ; SHRL:  == BIT15 ; SOFTWARE OCL HRL MARK
80.      ; SXTDEX: == BIT15 ; SOFTWARE XLIST OEX MARK
81.      ; SXTSLB: == BIT14 ; SOFTWARE XLIST SDLB MARK
82.      ; HXTSLB: == BIT15 ; HARDWARE XLIST SDLB MARK
83.      ; QXCNTL: == BIT15 ; QEX POINTER MARK IN SOFT XLIST
84.      ; HRLPNT: == BIT13 ; MARK FOR HRL POINTER IN HQR FAL
85.      ; LDCT:   == 140000 ; AMD2900 'LDCT' INSTR. = X'0C' IN TOP 4 BITS
86.      ; POLB:  == BIT5 ; QLB BLOCK GENERATED FLAG
87.      ; SDMAP: == BIT6 ; HQRMAP TYPE FLAG
88.      ; SHMN:  == BIT3 ; LOCAL FLAG - EOR GROUP HAS SUBELEMENT
89.      ; SHLN:  == BIT11 ; HQR SDLB MULTI-NOT
90.      ; SHLMN: == BIT12 ; HQR SDLB LAST MULTI-NOT
91.      ; SHLNE: == BIT13 ; HQR SDLB END TOP OR
92.      ; SHEOD: == BIT15+BIT14 ; HQR SDLB END OF SUBQUERY
93.      ;
94.      ; ALLHRL == 26 ; X'16' ADDR IN CD MEMORY OF ALLHRL FLAG
95.      ; NUHLCD == 724 ; X'1D4' ADDR OF NO-UHL DUMMY HRL IN CD MEM
96.      ; OCLBUF == 115 ; X'4D' ADDR OF OCL IN CD MEM
97.      ; QXLOW: == 76000 ; X'7C00' LOW ADDRESS OF QEX MEMORY
98.      ; SIDSIZ: == 4096 ; NUMBER OF WORDS IN SIDMEM
99.      ;
100.     ; .NLIST BEX
101.     MYSELF: .RAD50 /HQRQLS/
102.     TSKTCB: .WORD 0
103.     O_DVEC: .WORD 0
104.     E$BUF:  .BLKW 4
105.     L$M:    .RAD50 /LDM$/ ; MRP MICROPROGRAM MEMORY LOADABLE FILE
106.     L$S:    .RAD50 /LDCS/ ; CP CONTROL STORE LOADABLE FILE
107.     L$D:    .RAD50 /LDCD/ ; CP DATA MEMORY LOADABLE FILE
108.     E$X:    .RAD50 /EMATRIX/ ; *** TEMPORARY ***
109.     Q_SR50: .RAD50 /QLS/ ; *** TEMPORARY ***

```

```

109
110 000044 000000      VIRT:: .WORD. 0      ;VIRTUAL-BLOCK-COUNTER.
111 000046 000001      .WORD. 1
112 000050      STAT:: .BLKW. 2
113 000054 000000      ERWORD:: .WORD. 0      ;INDEX-VALUE-FOR-ERROR-MESSAGE-TABLE.
114 000056 000000      SELECT:: .WORD. 0      ;ALL-PURPOSE-FLAG.
115 000060 000000      APLACE:: .WORD. 0      ;PLACE-TO-MESS-WITH-CSR1
116 000062 000000      LCOUNT:: .WORD. 0      ;NUMBER-OF-WORDS-TO-LOAD.
117 000064 000000      WCOUNT:: .WORD. 0      ;WORKING-COUNTER.
118 000066 000000      CODE:: .WORD. 0      ;MEMORY-SELECT-CODE.
119 000070
120 000070 000000      ADDR:: .WORD. 0      ;ADDRESS-FOR-MEMORY-WRITE.
121 000072      DATA1::
122 000072 000000      DATA:: .WORD. 0      ;DATA-FOR-MEMORY-WRITE.
123 000074 000000      QXCNT:: .WORD. 0      ;NUMBER-OF-QEX-ADDR/DATA-PAIRS-IN-CD-MEMORY.
124 000076 076000      QXADD:: .WORD. 76000      ;START-OF-QEX-IN-HQR = X'7C00'
125 000100 000000      TRANSF:: .WORD. 0      ;DMA-TRANSFER-COUNT.
126 000102 000000      INSAVE:: .WORD. 0      ;DMA-POINTER.
127 000104
128
129 010104 006212*      PSQLB:: .WORD. QLB      ;POINTER-TO-SOFTWARE-QLB-(SQLB)
130 010106 000000      NPSQLB:: .WORD. 0      ;POINTER-TO-NEXT-SQLB-ENTRY.
131 010110 000000      QENTL:: .WORD. 0      ;NUMBER-OF-BYTES-IN-CURRENT-SQLB-ENTRY.
132 010112 000000      WQENTL:: .WORD. 0      ;LOOP-COUNTER-BASED-ON-QENTL.
133 010114 000000      XQID:: .WORD. 0      ;EXTERNAL-QID-(FROM-SQLB-ENTRY)
134 010116 000000      LXQID:: .WORD. 0      ;XQID-FROM-PREVIOUS-SQLB-ENTRY
135 010120 000000      IQID:: .WORD. 0      ;INTERNAL-(CONTIGUOUS-AND-0-RELATIVE) QID.
136 010122 000000      WIQID:: .WORD. 0      ;IQID-X-2-(WORD-OFFSET)
137 010124 000000      GFLAG:: .WORD. 0      ;GENERAL-PURPOSE-FLAG-WORD.
138 010126 000000      CHQA:: .WORD. 0      ;ADDRESS-OF-CURRENT-HQR-QLB-REF-PAGE-WORD.
139 010130 000000      WQW:: .WORD. 0      ;WORKING-HQR-QLB-REF-PAGE-WORD.
140 010132 000004      BITPOS:: .WORD. BIT2      ;POSITION-OF-BIT-IN-HQR-REF-PAGE-WORD.
141 010134 000001      BITADD:: .WORD. 1      ;BIT-ADDR-PORTION-OF-HQR-XLIST-QLB-POINTER.
142 010136 000000      WQA:: .WORD. 0      ;WORD-ADDR-PORTION-OF-HQR-XLIST-QLB-POINTER.
143 010140 000000      RELPOS:: .WORD. 0      ;RELATIVE-POSITION-OF-QLB-ELEMENT.
144 010142 000000      BFIRST:: .WORD. 0      ;HQRMAP-ARGUMENT -- START-BIT.
145 010144 000000      BLAST:: .WORD. 0      ;HQRMAP-ARGUMENT -- STOP-BIT.
146 010146 000000      BSHIFT:: .WORD. 0      ;HQRMAP-ARGUMENT -- SHIFT-COUNT.
147
148 010150 000712*      PSTT:: .WORD. TTABLE      ;POINTER-TO-SOFTWARE-TTABLE.
149 010152 000000      STTENT:: .WORD. 0      ;NUMBER-OF-STT-ENTRIES.
150 010154 000000      PHTT:: .WORD. 0      ;POINTER-TO-HQR-TTABLE.
151 010156 000000      PHXT:: .WORD. 0      ;POINTER-TO-HQR-XTABLE.
152 010160 000000      NULXT:: .WORD. 0      ;POINTER-TO-HQR-NULL-XLIST.
153 010162 000000      QEXCT:: .WORD. 0      ;COUNTER-FOR-HQR-XLIST-QEX-ENTRIES.
154
155 010164 000130*      PSQCL:: .WORD. QCL      ;POINTER-TO-SOFTWARE-QCL.
156 010166 000115      PHQCL:: .WORD. QCLBUF      ;POINTER-TO-HQR-QCL.
157 010170 000000      PHFAL:: .WORD. 0      ;POINTER-TO-HQR-FAL.
158 010172 010345*      PNUHRL:: .WORD. NUHRL+1      ;POINTER-TO-QID-PORTION-OF-DUMMY-HRL.
159 010174 177777      SIDBUF:: .WORD. -1      ;LAST-ALLOCATED-SIDMEM-ADDRESS.
160 010176 000000      SIDBLK:: .WORD. 0      ;SIZE-OF-SINGLE-SID-BLOCK-ALLOCATION.
161 010200      SDBUFS:: .BLKW. 0RYMSZ      ;SUBREAD-BUFFER.
162 010262      SDBEND::
163 010262      HEAT:: .BLKW. 0RYMSZ      ;ADDRESSES-OF-LOCAL-HLR-WORDS-IN-HQR-QLB.
164 010344      NUHRL:: .BLKW. 0RYMSZ+1      ;NO-UHL-DUMMY-HRL-ENTRY.
165

```

```

166
167
168
169 010376 007777
170 010400 007777
171 010402 077777
172 010404 007777
173 010406 003777
174 010410 000377
175
176
177
178 010412 015 012
179 010414
180 000116
181
182
183
184
185
186 010532 000
187 010533 015 012
188 010535 122 105
189 010557 015 012
190 010561 117 105
191 010603 013 012
192 010605 123 101
193 010623 015 012
194 010625 121 130
195 010642 015 012
196 010644 105 122
197 010672 377
198
199
200
201
202
203
204 010674
205 011034
206 011034
207 011034
208 011034
209
210
211
212 011072
213 011232
214 011232
215 011232
216 011232

```

```

:
: MEMORY LIMITS
:
MDHIGH: .WORD 4095. :MRP MICRO MEMORY
CDHIGH: .WORD 4095. :CP CONTROL STORE
OXHIGH: .WORD 77777 :DEX WINDOW MEMORY (=X'7FFF')
FAHIGH: .WORD 4095. :FAL MEMORY
LHHIGH: .WORD 3777 :GLB'S (=X'3FF')
SRHIGH: .WORD 255. :SUBREAD MEMORY
:
: PRINT LINE
:
PRINT: .BYTE 15,12 :PRECEDE PRINT LINE WITH CRLF
:REPT 78.
:BYTE 40
:ENDR
:
: TABLE OF MESSAGES
:
:BYTE 0
:BYTE 15,12
:ASCIZ /READ ERROR ON QLS/
:BYTE 15,12
:ASCIZ /OPEN ERROR ON QLS/
:BYTE 15,12
:ASCIZ /SOAT OVERFLOW/
:BYTE 15,12
:ASCIZ /DEX OVERFLOW/
:BYTE 15,12
:ASCIZ /ERROR ON NON-QLS READ/
ASCIZ: .BYTE 377
:EVEN
:LIST BEX
:NLIST CND
:
: INPUT FILE FDB
:
INFDB: FDBDF$
FDRCA$ FD,RWM
FDBKA$ DHR0,512,,,STAT
FDOPA$ INLUN,,INDNB
INDNB: NMBLK$ ,DAT
:
: EMATRIX,QLS,FDB
:
QLSFDB: FDBDF$
FDRCA$ FD,RWM
FDBKA$ QLSBUF,QLSIZ,,EFN,1,STAT
FDOPA$ QLSLUN
FSRSZ$ 0

```

```

218:
219:
220:
221: ENTER HERE
222:
223: START:
224: FINIT$
225: 011232 016767 000000G 166540 MOV $TKTCB,TSKTCB ;SAVE MY TCB
226: 011244 013767 000274 166534 MOV @#274,OLDVEC ;SAVE ADDRESS AT 274
227: 011252 012737 011566' 000274 MOV #BPTISR,@#274 ;MOVE IN MY ISR ADDRESS
228:
229:
230: RESET HQR
231: 011260 012746 177777 MOV #177777,-(SP) ;CLEAR CSR1
232: 011264 012746 000010 MOV #0$RSET,-(SP) ;SET RESET
233: 011270 CALL CSR1 ;RESET HQR
234: 011274 012746 000010 MOV #0$RSET,-(SP) ;CLEAR RESET
235: 011300 012746 176000 MOV #0$NCLK,-(SP) ;SET NO CLKS
236: 011304 CALL CSR1 ;MOVE TO CSR1
237:
238:
239: RESET MRP AND CP
240: 011310 005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
241: 011312 012746 000004 MOV #0$MSET,-(SP) ;SET RESET
242: 011316 CALL CSR1
243: 011322 012746 000004 MOV #0$MSET,-(SP) ;CLEAR RESET
244: 011326 005046 CLR -(SP) ;SET NOTHING
245: 011330 CALL CSR1
246:
247: 011334 005046 CLR -(SP) ;CLEAR NOTHING
248: 011336 012746 000002 MOV #0$CSET,-(SP) ;SET RESET
249: 011342 CALL CSR1
250: 011346 012746 000002 MOV #0$CSET,-(SP) ;CLEAR RESET
251: 011352 005046 CLR -(SP) ;SET NOTHING
252: 011354 CALL CSR1
253:
254: CALL CONVERSION MODULES IN SUCCESSION
255: FIRST CALL MODULES THAT DON'T NEED THE SOFTWARE OLS
256:
257:
258:
259:
260: WE NO LONGER LOAD THE MICROCODE ON EACH CYCLE
261:
262:
263: CALL MICRO ;LOAD MICROCODE
264:
265:
266:
267: 011360 CALL FRAME2 ;CLEAR FRAME 2 MEMORIES

```

```

269      ;
270      ;
271      ;
272      ;
273      ;
274      ;
275      ;
276      ;
277      ;
278      ;
279      ;
280      ;
281      ;
282      ;
283      ;
284      ;
285      ;
286      ;
287      ;
288      ;
289      ;
290      ;
291      ;
292      ;
293      ;
294      ;
295      ;
296      ;
297      ;
298      ;
299      ;
300      ;
301      ;
302      ;
303      ;
304      ;
305      ;
306      ;

```

```

      READ SOFTWARE QLS FROM FILE 'EMATRIX.QLS'
      EQLS:
      MOV #QLSFDB,R0      ;R0 -> FDB
      MOV #FDSQLS,R1     ;R1 -> FDSC
      CALL BLDFL
      MOV #QLSIZ,INFDB+F,BKDS
      MOV EMX,INDNB+N,FNAM
      MOV EMX+2,INDNB+N,FNAM+2
      MOV EMX+4,INDNB+N,FNAM+4
      MOV QLSR50,INDNB+N,FTYP
      OFNB$R #QLSFDB
      OPEN$R #INFDB
      *****
      BCC 1$      ;OPEN OK
      CALL ERR4
      JMP EXIT
      1$: READ$ R0      ;READ ENTIRE QLS
      WAIT$ R0
      BCC 2$      ;READ OK
      CALL ERR5
      JMP EXIT
      2$: CALL ,DLFNB      ;DELETE EMATRIX.QLS FILE
      CLR FDSQLS      ;RESET FDSC
      MOV FALAD,FOSCTL      ;INITIALIZE FOS WORD COUNT
      MOV QRYMAX,FOSCTL+2   ;GET LAST ALLOCATED QUERY ID
      INC FOSCTL+2         ;INITIALIZE NUMBER OF QUERIES

```

```

308      ;
309      ;
310      ;
311      ;
312      ; CALL CONVERSION MODULES IN SUCCESSION
313      ; CALL THOSE MODULES THAT NEED THE SOFTWARE QLS
314      ; NB. THE MODULE CD DOES NOT NEED THE QLS BUT IT
315      ; MUST BE CALLED AFTER THE QEX MODULE.
316      ;
317      ;
318      ;
319      ;
320      ;
321      ;
322      ;
323      ;
324      ;
325      ;
326      ;
327      ;
328      ;
329      ;
330      ;
331      ;
332      ;
333      ;
334      ;
335      ;

```

316	011502					CALL	QBLD		: CONVERT QLB
317	011506					CALL	SDLBLD		: CONVERT SDLB
318	011512					CALL	QEXLD		: CONVERT QEX
319	011516	103412				BCS	EXIT		: ERROR ON CONVERSION
320	011520	012767	076000	166350		MOV	#QXLOW,QXADD		: RESTORE START ADDRESS OF HQR QEX
321	011526					CALL	TTXT		: CONVERT TTABLE AND XTABLE
322	011532	103404				BCS	EXIT		: ERROR ON CONVERSION
323	011534					CALL	CD		: LOAD CP DATA MEMORY
324	011540					CALL	QCLFAL		: CONVERT QCL AND FAL
325									
326									
327									
328						EXIT			
329									
330	011544					EXIT:			
331	011544	016737	166236	000274		MOV	OLDVEC,@#274		: RESTORE OLD VECTOR ADDRESS
332									
333	011552					RST#C	HRLGET		
334									
335	011560					EXIT#S			

```
337 ;
338 ;
339 ; INTERRUPT SERVICE ROUTINE
340 TRAP INTERRUPTS FROM HDR THROUGH VECTOR ADDRESS 274
341 SET EVENT FLAG 3
342 ; CP DEBUGGING ROUTINES WILL READ CSR #2 AND DECODE THE INTERRUPT
343 ;
344 ;
345 011566 EPTISR::
346 011566 SAVE R0,R1,R2,R3,R4,R5
347 ;
348 011602 016705 166175 MOV TSKTCB,R5 ;LOAD MY TCB
349 011606 012700 000003 MOV #EFN,3,R0 ;EVENT FLAG TO BE SET
350 011612 CALL $CFI
351 011616 050011 BIS R0,(R1) ;SET LOCAL FLAG
352 011620 CALL $DRDSE ;DECLARE SIGNIFICANT EVENT
353 ;
354 011624 RESTOR R0,R1,R2,R3,R4,R5
355 011640 000002 RTI
```



```

386 ;
387 ;
388 ;
389 ; READ A 256-WORD BLOCK FROM ONE OF THE HQR LOAD
390 ; FILES WHICH FILE IS DETERMINED BY THE WAY IN
391 ; WHICH THE FILE NAME BLOCK HAS BEEN FILLED IN.
392 ;
393 GET::
394 READ$ #INFDB,.,#VIRT,#EFN.1,#STAT
395 BCC 1$
396 CALL ERR1
397 BR GETSX
398 ;
399 ;: WTSE$S #EFN.1
400 ;
401 CLEF$S #EFN.1
402 TSTB STAT
403 BGT GETCX ;GOOD COMPLETION
404 CALL ERR1
405 ;
406 GETSX:
407 SEC
408 BR GETX
409 GETCX: INC VIRT+2 ;INC BLOCK COUNTER
410 CLC
411 GETX: RETURN

```

393 011734
394 011734
395 012002 103003
396 012004
397 012010 000417
398
399 012012
400
401 012024
402 012036 105767 166006
403 012042 003004
404 012044
405
406 012050
407 012050 000261
408 012052 000403
409 012054 005267 165766
410 012060 000241
411 012062

```
413      ;
414      ;
415      ; PLACE A VALUE INTO HOR CONTROL AND STATUS REGISTER #1.
416      ; READ CSR1 INTO A WORK AREA, CLEAR BITS AT 4(SP), SET
417      ; BITS AT 2(SP), RE-WRITE CSR1.
418      ;
419      ; INPUT:
420      ; 2(SP) = BITS TO BE SET IN CSR1
421      ; 4(SP) = BITS TO BE CLEARED IN CSR1
422      ;
423      ;
424 012064      ; CSR1::
425 012064 016767 176420 165766      MOV.   OR$CR1, APLACE      ; GET THE CURRENT VALUE
426 012072 046667 000004 165760      BIC.   4(SP), APLACE      ; CLEAR FIRST
427 012100 056667 000002 165752      BIS.   2(SP), APLACE      ; THEN SET
428 012106 016767 165746 176420      MOV.   APLACE, OR$CR1      ; NOW RETURN IT
429 012114 011666 000004              MOV.   (SP), 4(SP)         ; MOVE RETURN ADDR TO TOP OF STACK
430 012120 022626              CMP.   (SP)+, (SP)+        ; BUMP STACK POINTER PAST ARGS
431 012122      RETURN.              ; SPLIT
```

```

433 ;
434 ;
435 ; WRITE TO TT0 AND PROMPT.
436 ;
437 ;
438 012124 005267 165724 ERR5:: INC ERWORD
439 012130 005267 165720 ERR4:: INC ERWORD
440 012134 005267 165714 ERR3:: INC ERWORD
441 012140 005267 165710 ERR2:: INC ERWORD
442 012144 005267 165704 ERR1:: INC ERWORD
443 ;
444 ; USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF
445 ; THE MESSAGE TABLE. FIND THE END OF THE MESSAGE FIRST,
446 ; THEN THE BEGINNING, THEN GET THE LENGTH.
447 ;
448 012150 016702 165700 MOV ERWORD,R2 ;LOAD LOOP COUNT
449 012154 012701 010672 MOV *ASCIZ,R1 ;POINT TO END OF MESSAGE TABLE
450 012160 105741 1# TSTB -(R1) ;LOOK FOR END OF MESSAGE
451 012162 001376 BNE 1$
452 012164 005302 DEC R2 ;LOOP COUNT
453 012166 001374 BNE 1$ ;BACK UP ANOTHER MESSAGE
454 012170 010100 MOV R1,R0 ;SAVE POINTER TO END OF MESSAGE
455 012172 105741 2# TSTB -(R1) ;BACK UP TO BEGINNING OF MESSAGE
456 012174 001376 BNE 2$
457 012176 005201 INC R1 ;BUMP TO FIRST CHAR OF MESSAGE
458 012200 160100 SUB R1,R0 ;R0 NOW = MESSAGE LENGTH
459 ;
460 012202 QIOW$S #IO,WVB,#LUN:TT,#EFN.1, #STAT,<R1,R0>,ABEND
461 ;
462 012256 CLEF$S #EFN.1
463 012270 105767 165554 TSTB STAT ;GOOD RETURN
464 012274 003403 BLE ABEND ;NO
465 012276 005067 165552 CLR ERWORD ;CLEAR ERROR NUMBER INDICATOR
466 012302 RETURN ;AND RETURN
467 ;
468 012304 ABEND: ABRT$S #MYSELF
469 ;

```

```
1          ;
2          000104*      QLSHD==QLSXX1+QLSBUF          :CONVERT TO ADDRESS.
3          000110*      QLSBAT==QLSXX17+QLSBUF
4          000112*      QLBAD==QLSXX2+QLSBUF
5          000114*      SDLBAD==QLSXX3+QLSBUF
6          000116*      QEXAD==QLSXX4+QLSBUF
7          000120*      QRYMAX==QLSXX5+QLSBUF
8          000122*      FALAD==QLSXX6+QLSBUF
9          000124*      TTSIZ==QLSXX7+QLSBUF
10         000126*      XTSIZ==QLSXX8+QLSBUF
11         000130*      QCL==QLSXX10+QLSBUF
12         000212*      FAL==QLSXX11+QLSBUF
13         000712*      TTABLE==QLSXX12+QLSBUF
14         004202*      XTABLE==QLSXX13+QLSBUF
15         006212*      QLB==QLSXX14+QLSBUF
16         006730*      SDLB==QLSXX15+QLSBUF
17         007304*      QEX==QLSXX16+QLSBUF
18         ;
19         000104*      QLS==QLSBUF
```

LMAIN MA M1110 27-MAR-80 14:58 PAGE 24

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1 011232

.END. START.

ABEND = 012304R.	BYTE31 = 000037	BYTE83 = 000123	DB#CPP = 001457	F.BKEF = 000050
ADDR = 000070RG.	BYTE32 = 000040	BYTE84 = 000124	DB#SPT = 000026	F.BKP1 = 000051
ALLHRL = 000026 G.	BYTE33 = 000041	BYTE85 = 000125	DB#TPC = 000023	F.BKST = 000024
ALUCKE = 040000	BYTE34 = 000042	BYTE86 = 000126	DG.ERR = 001000	F.BKVB = 000064
ALUOE = 004000	BYTE35 = 000043	BYTE87 = 000127	DG.SDF = 002000	F.CHR = 000075
APLACE = 000060RG.	BYTE36 = 000044	BYTE88 = 000130	DG.TDF = 004000	F.CHTG = 000034
ASCIZ = 010672R.	BYTE37 = 000045	BYTE89 = 000131	DHR0 = ***** GX.	F.DFNB = 000046
A01 = 010000	BYTE38 = 000046	BYTE9 = 000011	DISPGS = 100000	F.DSPT = 000044
BFIRST = 010142RG.	BYTE39 = 000047	BYTE90 = 000132	DMA = ***** GX.	F.DVNM = 000134
BITADD = 010134RG.	BYTE4 = 000004	BYTE91 = 000133	DMAAWR = 000005	F.EFBK = 000010
BITPOS = 010132RG.	BYTE40 = 000050	BYTE92 = 000134	DMACL = 011666R.	F.EFN = 000050
BITVAL = 000000	BYTE41 = 000051	BYTE93 = 000135	DMARRD = 000003	F.EOBB = 000032
BIT0 = 000001	BYTE42 = 000052	BYTE94 = 000136	DMARWR = 000004	F.ERR = 000052
BIT1 = 000002	BYTE43 = 000053	BYTE95 = 000137	DMASET = ***** GX.	F.FACC = 000043
BIT10 = 002000	BYTE44 = 000054	BYTE96 = 000140	EFBUF = 000010RG.	F.FFBY = 000014
BIT11 = 004000	BYTE45 = 000055	BYTE97 = 000141	EFN.1 = 000001	F.FHAM = 000110
BIT12 = 010000	BYTE46 = 000056	BYTE98 = 000142	EFN.3 = 000003 G.	F.FNB = 000102
BIT13 = 020000	BYTE47 = 000057	BYTE99 = 000143	ENBR = 000034R.	F.FTYP = 000116
BIT14 = 040000	BYTE48 = 000060	BYTVAL = 000144	ENR1 = 010000	F.FVER = 000120
BIT15 = 100000	BYTE49 = 000061	B#EQ0 = 000004 G.	ERR1 = 012144RG.	F.HIBK = 000004
BIT2 = 000004	BYTE5 = 000005	B#EOTR = 000002 G.	ERR2 = 012140RG.	F.LUN = 000042
BIT3 = 000010	BYTE50 = 000062	B#FMN = 000020 G.	ERR3 = 012134RG.	F.MBCT = 000054
BIT4 = 000020	BYTE51 = 000063	B#FST = 020000 G.	ERR4 = 012130RG.	F.MBC1 = 000055
BIT5 = 000040	BYTE52 = 000064	B#HUL = 000040 G.	ERR5 = 012124RG.	F.MBFG = 000056
BIT6 = 000100	BYTE53 = 000065	B#HOT = 000100 G.	ERWORD = 000054R.	F.NRBD = 000024
BIT7 = 000200	BYTE54 = 000066	B#PUBH = 010000 G.	EXIT = 011544R.	F.NREC = 000030
BIT8 = 000400	BYTE55 = 000067	B#PUBL = 004000 G.	FAHIGH = 010404RG.	F.OVBS = 000030
BIT9 = 001000	BYTE56 = 000070	B#PUFH = 100000 G.	FAL = 000212RG.	F.RACC = 000016
BLAST = 010144RG.	BYTE57 = 000071	B#PUFL = 040000 G.	FALAD = 000122RG.	F.RATT = 000001
BLDEFL = ***** GX.	BYTE58 = 000072	B#SDEL = 000001 G.	FALMSZ = 000500 G.	F.RCNM = 000034
BPTISR = 011566RG.	BYTE59 = 000073	B#SUC = 000200 G.	FDSOLS = ***** GX.	F.RCTL = 000017
BSHIFT = 010146RG.	BYTE6 = 000006	CBKALL = 001000	FD.FID = 000000	F.RSIZ = 000002
BYTE0 = 000000	BYTE60 = 000074	CBKCLK = 000400	FD.FNB = 000006	F.SBCT = 000000
BYTE1 = 000001	BYTE61 = 000075	CD = ***** GX.	FD.FVR = 000004	F.SEQN = 000100
BYTE10 = 000012	BYTE62 = 000076	CDHIGH = 010400RG.	FD.LEN = 000010	F.SPDV = 000072
BYTE11 = 000013	BYTE63 = 000077	CF.COT = 000041	FD.RWM = ***** GX.	F.SPUN = 000074
BYTE12 = 000014	BYTE64 = 000100	CF.DGN = 000046	FIRST = 000001 G.	F.STBK = 000036
BYTE13 = 000015	BYTE65 = 000101	CF.DHR = 000042	FN.ACK = 000016	F.UNIT = 000136
BYTE14 = 000016	BYTE66 = 000102	CF.DMC = 000047	FN.FSA = 000000	F.URBD = 000020
BYTE15 = 000017	BYTE67 = 000103	CF.HBR = 000045	FN.FSB = 000002	F.VBN = 000064
BYTE16 = 000020	BYTE68 = 000104	CF.HRL = 000044	FN.FSD = 000004	F.VBSZ = 000060
BYTE17 = 000021	BYTE69 = 000105	CF.UPD = 000043	FN.FSD = 000020	004 GET = 011734RG.
BYTE18 = 000022	BYTE7 = 000007	CHQA = 010126RG.	FN.FSH = 000010	004 GETCX = 012054R.
BYTE19 = 000023	BYTE70 = 000106	CNOBRE = 100000	FN.NMB = 000022	004 GETSX = 012050R.
BYTE2 = 000002	BYTE71 = 000107	CODE = 000066RG.	FN.QLS = 000006	004 GETX = 012052R.
BYTE20 = 000024	BYTE72 = 000110	CPCCEN = 010000	FN.RDC = 000014	004 GFLAG = 010000RG.
BYTE21 = 000025	BYTE73 = 000111	CPREAD = 040000	FN.UPD = 000012	004 HEAT = 010262RG.
BYTE22 = 000026	BYTE74 = 000112	CPURTE = 020000	FOSCTL = ***** GX.	HE00 = 100000 G.
BYTE23 = 000027	BYTE75 = 000113	CSADDR = 000004	FO.RD = ***** GX.	HLMN = 000002 G.
BYTE24 = 000030	BYTE76 = 000114	CSEDC1 = 100000	FRAME2 = ***** GX.	HLNE = 040002 G.
BYTE25 = 000031	BYTE77 = 000115	CSDR = 000040	F.ACTL = 000076	HMN = 000001 G.
BYTE26 = 000032	BYTE78 = 000116	CSR1 = 012064RG.	F.ALDC = 000040	HRLPHT = 020000 G.
BYTE27 = 000033	BYTE79 = 000117	CSURTE = 000100	F.BBFS = 000062	HXTSLB = 100000 G.
BYTE28 = 000034	BYTE8 = 000010	DATA = 000072RG.	F.BDB = 000070	INDR = 010000RG.
BYTE29 = 000035	BYTE80 = 000120	DATA1 = 000072RG.	F.BGBC = 000067	INDR = 010000RG.
BYTE3 = 000003	BYTE81 = 000121	DBR.RD = 000001	F.BKDN = 000020	INLUN = 000003
BYTE30 = 000036	BYTE82 = 000122	DBSLEN = 000116	F.BKDS = 000020	INSAVE = 000102RG.

IO,UVB=***** GX	OLDVEC 000006R	QRYMAX=000120RG	Q#QLB = 000054	SR,NIP 000022 002
IQID 010120RG	ONECD 011652RG	QRYMSZ=000031 G	Q#QLR = 000001	SR,SDB 000032 002
LAST = 000002 G	ONECD2 011660RG	QR#CR1=176420	Q#QW = 000042	SR,SRC 000002 002
LCD 000030RG	ONEMD 011642RG	QR#CR2=176422	Q#RDCD=000005	SR,SUN 000000 002
LCOUNT 000062RG	PAR#*#* 000027	QR#LBR=176424	Q#RDMD=000005	SR,TUS 000056 002
LCS 000024RG	PHFAL 010170RG	QXADD 000076RG	Q#REBK=001000	SR,WSL 000052 002
LDCT = 140000 G	PHOCL 010166RG	QXCNT 000074RG	Q#RNC = 000000	SR,YR 000004 002
LHHIGH 010406RG	PHTT 010154RG	QXCNTL=100000 G	Q#RSC = 004000	SR,11N 000024 002
LMM 000020RG	PHXT 010156RG	QXHGH 010402RG	Q#RSET=000010	SR,11P 000016 002
LMM = 000001 G	PLB = 000010	QXLQW = 076000 G	Q#SM = 100000	START 011232R
LOC,EN=000100	PLC = 000020	Q#ATTN=000100	Q#SP = 000120	STAT 000050RG
LOC,UA=040000	PLD = 000030	Q#BCL = 000001	Q#SP2 = 000340	STTENT 010152RG
LOC,UB=100000	PLRWR = 000200	Q#CCCP = 000040	RELPOS 010140RG	SXTQEX=100000 G
LUN,TT=000001	PLR,EN=000200	Q#CHB = 000400	RGQ,EN=000200	SXTSLB=040000 G
LX,OID 010116RG	PNUHLR 010172RG	Q#CHRL=000200	RGQ,VA=020000	S#CLR = 000000
MAREN1=000001	PQLB = 000040 G	Q#CLR = 000040	R,OSGC=000015	S#LA = 000001
MAREN2=004000	PRINT 010414RG	Q#CNC = 030000	R,OSPC=000014	S#OB = 000005
MARL0D=010000	PSOCL 010164RG	Q#CP = 000060	R,OSPN=000005	S#OR = 000006
MAROUT=000002	PSQLB 010104RG	Q#CPCC=000010	R,OSPR=000012	S#OX = 000004
MAR,LO=002000	PSTT 010150RG	Q#CP2 = 000260	R,OSTN=000002	S#SR = 000007
MAR,OU=000040	QCL = 000130RG	Q#C9C = 010000	SDBEND 010262RG	S#S1 = 000010
MBKALL=001000	QCLBUF=000115 G	Q#CSEL=000360	SDBUFS 010200RG	S#S2 = 000014
MBKCLK=000400	QCLFAL=***** GX	Q#CSET=000002	SDLB = 006730RG	S,BFHD=000020
MDHIGH 010376RG	QENTL 010110RG	Q#CSP = 020000	SDLBAD=000114RG	S,FATT=000016
MMADRD=000100	QEX = 007304RG	Q#DMA = 000001	SDLBLD=***** GX	S,FDB = 000140
MMLEFT=000002	QEXAD = 000116RG	Q#ENBK = 040000	SDMAP = 000100 G	S,FNAM=000006
MMOE = 000004	QEXCT 010162RG	Q#ENOP=020000	SDMAP=000010 G	S,FNB = 000036
MMURTE=000010	QEXLD = ***** GX	Q#FAL = 004000	SELECT 000056RG	S,FNBW=000017
MMOBRE=100000	QEXMSZ=000600 G	Q#FC = 000045	SEQ,CI=000010	S,FNTY=000004
MREN1 = 000001	QLB = 006212RG	Q#FD = 000044	SHE00 = 140000 G	S,FTYP=000002
MREN2 = 020000	QLBAD = 000112RG	Q#FP = 000046	SLLMN = 010000 G	S,NFEN=000020
MSTR2 = 000070RG	QLBLD = ***** GX	Q#HBF = 000002	SLLNE = 020000 G	TD#CTR = 176370
MSYN = 000040	QLBMSZ=000516 G	Q#ICP = 000006	SLLN = 004000 G	TD#CTW = 176360
MS,DGN=010000	QLS = 000104RG	Q#IHB = 000003	SHRL = 100000 G	TD#INL = 004000
MYSELF 000000R	QLSBAT=000110RG	Q#IHLR=000002	SIDBLK 010176RG	TD#MEM = 000270
N = 000144	QLSBUF 000104RG	Q#IMRP=000007	SIDBUF 010174RG	TD#DAR = 176344
NPSQLB 010106RG	QLSFB 011072R	Q#LBD = 001000	SIBUZ = 010000 G	TD#OTR = 176346
NUHLCD=000724 G	QLSHD = 000104RG	Q#LBDP=001001	SLBMSZ=000354 G	TD#ORD = 000274
NUHRL 010344RG	QLSIZ = 010000 G	Q#LBP = 000001	SOLS 011364R	TD#RST = 176366
NULXT 010160RG	QLSLUN=000004	Q#LDCD=000003	SRHIGH 010410RG	TD#SWJ = 176376
N,BFAC=000004	QLSR50 000042R	Q#LDMD=000004	SR,ARE 000114	TD#STAR = 176372
N,BHGH=000006	QLSXX1=000000	Q#LDPP=002000	SR,ARS 000106	TD#TAW = 176362
N,BTCH=000004	QLSXX2=000006	Q#LHP = 010000	SR,DAY 000010	TD#TDR = 176374
N,BUFB=004000	QLSXX3=000010	Q#HNC = 140000	SR,DLT 000014	TD#TDW = 176364
N,BUFW=002000	QLSXX4=000012	Q#HNR = 000052	SR,ECH 000047	TD#TRSF = 000100RG
N,DIR = 000024	QLSXX5=000014	Q#HNP = 000040	SR,ECH 000046	TD#TTCB = 000004R
N,DVNM=000032	QLSXX6=000016	Q#HNP2=000240	SR,ECL 000050	TD#TABLE = 000712RG
N,FID = 000000	QLSXX7=000020	Q#HSC = 040000	SR,FIB 000012	TD#TMSZ=003270 G
N,FNAM=000006	QLSXX8=000022	Q#HSET=000004	SR,GRE 000100	TD#TISZ = 000124RG
N,FOS = 000764	QLSXX9=000024	Q#HSP = 100000	SR,GRS 000072	TD#TXT = ***** GX
N,FTYP=000014	QLSXX10=000106	Q#HCLK=176000	SR,LEN 000122	TD#TAD = 000020
N,FVER=000016	QLSXX11=000006	Q#HPP = 000100	SR,LIN 000066	TD#TBA = 000002
N,NEXT=000022	QLSXX12=004076	Q#HPPSU=000320	SR,LIP 000062	TD#TBD = 000010
N,ORY = 000031	QLSXX14=006106	Q#HPP2 = 000300	SR,MON 000006	TD#TBSD = 100000
N,STAT=000020	QLSXX15=006624	Q#HHLT=000013	SR,NDC 000042	TD#TBT = 000020
N,SUNT=000002	QLSXX16=007200	Q#QL = 000043	SR,NDS 000036	TD#BTAR = 000030
N,UNIT=000034	QLSXX17=000004	Q#QLA = 000057	SR,NIN 000030	TD#BTB = 002000

T\$GD	= 000100	UBD IN	= 000020	WORD35	= 000106	WORD65	= 000202	WORD95	= 000276
T\$CLK	= 002000	VIRT	= 000044RG	WORD36	= 000110	WORD66	= 000204	WORD96	= 000300
T\$DISK	= 000200	WCOUNT	= 000064RG	WORD37	= 000112	WORD67	= 000206	WORD97	= 000302
T\$DRD	= 000004	WIGID	= 010122RG	WORD38	= 000114	WORD68	= 000210	WORD98	= 000304
T\$EMEM	= 010000	WORD0	= 000000	WORD39	= 000116	WORD69	= 000212	WORD99	= 000306
T\$FSAA	= 000000	WORD1	= 000002	WORD4	= 000010	WORD7	= 000016	WQA	= 010136RG
T\$FSAB	= 000004	WORD10	= 000024	WORD40	= 000120	WORD70	= 000214	WQENTL	= 010112RG
T\$FSAC	= 000014	WORD11	= 000026	WORD41	= 000122	WORD71	= 000216	WQW	= 010130RG
T\$FSB2	= 000010	WORD12	= 000030	WORD42	= 000124	WORD72	= 000220	WRDVAL	= 000310
T\$IB	= 000026	WORD13	= 000032	WORD43	= 000126	WORD73	= 000222	XQID	= 010114RG
T\$IBAR	= 000024	WORD14	= 000034	WORD44	= 000130	WORD74	= 000224	XTABLE	= 004202RG
T\$IBE	= 020000	WORD15	= 000036	WORD45	= 000132	WORD75	= 000226	XTBMSZ	= 002010 G
T\$IBF	= 040000	WORD16	= 000040	WORD46	= 000134	WORD76	= 000230	XTREAD	= 001000
T\$ICD	= 000040	WORD17	= 000042	WORD47	= 000136	WORD77	= 000232	XTSIZ	= 000126RG
T\$MODE	= 004000	WORD18	= 000044	WORD48	= 000140	WORD78	= 000234	XTURTE	= 000400
T\$OB	= 000036	WORD19	= 000046	WORD49	= 000142	WORD79	= 000236	\$CEFI	= ***** GX
T\$OBE	= 004000	WORD2	= 000004	WORD5	= 000012	WORD8	= 000020	\$DRDSE	= ***** GX
T\$OBF	= 010000	WORD20	= 000050	WORD50	= 000144	WORD80	= 000240	\$TKTCB	= ***** GX
T\$OBRA	= 000034	WORD21	= 000052	WORD51	= 000146	WORD81	= 000242	\$\$\$	= 000000R 006
T\$OBWA	= 000032	WORD22	= 000054	WORD52	= 000150	WORD82	= 000244	\$\$\$ARG	= 000002
T\$OUTA	= 100000	WORD23	= 000056	WORD53	= 000152	WORD83	= 000246	\$\$\$OST	= 000016
T\$RBD0	= 000200	WORD24	= 000060	WORD54	= 000154	WORD84	= 000250	\$\$\$T1	= 000067
T\$RBD	= 000040	WORD25	= 000062	WORD55	= 000156	WORD85	= 000252	\$\$\$T2	= 000027
T\$RSET	= 040000	WORD26	= 000064	WORD56	= 000160	WORD86	= 000254	.DLFNB	= ***** GX
T\$SC	= 000022	WORD27	= 000066	WORD57	= 000162	WORD87	= 000256	.FINIT	= ***** G
T\$SCLK	= 020000	WORD28	= 000070	WORD58	= 000164	WORD88	= 000260	.FSRCB	= ***** G
T\$SEG1	= 000000	WORD29	= 000072	WORD59	= 000166	WORD89	= 000262	.OPFNB	= ***** G
T\$SEG2	= 000001	WORD3	= 000006	WORD6	= 000014	WORD9	= 000022	.READ	= ***** G
T\$SEG3	= 000002	WORD30	= 000074	WORD60	= 000170	WORD90	= 000264	.WAIT	= ***** G
T\$SO	= 000001	WORD31	= 000076	WORD61	= 000172	WORD91	= 000266	...PC1	= 011072R
T\$UBUS	= 100000	WORD32	= 000100	WORD62	= 000174	WORD92	= 000270	...PC2	= 011232R
T\$1CLK	= 000400	WORD33	= 000102	WORD63	= 000176	WORD93	= 000272	...PC3	= 010674R
T\$BBEN	= 000020	WORD34	= 000104	WORD64	= 000200	WORD94	= 000274	...TPC	= 000020

. ABS. 000000 000
012336 001
SRCOFF 000122 002
FDSOOF 000010 003
FNDFS 000022 004
\$FSR1 000000 005
\$DPB\$ 000016 006
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 8820 WORDS (35 PAGES)
DYNAMIC MEMORY: 10196 WORDS (39 PAGES)
ELAPSED TIME: 00:01:41
LMAIN LMAIN /-SP=C20.1JP.C.Q. IM.C20.1JLMAIN.QG.ENDST

1
2:000000
3
4
5
6
7
8
9
10
11

:
:
: .TITLE: DMASUB
: .PSECT: DMASUB
:
: HARDWARE QUERY RESOLVER
: DMA SUBROUTINES
:
: USED BY LOADER AND HQRLS
:
:
: .MCALL: RDAF\$S,CLEF\$S
:
:

```
13          ;          START UP DMA LOAD MICROCODE IN CP. MRP DOES NOTHING.
14          ;
15          ;          DMASET::
16 000000      012746  000377      MOV      #377, -(SP)          ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
17 000004          CALL     SEQMM
18          ;
19 000010      005046          CLR      -(SP)          ;RESET BR INHIBIT
20 000012      005046          CALL     MRPCR
21 000016      005046          CLR      -(SP)          ;START MICROCODE AT 0
22 000020      005046          CALL     SEQCS
23 000024      005046          CLR      -(SP)          ;RESET BR INHIBIT
24 000026          CALL     CPCR
25 000032      012767  001000  176422  MOV      #Q$REBK, Q$CR2      ;RE-ARM INTERRUPTS
26 000040      012767  120000  176422  MOV      *(Q$SM+Q$ENOP), Q$CR2 ;SET SEARCH MODE + ENABLE INTERRUPTS
27 000046      012746  000360          MOV      #Q$CSEL, -(SP)      ;CLEAR ALL SELECTIONS
28 000052      052716  001001          BIS      *(Q$LBD+Q$LBP), (SP) ;CLEAR DRIVE AND PULSE
29 000056      052716  170000          BIS      *(Q$MNC+Q$CNC), (SP) ;CLEAR CP NO-CLOCK
30 000062      005046          CLR      -(SP)          ;SET NOTHING
31 000064          CALL     CSR1
32          ;
33 000070          RETURN
```

```

35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 000072. ;
46 000100 012767 000002 176424 ;
47 000106 016701 176422 ;
48 000112 032701 000100 ;
49 000116 001373 ;
50 ;
51 000120 016767 000000G 176424 ;
52 000126 012767 120040 176422 ;
53 000134 016701 176422 ;
54 000140 032701 000040 ;
55 000144 001373 ;
56 ;
57 000146 016767 000000G 176424 ;
58 000154 012767 120040 176422 ;
59 000162 016701 176422 ;
60 000166 032701 000040 ;
61 000172 001373 ;
62 ;
63 000174 016767 000000G 176424 ;
64 000202 012767 120040 176422 ;
65 ;
66 ;
67 ;
68 000210 ;
69 ;
70 000222 032767 000004 000000G ;
71 000230 001767 ;
72 ;
73 000232 ;
74 ;
75 ;
76 ;
77 000244 012767 100400 176422 ;
78 000252 012767 101000 176422 ;
79 000260 012767 160000 176422 ;
80 ;
81 000266 011666 000002 ;
82 000272 005726 ;
83 000274 ;
84 000001 ;

```

```

PERFORM DMA LOAD OF HQR DATA MEMORIES (MRP AND CP).
2(SP) = MEMORY SELECT (ATTN) CODE.
FIELD 'MSTR2' = HQR MEMORY ADDRESS.
FIELD 'TRANSF' = NUMBER OF WORDS TO BE TRANSFERRED.
FIELD 'INSAVE' = ADDRESS OF CC BUFFER TO BE TRANSFERRED.
DMA:
MOV 2(SP),QR$LBR ;MOVE ATTN CODE TO LOD BUS REG
MOV *(Q$ATTN+Q$SM+Q$ENOP),QR$CR2 ;SET ATTN CODE READY.
MOV QR$CR2,R1 ;READ CSR2
BIT *Q$ATTN,R1 ;ATTN CLEAR
BNE 6$ ;NO READ AGAIN
MOV MSTR2,QR$LBR ;CD MEMORY START ADDRESS.
MOV *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2 ;SET CC TO CP.
MOV QR$CR2,R1 ;READ CSR2
BIT *Q$CCCP,R1 ;IS CC TO CP CLEAR.
BNE 7$ ;NO READ AGAIN
MOV TRANSF,QR$LBR ;TRANSFER COUNT.
MOV *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2 ;SET CC TO CP.
MOV QR$CR2,R1 ;READ CSR2
BIT *Q$CCCP,R1 ;IS CC TO CP CLEAR.
BNE 8$ ;NO READ AGAIN
MOV INSAVE,QR$LBR ;CC MEMORY DATA BUFFER.
MOV *(Q$CCCP+Q$SM+Q$ENOP),QR$CR2 ;SET CC TO CP.
WAIT FOR INTERRUPT FROM CP.
RDAF$S #EFBUF ;READ EVENT FLAGS.
BIT #BIT2,EFBUF ;WAS EF#3 SET.
BEQ 9$ ;NO READ AGAIN
GLEF$S #EFN.3
RE-ARM INTERRUPTS.
MOV *(Q$SM+Q$CHB),QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
MOV *(Q$SM+Q$REBK),QR$CR2 ;RE-ARM.
MOV *(Q$SM+Q$ENBK+Q$ENOP),QR$CR2 ;ENABLE.
MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK.
TST (SP)+ ;POINT TO RETURN ADDRESS.
RETURN
END

```

ALUCKE = 000000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 000100
CNOBRE = 100000
CPCCEN = 010000
CPCR = ***** GX
CPREAD = 040000
CPWRITE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSURTE = 000100
DBR_RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DMA = 000072RG
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
DMASET = 000000RG
EFBUF = ***** GX
EFN_3 = ***** GX
ENBR = 010000
INSAVE = ***** GX
LOC_EN = 000100
LOC_WA = 040000
LOC_WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR_1Q = 002000
MAR_1O = 000040
MBKALL = 000100
MBKCLK = 000400
MMADR = 000100
MMLEFT = 000002
MMODE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = ***** GX
MSTR2 = ***** GX
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
PLRW = 000200
PLR_EN = 000200
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IH = 000003
Q\$IHRP = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDM = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGO_EN = 000200
RGO_VA = 020000
SEQCS = ***** GX
SEQMM = ***** GX
SEQ_CI = 000010
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$OR = 000006
S\$OX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$QTR = 176346
TD\$QRD = 000274
TD\$SW = 176376
TD\$STAR = 176372
TD\$TAU = 176362
TD\$TDR = 176374
TD\$TDW = 176364
TRANSF = ***** GX
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTID = 002000
T\$CD = 000100
T\$CLK = 002000
T\$DLSK = 000200
T\$DRD = 000004
T\$FEMEM = 010000
T\$FSA = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBE = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036

DMASUB: MACRO:M1110 27-MAR-80 14:51 PAGE:7=2
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

T#DBE = 004000	WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240
T#DBF = 010000	WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242
T#DBRA = 000034	WORD17 = 000042	WORD39 = 000116	WORD50 = 000170	WORD82 = 000244
T#DBWA = 000032	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246
T#DUTA = 100000	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T#RBD0 = 000200	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T#RNB = 000040	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T#RSET = 040000	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T#SC = 000022	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T#SCLK = 020000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T#SEG1 = 000000	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T#SEG2 = 000001	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T#SEG3 = 000002	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T#SO = 000001	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T#SUBUS = 100000	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T#1CLK = 000400	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T#BEN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
UBD: IN = 000020	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD0 = 000000	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD1 = 000002	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD10 = 000024	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD11 = 000026	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORDVAL = 000310
WORD12 = 000030	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	XTREAD = 001000
WORD13 = 000032	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	XTWRITE = 000400
WORD14 = 000034				

. ABS. 000000 000
000000 001
DMASUB: 000276 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3200 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
DMASUB, DMASUB /-SP=C 20, 1 JIM, C 20, 1 JIM, DMASUB

```

1
2 000000      .TITLE  HQRREF-
3              .PSECT  HQRREF-
4              ;
5              ;
6              ;   HARDWARE QUERY RESOLVER TRANSLATOR
7              ;   INTERFACE SUBROUTINES TO LOAD FRAMES 2 AND 3
8              ;
9              ;   ENTRY POINTS
10             ;
11             ;   PPHALT - HALT PP
12             ;   SDHALT - HALT SD
13             ;   QLBREF - WRITE WORD TO QR
14             ;   SLBREF - WRITE WORD TO SD REFERENCE PAGE
15             ;   ONEFA - WRITE WORD TO FP
16             ;   STOP - LOAD QCL POINTER
17             ;   ONESR - LOAD SUBREAD MEMORY WORD
18             ;
19             ;
20             ;   PPHALT:
21             ;   MOV  #0$OHLT, -(SP)      ; SEND HALT CODE TO PPS
22             ;   CALL PPCR
23             ;   MOV  #0$CLR, -(SP)      ; CLEAR PPS
24             ;   CALL PPCR
25             ;   SDHALT:
26             ;   RETURN

```

```

27      :
28      :
29      :      WRITE A WORD TO THE HQR QLB REFERENCE PAGE
30      :
31      :
32      :      QLBREF::
33      000022 012757 000004 000000 MOV      #BIT2,BITPOS      ;RE-INIT-BIT-POSITION
34      000030 012746 000053      MOV      #0QLA,-(SP)      ;DDR-SELECT-FOR-QLB
35      000034      CALL     PPCR          ;SEND-TO-PP-COUNT-DL-REG
36      000040 016746 000000G      MOV      CHQA,-(SP)      ;QLB-ADDRESS
37      000044      CALL     LBPP          ;SEND-TO-PPS
38      000050 012746 000001      MOV      #0QLR,-(SP)      ;SELECT-REFERENCE-PAGE
39      000054      CALL     PPCR          ;
40      000060 016746 000000G      MOV      W0W,-(SP)      ;DATA-FOR-QLB-REF-PAGE
41      000064      CALL     LBPP          ;SEND-TO-PPS
42      :
43      000070 005067 000000G      CLR      W0W          ;CLEAR-WORKING-WORD
44      000074 005267 000000G      INC      CHQA         ;BUMP-ADDRESS
45      000100      RETURN
46      :

```



```
48 ;
49 ; WRITE A WORD TO THE SP REFERENCE PAGE.
50 ;
51 000102. SLBREF::
52 000102. 012767 000001 000000G. MOV. #BIT0.BITPOS. ;REINITIALIZE BIT POSITION.
53 000110 012746 000001 MOV. #S$LA,-(SP) ;ADDR SELECT FOR QLB
54 000114 CALL. SPCR ;SEND TO SP CONTROL REG.
55 000120 016746 000000G. MOV. CHQA,-(SP) ;QLB ADDRESS.
56 000124 CALL. LBSP ;SEND TO SP.
57 000130 012746 000006 MOV. #S$QR,-(SP) ;SELECT REFERENCE PAGE.
58 000134 CALL. SPCR
59 000140 016746 000000G. MOV. WQW,-(SP) ;DATA FOR QLB REF PAGE.
60 000144 CALL. LBSP ;SEND TO SP.
61 ;
62 000150 005067 000000G. CLR. WQW. ;CLEAR WORKING WORD.
63 000154 005267 000000G. INC. CHQA ;BUMP ADDRESS.
64 000160 RETURN.
65 ;
```

```
67 ;
68 ; LOAD ONE WORD OF FAL POINTER MEMORY.
69 ;
70 ;
71 000162. ONEFA:
72 000162. 016746 000000G. MOV. ADDR.-(SP) ;LOAD ADDR INTO OCL POINTER.
73 000166. CALL. STOP
74 000172. 012746 000046. MOV. #0$FP.-(SP) ;SELECT MEMORY.
75 000176. CALL. PPCR ;WRITE SELECTION TO CONTROL REG.
76 000202. 016746 000000G. MOV. DATA.-(SP) ;SEND DATA WORD TO FAL MEMORY.
77 000206. CALL. LBPP
78 000212. 012746 000040. MOV. #0$CLR.-(SP)
79 000216. CALL. PPCR
80 000222. RETURN.
81 ;
```

```
83      ;
84      ;      LOAD ONE WORD OF SUBREAD MEMORY.
85      ;
86 000224      ;      ONESR::
87 000224 012746 000001      MOV.   #S$LA, -(SP)      ; ADDR SELECT FOR SP.
88 000230      CALL.  SPCR      ; SEND TO SP CONTROL REG.
89 000234 016746 000000G     MOV.   ADDR, -(SP)      ; QLB ADDRESS
90 000240      CALL.  LBSP      ; SEND TO SP.
91 000244 012746 000007     MOV.   #S$SR, -(SP)     ; SELECT SUBREAD MEMORY.
92 000250      CALL.  SPCR
93 000254 016746 000000G     MOV.   DATA, -(SP)     ; DATA FOR QLB REF PAGE.
94 000260      CALL.  LBSP      ; SEND TO SP.
95
96 000264      ;      RETURN.
```

```

98      ;
99      ;      LOAD QCL POINTER
100     ;
101     ;
102-000266      ;      STOP::
103 000266 016667 000002 176424      MOV.    2(SP),QR#LBR.      ;MOVE POINTER WORD TO LOD BUS REG.
104 000274 012746 001001      MOV.    *(<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
105 000300 052716 000360      BIS.    #Q#CSEL,(SP)      ;CLEAR SELECTION BITS
106 000304 012746 176000      MOV.    #Q#NCLK,-(SP)     ;SET NO-CLOCKS
107 000310      CALL.   CSR1
108     ;
109 000314 005046      CLR.    -(SP)             ;CLEAR NOTHING
110 000316 012746 001300      MOV.    *(<Q#PP2+Q#LBD>,-(SP) ;SELECT PPS AND SET DRIVE
111 000322      CALL.   CSR1
112     ;
113     ;      SET FAL LOAD
114     ;
115 000326 012767 004000 176422      MOV.    #Q#FAL,QR#CR2.    ;SET FAL LOAD
116     ;
117     ;      EXTRA CLOCK FOR PPS
118     ;
119 000334 012746 000001      MOV.    #Q#LBP,-(SP)     ;CLEAR PULSE
120 000340 052716 006000      BIS.    #Q#RNC,(SP)      ;CLEAR PPS NO-CLOCK
121 000344 005046      CLR.    -(SP)            ;SET NOTHING
122 000346      CALL.   CSR1
123     ;
124     ;      TURN OFF FAL LOAD
125     ;
126 000352 005067 176422      CLR.    QR#CR2.
127     ;
128     ;      DE-SELECTION
129     ;
130 000356 012746 001001      MOV.    *(<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
131 000362 052716 000360      BIS.    #Q#CSEL,(SP)      ;CLEAR SELECTION BITS
132 000366 012746 176000      MOV.    #Q#NCLK,-(SP)     ;SET NO-CLOCKS
133 000372      CALL.   CSR1
134     ;
135 000376 011666 000002      MOV.    (SP),2(SP)        ;MOVE RETURN ADDRESS DOWN STACK
136 000402 005726      TST.   (SP)+             ;POINT TO RETURN ADDRESS
137 000404      RETURN.
138     ;
139     000001      .END

```

ADDR. = ***** GX.
ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CHQA = ***** GX.
CNOBRE = 100000
CPCCEN = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSECCI = 100000
CSOE = 000040
CSR1 = ***** GX.
CSWRTE = 000100
DATA = ***** GX.
DBR.RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DNAAUR = 000005
DNARRD = 000003
DNARUR = 000004
ENBR = 010000
LBSP = ***** GX.
LBP = ***** GX.
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARL0D = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADR1 = 000100
MMLFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
ONEFA = 000162RG.
ONESR = 00024RG.
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
PPCR = ***** GX.
PPHALT = 000000RG.
QLBREF = 000022RG.
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000045
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LCD = 000003
Q\$LMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MNC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QJ = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGD.EN = 000200
RGD.VA = 020000
SDHALT = 000020RG.
SEQ.CI = 000010
SLBREF = 000102RG.
SPCR = ***** GX.
STOP = 000266RG.
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$OR = 000006
S\$OX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$DAR = 176344
TD\$OTR = 176345
TD\$ORD = 000274
TD\$SW = 176376
TD\$STAR = 176372
TD\$TAU = 176362
TD\$TDR = 176374
TD\$TDW = 176364
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTB = 000000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DRD = 000004
T\$MEMH = 010000
T\$FSAR = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB = 000020
T\$IBAR = 000024
T\$IBE = 020000

T#IBF = .040000	WORD12 = .000030	WORD35 = .000106	WORD58 = .000164	WORD80 = .000240
T#ICD = .000040	WORD13 = .000032	WORD36 = .000110	WORD59 = .000166	WORD81 = .000242
T#MODE = .004000	WORD14 = .000034	WORD37 = .000112	WORD6 = .000014	WORD82 = .000244
T#OB = .000036	WORD15 = .000036	WORD38 = .000114	WORD60 = .000170	WORD83 = .000246
T#OBE = .004000	WORD16 = .000040	WORD39 = .000116	WORD61 = .000172	WORD84 = .000250
T#DBF = .010000	WORD17 = .000042	WORD4 = .000010	WORD62 = .000174	WORD85 = .000252
T#OBRA = .000034	WORD18 = .000044	WORD40 = .000120	WORD63 = .000176	WORD86 = .000254
T#OBWA = .000032	WORD19 = .000046	WORD41 = .000122	WORD64 = .000200	WORD87 = .000256
T#OUTA = .100000	WORD2 = .000004	WORD42 = .000124	WORD65 = .000202	WORD88 = .000260
T#RBDU = .000200	WORD20 = .000050	WORD43 = .000126	WORD66 = .000204	WORD89 = .000262
T#RNB = .000040	WORD21 = .000052	WORD44 = .000130	WORD67 = .000206	WORD9 = .000022
T#RSET = .040000	WORD22 = .000054	WORD45 = .000132	WORD68 = .000210	WORD90 = .000264
T#SC = .000022	WORD23 = .000056	WORD46 = .000134	WORD69 = .000212	WORD91 = .000266
T#SCLK = .020000	WORD24 = .000060	WORD47 = .000136	WORD7 = .000016	WORD92 = .000270
T#SEG1 = .000000	WORD25 = .000062	WORD48 = .000140	WORD70 = .000214	WORD93 = .000272
T#SEG2 = .000001	WORD26 = .000064	WORD49 = .000142	WORD71 = .000216	WORD94 = .000274
T#SEG3 = .000002	WORD27 = .000066	WORD5 = .000012	WORD72 = .000220	WORD95 = .000276
T#SO = .000001	WORD28 = .000070	WORD50 = .000144	WORD73 = .000222	WORD96 = .000300
T#UBUS = .100000	WORD29 = .000072	WORD51 = .000146	WORD74 = .000224	WORD97 = .000302
T#1CLK = .000400	WORD3 = .000006	WORD52 = .000150	WORD75 = .000226	WORD98 = .000304
T#BBEN = .000020	WORD30 = .000074	WORD53 = .000152	WORD76 = .000230	WORD99 = .000306
UBD, IN = .000020	WORD31 = .000076	WORD54 = .000154	WORD77 = .000232	WDJ = .*****GX
WORD0 = .000000	WORD32 = .000100	WORD55 = .000156	WORD78 = .000234	WORDVAL = .000310
WORD1 = .000002	WORD33 = .000102	WORD56 = .000160	WORD79 = .000236	XTREAD = .001000
WORD10 = .000024	WORD34 = .000104	WORD57 = .000162	WORD8 = .000020	XTWRITE = .000400
WORD11 = .000026				

. ABS. 000000 000
000000 001
HQRREF 000406 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3061 WORDS (12 PAGES)
DYNAMIC MEMORY: 3060 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
HQRREF, HQRREF / - SP = [20, 1] JIM, [20, 1] HQRREF

```

1          .TITLE- MICRO-
2 000000   .PSECT- MICRO-
3          :
4          :
5          :   HARDWARE-QUERY-RESOLVER-TRANSLATOR-
6          :   LOAD-MRP-AND-CP-MICROCODE-
7          :
8          :   CALLED-FROM-LMAIN-VIA-JSR-
9          :   RETURN-VIA-RTS-
10         :
11         :
12         :   .MCALL- OPEN$,CLOSE$,FINIT$
13         :
14 000000   MICRO::
15 000000   FINIT$
16 000004   CALL-   MRPMM-           ;LOAD-MRP-MICROCODE-
17 000010 042767 000000G-000000G-   ;CLEAR-FIRST-TIME-THROUGH-FLAG
18 000016 012767 000001 000002G-   ;RE-INIT-BLOCK-COUNT
19 000024   CALL-   CPCS           ;LOAD-CP-MICROCODE-
20 000030   RETURN-

```

```

22. ;
23. ; MRP MICROPROGRAM MEMORY.
24. ;
25. ;
26. ; FILL IN FILE NAME BLOCK FOR LDMM.DAT.
27. ; OPEN FILE.
28. 000032. MRPMM:
29. 000032. 016767 000000G 000000C. MOV. LMM,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
30. 000040. 016767 000002G 000000C. MOV. LMM+2,INDNB+N.FNAM+2.
31. 000046. OPEN$R. #INFDB.
32. ;
33. ; SET MRP SEQUENCER TO ZERO.
34. ; GET FIRST RECORD. THE FIRST WORD OF THE FIRST RECORD
35. ; CONTAINS THE NUMBER OF WORDS TO BE LOADED INTO AN
36. ; MRP COLUMN (SEE BELOW). SAVE THIS VALUE.
37. ;
38. 000064. 005067 000000G. CLR. MSTR2. ;INIT SEQUENCER = 0
39. 000070. 1$. CALL. GET. ;READ A RECORD.
40. 000074. 103511 BCS. MRPMX. ;ERROR, EXIT.
41. 000076. 016705 000000C. MOV. INFDB+F.BKDS+2,R5 ;POINT TO RECORD READ.
42. 000102. 012704 000400 MOV. #256,,R4 ;NUMBER OF WORDS IN RECORD (MAX)
43. 000106. 032767 000000G 000000G. BIT. #FIRST.SELECT. ;FIRST TIME THROUGH.
44. 000114. 001011 BNE. 2$. ;NO.
45. 000116. 052767 000000G 000000G. BIS. #FIRST.SELECT. ;SET FLAG FOR FIRST TIME THROUGH.
46. 000124. 012567 000000G. MOV. (R5)+,LCOUNT. ;SAVE NUMBER OF WORDS IN COLUMN.
47. 000130. 016767 000000G 000000G. MOV. LCOUNT,WCOUNT. ;INITIALIZE WORKING COUNTER.
48. 000136. 005304 DEC. R4 ;SUB FROM NUMBER OF WORDS IN RECORD.
49. ;
50. ; EACH LOCATION IN MRP MICROPROGRAM MEMORY CONSISTS OF TWO
51. ; WORDS, A LEFT WORD AND A RIGHT WORDS. IN LOADING, ALL LEFT
52. ; WORDS ARE LOADED FIRST (IE, A COLUMN) THEN ALL RIGHT WORDS.
53. ; THE PROGRAM 'CONVRT' HAS WRITTEN THE FILE LDMM.DAT TO CONTAIN
54. ; MRP MICROPGM MEMORY DATA IN COLUMNS.
55. ;
56. ; FILL THE LEFT COLUMN OF MRP MICROPGM MEMORY.
57. ;
58. 000140. 2$:
59. 000140. 016746 000000G. MOV. MSTR2,-(SP) ;INITIALIZE SEQUENCER ADDRESS.
60. 000144. SEQMM. ;SET ADDRESS.
61. 000150. 012746 000012. MOV. #(<HMMWRTEN+MMLEFT>,-(SP)
62. 000154. CALL. MRPCRA. ;DIRECT CONTROL WORD TO MRP.
63. 000160. 012546 MOV. (R5)+,-(SP) ;GET READY TO MOVE DATA TO MRP.
64. 000162. CALL. LBMSC. ;DO IT.
65. 000166. 005046 CLR. -(SP) ;CLEAR CONTROL REG.
66. 000170. CALL. MRPCR.
67. ;
68. 000174. 005367 000000G. DEC. WCOUNT. ;SUB FROM # WORDS IN A COLUMN.
69. 000200. 001405 BEQ. MRPRGT. ;DO RIGHT HAND COLUMN.
70. 000202. 005267 000000G. INC. MSTR2. ;ADVANCE SEQUENCER ADDRESS.
71. 000206. 005304 DEC. R4 ;FINISHED WITH THIS RECORD.
72. 000210. 001727 BEQ. 1$. ;YES, GET NEXT.
73. 000212. 000752. BR. 2$. ;NO, LOAD NEXT WORD.
74. ;
75. ; FILL THE RIGHT COLUMN OF MRP MICROPGM MEMORY.
76. ;
77. 000214. MRPRGT:
78. 000214. 016767 000000G 000000G. MOV. LCOUNT,WCOUNT. ;REINITIALIZE WORKING COUNTER.

```



```

79 000222 005067 000000G.
80 000226 005304
81 000230 001007
82 000232
83 000236 103430
84 000240 016705 000000C.
85 000244 012704 000400
86
87 000250
88 000250 016746 000000G.
89 000254
90 000260 012746 000010
91 000264
92 000270 012546
93 000272
94 000276 005046
95 000300
96
97 000304 005367 000000G.
98 000310 001403
99 000312 005267 000000G.
100 000316 000743
101
102 000320
103 000320
104 000330 105067 000000C.
105 000334

1$: CLR MSTR2 : INIT SEQUENCER = 0
DEC R4 : FINISHED WITH THIS RECORD
BNE 2$ : NO, CONTINUE
CALL GET : READ NEXT
BCS MRPXM : ERROR, EXIT
MOV INFDB+F,BKDS+2,R5 : POINT TO RECORD READ
MOV #256,R4 : R4 = NUMBER OF WORDS IN RECORD
;
2$: MOV MSTR2,-(SP) : INITIALIZE SEQUENCER ADDRESS
CALL SEQMM : SET ADDRESS
MOV #<MMURTEN>,-(SP)
CALL MRPCRA : DIRECT CONTROL WORD TO MRP
MOV (R5)+,-(SP) : GET READY TO MOVE DATA TO MRP
CALL LBMSC : DO IT
CLR -(SP) : CLEAR CONTROL REG
CALL MRPCR
;
DEC WCOUNT : FINISHED WITH THIS COLUMN
BEQ MRPXM : YES, DONE
INC MSTR2 : NO, ADVANCE SEQUENCER ADDRESS
BR 1$ : SET IT
MRPXM:
CLOSE$ #INFDB :
CLRB INDNB+N,FVER : RESET FILE VERSION NUMBER
RETURN

```

```

107      ;
108      ;      CP CONTROL STORE SUBROUTINE.
109      ;
110      ;
111      000336      CPCS:
112      ;      FILL IN FILE NAME BLOCK FOR LDCS.DAT.
113      ;      OPEN FILE.
114      ;
115      000336      016767      000000G-000000C      MOV      LCS,INDNB+N.FNAM      ;PLACE FILE NAME INTO INPUT DNB.
116      000344      016767      000002G-000000C      MOV      LCS+2,INDNB+N.FNAM+2
117      000352      ;      OPEN#R      #INFDB.
118      ;
119      ;      RESET CP, SET CP SEQUENCER TO ZERO.
120      ;      GET FIRST RECORD, THE FIRST WORD OF THE FIRST RECORD
121      ;      CONTAINS THE NUMBER OF WORDS TO BE LOADED INTO AN
122      ;      CP COLUMN (SEE BELOW). SAVE THIS VALUE.
123      ;
124      000370      005046      ;      CLR      -(SP)      ;CLEAR NOTHING.
125      000372      012746      000002      MOV      #0*CSSET,-(SP)      ;CP RESET.
126      000376      ;      CALL     CSR1
127      000402      012746      000002      MOV      #0*CSSET,-(SP)      ;CLEAR RESET.
128      000406      005046      ;      CLR      -(SP)      ;SET NOTHING.
129      000410      ;      CALL     CSR1
130      ;
131      000414      005067      000000G      CLR      MSTR2      ;SET SEQUENCER TO ZERO.
132      000420      15$      CALL     GET      ;READ A RECORD.
133      000424      103002      BCC     15$      ;BRANCH IF OK.
134      000426      000167      000432      JMP      CPCSX      ;ERROR, EXIT.
135      000432      15$:
136      000432      016705      000000C      MOV      INFDB+F,BKDS+2,R5      ;POINT TO RECORD READ.
137      000436      012704      000400      MOV      #256,R4      ;NUMBER OF WORDS IN RECORD (MAX)
138      000442      032767      000000G-000000G      BIT      #FIRST,SELECT      ;FIRST TIME THROUGH.
139      000450      001011      BNE     2$      ;NO.
140      000452      052767      000000G-000000G      BIS      #FIRST,SELECT      ;SET FLAG FOR FIRST TIME THROUGH.
141      000460      012567      000000G      MOV      (R5)+,LCOUNT      ;SAVE NUMBER OF WORDS IN COLUMN.
142      000464      016767      000000G-000000G      MOV      LCOUNT,WCOUNT      ;INITIALIZE WORKING COUNTER.
143      000472      005304      DEC     R4      ;SUB FROM NUMBER OF WORDS IN RECORD.
144      ;
145      ;      EACH LOCATION IN CP CONTROL STORE CONSISTS OF FOUR WORDS.
146      ;      SECTION 'A', SECTION 'B', SECTION 'C', SECTION 'D', IN
147      ;      WORDS ARE LOADED FIRST (IE, A COLUMN) THEN ALL RIGHT WORDS.
148      ;      LOADING, ALL OF SECTION 'A' IS LOADED FIRST, THEN SECTION
149      ;      'B', 'C', 'D'. THE PROGRAM 'CONVRT' HAS WRITTEN THE FILE
150      ;      LDCS.DAT TO CONTAIN CP CONTROL STORE IN COLUMNS.
151      ;
152      ;      FILL SECTION 'A' OF CP CONTROL STORE.
153      ;
154      000474      2$:
155      000474      016746      000000G      MOV      MSTR2,-(SP)      ;INITIALIZE SEQUENCER ADDRESS.
156      000500      ;      CALL     SEQCS      ;SET ADDRESS.
157      000504      012746      000100      MOV      #CSWRTEN,-(SP)      ;SELECT SECTION A.
158      000510      ;      CALL     CPCRA      ;DIRECT CONTROL WORD TO CP.
159      000514      012546      MOV      (R5)+,-(SP)      ;GET READY TO MOVE DATA TO MRP.
160      000516      ;      CALL     LBCSC      ;DO IT.
161      000522      005046      CLR      -(SP)      ;CLEAR CP CONTROL REG.
162      000524      ;      CALL     CPCR
163

```

```

164 000530 005367 000000G..... DEC.....WCOUNT..... :SUB FROM # WORDS IN A COLUMN.
165 000534 001405 BEQ CPB :DO SECT B
166 000536 005267 000000G. INC MSTR2 :ADVANCE SEQUENCER ADDRESS.
167 000542 005304 DEC R4 :FINISHED WITH THIS RECORD.
168 000544 001725 BEQ 1$ :YES, GET NEXT.
169 000546 000752 BR 2$ :NO, RESET SEQUENCER ADDRESS.
170 :
171 :
172 :
173 000550 :
174 000550 016767 000000G.000000G. CPB: MOV LCOUNT,WCOUNT :REINIT WORKING COUNTER.
175 000556 005067 000000G. CLR MSTR2 :SET SEQUENCER TO ZERO.
176 000562 005304 1$: DEC R4 :FINISHED WITH THIS RECORD.
177 000564 001007 BNE 2$ :NO, CONTINUE.
178 000566 CALL GET :READ NEXT.
179 000572 103534 BCS CPCSX :ERROR, EXIT.
180 :
181 000574 016705 000000C. MOV INFDB+F,BKDS+2,R5 :POINT TO RECORD READ.
182 000600 012704 000400 MOV #256.,R4 :R4 = NUMBER OF WORDS IN RECORD.
183 :
184 000604 :
185 000604 016746 000000G. 2$: MOV MSTR2,-(SP) :INITIALIZE SEQUENCER ADDRESS.
186 000610 CALL SEQCS :SET ADDRESS.
187 000614 012746 000110 MOV #<CSURTEN+PLB>,-(SP) :SELECT SECTION B.
188 000620 CALL CPCRA :DIRECT CONTROL WORD TO CP.
189 000624 012546 MOV (R5)+,-(SP) :GET READY TO MOVE DATA TO MRP.
190 000626 CALL LBCSC :DO IT.
191 000632 005046 CLR -(SP) :CLEAR CP CONTROL REG.
192 000634 CALL CPCR
193 :
194 000640 005367 000000G. DEC WCOUNT :FINISHED WITH THIS COLUMN.
195 000644 001403 BEQ CPC :YES, GET NEXT.
196 000646 005267 000000G. INC MSTR2 :NO, ADVANCE SEQUENCER ADDRESS.
197 000652 000743 BR 1$ :SET IT.
198 :
199 :
200 :
201 000654 :
202 000654 016767 000000G.000000G. CPC: MOV LCOUNT,WCOUNT :REINITIALIZE WORKING COUNTER.
203 000662 005067 000000G. CLR MSTR2 :INIT SEQUENCER = 0.
204 000666 005304 1$: DEC R4 :FINISHED WITH THIS RECORD.
205 000670 001007 BNE 2$ :NO, CONTINUE.
206 000672 CALL GET :READ NEXT.
207 000676 103472 BCS CPCSX :ERROR, EXIT.
208 000700 016705 000000C. MOV INFDB+F,BKDS+2,R5 :POINT TO RECORD READ.
209 000704 012704 000400 MOV #256.,R4 :R4 = NUMBER OF WORDS IN RECORD.
210 :
211 000710 :
212 000710 016746 000000G. 2$: MOV MSTR2,-(SP) :INITIALIZE SEQUENCER ADDRESS.
213 000714 CALL SEQCS :SET ADDRESS.
214 000720 012746 000120 MOV #<CSURTEN+PLC>,-(SP) :SELECT SECTION C.
215 000724 CALL CPCRA :DIRECT CONTROL WORD TO CP.
216 000730 012546 MOV (R5)+,-(SP) :GET READY TO MOVE DATA TO MRP.
217 000732 CALL LBCSC :DO IT.
218 000736 005046 CLR -(SP) :CLEAR CP CONTROL REG.
219 000740 CALL CPCR
220 :

```

```

221 000744 005367 000000G. DEC WCOUNT. ;FINISHED WITH THIS COLUMN.
222 000750 001403 BEQ CPD. ;YES, GET NEXT.
223 000752 005267 000000G. INC MSTR2. ;NO, ADVANCE SEQUENCER ADDRESS
224 000756 000743 BR 1$. ;SET IT.
225 ;
226 ;
227 ;
228 000760 ;
229 000760 016767 000000G-000000G. CPD: MOV LCOUNT,WCOUNT. ;REINITIALIZE WORKING COUNTER.
230 000766 005067 000000G. CLR MSTR2. ;INIT SEQUENCER = 0
231 000772 005304 1$: DEC R4 ;FINISHED WITH THIS RECORD.
232 000774 001007 BNE 2$. ;NO, CONTINUE.
233 000776 CALL GET. ;READ NEXT
234 001002 103430 BCS CPCSX. ;ERROR, EXIT.
235 001004 016705 000000C. MOV INFDB+F.BKDS+2,R5 ;POINT TO RECORD READ.
236 001010 012704 000400 MOV #256,,R4 ;R4 = NUMBER OF WORDS IN RECORD.
237 ;
238 001014 2$:
239 001014 016746 000000G. MOV MSTR2,-(SP) ;INITIALIZE SEQUENCER ADDRESS.
240 001020 CALL SEQCS ;SET ADDRESS.
241 001024 012746 000130 MOV #<CSWRTEN+PLD>,-(SP) ;SELECT SECTION D.
242 001030 CALL CPCRA. ;DIRECT CONTROL WORD TO CP.
243 001034 012546 MOV (R5)+,-(SP). ;GET READY TO MOVE DATA TO MRP
244 001036 CALL LBCSC. ;DO IT.
245 001042 005046 CLR -(SP) ;CLEAR CP CONTROL REG.
246 001044 CALL CPCR
247 ;
248 001050 005367 000000G. DEC WCOUNT. ;FINISHED WITH THIS COLUMN.
249 001054 001403 BEQ CPCSX. ;YES, ALL DONE.
250 001056 005267 000000G. INC MSTR2. ;NO, ADVANCE SEQUENCER ADDRESS
251 001062 000743 BR 1$. ;SET IT.
252 ;
253 001064 CPCSX:
254 001064 CLR #INFDB.
255 001074 105067 000000C. CLR INDNB+N.FVER. ;RESET FILE VERSION NUMBER.
256 001100 RETURN.
257 001102 RETURN.
258 ;
259 000001 .END.

```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CNOBRE = 100000
CPB = 000550R
CPC = 000654R
CPCCEN = 010000
CPCR = * * * * * GX
CPCRA = * * * * * GX
CPCS = 000336RG
CPCSX = 001064R
CPD = 000760R
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = * * * * * GX
CSURTE = 000100
DBR.RD = 000001
DB#CPR = 001457
DB#SPT = 000026
DB#TPC = 000023
DISPGS = 100000
DMAUR = 000005
DMARRD = 000003
DMARUR = 000004
ENBR = 010000
FIRST = * * * * * GX
FO.RD = * * * * * GX
F.BKDS = * * * * * GX
F.FACC = * * * * * GX
GET = * * * * * GX
INDNB = * * * * * GX
INFDN = * * * * * GX
LBCSC = * * * * * GX
LBMS = * * * * * GX
LCOUNT = * * * * * GX
LCS = * * * * * GX
LHM = * * * * * GX
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MICRO = 000000RG
MMADDR = 000100
MMLEFT = 000002
MMOE = 000004
MMURTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = * * * * * GX
MRPCRA = * * * * * GX
MRPMM = 000032RG
MRPMX = 000320R
MRPRGT = 000214R
MSTR2 = * * * * * GX
MSYN = 000040
002.N = 000144
002.N.FNAM = * * * * * GX
002.N.FVER = * * * * * GX
PAR### = 000027
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
QR#CR1 = 176420
QR#CR2 = 176422
QR#LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPCC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
002.Q\$NCLK = 176000
002.Q\$PP = 000100
002.Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$PHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGD.EN = 000200
RGD.VA = 020000
SELECT = * * * * * GX
SEQCS = * * * * * GX
SEGNM = * * * * * GX
SEQ.CI = 000010
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$QR = 000006
S\$QX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD#CTR = 176370
TD#CTW = 176360
TD#ENM = 004000
TD#NAM = 000270
TD#OAR = 176344
TD#QTR = 176346
TD#ORD = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDW = 176364
T\$AD = 000020
T\$BA = 000002

T\$BD = 000010	T\$SCLK = 020000	WORD26 = 000064	WORDS2 = 000150	WORD79 = 000236
T\$BS0 = 100000	T\$SEG1 = 000000	WORD27 = 000066	WORDS3 = 000152	WORD8 = 000020
T\$BT = 000020	T\$SEG2 = 000001	WORD28 = 000070	WORDS4 = 000154	WORD80 = 000240
T\$BTAR = 000030	T\$SEG3 = 000002	WORD29 = 000072	WORDS5 = 000156	WORD81 = 000242
T\$BTD = 002000	T\$SO = 000001	WORD3 = 000006	WORDS6 = 000160	WORD82 = 000244
T\$CD = 000100	T\$UBUS = 100000	WORD30 = 000074	WORDS7 = 000162	WORD83 = 000246
T\$CLK = 002000	T\$ICLK = 000400	WORD31 = 000076	WORDS8 = 000164	WORD84 = 000250
T\$DISK = 000200	T\$EBEN = 000020	WORD32 = 000100	WORDS9 = 000166	WORD85 = 000252
T\$DRD = 000004	UBD.IN = 000020	WORD33 = 000102	WORD6 = 000014	WORD86 = 000254
T\$EMEM = 010000	VIRT = ***** GX	WORD34 = 000104	WORD60 = 000170	WORD87 = 000256
T\$FSAH = 000000	WCOUNT = ***** GX	WORD35 = 000106	WORD61 = 000172	WORD88 = 000260
T\$FSAB = 000004	WORD0 = 000000	WORD36 = 000110	WORD62 = 000174	WORD89 = 000262
T\$FSAC = 000014	WORD1 = 000002	WORD37 = 000112	WORD63 = 000176	WORD9 = 000022
T\$FSB2 = 000010	WORD10 = 000024	WORD38 = 000114	WORD64 = 000200	WORD90 = 000264
T\$IB = 000026	WORD11 = 000026	WORD39 = 000116	WORD65 = 000202	WORD91 = 000266
T\$IBAR = 000024	WORD12 = 000030	WORD4 = 000010	WORD66 = 000204	WORD92 = 000270
T\$IBE = 020000	WORD13 = 000032	WORD40 = 000120	WORD67 = 000206	WORD93 = 000272
T\$IBF = 040000	WORD14 = 000034	WORD41 = 000122	WORD68 = 000210	WORD94 = 000274
T\$ICD = 000040	WORD15 = 000036	WORD42 = 000124	WORD69 = 000212	WORD95 = 000276
T\$MODE = 004000	WORD16 = 000040	WORD43 = 000126	WORD7 = 000016	WORD96 = 000300
T\$OB = 000036	WORD17 = 000042	WORD44 = 000130	WORD70 = 000214	WORD97 = 000302
T\$OBE = 004000	WORD18 = 000044	WORD45 = 000132	WORD71 = 000216	WORD98 = 000304
T\$OBF = 010000	WORD19 = 000046	WORD46 = 000134	WORD72 = 000220	WORD99 = 000306
T\$OBRA = 000034	WORD2 = 000004	WORD47 = 000136	WORD73 = 000222	WORDVAL = 000310
T\$OBWA = 000032	WORD20 = 000050	WORD48 = 000140	WORD74 = 000224	XTREAD = 001000
T\$OUTA = 100000	WORD21 = 000052	WORD49 = 000142	WORD75 = 000226	XTWRITE = 000400
T\$RBD0 = 000200	WORD22 = 000054	WORD5 = 000012	WORD76 = 000230	.CLOSE = ***** G
T\$RNB = 000040	WORD23 = 000056	WORD50 = 000144	WORD77 = 000232	.FINIT = ***** G
T\$RSET = 040000	WORD24 = 000060	WORD51 = 000146	WORD78 = 000234	.OPEN = ***** G
T\$SC = 000022	WORD25 = 000062			

. ABS. 000000 000
000000 001
MICRO: 001104 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4015 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:51
MICRO MICRO--SP=C20.1JIM.C20.1MICRO

```

1
2 000000          .TITLE  FRAME2.
3                    .PSECT  FRAME2.
4
5
6                    ;
7                    ;
8                    ;
9                    ;
10                   ;
11                   ;
12                   ;
13                   ;
14                   ;
15                   ;
16                   ;
17                   ;
18 000000          FRAME2:
19 000000  012746  000052      MOV.   #Q$MR,-(SP)          ;SELECT MASK WORD.
20 000004                          CALL.  PPCR
21 000010  012746  037774      MOV.   #37774,-(SP)        ;SET MASK WORD.
22 000014                          CALL.  LBPSC
23 000020  012746  000040      MOV.   #Q$CLR,-(SP)        ;CLEAR PPS
24 000024                          CALL.  PPCR
25
26
27                   ;
28                   ;
29                   ;
30 000030  012746  001001      MOV.   #<Q$LBD+Q$LBP>,-(SP)  ;CLEAR DRIVE AND PULSE.
31 000034  052716  000360      BIS.   #Q$CSEL,(SP)         ;CLEAR SELECTION BITS.
32 000040  012746  176000      MOV.   #Q$NCLK,-(SP)       ;SET NO-CLOCKS.
33 000044                          CALL.  CSR1
34
35                   ;
36                   ;
37 000050  012746  006000      MOV.   #Q$RNC,-(SP)         ;CLEAR PPS NO-CLOCKS (START PPS)
38 000054  005046                          CLR.   -(SP)               ;SET NOTHING.
39 000056                          CALL.  CSR1
40
41                   ;
42                   ;
43                   ;
44 000062  012746  000025      MOV.   #025,-(SP)           ;SET PPCR = X'15'
45 000066                          CALL.  PPCR
46
47                   ;
48                   ;
49                   ;
50 000072  005067  176424      CLR.   QR$LBR
51 000076  005046                          CLR.   -(SP)               ;CLEAR NOTHING
52 000100  012746  001300      MOV.   #<Q$PP2+Q$LBD>,-(SP)  ;SELECT PPS AND SET DRIVE
53 000104                          CALL.  CSR1
54
55                   ;
56                   ;
57 000110  012767  004000  176422  MOV.   #0$FAL,0R$CR2
    
```

```

58 000116 012767 077777 176424 MOV. #077777,QR#LBR ;WRITE "FAL-PARK" TO LOD-BUS-REG.
59 000124 005067 176422 CLR. QR#CR2. ;CLEAR.
60 ;
61 ; WAIT. 2. MS. FOR. FAL. TO. FINISH.
62 ; ADD. = 3. 17US
63 ; DEC. = 2. 65US
64 ; BNE. = 1. 87US
65 ;
66 000130 012701 000454 MOV. #300.,R1 ;EXECUTE 300 TIMES.
67 000134 005000 CLR. R0 ;ADD. NOTHING.
68 000136 060000 1$: ADD. R0,R0
69 000140 005301 DEC. R1
70 000142 001375 BNE. 1$
71 ;
72 ; CLEAR. LOD. BUS. DRIVE, DESELECT. PPS.
73 ;
74 000144 012746 001300 MOV. #<0$PP2+0$LBD>,-(SP) ;CLEAR. DRIVE. AND. DESELECT.
75 000150 012746 176000 MOV. #0$NCLK,-(SP) ;SET. NO-CLOCKS.
76 000154 CALL. CSR1
77 ;
78 ; CLEAR. PP. CR.
79 ;
80 000160 005046 CLR. -(SP) ;CODE. FOR. CLEAR.
81 000162 CALL. PPCR

```



```

83      ;
84      ;
85      ;
86      ;
87      ;
88 000166 012746 001001      MOV.    #<0$LBD+0$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE
89 000172 052716 000360      BIS.    #0$CSEL,(SP)             ;CLEAR SELECTION BITS
90 000176 012746 176000      MOV.    #0$NCLK,-(SP)          ;SET NO-CLOCKS
91 000202      CALL.   CSR1
92      ;
93      ;
94      ;
95 000206 012746 006000      MOV.    #0$RNC,-(SP)           ;CLEAR PPS NO-CLOCKS (START PPS)
96 000212 005046      CLR.    -(SP)                  ;SET NOTHING
97 000214      CALL.   CSR1
98      ;
99      ;
100     ;
101 000220 012746 000141      MOV.    #141,-(SP)             ;SET PPCR = X'61'
102 000224      CALL.   PPCR
103     ;
104     ;
105     ;
106     ;
107 000230 005067 176424      CLR.    QR$LBR                 ;SET LBR = 0
108 000234 005046      CLR.    -(SP)                 ;CLEAR NOTHING
109 000236 012746 001300      MOV.    #<0$PP2+0$LBD>,-(SP)   ;SELECT PPS AND SET DRIVE
110 000242      CALL.   CSR1
111     ;
112     ;
113     ;
114 000246 012767 010000 176422 MOV.    #0$LHP,QR$CR2          ;START LHP
115 000254 005067 176422      CLR.    QR$CR2                ;CLEAR
116     ;
117     ;
118     ;
119     ;
120     ;
121     ;
122 000260 012701 000454      MOV.    #300,R1                ;EXECUTE 300 TIMES
123 000264 005000      CLR.    R0                    ;ADD NOTHING
124 000266 006000      ADD.    R0,R0
125 000270 005301      DEC.    R1
126 000272 001375      BNE.    2$
127     ;
128     ;
129     ;
130 000274 012746 001300      MOV.    #<0$PP2+0$LBD>,-(SP)   ;CLEAR DRIVE AND DESELECT
131 000300 012746 176000      MOV.    #0$NCLK,-(SP)          ;SET NO-CLOCKS
132 000304      CALL.   CSR1
133     ;
134     ;
135     ;
136 000310 012746 000040      MOV.    #0$CLR,-(SP)           ;CODE FOR CLEAR
137 000314      CALL.   PPCR
138     ;
139     ;

```

```

140      ;      CLEAR QLB MEMORIES
141      ;
142      ;
143 000320 005046      CLR      -(SP)      ; SELECT PAGE 0
144 000322      CALL    SELPG      ;
145 000326      CALL    QLBCLR     ; CLEAR QLB PAGE
146      ;
147 000332 012746 000001  MOV     #1, -(SP)      ; SELECT PAGE 1
148 000336      CALL    SELPG      ;
149 000342      CALL    QLBCLR     ; CLEAR QLB PAGE
150      ;
151 000346 012746 000002  MOV     #2, -(SP)      ; SELECT PAGE 2
152 000352      CALL    SELPG      ;
153 000356      CALL    QLBCLR     ; CLEAR QLB PAGE
154      ;
155 000362 012746 000013  MOV     #0$OHLT, -(SP) ; SEND HALT CODE TO PPS
156 000366      CALL    PPCR      ;
157 000372 012746 000040  MOV     #0$CLR, -(SP)  ; CLEAR PPS
158 000376      CALL    PPCR      ;
159      ;
160      ;
161      ;
162      ;      CLEAR FRAME3 MEMORIES
163 000402      FM3:
164 000402 012767 000377 000000G  MOV     #255, CHQA     ; SET REFERENCE PAGE HIGH ADDRESS
165 000410      CALL    SPCLR     ;
166      ;
167 000414 005067 000000G  CLR     DATA          ;
168 000420 012767 000377 000000G  MOV     #255, ADDR     ; SUBREAD HIGH ADDRESS
169 000426      ; 13$:
170 000426      CALL    ONESR     ; CLEAR SUBREAD WORD
171 000432 005367 000000G  DEC     ADDR           ; NEXT LOWER ADDRESS
172 000436 002373      BGE     10$            ; BRANCH IF MORE
173      ;
174 000440      RETURN

```

```

176 ;
177 ;
178 ; CLEAR QLB MEMORY
179 ;
180 ;
181 000442 QLBCLR:
182 000442 012746 001001 MOV #<0$LBD+0$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
183 000446 052716 000360 BIS #0$CSEL,(SP) ;CLEAR SELECTION BITS
184 000452 012746 176000 MOV #0$NCLK,-(SP) ;SET NO CLOCKS
185 000456 CALL CSR1
186 ;
187 ;
188 ; TURN ON PPS CLOCKS
189 000462 012746 006000 MOV #0$RNC,-(SP) ;CLEAR PPS NO CLOCKS (START PPS)
190 000466 005046 CLR -(SP) ;SET NOTHING
191 000470 CALL CSR1
192 ;
193 ;
194 ; SET PPS CONTROL REGISTER
195 000474 012746 000155 MOV #155,-(SP) ;SET PPCR = X'6D'
196 000500 CALL PPCR
197 ;
198 ;
199 ; SET LOD BUS DRIVE, SELECT PPS
200 000504 005046 CLR -(SP) ;CLEAR NOTHING
201 000506 012746 001300 MOV #<0$PP2+0$LBD>,-(SP) ;SELECT PPS AND SET DRIVE
202 000512 CALL CSR1
203 ;
204 ;
205 ; SET LHP START, THEN CLEAR
206 000516 012767 010000 176422 MOV #0$LHP,QR$CR2 ;START LHP
207 000524 005067 176422 CLR QR$CR2 ;CLEAR
208 ;
209 ;
210 ; WAIT 2 MS FOR LHP TO FINISH
211 ;
212 ; ADD = 3.17US
213 ;
214 ; DEC = 2.65US
215 ;
216 ; BNE = 1.87US
217 ;
218 ;
219 ;
220 ;
221 ;
222 000544 012701 000454 MOV #300,R1 ;EXECUTE 300 TIMES
223 000550 012746 176000 CLR R0 ;ADD NOTHING
224 000554 ADD R0,R0
225 000540 005301 DEC R1
226 000542 001375 BNE 1$
227 ;
228 ;
229 ; CLEAR LOD BUS DRIVE, DESELECT PPS
230 000544 012746 001300 MOV #<0$PP2+0$LBD>,-(SP) ;CLEAR DRIVE AND DESELECT
231 000550 012746 176000 MOV #0$NCLK,-(SP) ;SET NO CLOCKS
232 000554 CALL CSR1
233 ;
234 ;
235 ; CLEAR PP CR
236 000560 012746 000040 MOV #0$CLR,-(SP) ;CODE FOR CLEAR
237 000564 CALL PPCR
238 000570 RETURN

```

```

232.                                     : CLEAR SUBDOCUMENT REFERENCE PAGE MEMORY.
233.                                     :
234 000572.                               : SPCLR:
235 000572. 012746 000001                : MOV.   #S$LA, -(SP)           : SET ADDRESS SELECT.
236 000576.                               : CALL.  SPCR                   :
237 000602. 016746 000000G.              : MOV.   CHQA, -(SP)           : SET ADDRESS.
238 000606.                               : CALL.  LBSP                   :
239 000612.                               :
240 000612. 012746 000006                : 13$: MOV.   #S$OR, -(SP)       : SET MEMORY SELECT CODE.
241 000616.                               : CALL.  SPCR                   :
242 000622. 005046                        : CLR.   -(SP)                 : ZERO IS MEMORY VALUE.
243 000624.                               : CALL.  LBSP                   :
244 000630. 005367 000000G.              : DEC.   CHQA                   : NEXT LOWER ADDRESS.
245 000634. 002356                        : BGE.   SPCLR                  : ZERO IT.
246.                                     :
247 000636. 005067 000000G.              : CLR.   CHQA                   : REINITIALIZE.
248 000642.                               : RETURN.
249.                                     :
250.                                     :
251      000001                            : .END.

```


T\$MODE = 004000	WORD13 = 000032	WORD36 = 000110	WORD59 = 000166	WORD80 = 000240
T\$OB = 000036	WORD14 = 000034	WORD37 = 000112	WORD6 = 000014	WORD81 = 000242
T\$OBE = 004000	WORD15 = 000036	WORD38 = 000114	WORD60 = 000170	WORD82 = 000244
T\$OBF = 010000	WORD16 = 000040	WORD39 = 000116	WORD61 = 000172	WORD83 = 000246
T\$OBRA = 000034	WORD17 = 000042	WORD4 = 000010	WORD62 = 000174	WORD84 = 000250
T\$OBWA = 000032	WORD18 = 000044	WORD40 = 000120	WORD63 = 000176	WORD85 = 000252
T\$OUTA = 100000	WORD19 = 000046	WORD41 = 000122	WORD64 = 000200	WORD86 = 000254
T\$RBD0 = 000200	WORD2 = 000004	WORD42 = 000124	WORD65 = 000202	WORD87 = 000256
T\$RNB = 000040	WORD20 = 000050	WORD43 = 000126	WORD66 = 000204	WORD88 = 000260
T\$RSET = 040000	WORD21 = 000052	WORD44 = 000130	WORD67 = 000206	WORD89 = 000262
T\$SC = 000022	WORD22 = 000054	WORD45 = 000132	WORD68 = 000210	WORD9 = 000022
T\$SCLK = 020000	WORD23 = 000056	WORD46 = 000134	WORD69 = 000212	WORD90 = 000264
T\$SEG1 = 000000	WORD24 = 000060	WORD47 = 000136	WORD7 = 000016	WORD91 = 000266
T\$SEG2 = 000001	WORD25 = 000062	WORD48 = 000140	WORD70 = 000214	WORD92 = 000270
T\$SEG3 = 000002	WORD26 = 000064	WORD49 = 000142	WORD71 = 000216	WORD93 = 000272
T\$SO = 000001	WORD27 = 000066	WORD5 = 000012	WORD72 = 000220	WORD94 = 000274
T\$UBUS = 100000	WORD28 = 000070	WORD50 = 000144	WORD73 = 000222	WORD95 = 000276
T\$1CLK = 000400	WORD29 = 000072	WORD51 = 000146	WORD74 = 000224	WORD96 = 000300
T\$BBEN = 000020	WORD3 = 000006	WORD52 = 000150	WORD75 = 000226	WORD97 = 000302
UBD.IN = 000020	WORD30 = 000074	WORD53 = 000152	WORD76 = 000230	WORD98 = 000304
WORD0 = 000000	WORD31 = 000076	WORD54 = 000154	WORD77 = 000232	WORD99 = 000306
WORD1 = 000002	WORD32 = 000100	WORD55 = 000156	WORD78 = 000234	WORDVAL = 000310
WORD10 = 000024	WORD33 = 000102	WORD56 = 000160	WORD79 = 000236	XTWRITE = 001000
WORD11 = 000026	WORD34 = 000104	WORD57 = 000162	WORD8 = 000020	
WORD12 = 000030	WORD35 = 000106	WORD58 = 000164		

. ABS: 000000 000
000000 001
FRAME2: 000644 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3175 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:45
FRAME2, FRAME2, SP=C20.111M, C20.11FRAME2

```

1
2 000000 .TITLE--QLBLD...
3 .PSECT--QLBLD.
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 000000 QLBLD::
12 000000 005003 CLR R3 ; INITIALIZE ELEMENT POINTER.
13 000002 CALL QBMARK ; MARK BLOCKS WHICH USE SUBDOC LOGIC.
14 000006 005767 000000G TST QLBAD ; FINISHED PROCESSING QLB?.
15 000012 ;
16 000012 003427 O:STRT: BLE QLBPT ; BRANCH IF YES.
17 ;
18 ;
19 ;
20 ;
21 ;
22 000014 016700 000000G MOV PSQLB,R0 ;R0 -> SOLB ENTRY.
23 000020 112067 000000G MOVB (R0)+,QENTL ;GET LENGTH OF ENTRY
24 000024 042767 177400 000000G BIC #177400,QENTL ;CLEAR UPPER BITS.
25 000032 016767 000000G 000000G MOV QENTL,WQENTL ;COPY FOR LOOP CONTROL COUNTER
26 000040 066700 000000G ADD QENTL,R0 ;POINT TO QID AT END OF SOLB ENTRY.
27 000044 112067 000000G MOVB (R0)+,XQID ;SAVE AS EXTERNAL QID.
28 000050 010067 000000G MOV R0,NPSQLB ;SAVE AS POINTER TO NEXT ENTRY
29 ;
30 ;
31 ;
32 ;
33 ;
34 ;
35 ;
36 000054 QLBLT:
37 000054 026767 000000G 000000G CMP LXQID,XQID ;PREVIOUS == CURRENT.?
38 000062 002403 BLT QLBPT ;LAST QLB OR DUMMY QUERY.
39 000064 001416 BEQ QLB.2 ;NOT LAST QLB OF QUERY.
40 000066 000167 000502 JMP QLBXT ;GENERATE TERMINAL WORD OF QLB
41 ;
42 000072 QLBPT:
43 000072 032767 000000G 000000G BIT #POLB,GFLAG ;WAS THERE A QLB BLOCK FOR QUERY?
44 000100 BON JOINX ;BRANCH IF YES.
45 ;
46 000102 012767 137774 000000G MOV #137774,WQW ;PAD WORD.
47 000110 CALL QLBREF ;WRITE IN HQR QLB REF PAGE.
48 000114 005267 000000G INC LXQID ;SET NEXT QUERY ID.
49 000120 000755 BR QLBLT ;CATCH ALL SKIPPED QID'S.
50 ;
51 000122 QLB.2:
52 000122 052767 000000G 000000G BIS #POLB,GFLAG ;SET QLB PROCESSED FLAG.
53 000130 016700 000000G MOV PSQLB,R0 ;R0 -> SOLB ENTRY.
54 000134 005200 INC R0 ;BUMP PAST ENTRY LENGTH.

```

```

56      ;          CONVERT ONE SOLB ENTRY ELEMENT BY ELEMENT.
57      ;
58 000136 132710 000000G. ELEMENT: BITB.  #B$FMN.(R0)      ;FIRST MULTI-NOT.?
59 000142. 001432.      BEQ.      NOTFMN.      ;NO.
60      ;
61      ;          FIRST MULTI-NOT BIT SET IN SOLB ELEMENT.
62      ;
63 000144 032767 000000G.000000G. BIT.      #LMN.GFLAG.      ;WAS PREVIOUS ELEMENT A MULTI-NOT.
64 000152. 001411      BEQ.      1$      ;NO.
65 000154 052767 000000G.000000G. BIS.      #HLMN.WQW.      ;SET HOR LAST MULTI-NOT FLAG.
66 000162.      CALL.      QLBREF.      ;WRITE WORD TO HOR QLB.
67 000166 052767 000000G.000000G. BIS.      #HMN.WQW.      ;SET MULTI-NOT FLAG IN NEW WORKING QLB WORD.
68 000174 000440      BR.      JOIN
69      ;
70      ;          FIRST MULTI-NOT BIT SET IN SOLB ELEMENT AND PREVIOUS
71      ;          ELEMENT WAS NOT A MULTI-NOT.
72      ;
73 000176 052767 000000G.000000G. 1$: BIS.      #LMN.GFLAG.      ;SET LOCAL FLAG FOR MULTI-NOT.
74 000204 022767 000004. 000000G. CMP.      #BIT2.BITPOS.      ;IS WQW EMPTY?.
75 000212. 001431      BEQ.      JOIN      ;PROCESS CURRENT ELEMENT.
76 000214      CALL.      QLBREF.      ;WRITE REF PAGE WORD
77 000220 052767 000000G.000000G. BIS.      #HMN.WQW.      ;SET MULTI-NOT FLAG IN NEW WQW
78 000226 000423      BR.      JOIN      ;PROCESS CURRENT ELEMENT.
79      ;
80      ;          FIRST MULTI-NOT BIT NOT SET IN SOLB ELEMENT.
81      ;
82 000230 132710 000000G. NOTFMN: BITB.  #B$MUL.(R0)      ;IS THIS ELEMENT A MULTI-NOT.
83 000234 001404      BEQ.      1$      ;NO.
84 000236 052767 000000G.000000G. BIS.      #HMN.WQW.      ;SET HOR MULTI-NOT FLAG.
85 000244 000414      BR.      JOIN      ;PROCESS CURRENT ELEMENT.
86      ;
87      ;          FIRST MULTI-NOT BIT NOT SET IN SOLB ELEMENT AND ELEMENT
88      ;          IS NOT A MULTI-NOT.
89      ;
90 000246 032767 000000G.000000G. 1$: BIT.      #LMN.GFLAG.      ;WAS PREVIOUS ELEMENT PART OF A M-NOT.
91 000254 001410      BEQ.      JOIN      ;NO.
92 000256 042767 000000G.000000G. BIC.      #LMN.GFLAG.      ;CLEAR LOCAL FLAG.
93 000264 052767 000000G.000000G. BIS.      #HLMN.WQW.      ;SET FLAG FOR HOR LAST MULTI-NOT.
94 000272.      CALL.      QLBREF.      ;WRITE WQW TO HOR REF PAGE.
95      ;

```



```

97 000276          JOIN:
98 000276 132710 000000G BITB  #B$DEL,(R0)      ;IS THIS A SUBDOC USING ELEMENT?
99 000302          BOFF  15$          ;BRANCH IF NO
100 000304 010003          MOV  R0,R3         ;SAVE ADDRESS OF QLB ELEMENT
101 000306 052767 000000G 000000G BIS  #SDUMAP,GFLAG  ;SET FLAG TO MARK HQR EOR WORD
102 000314          ;
103 000314 132720 000000G          15$: BITB  #B$NOT,(R0)+  ;IS CURRENT ELEMENT IN SOLB A NOT?
104 000320 001003          BNE  1$          ;YES, DO NOT SET BIT IN WQW
105 000322 056767 000000G 000000G BIS  BITPOS,WQW  ;SET A BIT FOR REF PAGE
106
107 000330          ;
108 000330 005367 000000G          1$: DEC  WQENTL      ;DEC LOOP COUNT (# ELEMENTS IN SOLB ENTRY)
109 000334 001414          BEQ  2$          ;FINISHED WITH CURRENT ENTRY
110 000336 016701 000000G          MOV  BITPOS,R1   ;PREPARE TO SHIFT
111 000342 006301          ASL  R1          ;SHIFT TO NEXT POSITION
112 000344 010167 000000G          MOV  R1,BITPOS  ;SET NEW BIT POSITION
113
114 000350 032767 040000 000000G BIT  #BIT14,BITPOS ;SHIFTED OUT OF BIT13 ?
115 000356          BOFF  ELEMNT      ;NO, PROCESS NEXT ELEMENT
116 000360          CALL  QLBREF     ;WRITE WQW TO HQR
117 000364 000664          BR   ELEMNT     ;PROCESS NEXT ELEMENT
118
119 000366 052767 000000G 000000G 2$: BIS  #HLNE,WQW   ;SET LAST NON-EQW WORD
120 000374 032767 000000G 000000G BIT  #SDUMAP,GFLAG ;DOES THIS QLB BLOCK HAVE MAPPED SUBELEMENT?
121 000402          BON  3$          ;BRANCH IF YES
122 000404 042767 000002 000000G BIC  #BIT1,WQW  ;REMOVE SUBDOC USING CODE IN QLB
123 000412 000411          BR   4$
124 000414          ;
125 000414 152713 000000G          BISB #B$EOTR,@R3 ;MARK SOFTWARE QLB AS LAST MAPPED TOR ELEMENT
126 000420 016701 000000G          MOV  XQID,R1    ;GET QUERY ID
127 000424 006301          ASL  R1         ;MAKE INTO WORD OFFSET
128 000426 005261 000000G          INC  SDBUFS(R1) ;ALLOCATE SID BLOCK FOR QUERY
129 000432 005267 000000G          INC  SIDBLK    ;INCREMENT COUNT OF ALLOCATED SID BLOCKS
130 000436          ;
131 000436          4$: CALL  QLBREF     ;WRITE WQW TO QLB REF
132 000442 042767 000000G 000000G BIC  #LMN+SDUMAP,GFLAG ;CLEAR LOCAL FLAG
133 000450 000436          BR   JOIN1

```

```

135 000452.          JOINX:
136 000452. 016701 000000G.      MOV.   LXQID,R1          ;LOAD EXTERNAL QID IN WORD OFFSET FORM
137 000456 006301          ASL.   R1
138 000460 032761 000000G.000000G. BIT.   *SHRL,QCL(R1)    ;IS THERE AN HRL MARK IN SOFTWARE QCL
139 000466 001406          BEQ.   3$              ;NO
140 000470 016761 000000G.000000G. MOV.   CHQA,HEAT(R1)    ;SAVE ADDRESS OF HRL MARK WORD
141 000476 052767 000004 000000G. BIS.   #BIT2,WQW        ;SET HRL MARK IN QLB REF WORD
142.                  ;
143.                  ;
144.                  ;
145.                  ;
146 000504 052767 000000G.000000G.3$: BIS.   #HEQ0,WQW        ;SET END-OF-QUERY BIT
147 000512.          CALL.  QLBREF           ;WRITE REF QLB WORD
148 000516 132713 000000G.      BITB.  #B#EOTR,@R3     ;SUBELEMENT MAPPED TO BLOCK?
149 000522.          BOFF.  4$              ;BRANCH IF NOT
150 000524 152713 000000G.      BISB.  #B#E00,@R3     ;MARK QLB END-OF-QUERY ELEMENT
151 000530          ;
152.000530 042767 000000G.000000G. BIC.   #POLB,GFLAG     ;RESET QLB PROCESSED FLAG
153 000536 005267 000000G.      INC.   LXQID            ;SET NEXT QID TO BE PROCESSED
154 000542. 000167 177306      JMP.   QLBRT          ;PROCESS NEXT QLB BLOCK
155.                  ;
156.                  ;
157.                  ;
158.                  ;
159.                  ;
160.                  ;
161.                  ;
162.                  ;
163.                  ;
164.                  ;
165 000546          JOIN1:
166 000546 016767 000000G.000000G. MOV.   NPSQLB,PSQLB    ;POINT TO NEXT SQLB ENTRY
167 000554 166767 000000G.000000G. SUB.   QENTL,QLBAD     ;SUB LENGTH OF ENTRY FROM SQLB LENGTH
168 000562. 162767 000002 000000G. SUB.   #2,QLBAD        ;ADJUST FOR LENGTH BYTE + QID
169 000570 000167 177216      JMP.   QSTRT
170.                  ;
171.                  ;
172.                  ;
173 000574          QLBXT:
174.                  ;
175.                  ;
176.                  ;
177 000574 012700 000000G.      MOV.   #SIDSIZ,R0       ;SIZE OF SID MEMORY
178 000600 016701 000000G.      MOV.   SIDBLK,R1       ;DIVIDE BY NUMBER OF BLOCKS ALLOCATED
179 000604          CALL.  $DIV
180 000610 010067 000000G.      MOV.   R0,SIDBLK      ;SIZE OF A SINGLE SID BLOCK ALLOCATION
181.                  ;
182.                  ;
183.                  ;
184 000614          HALT. PP. PROCESSOR
185 000620          CALL.  PPHALT           ;HALT PP
186.                  ;
186.                  ;

```

```

188
189
190
191 000622
192 000622
193 000630 012700 000000G
194 000634 016701 000000G
195 000640
196 000640 003417
197 000642 112002
198 000644 160201
199 000646 060200
200 000650 032700 000001
201 000654
202 000656 005200
203 000660 005301
204 000662
205 000662 012002
206 000664 152762 000000G 000000G
207 000672 162701 000003
208 000676 000760
209 000700
210 000700
211 000706
212
213 000001

```

```

:
: SUBROUTINE TO MARK SOFTWARE QLB ELEMENTS WHICH HAVE SUBDOC MAPPINGS
:
: Q3MARK:
SAVE R0,R1,R2
MOV *SDLB,R0 ;SET POINTER TO START OF SDLB
MOV SDLBAD,R1 ;GET BYTE SIZE OF SDLB
10$:
BLE 50$ ;SDLB EXHAUSTED -- EXIT SUBROUTINE
MOVB (R0)+,R2 ;GET ELEMENT COUNT IN THIS BLOCK
SUB R2,R1 ;ADJUST SDLB SPACE REMAINING
ADD R2,R0 ;POINT PAST ELEMENTS
BIT #BIT0,R0 ;IS ADDRESS ON WORD BOUNDARY?
BOFF 20$ ;BRANCH IF SO
INC R0 ;SKIP PAD BYTE
DEC R1 ;ALSO ADJUST SDLB SPACE
20$:
MOV (R0)+,R2 ;GET POINTER TO QLB
BISB #B$SDEL,QLB(R2) ;MARK QLB ELEMENT AS SUBDOC USING
SUB #3,R1 ;ADJUST SDLB SPACE FOR COUNT + POINTER
BR 10$ ;CHECK FOR SDLB EXHAUSTED
50$:
RESTOR R0,R1,R2
RETURN
:
.END

```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITPOS = ***** GX
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
B\$EOD = ***** GX
B\$EODTR = ***** GX
B\$FMN = ***** GX
B\$MUL = ***** GX
B\$NOT = ***** GX
B\$SDEL = ***** GX
CBKALL = 001000
CBKCLK = 000400
CHOA = ***** GX
CHOBRE = 100000
CPCCRE = 010000
CPREAD = 040000
CPWRTE = 020000
CSADRD = 000004
CSEDCI = 100000
C\$ODE = 000040
CSWRTE = 000100
DBR.RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPGS = 100000
DMAARR = 000005
DMARRD = 000003
DMARRU = 000004
ELEMNT = 000136R
ENBR = 010000
FLAG = ***** GX
HEAT = ***** GX
HEQQ = ***** GX
HLMN = ***** GX
HLNE = ***** GX
HMN = ***** GX
JOIN = 000276R
JOINX = 000452R
JOIN1 = 000546R
LMN = ***** GX
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LXQID = ***** GX
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADDR = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
NOTFMN = 000230R
NPSQLB = ***** GX
PLB = 000010
PLC = 000020
PLD = 000030
PLRWJ = 000200
PLR.EN = 000200
PPHALT = ***** GX
PDLB = ***** GX
PSOLB = ***** GX
QBMARK = 000622R
OCL = ***** GX
OENTL = ***** GX
OQLB = ***** GX
QLBAD = ***** GX
QLBLD = 000000RG
QLBLT = 000054R
QLBPT = 000072R
QLBREF = ***** GX
QLBXT = 000574R
QLB.2 = 000122R
QR\$CR1 = 176420
QR\$CR2 = 176422
OR\$LBR = 176424
002.Q\$TRT = 000012R
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
002.Q\$CP = 000060
002.Q\$CPCC = 000010
002.Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IH = 000003
Q\$IHRL = 000002
Q\$INRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
002.Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
002.Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
002.Q\$QLB = 000054
002.Q\$QLR = 000001
002.Q\$QJ = 000042
Q\$RDCD = 000005
002.Q\$RDMD = 000006
002.Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
002.Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGO.EN = 000200
RGO.VA = 020000
SDBUFS = ***** GX
SDLB = ***** GX
SDLBAD = ***** GX
SDUMAR = ***** GX
SEQ.CI = 000010
SHRL = ***** GX
SIDSLZ = ***** GX
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$OR = 000006
S\$OX = 000004
S\$SR = 000007
S\$S1 = 000010
TD\$CTW = 176300
TD\$INL = 004000

TD#MEM= 000270	T#OBE = 004000	WORD21= 000052	WORD5 = 000012	WORD78= 000234
TD#OAR= 176344	T#OBF = 010000	WORD22= 000054	WORD50= 000144	WORD79= 000236
TD#OTR= 176346	T#OBRA= 000034	WORD23= 000056	WORD51= 000146	WORD8 = 000020
TD#ORD= 000274	T#OBWA= 000032	WORD24= 000060	WORD52= 000150	WORD80= 000240
TD#SW = 176376	T#OUTA= 100000	WORD25= 000062	WORD53= 000152	WORD81= 000242
TD#TAR= 176372	T#RBD0= 000200	WORD26= 000064	WORD54= 000154	WORD82= 000244
TD#TAW= 176362	T#RNB = 000040	WORD27= 000066	WORD55= 000156	WORD83= 000246
TD#TDR= 176374	T#RSET= 040000	WORD28= 000070	WORD56= 000160	WORD84= 000250
TD#TDW= 176364	T#SC = 000022	WORD29= 000072	WORD57= 000162	WORD85= 000252
T#AD = 000020	T#SCLK= 020000	WORD3 = 000006	WORD58= 000164	WORD86= 000254
T#BA = 000002	T#SEG1= 000000	WORD30= 000074	WORD59= 000166	WORD87= 000256
T#BD = 000010	T#SEG2= 000001	WORD31= 000076	WORD6 = 000014	WORD88= 000260
T#BS0 = 100000	T#SEG3= 000002	WORD32= 000100	WORD60= 000170	WORD89= 000262
T#BT = 000020	T#S0 = 000001	WORD33= 000102	WORD61= 000172	WORD9 = 000022
T#BTAR= 000030	T#UBUS= 100000	WORD34= 000104	WORD62= 000174	WORD90= 000264
T#BTD = 002000	T#ICLK= 000400	WORD35= 000106	WORD63= 000176	WORD91= 000266
T#CD = 000100	T#BEN= 000020	WORD36= 000110	WORD64= 000200	WORD92= 000270
T#CLK = 002000	UBD.IN= 000020	WORD37= 000112	WORD65= 000202	WORD93= 000272
T#DISK= 000200	WORD0 = 000000	WORD38= 000114	WORD66= 000204	WORD94= 000274
T#DRD = 000004	WORD1 = 000002	WORD39= 000116	WORD67= 000206	WORD95= 000276
T#ENEM= 010000	WORD10= 000024	WORD4 = 000010	WORD68= 000210	WORD96= 000300
T#FSAA= 000000	WORD11= 000026	WORD40= 000120	WORD69= 000212	WORD97= 000302
T#FSAB= 000004	WORD12= 000030	WORD41= 000122	WORD7 = 000016	WORD98= 000304
T#FSAC= 000014	WORD13= 000032	WORD42= 000124	WORD70= 000214	WORD99= 000306
T#FSB2= 000010	WORD14= 000034	WORD43= 000126	WORD71= 000216	WQENTL= ***** GX
T#IB = 000026	WORD15= 000036	WORD44= 000130	WORD72= 000220	WQW = ***** GX
T#IBAR= 000024	WORD16= 000040	WORD45= 000132	WORD73= 000222	WRDVAL= 000310
T#IBE = 020000	WORD17= 000042	WORD46= 000134	WORD74= 000224	XO ID = ***** GX
T#IBF = 040000	WORD18= 000044	WORD47= 000136	WORD75= 000226	XTREAD= 001000
T#ICD = 000040	WORD19= 000046	WORD48= 000140	WORD76= 000230	XTWRITE= 000400
T#MODE= 004000	WORD2 = 000004	WORD49= 000142	WORD77= 000232	\$D IV = ***** GX
T#OB = 000036	WORD20= 000050			

. ABS. 000000 000
 000000 001
 QLBLD 000710 002
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3200 WORDS (13 PAGES)
 DYNAMIC MEMORY: 3060 WORDS (14 PAGES)
 ELAPSED TIME: 00:00:44
 QLBLD,QLBLD /-SP=C20,11M,C20,1QLBLD

```

1
2 000000 .TITLE QEXLD
3 .PSECT QEXLD
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 000000 QEXLD::
27 000000 COEX:
28 000000 005767 000000G TST QEXAD ; IS QEX EMPTY ?
29 000004 001470 BEQ QCEX ; YES, EXIT
30 000006 CALL DMASET ; START UP MICROCODE
31 ;
32 ;
33 ;
34 000012 012767 000002 000000G MOV #2,ADDR ; WRITE INTO CD STARTING AT LOC 2
35 000020 016700 000000G MOV QEXAD,R0 ; NUMBER OF BYTES IN QEX
36 000024 006200 ASR R0 ; GET NUMBER OF WORDS
37 000026 012701 000000G MOV #QEX,R1 ; R1 -> SOFT QEX
38 000032 016767 000000G 000000G 255: MOV QXADD,DATA ; FIRST WRITE QEX ADDR INTO CD
39 000040 CALL ONECD2
40 000044 005267 000000G INC ADDR ; BUMP CD ADDRESS
41 000050 012167 000000G MOV (R1),DATA ; WRITE SQEX TO HQEX
42 000054 CALL ONECD2
43 000060 005267 000000G INC ADDR ; BUMP CD ADDRESS
44 000064 005267 000000G INC QXCNT ; BUMP QEX ADDR/DATA PAIRS COUNT
45 000070 005300 DEC R0 ; SUB FROM WORD COUNT
46 000072 001412 BEQ LBQX ; FINISHED, NOW LOAD QEX
47 000074 005267 000000G INC QXADD ; BUMP TO NEXT HQR QEX ADDRESS
48 000100 026767 000000G 000000G CMP #HIGH,QXADD ; FINISHED WITH QEX
49 000106 103351 BHIS 2# ; NO, CONTINUE
50 000110 CALL ERR2 ; OVERFLOW
51 000114 000167 000052 JMP QSEX
52 ;
53 ;
54 ;
55 ;
56 000120 LDQX:
57 000120 005067 000000G CLR ADDR ; WRITE FLAG TO CD ADDR ZERO

```

```

58 000124 012767 000001 000000G. MOV. #1,DATA. ;FLAG FOR 'WRITE QEX'
59 000132. CALL. ONECD2.
60 000136 005267 000000G. INC. ADDR ;BUMP ADDRESS TO 1
61 000142 016767 000000G.000000G. MOV. QXCNT,DATA. ;WRITE QEX ADDR/DATA PAIRS COUNT TO CD ADDR 1
62 000150 CALL. ONECD2.
63 ;
64 ;
65 ; START THE MICROCODE THAT DOES THE LOADING OF THE QEX.
66 000154 012767 000042 000000G. MOV. #Q$QW, CODE. ;SELECT WINDOW MEMORY.
67 000162. CALL. LOADQX.
68 ;
69 ;
70 ; GOOD EXIT.
71 000166 000241 QCEX: CLC.
72 000170 RETURN.
73 ;
74 ;
75 ; EXIT ON ERROR.
76 000172 000261 QSEX: SEC.
77 000174 RETURN.
78 ;
79 000001 .END.

```

ADDR = ***** GX.	BYTE41 = 000051	BYTE93 = 000135	PLD = 000030	Q\$PP2 = 000300
ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLRWR = 000200	Q\$GHLT = 000013
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLR,EN = 000200	Q\$QL = 000043
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QCEX = 000166R	002 Q\$QLA = 000053
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	QEX = ***** GX.	Q\$QLB = 000054
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	QEXAD = ***** GX.	Q\$QLR = 000051
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	QEXLD = 000000RG	002 Q\$QJ = 000042
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	QR\$CR1 = 176420	Q\$RDCD = 000005
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	QR\$CR2 = 176422	Q\$RDM = 000006
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	QR\$CLR = 176424	Q\$REBK = 001000
BIT13 = 020000	BYTE50 = 000062	CNOBRE = 100000	QSEX = 000172R	002 Q\$RNC = 006000
BIT14 = 040000	BYTE51 = 000063	CODE = ***** GX.	QXADD = ***** GX.	Q\$RSC = 004000
BIT15 = 100000	BYTE52 = 000064	CPCCEN = 010000	QXCNT = ***** GX.	Q\$RSET = 000010
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	QXHIGH = ***** GX.	Q\$SM = 100000
BIT3 = 000010	BYTE54 = 000066	CPURTE = 020000	Q\$ATTN = 000100	Q\$SP = 000120
BIT4 = 000020	BYTE55 = 000067	COEX = 000000R	002 Q\$BCL = 000001	Q\$SP2 = 000340
BIT5 = 000040	BYTE56 = 000070	CSADRD = 000004	Q\$CCCP = 000040	RGD,EN = 000200
BIT6 = 000100	BYTE57 = 000071	CSEDCI = 100000	Q\$CHB = 000400	RGD,VA = 020000
BIT7 = 000200	BYTE58 = 000072	CSOE = 000040	Q\$CHRL = 000200	SEQ,CI = 000010
BIT8 = 000400	BYTE59 = 000073	CSURTE = 000100	Q\$CLR = 000040	S\$CLR = 000000
BIT9 = 001000	BYTE6 = 000006	DATA = ***** GX.	Q\$CNC = 030000	S\$LA = 000001
BYTE0 = 000000	BYTE60 = 000074	DBR, RD = 000001	Q\$CP = 000060	S\$QB = 000005
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$CPCC = 000010	S\$QR = 000006
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$CP2 = 000260	S\$QX = 000004
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$CSC = 010000	S\$SR = 000007
BYTE12 = 000014	BYTE64 = 000100	DISPGS = 100000	Q\$CSEL = 000360	S\$S1 = 000010
BYTE13 = 000015	BYTE65 = 000101	DNARUR = 000005	Q\$CSET = 000002	S\$S2 = 000014
BYTE14 = 000016	BYTE66 = 000102	DNARRD = 000003	Q\$CSP = 020000	TD\$CTR = 176370
BYTE15 = 000017	BYTE67 = 000103	DNARUR = 000004	Q\$DMA = 000001	TD\$CTU = 176360
BYTE16 = 000020	BYTE68 = 000104	DNASET = ***** GX.	Q\$ENBK = 040000	TD\$INL = 004000
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$ENOP = 020000	TD\$MEM = 000270
BYTE18 = 000022	BYTE7 = 000007	ERR2 = ***** GX.	Q\$FAL = 004000	TD\$OAR = 176344
BYTE19 = 000023	BYTE70 = 000106	LDQX = 000120R	002 Q\$FC = 000045	TD\$QTR = 176345
BYTE2 = 000002	BYTE71 = 000107	LOADQX = ***** GX.	Q\$FO = 000044	TD\$QRD = 000274
BYTE20 = 000024	BYTE72 = 000110	LOC,EN = 000100	Q\$FP = 000046	TD\$SW = 176376
BYTE21 = 000025	BYTE73 = 000111	LOC,UB = 040000	Q\$HBF = 000002	TD\$STAR = 176372
BYTE22 = 000026	BYTE74 = 000112	LOC,UB = 100000	Q\$ICP = 000006	TD\$TAU = 176362
BYTE23 = 000027	BYTE75 = 000113	MAREN1 = 000001	Q\$IHB = 000003	TD\$TDR = 176374
BYTE24 = 000030	BYTE76 = 000114	MAREN2 = 004000	Q\$IHLR = 000002	TD\$TDW = 176364
BYTE25 = 000031	BYTE77 = 000115	MARLOD = 010000	Q\$IMRP = 000007	T\$AD = 000020
BYTE26 = 000032	BYTE78 = 000116	MAROUT = 000002	Q\$LBD = 001000	T\$BA = 000002
BYTE27 = 000033	BYTE79 = 000117	MAR,LO = 002000	Q\$LBDP = 001001	T\$BD = 000010
BYTE28 = 000034	BYTE8 = 000010	MAR,OU = 000040	Q\$LBP = 000001	T\$BSO = 100000
BYTE29 = 000035	BYTE80 = 000120	MBKALL = 001000	Q\$LDCD = 000003	T\$BT = 000020
BYTE3 = 000003	BYTE81 = 000121	MBKCLK = 000400	Q\$LDMD = 000004	T\$BTAR = 000030
BYTE30 = 000036	BYTE82 = 000122	MMAADR = 000100	Q\$LDPP = 002000	T\$STD = 002000
BYTE31 = 000037	BYTE83 = 000123	MNLEFT = 000002	Q\$LHP = 010000	T\$CD = 000100
BYTE32 = 000040	BYTE84 = 000124	MNOE = 000004	Q\$MNC = 140000	T\$CLK = 002000
BYTE33 = 000041	BYTE85 = 000125	MNURTE = 000010	Q\$MR = 000052	T\$DISK = 000200
BYTE34 = 000042	BYTE86 = 000126	MNOBRE = 100000	Q\$MRP = 000040	T\$DRD = 000004
BYTE35 = 000043	BYTE87 = 000127	MREN1 = 000001	Q\$MRP2 = 000240	T\$MEMM = 010000
BYTE36 = 000044	BYTE88 = 000130	MREN2 = 020000	Q\$MSC = 040000	T\$FSAB = 000000
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q\$MSET = 000004	T\$FSAB = 000004
BYTE38 = 000046	BYTE9 = 000011	N = 000144	Q\$MSP = 100000	T\$FSAC = 000014
BYTE39 = 000047	BYTE90 = 000132	ONECD2 = ***** GX.	Q\$NCLK = 176000	T\$FSB2 = 000010
BYTE4 = 000004	BYTE91 = 000133	PLB = 000010	Q\$PP = 000100	T\$IB = 000026
BYTE40 = 000050	BYTE92 = 000134	PLC = 000020	Q\$PPSW = 000320	T\$IBAR = 000024

T#IBE = .020000	WORD11 = .000026	WORD34 = .000104	WORD57 = .000162	WORD8 = .000020
T#IBF = .040000	WORD12 = .000030	WORD35 = .000106	WORD58 = .000164	WORD80 = .000240
T#ICD = .000040	WORD13 = .000032	WORD36 = .000110	WORD59 = .000166	WORD81 = .000242
T#MODE = .004000	WORD14 = .000034	WORD37 = .000112	WORD6 = .000014	WORD82 = .000244
T#OB = .000036	WORD15 = .000036	WORD38 = .000114	WORD60 = .000170	WORD83 = .000246
T#OBE = .004000	WORD16 = .000040	WORD39 = .000116	WORD61 = .000172	WORD84 = .000250
T#OBF = .010000	WORD17 = .000042	WORD4 = .000010	WORD62 = .000174	WORD85 = .000252
T#OBRA = .000034	WORD18 = .000044	WORD40 = .000120	WORD63 = .000176	WORD86 = .000254
T#OBWA = .000032	WORD19 = .000046	WORD41 = .000122	WORD64 = .000200	WORD87 = .000256
T#OUTA = .100000	WORD2 = .000004	WORD42 = .000124	WORD65 = .000202	WORD88 = .000260
T#RBD = .000200	WORD20 = .000050	WORD43 = .000126	WORD66 = .000204	WORD89 = .000262
T#RNB = .000040	WORD21 = .000052	WORD44 = .000130	WORD67 = .000206	WORD9 = .000022
T#RSET = .040000	WORD22 = .000054	WORD45 = .000132	WORD68 = .000210	WORD90 = .000264
T#SC = .000022	WORD23 = .000056	WORD46 = .000134	WORD69 = .000212	WORD91 = .000266
T#SCLK = .020000	WORD24 = .000060	WORD47 = .000136	WORD7 = .000016	WORD92 = .000270
T#SEG1 = .000000	WORD25 = .000062	WORD48 = .000140	WORD70 = .000214	WORD93 = .000272
T#SEG2 = .000001	WORD26 = .000064	WORD49 = .000142	WORD71 = .000216	WORD94 = .000274
T#SEG3 = .000002	WORD27 = .000066	WORD5 = .000012	WORD72 = .000220	WORD95 = .000276
T#SO = .000001	WORD28 = .000070	WORD50 = .000144	WORD73 = .000222	WORD96 = .000300
T#SUBS = .100000	WORD29 = .000072	WORD51 = .000146	WORD74 = .000224	WORD97 = .000302
T#1CLK = .000400	WORD3 = .000006	WORD52 = .000150	WORD75 = .000226	WORD98 = .000304
T#BBEN = .000020	WORD30 = .000074	WORD53 = .000152	WORD76 = .000230	WORD99 = .000306
UBD.IN = .000020	WORD31 = .000076	WORD54 = .000154	WORD77 = .000232	WORDVAL = .000310
WORD0 = .000000	WORD32 = .000100	WORD55 = .000156	WORD78 = .000234	XTREAD = .001000
WORD1 = .000002	WORD33 = .000102	WORD56 = .000160	WORD79 = .000236	XTWRITE = .000400
WORD10 = .000024				

. ABS. 000000 000
000000 001
QEXLD. 000176 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3170 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
QEXLD, QEXLD /-SP=C20, IJIM, C20, IJQEXLD.

```

1
2 000000 .TITLE--CD--
3 .PSECT CD
4
5
6
7
8
9
10
11
12 .MCALL RDAF$,CLEF$,OPEN$,CLOSE$
13
14
15
16
17
18
19
20
21
22
23 CD::
24 000000 042767 000000G-000000G BIC #FIRST,SELECT ;CLEAR FIRST TIME THROUGH FLAG
25 000006 012767 000001 000002G MOV #1,VIRT+2 ;RE-INIT BLOCK COUNT
26 000014 005067 000000G CLR ADDR ;CLEAR CP DATA MEMORY ADDRESS
27 000020 016767 000000G-000000C MOV LCD,INDNB+N.FNAM ;PLACE FILE NAME INTO INPUT DNB
28 000026 016767 000002G-000000C MOV LCD+2,INDNB+N.FNAM+2
29 000034 OPEN$, #INFDB
30
31 000052 CALL DMASET ;START UP DMA MICROCODE
32
33
34
35
36
37 000056 CDNEXT: CALL GET ;READ A RECORD
38 000062 103002 BCC 1$
39 000064 000167 000150 JMP CPDX ;ERROR, EXIT
40 000070
41 000070 016705 000000C 1$: MOV INFDB+F.BKDS+2,R5 ;POINT TO RECORD READ
42 000074 032767 000000G-000000G BIT #FIRST,SELECT ;FIRST TIME THROUGH
43 000102 001014 BNE 3$ ;NO
44 000104 052767 000000G-000000G BIS #FIRST,SELECT ;SET FLAG FOR FIRST TIME THROUGH
45 000112 012567 000000G MOV (R5)+,LCOUNT ;GET NUMBER OF DATA WORDS
46 000116 012767 177777 000000G MOV #-1,TRANSF ;START TRANSFER COUNT AT -1
47 000124 026727 000000G-000400 CMP LCOUNT,#256 ;256 WORDS TO TRANSFER
48 000132 001413 BEQ 4$ ;YES, 2 TRANSFERS NEEDED (COUNT WORD)
49
50
51
52
53
54 000134 026727 000000G-000400 3$: CMP LCOUNT,#256 ;MORE THAN A FULL BLOCK LEFT TO TRANSFER
55 000142 003007 BGT 4$ ;YES, TRANSFER 256 WORDS THIS TIME
56 000144 052767 000000G-000000G BIS #LAST,SELECT ;LAST TRANSFER
57 000152 066767 000000G-000000G ADD LCOUNT,TRANSF ;TRANSFER REMAINING WORDS

```

```

58 000160 000406 BR 5$ :LOAD CD BY DMA
59 000162 062767 000400 000000G 4$ :ADD #256, TRANSF :NUMBER OF WORDS TO TRANSFER
60 000170 166767 000000G 000000G :SUB TRANSF,LCOUNT :SUB FROM TOTAL
61 :
62 : SIGNAL MICROCODE TO ACCEPT CP DATA MEMORY DATA
63 :
64 000176 010567 000000G 5$ :MOV R5,INSAVE :SAVE POINTER TO INPUT DATA
65 000202 012746 000003 :MOV #0$LCD, -(SP) :MOVE ATTN CODE TO STACK
66 000206 :CALL DMA :PERFORM DMA LOAD
67 :
68 : IF NOT FINISHED WITH ALL DATA, GO BACK AND PREPARE TO
69 : TRANSFER NEXT BLOCK
70 :
71 000212 032767 000000G 000000G :BIT #LAST,SELECT :FINISHED?
72 000220 001007 :BNE CPDX :YES
73 000222 066767 000000G 000000G :ADD TRANSF,ADDR :SET CD ADDRESS FOR NEXT TRANSFER
74 000230 005067 000000G :CLR TRANSF
75 000234 000167 177616 :JMP CDNEXT :GET NEXT RECORD
76 :
77 000240 :CPDX
78 000240 005046 :CLR -(SP) :CLEAR NOTHING IN CSR1
79 000242 012746 176000 :MOV #0$CLK, -(SP) :SET NO-CLOCKS
80 000246 :CALL CSR1
81 000252 005067 176422 :CLR QR$CR2 :SET LOAD MODE
82 :
83 000256 :CLOSE$ #INFD
84 000266 105067 000000C :CLR INDNB+N,FVER :RESET FILE VERSION NUMBER
85 000272 :RETURN
86 :
87 000001 :END

```

ADDR = ***** GX.	BYTE41 = 000051	BYTE93 = 000135	MMOE = 000004	Q\$MSP = 100000
ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MMWRTE = 000010	Q\$NCLK = 176000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	MNOBRE = 100000	Q\$PP = 000100
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	MREN1 = 000001	Q\$PPSW = 000320
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	MREN2 = 020000	Q\$PP2 = 000300
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	MSYN = 000040	Q\$QHLT = 000013
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	N = 000144	Q\$QL = 000043
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	N.FNAM = ***** GX.	Q\$QLA = 000053
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	N.FVER = ***** GX.	Q\$QLB = 000054
BIT12 = 010000	BYTES = 000005	CBKCLK = 000400	PAR\$\$\$ = 000027	Q\$QLR = 000001
BIT13 = 020000	BYTE50 = 000062	CD = 000000RG	002.PLB = 000010	Q\$QW = 000042
BIT14 = 040000	BYTE51 = 000063	CBNEXT = 000056R	002.PLC = 000020	Q\$RDCD = 000005
BIT15 = 100000	BYTE52 = 000064	CHOBRE = 100000	PLD = 000030	Q\$RDMD = 000006
BIT2 = 000004	BYTE53 = 000065	CPCCEN = 010000	PLRWR = 000200	Q\$REBK = 001000
BIT3 = 000010	BYTE54 = 000066	CPDX = 000240R	002.PLR.EN = 000200	Q\$REB = 000000
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	QR\$CR1 = 176420	Q\$RSC = 004000
BIT5 = 000040	BYTE56 = 000070	CPURTE = 020000	QR\$CR2 = 176422	Q\$RSET = 000010
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	QR\$LBR = 176424	Q\$SM = 100000
BIT7 = 000200	BYTE58 = 000072	CSEDCI = 100000	Q\$ATTH = 000100	Q\$SP = 000120
BIT8 = 000400	BYTE59 = 000073	CSDOE = 000040	Q\$BCL = 000001	Q\$SP2 = 000340
BIT9 = 001000	BYTE6 = 000006	CSR1 = ***** GX.	Q\$CCCP = 000040	RGD.EN = 000200
BYTE0 = 000000	BYTE60 = 000074	CSURTE = 000100	Q\$CHB = 000400	RGD.VA = 020000
BYTE1 = 000001	BYTE61 = 000075	DBR.RD = 000001	Q\$CHRL = 000200	SELECT = ***** GX.
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	Q\$CLR = 000040	SEQ.CI = 000010
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	Q\$CNC = 030000	S\$CLR = 000000
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	Q\$CP = 000060	S\$LA = 000001
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q\$CPCC = 000010	S\$QB = 000005
BYTE14 = 000016	BYTE66 = 000102	DMA = ***** GX.	Q\$CP2 = 000260	S\$QR = 000006
BYTE15 = 000017	BYTE67 = 000103	DMAAUR = 000005	Q\$CSC = 010000	S\$QX = 000004
BYTE16 = 000020	BYTE68 = 000104	DHARRD = 000003	Q\$CSEL = 000360	S\$SR = 000007
BYTE17 = 000021	BYTE69 = 000105	DHARRR = 000004	Q\$CSET = 000002	S\$S1 = 000010
BYTE18 = 000022	BYTE7 = 000007	DMASET = ***** GX.	Q\$CSP = 020000	S\$S2 = 000014
BYTE19 = 000023	BYTE70 = 000106	ENBR = 010000	Q\$DMA = 000001	TD\$CTR = 176370
BYTE2 = 000002	BYTE71 = 000107	FIRST = ***** GX.	Q\$ENBK = 040000	TD\$CTW = 176360
BYTE20 = 000024	BYTE72 = 000110	FO.RD = ***** GX.	Q\$ENOP = 020000	TD\$INL = 004000
BYTE21 = 000025	BYTE73 = 000111	F.BKDS = ***** GX.	Q\$FAL = 004000	TD\$NEM = 000270
BYTE22 = 000026	BYTE74 = 000112	F.FACC = ***** GX.	Q\$FC = 000045	TD\$OAR = 176344
BYTE23 = 000027	BYTE75 = 000113	GET = ***** GX.	Q\$FO = 000044	TD\$OTR = 176345
BYTE24 = 000030	BYTE76 = 000114	INDNB = ***** GX.	Q\$FP = 000046	TD\$ORD = 000274
BYTE25 = 000031	BYTE77 = 000115	INFDB = ***** GX.	Q\$HBF = 000002	TD\$SW = 176375
BYTE26 = 000032	BYTE78 = 000116	INSVE = ***** GX.	Q\$ICP = 000006	TD\$TAR = 176372
BYTE27 = 000033	BYTE79 = 000117	LAST = ***** GX.	Q\$IHB = 000003	TD\$TAW = 176362
BYTE28 = 000034	BYTE8 = 000010	LCD = ***** GX.	Q\$IHRL = 000002	TD\$TDR = 176374
BYTE29 = 000035	BYTE80 = 000120	LCOUNT = ***** GX.	Q\$IMRP = 000007	TD\$TDW = 176364
BYTE3 = 000003	BYTE81 = 000121	LOC.EN = 000100	Q\$LBD = 001000	TRANSF = ***** GX.
BYTE30 = 000036	BYTE82 = 000122	LOC.WA = 040000	Q\$LBDP = 001001	T\$AD = 000020
BYTE31 = 000037	BYTE83 = 000123	LOC.WB = 100000	Q\$LBP = 000001	T\$BA = 000002
BYTE32 = 000040	BYTE84 = 000124	MAREN1 = 000001	Q\$LDCD = 000003	T\$BD = 000010
BYTE33 = 000041	BYTE85 = 000125	MAREN2 = 004000	Q\$LDMD = 000004	T\$B50 = 100000
BYTE34 = 000042	BYTE86 = 000126	MARLDD = 010000	Q\$LDPP = 000200	T\$BT = 000020
BYTE35 = 000043	BYTE87 = 000127	MAROUT = 000002	Q\$LHP = 010000	T\$BTAR = 000030
BYTE36 = 000044	BYTE88 = 000130	MAR.LO = 002000	Q\$MNC = 140000	T\$BTD = 002000
BYTE37 = 000045	BYTE89 = 000131	MAR.OU = 000040	Q\$MR = 000052	T\$CD = 000100
BYTE38 = 000046	BYTES = 000011	MBKALL = 001000	Q\$MRP = 000040	T\$C = 000100
BYTE39 = 000047	BYTE90 = 000132	MBKCLK = 000400	Q\$MRP2 = 000240	T\$C = 000100
BYTE4 = 000004	BYTE91 = 000133	MNADRD = 000100	Q\$MRC = 040000	T\$C = 000100
BYTE40 = 000005	BYTE92 = 000134	MNLEFT = 000002	Q\$MSET = 000004	T\$DISK = 000200

SYMBOL TABLE

T\$FSAA= 000000	T\$BBEN= 000020	WORD30= 000074	WORD55= 000156	WORD8 = 000020
T\$FSAB= 000004	UBD, IN= 000020	WORD31= 000076	WORD56= 000160	WORD80= 000240
T\$FSAC= 000014	VIRT. = ***** GX	WORD32= 000100	WORD57= 000162	WORD81= 000242
T\$FSB2= 000010	WORD0 = 000000	WORD33= 000102	WORD58= 000164	WORD82= 000244
T\$IB = 000026	WORD1 = 000002	WORD34= 000104	WORD59= 000166	WORD83= 000246
T\$IBAR= 000024	WORD10= 000024	WORD35= 000106	WORD6 = 000014	WORD84= 000250
T\$IBE = 020000	WORD11= 000026	WORD36= 000110	WORD60= 000170	WORD85= 000252
T\$IBF = 040000	WORD12= 000030	WORD37= 000112	WORD61= 000172	WORD86= 000254
T\$ICD = 000040	WORD13= 000032	WORD38= 000114	WORD62= 000174	WORD87= 000256
T\$NODE= 004000	WORD14= 000034	WORD39= 000116	WORD63= 000176	WORD88= 000260
T\$OB = 000036	WORD15= 000036	WORD4 = 000010	WORD64= 000200	WORD89= 000262
T\$OBE = 004000	WORD16= 000040	WORD40= 000120	WORD65= 000202	WORD9 = 000022
T\$OBF = 010000	WORD17= 000042	WORD41= 000122	WORD66= 000204	WORD90= 000264
T\$OBRA= 000034	WORD18= 000044	WORD42= 000124	WORD67= 000206	WORD91= 000266
T\$OBWA= 000032	WORD19= 000046	WORD43= 000126	WORD68= 000210	WORD92= 000270
T\$OUTA= 100000	WORD2 = 000004	WORD44= 000130	WORD69= 000212	WORD93= 000272
T\$RBD0= 000200	WORD20= 000050	WORD45= 000132	WORD7 = 000016	WORD94= 000274
T\$RNB = 000040	WORD21= 000052	WORD46= 000134	WORD70= 000214	WORD95= 000276
T\$RSET= 040000	WORD22= 000054	WORD47= 000136	WORD71= 000216	WORD96= 000280
T\$SC = 000022	WORD23= 000056	WORD48= 000140	WORD72= 000220	WORD97= 000302
T\$SCLK= 020000	WORD24= 000060	WORD49= 000142	WORD73= 000222	WORD98= 000304
T\$SEG1= 000000	WORD25= 000062	WORD5 = 000012	WORD74= 000224	WORD99= 000306
T\$SEG2= 000001	WORD26= 000064	WORD50= 000144	WORD75= 000226	WORDVAL= 000310
T\$SEG3= 000002	WORD27= 000066	WORD51= 000146	WORD76= 000230	XTREAD= 001000
T\$SD = 000001	WORD28= 000070	WORD52= 000150	WORD77= 000232	XTWRITE= 000400
T\$UBUS= 100000	WORD29= 000072	WORD53= 000152	WORD78= 000234	.CLOSE= ***** G
T\$1CLK= 000400	WORD3 = 000006	WORD54= 000154	WORD79= 000236	.OPEN = ***** G

. ABS. 000000 000
000000 001
CD. 000274 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3981 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:44
CD, EB, SP=C20, 1JIM, C20, 1JCD

```

1
2 000000 .TITLE TTXT
3 .PSECT TTXT
4
5
6
7
8
9
10
11
12
13
14
15
16
17 000000 TTXT:
18 000000 016700 000000G MOV TTSIZ,R0 ;GET-BYTE-SIZE-OF-SOFTWARE-TT
19 000004 006200 ASR R0 ;
20 000006 006200 ASR R0 ;NUMBER-OF-HQR-WORDS
21 000010 010067 000000G MOV R0,STTENT ;SAVE-FOR-LOOP-CONTROL
22 000014 010067 000000G MOV R0,PHXT ;ALSO-FOR-HQR-XLIST-START-ADDRESS
23 000020 010067 000000G MOV R0,NULXT ;ALSO-ADDRESS-OF-THE-NULL-XLIST
24
25
26
27
28 000024 012767 000000G 000000G MOV #LDCT,DATA ;SET-INITIAL-COUNT-WORD
29 000032 010067 000000G MOV R0,ADDR ;SET-NULL-XLIST-ADDRESS
30 000036 CALL ONEMD ;WRITE-FIRST-WORD-OF-ENTRY
31 000042 005267 000000G INC PHXT ;BUMP-HQR-XTABLE-POINTER
32
33 000046 012767 003777 000000G MOV #3777,DATA ;INNOCUOUS-OLB-ADDRESS(3FF,1)
34 000054 016767 000000G 000000G MOV PHXT,ADDR ;ADDRESS-OF-WORD-2-OF-DUMMY-XLIST
35 000062 CALL ONEMD ;WRITE-SECOND-WORD-OF-ENTRY
36 000066 005267 000000G INC PHXT ;BUMP-HQR-XTABLE-POINTER
37

```

```

39      ; WRITE THE ADDRESS OF THE NEXT AVAILABLE HQR XTABLE SLOT
40      ; (I.E., THE START OF AN XTABLE BLOCK) INTO THE HQR TTABLE.
41      ; IF FLU CODE IS USED; ELSE WRITE ADDRESS OF THE DUMMY XLIST.
42      ;
43      ; ONET:
44      000072 016700 000000G MOV PS TT,R0 ;POINT TO SOFT TTABLE.
45      000076 005760 000002 TST 2(R0) ;IS COUNT WORD = 0 ?
46      000102 001012 BNE 1# ;NO, CONTINUE (NOT A PLACE-HOLDER ENTRY)
47      ;
48      000104 016767 000000G-000000G MOV NULXT,DATA ;WRITE HQR DUMY XLIST ADDRESS.
49      000112 016767 000000G-000000G MOV PH TT,ADDR ; INTO HQR TTABLE.
50      000120 CALL ONEMD
51      000124 000167 000260 JMP BT ;BUMP TTABLE POINTERS.
52      ;
53      ; NON-TRIVIAL FLU CODE.
54      ;
55      000130 016767 000000G-000000G-1# MOV PH XT,DATA ;WRITE HQR XTABLE ADDRESS.
56      000136 016767 000000G-000000G MOV PH TT,ADDR ; INTO HQR TTABLE.
57      000144 CALL ONEMD
58      ;
59      ; MOVE XLIST COUNT FROM THE SOFT TTABLE (STT) TO THE HQR XTABLE.
60      ;
61      000150 016700 000000G MOV PS TT,R0 ;R0 -> SOFTWARE TT.
62      000154 016001 000002 MOV 2(R0),R1 ;GET XLIST COUNT.
63      000160 005301 DEC R1 ;ADJUST FOR HQR.
64      000162 052701 000000G BIS #LDCT,R1 ;'OR' IN AMD2900 LDCT INSTR.
65      000166 010167 000000G MOV R1,DATA ;WRITE COUNT TO HQR XTABLE.
66      000172 016767 000000G-000000G MOV PH XT,ADDR ;HQR XT ADDRESS.
67      000200 CALL ONEMD
68      000204 005267 000000G INC PH XT ;BUMP HQR XT POINTER
69      ;
70      ;
71      ; CONVERT THE XTABLE ENTRIES FOR ONE FLU FROM SOFTWARE
72      ; TO HARDWARE FORMAT.
73      000210 016001 000002 MOV 2(R0),R1 ;GET XLIST COUNT FOR LOOP CONTROL.
74      000214 011000 MOV (R0),R0 ;GET BYTE OFFSET TO XLIST.
75      000216 032760 000000G-000000G-ONEX BIT #SXTQEX,XTABLE(R0) ;DEX POINTER ?
76      000224 001430 BEQ QLBX ;NO, MUST BE QLB POINTER.
77      ;

```

```

79      ;          CONVERT A QEX POINTER.
80      ;
81      ;
82 000226 016002 000000G      MOV      XTABLE(R0),R2      ;GET OFFSET TO QEX POINTER.
83 000232 066702 000000G      ADD      QXADD,R2          ;ADD QEX MEMORY START ADDRESS.
84 000236 010267 000000G      MOV      R2,DATA          ;PREPARE TO WRITE HQR XLIST.
85      ;
86      ;
87      ;          IF THE QEX POINTER IS THE FIRST OF A PAIR.
88      ;          THEN SET THE HQR QEX POINTER MARK.
89 000242 005267 000000G      INC      QEXCT            ;COUNT QEX POINTER PAIRS.
90 000246 022767 000001 000000G  CMP      #1,QEXCT         ;FIRST OF PAIR?
91 000254 001407 000000G      BEQ      10$             ;YES, INSERT QEX MARK.
92 000256 005067 000000G      CLR      QEXCT          ;ELSE CLEAR COUNTER.
93 000262 042767 000000G 000000G  BIC      #QXCNTL,DATA     ;AND CLEAR QEX BIT.
94 000270 000167 000062 000000G  JMP      WRITEX          ;AND WRITE HQR XTABLE.
95 000274 052767 000000G 000000G 10$  BIS      #QXCNTL,DATA     ;SET QEX POINTER MARK FOR HQR.
96 000302 000167 000050 000000G  JMP      WRITEX          ;WRITE XTABLE.
97      ;
98      ;
99      ;          CONVERT A QLB OR AN SDLB POINTER
100     ;
101     ;
102     ;          R0 = BYTE OFFSET FROM TOP OF SXT TO QLB POINTER.
103     ;
104     ;          CLEAR THE LOCAL 'LAST MULTI-NOT' FLAG.
105     ;          COMPARE THE XTABLE ENTRY WITH THE SOFTWARE QLB ADDRESS.
106     ;          TABLE ENTRIES (SQAT). SET THE INTERNAL QID (IQID) = TO
107     ;          THE NUMBER OF SQAT ENTRIES WHOSE VALUES WERE LESS THAN
108     ;          THE XTABLE ENTRY - 1.
109     ;
110 000306 005067 000000G      0_LBX: CLR      QEXCT          ;CLEAR QEX PAIRS COUNTER.
111 000312 000312 000000G      SAVE   R2
112     ;
113 000314 016002 000000G      MOV      XTABLE(R0),R2   ;GET OFFSET INTO QLB OR SDLB.
114 000320 032702 000000G      BIT      #SXTSLB,R2     ;IS THIS AN SDLB POINTER?
115 000324 000324 000000G      BRNCH  10$             ;BRANCH IF YES.
116     ;
117 000326 000326 000000G      CALL   HMAPQ           ;GET QLB HARDWARE ADDRESS.
118 000332 000406 000000G      BR     20$             ;SET UP FOR XTABLE LOAD.
119 000334 000334 000000G      10$:
120 000334 042702 000000G      BIC      #SXTSLB,R2     ;CLEAR SDLB POINTER FLAG.
121 000340 000340 000000G      CALL   HMAPS           ;GET SDLB HARDWARE ADDRESS.
122 000344 052702 000000G      BIS      #HXTSLB,R2     ;SET HARDWARE XLIST SLB POINTER FLAG.
123 000350 000350 000000G      20$:
124 000350 010267 000000G      MOV      R2,DATA        ;PREPARE TO WRITE.
125 000354 000354 000000G      RESTOR R2
126     ;

```



```

128. ; WRITE XTABLE ENTRY INTO HQR.
129. ; UPDATE POINTERS TO SXT AND HXT.
130. ; DECREMENT XTABLE COUNT (FOR CURRENT TTBALE ENTRY)
131. ; IF COUNT NOT = 0
132. ; THEN GOTO 'ONEX' (PROCESS NEXT XTABLE ENTRY)
133. ;
134. ;
135. 000356 016767 000000G 000000G WRITEX: MOV. PHXT,ADDR. ;HQR XT ADDRESS.
136. 000364 CALL. ONEMD.
137. 000370 005267 000000G INC. PHXT ;BUMP HQR XT POINTER
138. 000374 062700 000002 ADD. #2,R0 ;BUMP SXT POINTER.
139. 000400 005301 DEC. R1 ;DEC SXT COUNT.
140. 000402 001402 BEQ. BT. ;FINISHED WITH ONE SXT ENTRY.
141. 000404 000167 177606 JMP. ONEX ;PROCESS NEXT SXT ENTRY.
142. ;
143. ;
144. ; UPDATE POINTERS TO HTT AND STT.
145. ; DECREMENT STT ENTRY COUNT.
146. ; IF COUNT NOT = 0
147. ; THEN GOTO 'ONET' (PROCESS NEXT STT ENTRY)
148. 000410 005267 000000G BTT: INC. PHTT ;BUMP HQR TTABLE POINTER.
149. 000414 062767 000004 000000G ADD. #4,PSTT. ;BUMP STT POINTER.
150. 000422 005367 000000G DEC. STTENT. ;FINISHED WITH STT.?
151. 000426 001402 BEQ. 1# ;YES.
152. 000430 000167 177436 JMP. ONET
153. ;
154. 000434 000241 15: CLC.
155. 000436 RETURN.
156. ;
157. ;
158. ; EXIT ON ERROR.
159. ;
160. 000440 000261 TTXTSX: SEC.
161. 000442 RETURN.
162. ;
163. 000001 .END.

```

ADDR = ***** GX	BYTE40 = 000050	BYTE92 = 000134	PHTT = ***** GX	Q\$PPSW = 000320
ALUCKE = 000000	BYTE41 = 000051	BYTE93 = 000135	PHXT = ***** GX	Q\$PP2 = 000300
ALUDE = 000000	BYTE42 = 000052	BYTE94 = 000136	PLB = 000010	Q\$PHLT = 000013
A01 = 010000	BYTE43 = 000053	BYTE95 = 000137	PLC = 000020	Q\$QL = 000043
BITVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	PLD = 000030	Q\$QLA = 000053
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	PLRWR = 000200	Q\$QLB = 000054
BIT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	PLREN = 000200	Q\$QLR = 000001
BIT10 = 002000	BYTE47 = 000057	BYTE99 = 000143	PSTT = ***** GX	Q\$QW = 000042
BIT11 = 004000	BYTE48 = 000060	BYTVAL = 000144	QEXCT = ***** GX	Q\$RDCD = 000005
BIT12 = 010000	BYTE49 = 000061	CBKALL = 000100	QLBX = 000306R	002 Q\$RDMD = 000006
BIT13 = 020000	BYTE50 = 000062	CHOBRE = 100000	QR\$CR1 = 176420	Q\$REBK = 001000
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	QR\$CR2 = 176422	Q\$RNC = 006000
BIT15 = 100000	BYTE52 = 000064	CPREAD = 040000	QR\$LBR = 176424	Q\$RSC = 004000
BIT2 = 000004	BYTE53 = 000065	CPURTE = 020000	QXADD = ***** GX	Q\$RSET = 000010
BIT3 = 000010	BYTE54 = 000066	CSADRD = 000004	QXCNTL = ***** GX	Q\$SM = 100000
BIT4 = 000020	BYTE55 = 000067	CSEQCI = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BIT5 = 000040	BYTE56 = 000070	CSOE = 000040	Q\$BCL = 000001	Q\$SP2 = 000340
BIT6 = 000100	BYTE57 = 000071	CSURTE = 000100	Q\$CCCP = 000040	RG0.EN = 000200
BIT7 = 000200	BYTE58 = 000072	DATA = ***** GX	Q\$CHB = 000400	RG0.VA = 020000
BIT8 = 000400	BYTE59 = 000073	DBR.RD = 000001	Q\$CHRL = 000200	SEQ.CI = 000010
BIT9 = 001000	002 BYTE60 = 000074	DB\$CPP = 001457	Q\$CLR = 000040	STTENT = ***** GX
BT = 000410R	BYTE61 = 000075	DB\$SPT = 000026	Q\$CNC = 030000	SXTOEX = ***** GX
BYTE0 = 000000	BYTE62 = 000076	DB\$TPC = 000023	Q\$CP = 000050	SXTSLB = ***** GX
BYTE1 = 000001	BYTE63 = 000077	DISPGS = 100000	Q\$CPC = 000010	S\$CLR = 000000
BYTE10 = 000012	BYTE64 = 000100	DMAUR = 000005	Q\$CP2 = 000260	S\$LA = 000001
BYTE11 = 000013	BYTE65 = 000101	DMARRD = 000003	Q\$CSC = 010000	S\$OB = 000005
BYTE12 = 000014	BYTE66 = 000102	DMARUR = 000004	Q\$CSEL = 000350	S\$QR = 000006
BYTE13 = 000015	BYTE67 = 000103	ENBR = 010000	Q\$CSET = 000002	S\$OX = 000004
BYTE14 = 000016	BYTE68 = 000104	HMAP0 = ***** GX	Q\$CSP = 020000	S\$SR = 000007
BYTE15 = 000017	BYTE69 = 000105	HMAPS = ***** GX	Q\$DMA = 000001	S\$S1 = 000010
BYTE16 = 000020	BYTE70 = 000106	HXTSLB = ***** GX	Q\$ENBK = 040000	S\$S2 = 000014
BYTE17 = 000021	BYTE71 = 000107	LDCT = ***** GX	Q\$ENOP = 020000	TD\$CTR = 176370
BYTE18 = 000022	BYTE72 = 000110	LOC.EN = 000100	Q\$FAL = 004000	TD\$CTW = 176360
BYTE19 = 000023	BYTE73 = 000111	LOC.WA = 040000	Q\$FC = 000045	TD\$INL = 004000
BYTE2 = 000002	BYTE74 = 000112	LOC.WB = 100000	Q\$FO = 000044	TD\$MEM = 000270
BYTE20 = 000024	BYTE75 = 000113	MAREN1 = 000001	Q\$FP = 000046	TD\$OAR = 176344
BYTE21 = 000025	BYTE76 = 000114	MAREN2 = 000000	Q\$HBF = 000002	TD\$QTR = 176346
BYTE22 = 000026	BYTE77 = 000115	MARLOD = 010000	Q\$ICP = 000006	TD\$QRD = 000274
BYTE23 = 000027	BYTE78 = 000116	MAROUT = 000002	Q\$IHB = 000003	TD\$R = 176376
BYTE24 = 000030	BYTE79 = 000117	MAR.LO = 000000	Q\$IHL = 000002	TD\$TAR = 176372
BYTE25 = 000031	BYTE80 = 000120	MAR.OU = 000040	Q\$IMRP = 000007	TD\$TAW = 176362
BYTE26 = 000032	BYTE81 = 000121	MBKALL = 000100	Q\$LBD = 001000	TD\$TDR = 176374
BYTE27 = 000033	BYTE82 = 000122	MBKCLK = 000400	Q\$LBDP = 001001	TD\$TDW = 176364
BYTE28 = 000034	BYTE83 = 000123	MHADRD = 000100	Q\$LBP = 000001	TT\$IZ = ***** GX
BYTE29 = 000035	BYTE84 = 000124	MILEFT = 000002	Q\$LDCD = 000003	TTXT 000000R6 002
BYTE3 = 000003	BYTE85 = 000125	MMD = 000004	Q\$LDMD = 000004	TTXTSX 000440R 002
BYTE30 = 000036	BYTE86 = 000126	MMDRTE = 000010	Q\$LDPP = 002000	T\$AD = 000020
BYTE31 = 000037	BYTE87 = 000127	MNOBRE = 100000	Q\$LHP = 000100	T\$BA = 000002
BYTE32 = 000040	BYTE88 = 000130	MREN1 = 000001	Q\$MNC = 140000	T\$BD = 000010
BYTE33 = 000041	BYTE89 = 000131	MREN2 = 020000	Q\$MR = 000052	T\$BSO = 100000
BYTE34 = 000042	BYTE90 = 000132	MSYN = 000040	Q\$MRP = 000040	T\$BT = 000020
BYTE35 = 000043	BYTE91 = 000133	N = 000144	Q\$MRP2 = 000240	T\$BTAR = 000030
BYTE36 = 000044		NULXT = ***** GX	Q\$MSC = 040000	T\$BD = 002000
BYTE37 = 000045		ONEMD = ***** GX	Q\$MSET = 000004	T\$CD = 000100
BYTE38 = 000046		ONET = 000072R	Q\$MSP = 100000	T\$CLK = 002000
BYTE39 = 000047		ONEX = 000216R	002 Q\$NCLK = 176000	TD\$ISK = 000200
BYTE4 = 000004			002 Q\$PP = 000100	T\$DRD = 000004

T\$EMEM- 010000	T\$1CLK- 000400	WORD30- 000074	WORD55- 000156	WORD8 = 000020
T\$FSAA- 000000	T\$BEN- 000020	WORD31- 000076	WORD56- 000160	WORD80- 000240
T\$FSAB- 000004	UBD IN- 000020	WORD32- 000100	WORD57- 000162	WORD81- 000242
T\$FSAC- 000014	WORD0 = 000000	WORD33- 000102	WORD58- 000164	WORD82- 000244
T\$FSB2- 000010	WORD1 = 000002	WORD34- 000104	WORD59- 000166	WORD83- 000246
T\$IB = 000026	WORD10- 000024	WORD35- 000106	WORD6 = 000014	WORD84- 000250
T\$IBAR- 000024	WORD11- 000026	WORD36- 000110	WORD60- 000170	WORD85- 000252
T\$IBE = 020000	WORD12- 000030	WORD37- 000112	WORD61- 000172	WORD86- 000254
T\$IBF = 040000	WORD13- 000032	WORD38- 000114	WORD62- 000174	WORD87- 000256
T\$ICD = 000040	WORD14- 000034	WORD39- 000116	WORD63- 000176	WORD88- 000260
T\$HODE- 004000	WORD15- 000036	WORD4 = 000010	WORD64- 000200	WORD89- 000262
T\$OB = 000036	WORD16- 000040	WORD40- 000120	WORD65- 000202	WORD9 = 000022
T\$OBE = 004000	WORD17- 000042	WORD41- 000122	WORD66- 000204	WORD90- 000264
T\$OBF = 010000	WORD18- 000044	WORD42- 000124	WORD67- 000206	WORD91- 000266
T\$OBRA- 000034	WORD19- 000046	WORD43- 000126	WORD68- 000210	WORD92- 000270
T\$OBWA- 000032	WORD2 = 000004	WORD44- 000130	WORD69- 000212	WORD93- 000272
T\$OUTA- 100000	WORD20- 000050	WORD45- 000132	WORD7 = 000016	WORD94- 000274
T\$RBD0- 000200	WORD21- 000052	WORD46- 000134	WORD70- 000214	WORD95- 000276
T\$RNB = 000040	WORD22- 000054	WORD47- 000136	WORD71- 000216	WORD96- 000276
T\$RSET- 040000	WORD23- 000056	WORD48- 000140	WORD72- 000220	WORD97- 000300
T\$SC = 000022	WORD24- 000060	WORD49- 000142	WORD73- 000222	WORD98- 000304
T\$SCLK- 020000	WORD25- 000062	WORD5 = 000012	WORD74- 000224	WORD99- 000306
T\$SEG1- 000000	WORD26- 000064	WORD50- 000144	WORD75- 000226	WRDVAL- 000310
T\$SEG2- 000001	WORD27- 000066	WORD51- 000146	WORD76- 000230	WRITEX- 000356R 002
T\$SEG3- 000002	WORD28- 000070	WORD52- 000150	WORD77- 000232	XTABLE- ***** GX
T\$SD = 000001	WORD29- 000072	WORD53- 000152	WORD78- 000234	XTREAD = 001000
T\$UBUS- 100000	WORD3 = 000006	WORD54- 000154	WORD79- 000236	XTWRTE = 000400

. ABS. 000000 000
000000 001
TTXT: 000444 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3135 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
TTXT>TTXT<-SP=[20,1]IM,[20,1]TTXT

```

1
2 000000 .TITLE: QCLFAL
3 .PSECT: QCLFAL
4
5
6
7
8
9
10
11
12
13
14
15
16 QCLFAL:
17 000000 016700 000000G MOV: QRYMAX,R0 ;GET NUMBER OF WORDS IN QCL
18 000004 005200 INC: R0 ;QUERY NUMBERS START AT ZERO
19 000006 005001 CLR: R1 ;QCL/HEAT OFFSET
20 000010 005067 000000G CLR: IQID ;QUERY ID
21 000014 016767 000000G-000000G QCLF2: MOV: PHFAL,DATA ;MOVE HQR FAL ADDR TO HQR QCL
22 000022 016767 000000G-000000G MOV: PHOCL,ADDR ;SET UP HQR QCL ADDRESS
23 000030 000000G CALL: ONECD ;GENERATE HQR QCL ENTRY
24
25
26
27
28
29 000034 017702 000000G MOV: @PSOCL,R2 ;GET SOCL ENTRY (= OFFSET INTO FAL)
30 000040 042702 100000 BIC: #BIT15,R2 ;ALLOW FOR POSSIBLE UHL FLAG
31 000044 062702 000000G ADD: #FAL,R2 ;ADD FAL START ADDRESS
32 000050 012203 MOV: (R2)+,R3 ;GET FLU ID COUNT FROM SOFT FAL
33 000052 003413 BLE: 2$ ;BRANCH IF NULL QUERY
34 000054 012267 000000G 1$: MOV: (R2)+,DATA ;MOVE FLU ID TO WORD FIELD
35 000060 016767 000000G-000000G MOV: PHFAL,ADDR ;HQR FAL ENTRY ADDRESS
36 000066
37 000072 005267 000000G INC: PHFAL ;INC HQR FAL POINTER
38 000076 005303 DEC: R3 ;DEC FLU ID COUNT
39 000100 001365 BNE: 1$
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55 000102 2$:
56 000102 032777 000000G-000000G BIT: #SHRL,@PSOCL ;TEST SOCL ENTRY FOR HRL BIT
57 000110 BON: MARK ;PUT HRL MARK POINTER IN FAL

```

```

58 000112 012767 001777G.000000G. MOV.   *(<HRLPNT+1777>).DATA.  ;POINT TO DUMMY WORD IN HOR OLB
59 000120 016767 000000G.000000G. MOV.   PHFAL,ADDR.          ;NEXT HOR FAL ENTRY ADDRESS
60 000126                                     CALL.  ONEFA.              ;
61 000132 005267 000000G.          INC.   PHFAL.              ;BUMP HOR FAL POINTER
62.                                     ;
63.                                     ;
64.                                     ;
65 000136 116777 000000G.000000G. MOV.   IQID,@PNUHRL.        ;PUT QID IN DUMMY HRL ENTRY
66 000144 005267 000000G.          INC.   PNUHRL.            ;BUMP POINTER
67 000150 105267 000000G.          INCB.  NUHRL.             ;INCREMENT QID COUNT
68 000154 000415                                     BR     JOIN3
69.                                     ;
70.                                     ;
71.                                     ;
72.                                     ;
73 000156                                     MARK:
74 000156 016167 000000G.000000G. MOV.   HEAT(R1).DATA.      ;GET ADDRESS OF HRL WORD IN OLB
75 000164 062767 004000G.000000G. ADD.   *(<04000+HRLPNT>).DATA. ;ADD BIT ADDR - BIT 2
76 000172 016767 000000G.000000G. MOV.   PHFAL,ADDR.        ;FAL ADDRESS
77 000200                                     CALL.  ONEFA.              ;
78 000204 005267 000000G.          INC.   PHFAL.              ;BUMP HOR FAL POINTER
79.                                     ;
80.                                     ;
81.                                     ;
82.                                     ;
83.                                     ;
84.                                     ;
85.                                     ;
86 000210                                     JOIN3:
87 000210 005267 000000G.          INC.   PHOCL.             ;BUMP POINTER TO HOR OCL
88 000214 005300          DEC.   R0.                ;SUB FROM OCL ENTRY COUNT
89 000216 001410          BEQ.   1$.                ;FINISHED
90 000220 005267 000000G.          INC.   IQID.             ;SET NEXT QUERY ID
91 000224 005721          TST.   (R1)+.            ;INCREMENT OCL/HEAT OFFSET
92 000226 062767 000002 000000G.  ADD.   #2,PSOCL.         ;BUMP POINTER TO SOCL
93 000234 000167 177554          JMP.   QCLF2.            ;CONVERT NEXT SET OF FAL ENTRIES
94.                                     ;
95.                                     ;
96.                                     ;
97.                                     ;
98.                                     ;
99.                                     ;
100 000240 012767 077777 000000G. 1$: MOV.   #077777.DATA.      ;HALT CODE - X'7777'
101 000246 016767 000000G.000000G. MOV.   PHOCL,ADDR.        ;OCL ADDRESS
102 000254                                     CALL.  ONECD.              ;
103.                                     ;
104.                                     ;
105.                                     ;
106.                                     ;
107.                                     ;
108 000260 105767 000000G.          TSTB.  NUHRL.            ;ANY NO-UHL HRL QID'S TABLED
109 000264 001011          BNE.   2$.                ;YES, WRITE TO CP DATA MEMORY
110 000266 012767 000000G.000000G. MOV.   #ALLHRL,ADDR.      ;ADDR OF ALLHRL FLAG IN CP DATA MEMORY
111 000274 012767 000001 000000G.  MOV.   #1,DATA.          ;SET FLAG
112 000302                                     CALL.  ONECD.              ;
113 000306 000430          BR     QFEX.             ;
114.                                     ;

```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

115      : MOVE THE NO-UHL DUMMY HRL ENTRY TO CP DATA MEMORY.
116      :
117 000310 116700 000000G. 2#: MOVB NUHRL,R0      ;GET NO-UHL HRL QID COUNT.
118 000314 006200      ASR R0
119 000316 005200      INC R0      ;GET NUMBER OF WORDS TO TRANSFER.
120 000320 012701 000000G.      MOV #NUHRL,R1      ;R1 -> NO-UHL HRL.
121 000324 012702 000000G.      MOV #NUHLCD,R2      ;R2 -> ADDRESS IN CP DATA MEMORY.
122 000330 012167 000000G. 3#: MOV (R1)+,DATA      ;NO-UHL HRL WORD TO BE WRITTEN
123 000334 010267 000000G.      MOV R2,ADDR      ;TO SLOT IN CP DATA MEMORY.
124 000340      CALL ONECD
125 000344 005202      INC R2      ;BUMP POINTER.
126 000346 005300      DEC R0      ;DEC WORD COUNT.
127 000350 001367      BNE 3#
128      :
129      : CLEAR THE 'ALLHRL' FLAG IN CP DATA MEMORY.
130      :
131 000352 012767 000000G.000000G.      MOV #ALLHRL,ADDR      ;ADDR OF ALLHRL FLAG IN CP DATA MEMORY.
132 000360 005067 000000G.      CLR DATA      ;CLEAR FLAG.
133 000364      CALL ONECD
134      :
135 000370      QFEX: RETURN
136      :
137      000001      .END
    
```

SYMBOL TABLE
ADDR. = ***** GX.
ALLHRL = ***** GX.
ALUCKE = 000000
ALUOE = 0004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004

BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133

BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CHOBRE = 100000
CPCCEN = 010000
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSURTE = 000100
DATA = ***** GX.
DBR.RD = 000001
DB*CPP = 001457
DB*SPT = 000026
DB*TPC = 000023
DISPGS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
FAL = ***** GX.
HEAT = ***** GX.
HRLPNT = ***** GX.
IQID = ***** GX.
JOIN3 = 000210R.
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARK = 000156R.
MARLOD = 010000
MAROUT = 000002
MAR.LO = 000200
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MMADRD = 000100
MMLEFT = 000002
MMOE = 000004
MMURTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
NUHLCD = ***** GX.
NUHRL = ***** GX.

ONECD = ***** GX.
ONEFA = ***** GX.
PHFAL = ***** GX.
PHOCL = ***** GX.
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
PNUHRL = ***** GX.
PSOCL = ***** GX.
QCLFAL = 000000RG.
QCLF2 = 000014R.
QFEX = 000370R.
QRYMAX = ***** GX.
QR*CR1 = 176420
QR*CR2 = 176422
QR*LBR = 176424
Q*ATTN = 000100
Q*BCL = 000001
Q*CCCP = 000040
Q*CHB = 000400
Q*CHRL = 000200
Q*CLR = 000040
Q*CNC = 030000
Q*CP = 000060
Q*CPCC = 000010
Q*CP2 = 000260
Q*CSC = 010000
Q*CSEL = 000360
Q*CSET = 000002
Q*CSP = 020000
Q*DMA = 000001
Q*ENBK = 040000
Q*ENOP = 020000
Q*FAL = 004000
Q*FC = 000045
Q*FO = 000044
Q*FP = 000046
Q*HBF = 000002
Q*ICP = 000006
Q*IHB = 000003
Q*IHRL = 000002
Q*IMRP = 000007
Q*LBD = 001000
Q*LBDP = 001001
Q*LBP = 000001
Q*LCD = 000003
Q*LDMD = 000004
Q*LDPP = 002000
Q*LHP = 010000
Q*HNC = 140000
Q*HR = 000052
Q*MRP = 000040
Q*MRP2 = 0000240
Q*MSC = 040000
Q*MSSET = 000004

Q*MSP = 100000
Q*NCCLK = 176000
Q*PP = 000100
Q*PPSW = 000320
Q*PP2 = 000300
Q*QHLT = 000013
Q*QL = 000043
Q*QLA = 000053
Q*QLB = 000054
Q*QLR = 000001
Q*QW = 000042
002 Q*RD CD = 000005
002 Q*RDMD = 000006
002 Q*REBK = 001000
Q*RNC = 006000
Q*RSR = 004000
Q*RSCT = 000010
Q*SM = 100000
Q*SP = 000120
Q*SP2 = 000340
RGQ.EN = 000200
RGQ.VA = 020000
SEQ.CI = 000010
SHRL = ***** GX.
S*CLR = 000000
S*LA = 000001
S*OB = 000005
S*OR = 000006
S*OX = 000004
S*SR = 000007
S*S1 = 000010
S*S2 = 000014
TD*CTR = 176370
TD*CTW = 176360
TD*INL = 004000
TD*MEM = 000270
TD*ORD = 176344
TD*QTR = 176346
TD*QDR = 000274
TD*SW = 176376
TD*STAR = 176372
TD*TAJW = 176362
TD*TDR = 176374
TD*TDJW = 176364
T*AD = 000020
T*BA = 000002
T*BD = 000010
T*BSO = 100000
T*BT = 000020
T*BTAR = 000030
T*BTB = 002000
T*CD = 000100
T*CLK = 002000
T*DISK = 000200
T*DRD = 000004
T*HEM = 010000
T*FSAA = 000000

T\$FSAB= 000004	UBD.IN= 000020	WORD31= 000076	WORD55= 000156	WORD79= 000236
T\$FSAC= 000014	WORD0 = 000000	WORD32= 000100	WORD56= 000160	WORD8 = 000020
T\$FSB2= 000010	WORD1 = 000002	WORD33= 000102	WORD57= 000162	WORD80= 000240
T\$IB = 000026	WORD10= 000024	WORD34= 000104	WORD58= 000164	WORD81= 000242
T\$IBAR= 000024	WORD11= 000026	WORD35= 000106	WORD59= 000166	WORD82= 000244
T\$IBE = 020000	WORD12= 000030	WORD36= 000110	WORD6 = 000014	WORD83= 000246
T\$IBF = 040000	WORD13= 000032	WORD37= 000112	WORD60= 000170	WORD84= 000250
T\$ICD = 000040	WORD14= 000034	WORD38= 000114	WORD61= 000172	WORD85= 000252
T\$MODE= 004000	WORD15= 000036	WORD39= 000116	WORD62= 000174	WORD86= 000254
T\$OB = 000036	WORD16= 000040	WORD4 = 000010	WORD63= 000176	WORD87= 000256
T\$OBE = 004000	WORD17= 000042	WORD40= 000120	WORD64= 000200	WORD88= 000260
T\$OBF = 010000	WORD18= 000044	WORD41= 000122	WORD65= 000202	WORD89= 000262
T\$OBRA= 000034	WORD19= 000046	WORD42= 000124	WORD66= 000204	WORD9 = 000022
T\$OBWA= 000032	WORD2 = 000004	WORD43= 000126	WORD67= 000206	WORD90= 000264
T\$OUTA= 100000	WORD20= 000050	WORD44= 000130	WORD68= 000210	WORD91= 000266
T\$RBD0= 000200	WORD21= 000052	WORD45= 000132	WORD69= 000212	WORD92= 000270
T\$RNB = 000040	WORD22= 000054	WORD46= 000134	WORD7 = 000016	WORD93= 000272
T\$RSET= 040000	WORD23= 000056	WORD47= 000136	WORD70= 000214	WORD94= 000274
T\$SC = 000022	WORD24= 000060	WORD48= 000140	WORD71= 000216	WORD95= 000276
T\$SCLK= 020000	WORD25= 000062	WORD49= 000142	WORD72= 000220	WORD96= 000300
T\$SEG1= 000000	WORD26= 000064	WORD5 = 000012	WORD73= 000222	WORD97= 000302
T\$SEG2= 000001	WORD27= 000066	WORD50= 000144	WORD74= 000224	WORD98= 000304
T\$SEG3= 000002	WORD28= 000070	WORD51= 000146	WORD75= 000226	WORD99= 000306
T\$SO = 000001	WORD29= 000072	WORD52= 000150	WORD76= 000230	WRDVAL= 000310
T\$UBUS= 100000	WORD3 = 000006	WORD53= 000152	WORD77= 000232	XTREAD= 001000
T\$1CLK= 000400	WORD30= 000074	WORD54= 000154	WORD78= 000234	XTURTE= 000400
T\$0BEN= 000020				

. ABS. 000000 000
000000 001
QCLFAL 000372 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3112 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
QCLFAL QCLFAL /-SP=[20,131M,[20,130]QCLFAL


```

1
2 000000 .TITLE..SDLBLD.....
3 .PSECT..SDLBLD.
4
5 ;
6 ;
7 ;
8 ;
9 ;
10 000000 SDLBLD::
11 000000 012767 000000G 000000G MOV #SDLB,PSQLB ;SET POINTER TO FIRST BLOCK IN SDLB.
12 000006 005067 000000G CLR GFLAG ;CLEAR LOCAL FLAGS.
13 000012 005067 000000G CLR LXQID ;INITIALIZE LAST QLB POINTER.
14 000016 005067 000000G CLR XQID ;INITIALIZE QLB POINTER.
15 000022 005067 000000G CLR IQID ;INITIALIZE QLB BLOCK INDEX.
16 000026 005067 000000G CLR WQW ;INITIALIZE SDLB MEMORY DATA WORD.
17 000032 005067 000000G CLR CHQA ;TEMPORARY SDLB MEMORY START.
18 000036 012767 000001 000000G MOV #BIT0,BITPOS ;SET INITIAL BIT POSITION.
19 000044
20 000044 005767 000000G QSTRT: TST SDLBAD ;FINISHED PROCESSING SDLB?.
21 000050 003440 BLE QLBPT ;BRANCH IF YES.
22
23 ;
24 ;
25 ;
26 ;
27 000052 016700 000000G MOV PSQLB,R0 ;R0 -> SOLB ENTRY.
28 000056 112067 000000G MOVB (R0)+,QENTL ;GET LENGTH OF ENTRY.
29 000062 042767 177400 000000G BIC #177400,QENTL ;CLEAR UPPER BITS.
30 000070 166767 000000G 000000G SUB QENTL,SDLBAD ;ADJUST REMAINING SDLB SPACE.
31 000076 066700 000000G ADD QENTL,R0 ;POINT TO QID AT END OF SOLB ENTRY.
32 000102 032700 000001 BIT #BIT0,R0 ;IS ADDRESS ON WORD BOUNDARY?
33 000106 BOFF 1$ ;BRANCH IF SO.
34 000110 005200 INC R0 ;SKIP PAD BYTE.
35 000112 005367 000000G DEC SDLBAD ;ALSO ADJUST SDLB SPACE.
36 000116
37 000116 012067 000000G 1$: MOV (R0)+,XQID ;SAVE AS EXTERNAL QID.
38 000122 162767 000003 000000G SUB #3,SDLBAD ;ADJUST FOR LENGTH + QLB POINTER.
39 000130 010067 000000G MOV R0,NPSQLB ;SAVE AS POINTER TO NEXT ENTRY.
40
41 000134
42 000134 026767 000000G 000000G QLBLT: CMP LXQID,XQID ;PREVIOUS = CURRENT?.
43 000142 002403 BLT QLBPT ;LAST QLB OR DUMMY QUERY.
44 000144 001411 BEQ QLB.2 ;NOT LAST QLB OF QUERY.
45 000146 000167 000540 JMP QLBXT ;GENERATE TERMINAL WORD OF QLB.
46 000152 QLBPT:
47 000152 032767 000000G 000000G BIT #PQLB,GFLAG ;WAS THERE A QLB BLOCK FOR QUERY?
48 000160 BON ;BRANCH IF YES.
49 000162 005267 000000G INC LXQID ;SET NEXT QUERY ID.
50 000166 000762 BR QBLT ;CATCH ALL SKIPPED QID'S.
51
52 000170
53 000170 052767 000000G 000000G QLB.2: BIS #PQLB,GFLAG ;SET QLB PROCESSED FLAG.
54 000176 016700 000000G MOV PSQLB,R0 ;R0 -> SOLB ENTRY.
55 000202 005200 INC R0 ;BUMP PAST ENTRY LENGTH.

```

```

57          :          CONVERT ONE SQLB ENTRY ELEMENT BY ELEMENT.
58          :
59          :
60 000204 132710 000000G. ELEMNT: BITB.  #B$FMN.(R0)          :FIRST MULTI-NOT.?
61 000210 001432.      BEQ.   NOTFMN.          :NO.
62          :
63          :          FIRST MULTI-NOT BIT SET IN SDLB ELEMENT.
64          :
65 000212. 032767 000000G.000000G. BIT.   #LMN.GFLAG.          :WAS PREVIOUS ELEMENT A MULTI-NOT.
66 000220. 001411      BEQ.   1$.          :NO.
67 000222. 052767 000000G.000000G. BIS.   #SHLMN.WQW.          :SET HQR LAST MULTI-NOT FLAG.
68 000230.      CALL.  SLBREF.          :WRITE WORD TO HQR SDLB.
69 000234 052767 000000G.000000G. BIS.   #SHMN.WQW.          :SET MULTI-NOT FLAG IN NEW WORKING SDLB WORD.
70 000242. 000440      BR.    JOIN.
71          :
72          :          FIRST MULTI-NOT BIT SET IN SDLB ELEMENT AND PREVIOUS
73          :          ELEMENT WAS NOT A MULTI-NOT.
74          :
75 000244 052767 000000G.000000G. 1$. BIS.   #LMN.GFLAG.          :SET LOCAL FLAG FOR MULTI-NOT.
76 000252. 032767 000001.000000G. BIT.   #BIT0.BITPOS.          :IS WQW EMPTY?.
77 000260.      BON.   JOIN.          :PROCESS CURRENT ELEMENT.
78 000262.      CALL.  SLBREF.          :WRITE REF PAGE WORD.
79 000266 052767 000000G.000000G. BIS.   #SHMN.WQW.          :SET MULTI-NOT FLAG IN NEW WQW.
80 000274 000423      BR.    JOIN.          :PROCESS CURRENT ELEMENT.
81          :
82          :          FIRST MULTI-NOT BIT NOT SET IN SDLB ELEMENT.
83          :
84 000276 132710 000000G. NOTFMN: BITB.  #B$MUL.(R0)          :IS THIS ELEMENT A MULTI-NOT.
85 000302. 001404      BEQ.   1$.          :NO.
86 000304 052767 000000G.000000G. BIS.   #SHMN.WQW.          :SET HQR MULTI-NOT FLAG.
87 000312. 000414      BR.    JOIN.          :PROCESS CURRENT ELEMENT.
88          :
89          :          FIRST MULTI-NOT BIT NOT SET IN SDLB ELEMENT AND ELEMENT
90          :          IS NOT A MULTI-NOT.
91          :
92 000314 032767 000000G.000000G. 1$. BIT.   #LMN.GFLAG.          :WAS PREVIOUS ELEMENT PART OF A M-NOT.
93 000322. 001410      BEQ.   JOIN.          :NO.
94 000324 042767 000000G.000000G. BIC.   #LMN.GFLAG.          :CLEAR LOCAL FLAG.
95 000332. 052767 000000G.000000G. BIS.   #SHLMN.WQW.          :SET FLAG FOR HQR LAST MULTI-NOT.
96 000340.      CALL.  SLBREF.          :WRITE WQW TO HQR REF PAGE.

```

```

98      ;
99      ;
100     ;
101     000344 132720 000000G JOIN: BITB  #B#NOT,(R0)+ ; IS CURRENT ELEMENT IN SDLB A 'NOT'
102     000350 001003 BNE  1$ ; YES, DO NOT SET BIT IN WQW
103     000352 056767 000000G-000000G BIC  BITPOS,WQW ; SET A BIT FOR REF PAGE
104     ;
105     ;
106     ;
107     ;
108     ;
109     000360 005367 000000G 1$: DEC  QENTL ; DEC LOOP COUNT (* ELEMENTS IN SDLB ENTRY)
110     000364 001414 BEQ  2$ ; FINISHED WITH CURRENT ENTRY
111     000366 016701 000000G MOV  BITPOS,R1 ; PREPARE TO SHIFT
112     000372 006301 ASL  R1 ; SHIFT TO NEXT POSITION
113     000374 010167 000000G MOV  R1,BITPOS ; SET NEW BIT POSITION
114     ;
115     ;
116     ;
117     ;
118     ;
119     000400 032767 004000 000000G BIT  #BIT11,BITPOS ; SHIFTED OUT OF BIT13 ?
120     000406 BOFF ELEMNT ; NO, PROCESS NEXT ELEMENT
121     000410 CALL SLBREF ; WRITE WQW TO HQR
122     000414 000673 BR  ELEMNT ; PROCESS NEXT ELEMENT
123     ;
124     ;
125     ;
126     ;
127     ;
128     000416 2$: BIT  #LMN,GFLAG ; IS THIS MULTI-NOT?
129     000416 032767 000000G-000000G BOFF 3$ ; BRANCH IF NOT
130     000424 BIC  #SHLMN,WQW ; SET LAST MULTI-NOT FLAG
131     000426 052767 000000G-000000G BIC  #LMN,GFLAG ; CLEAR LOCAL FLAG
132     000434 042767 000000G-000000G 3$: BIS  #SHLNE,WQW ; SET LAST NON-EQD WORD
133     000442 052767 000000G-000000G CALL SLBREF ; WRITE WQW TO QLB REF
134     000442 052767 000000G-000000G MOV  NPSQLB,PSQLB ; POINT TO NEXT SSDLB ENTRY
135     000450 016767 000000G-000000G MOV  R0,IOID ; SAVE OFFSET
136     000454 016767 000000G-000000G MOV  R0,IOID ; SAVE OFFSET
137     000462 010057 000000G JMP  QSTR
138     000466 000167 177352
139     ;

```

```

141 ;
142 ;
143 ;
144 ;
145 000472. ;
146 000472. 016702. 000000G. ; COINX:
147 000476 ;
148 000502. 010267 000000G. ; MOV. LXQID,R2. ; SET SDLB POINTER AS ARGUMENT.
149 000506 016702. 000000G. ; CALL. HMAPQ. ; CONVERT SDLB POINTER TO HARDWARE ADDRESS.
150 000512. 132762. 000000G. 000000G. ; MOV. R2,WQW. ; SET HARDWARE ADDRESS.
151 000520 ; MOV. LXQID,R2. ; GET QLB POINTER.
152 000522. 052767 000000G. 000000G. ; BITB. #B#EOTR,QLB(R2) ; IS IT LAST ELEMENT OF A QUERY TOP OR?
153 000530 132762. 000000G. 000000G. ; BOFF. 10$ ; BRANCH IF NOT.
154 000536 ; BIS. #SHEEQ,WQW. ; SET END-OF-QUERY BIT.
155 000540 042767 040000 000000G. ; BITB. #B#EQQ,QLB(R2) ; IS IT LAST ELEMENT OF A QUERY?
156 000546 000450 ; BON. 15$ ; BRANCH IF SO.
157 000550 ; BIC. #BIT14,WQW. ; SET END-OF-TOP-OR CODE.
158 000550 052767 040000 000000G. 10$: ; BR. 20$
159 000556 000444 ; BIS. #BIT14,WQW. ; SET END-OF-SUBBLOCK CODE.
160 000560 ; BR. 20$
161 000560 ;
162 000564 016767 000000G. 000000G. ; CALL. SLBREF. ; WRITE QLB POINTER WORD.
163 000572. 005267 000000G. ; MOV. SIDBUF,WQW. ; GET LAST ALLOCATED SIDMEM.
164 000576 016700 000000G. ; INC. SIDBUF. ; SET START OF NEXT ALLOCATION.
165 000602. 010067 000000G. ; MOV. IQID,R0 ; GET QUERY ID (RETURNED BY HMAPQ)
166 000606 006300 ; MOV. R0,ADDR. ; SAVE ADDR FOR SUBREAD LOAD.
167 000610 016001 000000G. ; ASL. R0 ; CHANGE TO WORD OFFSET.
168 000614 ; MOV. SDBUFS(R0),R1 ; GET NUMBER OF ALLOCATED SID BLOCKS.
169 000616 016700 000000G. ; SAVE. R0
170 000622. ; MOV. SIDBLK,R0 ; GET SIZE OF SIDMEM ALLOCATION FOR QUERY
171 000626 ; CALL. #NIL ;
172 000630 060167 000000G. ; RESTOR. R0
173 000634 016767 000000G. 000000G. ; ADD. R1,WQW. ; SET LAST ALLOCATED SIDMEM.
174 000642. 052767 000000G. 000000G. ; MOV. SIDBUF,DATA. ; SET START OF QUERY'S ALLOCATION.
175 000650 ; BIS. #HXTSLB,DATA. ; SET SUBDOC USING QUERY FLAG.
176 000654 016767 000000G. 000000G. ; CALL. ONESR. ; WRITE SUBREAD MEMORY.
177 000662. 052767 000000G. 000000G. ; MOV. WQW,SIDBUF. ; SET LAST ALLOCATED SIDMEM.
178 000670 ; BIS. #SHEEQ,WQW. ; SET CONTROL BITS.
179 000670 ;
180 000674 042767 000000G. 000000G. ; CALL. SLBREF. ; WRITE REF SDLB WORD
181 000702. 005267 000000G. ; BIC. #PQLB,GFLAG. ; RESET SDLB PROCESSED FLAG.
182 000706 000167 172222 ; INC. LXQID. ; SET NEXT QLB POINTER TO BE PROCESSED.
183 ; JMP. QLBLT. ; PROCESS NEXT SDLB BLOCK.
184 ;
185 ;
186 000712. ; CO_LBXT:
187 ;
188 ;
189 ;
190 000712. ; HALT SD PROCESSOR.
191 000716 ;
192. 000001 ; CALL. SDHALT. ; HALT SD PROCESSOR.
; RETURN.
; END.

```

ADDR = .***** GX.	BYTE40 = .000050	BYTE92 = .000134	MMDE = .000004	Q\$LDMD = .000004
ALUCKE = .040000	BYTE41 = .000051	BYTE93 = .000135	MMWRTE = .000010	Q\$LDPP = .002000
ALUOE = .004000	BYTE42 = .000052	BYTE94 = .000136	MNOBRE = .100000	Q\$LHP = .010000
A01 = .010000	BYTE43 = .000053	BYTE95 = .000137	MREN1 = .000001	Q\$MNC = .140000
BITVDS = .***** GX.	BYTE44 = .000054	BYTE96 = .000140	MREN2 = .020000	Q\$MRP = .000052
BITVAL = .000000	BYTE45 = .000055	BYTE97 = .000141	MSYN = .000040	Q\$MRP2 = .000040
BIT0 = .000001	BYTE46 = .000056	BYTE98 = .000142	N = .000144	Q\$MSC = .040000
BIT1 = .000002	BYTE47 = .000057	BYTE99 = .000143	NOTFMN = .000276R	Q\$MSET = .000004
BIT10 = .002000	BYTE48 = .000060	BYTVAL = .000144	NPSQLB = .***** GX.	Q\$MSP = .100000
BIT11 = .004000	BYTE49 = .000061	B\$E00 = .***** GX.	ONESR = .***** GX.	Q\$MSP = .100000
BIT12 = .010000	BYTE5 = .000005	B\$EDTR = .***** GX.	PLB = .000010	Q\$NCLK = .176000
BIT13 = .020000	BYTE50 = .000062	B\$FMN = .***** GX.	PLC = .000020	Q\$PP = .000100
BIT14 = .040000	BYTE51 = .000063	B\$MUL = .***** GX.	PLD = .000030	Q\$PPSW = .000320
BIT15 = .100000	BYTE52 = .000064	B\$NOT = .***** GX.	PLRWR = .000200	Q\$PP2 = .000300
BIT2 = .000004	BYTE53 = .000065	CBKALL = .001000	PLR.EN = .000200	Q\$QHLT = .000013
BIT3 = .000010	BYTE54 = .000066	CBKCLK = .000400	PQLB = .***** GX.	Q\$QL = .000043
BIT4 = .000020	BYTE55 = .000067	CHOA = .***** GX.	PSQLB = .***** GX.	Q\$QLA = .000053
BIT5 = .000040	BYTE56 = .000070	CNOBRE = .100000	QENTL = .***** GX.	Q\$QLB = .000054
BIT6 = .000100	BYTE57 = .000071	CPCCEN = .010000	QLB = .***** GX.	Q\$QLR = .000001
BIT7 = .000200	BYTE58 = .000072	C\$READ = .040000	QLBLT = .000134R	002 Q\$QU = .000042
BIT8 = .000400	BYTE59 = .000073	CPURTE = .020000	QLBPT = .000152R	002 Q\$RDCD = .000005
BIT9 = .001000	BYTE6 = .000006	CSADRD = .000004	QLBXT = .000712R	002 Q\$RDMD = .000006
BYTE0 = .000000	BYTE60 = .000074	CSEQC1 = .100000	QLB.2 = .000170R	002 Q\$REBK = .001000
BYTE1 = .000001	BYTE61 = .000075	CSOE = .000040	QR\$CR1 = .176420	Q\$RNC = .006000
BYTE10 = .000012	BYTE62 = .000076	CSURTE = .000100	QR\$CR2 = .176422	Q\$RSC = .004000
BYTE11 = .000013	BYTE63 = .000077	DATA = .***** GX.	QR\$LBR = .176424	Q\$RSET = .000010
BYTE12 = .000014	BYTE64 = .000100	DBR.RD = .000001	QSTRT = .000044R	002 Q\$SM = .100000
BYTE13 = .000015	BYTE65 = .000101	DB\$COP = .001457	Q\$ATTN = .000100	Q\$SP = .000120
BYTE14 = .000016	BYTE66 = .000102	DB\$SPT = .000026	Q\$BCL = .000001	Q\$SP2 = .000340
BYTE15 = .000017	BYTE67 = .000103	DB\$TPC = .000023	Q\$CCDP = .000040	RGQ.EN = .000200
BYTE16 = .000020	BYTE68 = .000104	D\$SPGS = .100000	Q\$CHB = .000400	RGQ.VA = .020000
BYTE17 = .000021	BYTE69 = .000105	D\$MAUR = .000005	Q\$CHRL = .000200	SDBUFS = .***** GX.
BYTE18 = .000022	BYTE7 = .000007	D\$MRD = .000003	Q\$CLR = .000040	SDHALT = .***** GX.
BYTE19 = .000023	BYTE70 = .000106	D\$MUR = .000004	Q\$CNC = .030000	SDLB = .***** GX.
BYTE2 = .000002	BYTE71 = .000107	ELEMNT = .000204R	002 Q\$CP = .000060	SDLBAD = .***** GX.
BYTE20 = .000024	BYTE72 = .000110	ENBR = .010000	Q\$CPCC = .000010	SDLBLD = .000000RG
BYTE21 = .000025	BYTE73 = .000111	GFLAG = .***** GX.	Q\$CP2 = .000260	SEQ.CI = .000010
BYTE22 = .000026	BYTE74 = .000112	HMAPO = .***** GX.	Q\$CSC = .010000	SHE00 = .***** GX.
BYTE23 = .000027	BYTE75 = .000113	HXTSLB = .***** GX.	Q\$CSEL = .000360	SHLMN = .***** GX.
BYTE24 = .000030	BYTE76 = .000114	IDID = .***** GX.	Q\$CSET = .000002	SHLNE = .***** GX.
BYTE25 = .000031	BYTE77 = .000115	JOIN = .000344R	002 Q\$CSP = .020000	SHPN = .***** GX.
BYTE26 = .000032	BYTE78 = .000116	JOINX = .000472R	002 Q\$DMA = .000001	SIDLK = .***** GX.
BYTE27 = .000033	BYTE79 = .000117	LMN = .***** GX.	Q\$ENBK = .040000	SIDBUF = .***** GX.
BYTE28 = .000034	BYTE8 = .000010	LOC.EN = .000100	Q\$ENOP = .020000	SLBREF = .***** GX.
BYTE29 = .000035	BYTE80 = .000120	LOC.WA = .040000	Q\$FAL = .004000	S\$CLR = .000000
BYTE3 = .000003	BYTE81 = .000121	LOC.WB = .100000	Q\$FC = .000045	S\$LA = .000000
BYTE30 = .000036	BYTE82 = .000122	LXQID = .***** GX.	Q\$FO = .000044	S\$QB = .000005
BYTE31 = .000037	BYTE83 = .000123	MAREN1 = .000001	Q\$FP = .000046	S\$QR = .000006
BYTE32 = .000040	BYTE84 = .000124	MAREN2 = .004000	Q\$HBF = .000002	S\$QX = .000004
BYTE33 = .000041	BYTE85 = .000125	MARL0D = .010000	Q\$ICP = .000006	S\$SR = .000007
BYTE34 = .000042	BYTE86 = .000126	MAROUT = .000002	Q\$IHB = .000003	S\$S1 = .000010
BYTE35 = .000043	BYTE87 = .000127	MAR.LO = .000200	Q\$IHRL = .000002	S\$S2 = .000014
BYTE36 = .000044	BYTE88 = .000130	MAR.OU = .000040	Q\$IMRP = .000007	TD\$CTR = .176370
BYTE37 = .000045	BYTE89 = .000131	MBKALL = .001000	Q\$LBD = .001000	TD\$INL = .004000
BYTE38 = .000046	BYTE9 = .000011	MBKCLK = .000400	Q\$LBDP = .001001	TD\$MEM = .000270
BYTE39 = .000047	BYTE90 = .000132	MHADRF = .000100	Q\$LBP = .000001	TD\$PAR = .176344
BYTE4 = .000004	BYTE91 = .000133	MHLFT = .000002	Q\$LD = .000003	

TD\$QTR= 176346	T\$QBF= 010000	WORD21= 000052	WORDS= 000012	WORD78= 000234
TD\$QRD= 000274	T\$QBR= 000034	WORD22= 000054	WORD50= 000144	WORD79= 000236
TD\$SW= 176376	T\$QBU= 000032	WORD23= 000056	WORD51= 000146	WORD8= 000020
TD\$STAR= 176372	T\$QDU= 100000	WORD24= 000060	WORD52= 000150	WORD80= 000240
TD\$TAW= 176362	T\$QDU= 000200	WORD25= 000062	WORD53= 000152	WORD81= 000242
TD\$TDR= 176374	T\$QDU= 000040	WORD26= 000064	WORD54= 000154	WORD82= 000244
TD\$TDW= 176364	T\$RSET= 040000	WORD27= 000066	WORD55= 000156	WORD83= 000246
T\$AD= 000020	T\$SC= 000022	WORD28= 000070	WORD56= 000160	WORD84= 000250
T\$BA= 000002	T\$SCLK= 020000	WORD29= 000072	WORD57= 000162	WORD85= 000252
T\$BD= 000010	T\$SEG1= 000000	WORD3= 000006	WORD58= 000164	WORD86= 000254
T\$BSD= 100000	T\$SEG2= 000001	WORD30= 000074	WORD59= 000166	WORD87= 000256
T\$BT= 000020	T\$SEG3= 000002	WORD31= 000076	WORD6= 000014	WORD88= 000260
T\$BTAR= 000030	T\$SO= 000001	WORD32= 000100	WORD60= 000170	WORD89= 000262
T\$BDT= 000000	T\$UBUS= 100000	WORD33= 000102	WORD61= 000172	WORD9= 000022
T\$CD= 000100	T\$ICLK= 000400	WORD34= 000104	WORD62= 000174	WORD90= 000264
T\$CLK= 000000	T\$BBEN= 000020	WORD35= 000106	WORD63= 000176	WORD91= 000266
T\$DISK= 000200	UB, IN= 000020	WORD36= 000110	WORD64= 000200	WORD92= 000270
T\$DRD= 000004	WORD0= 000000	WORD37= 000112	WORD65= 000202	WORD93= 000272
T\$MEM= 010000	WORD1= 000002	WORD38= 000114	WORD66= 000204	WORD94= 000274
T\$FSA= 000000	WORD10= 000024	WORD39= 000116	WORD67= 000206	WORD95= 000276
T\$FSAB= 000004	WORD11= 000026	WORD4= 000010	WORD68= 000210	WORD96= 000300
T\$FSAC= 000014	WORD12= 000030	WORD40= 000120	WORD69= 000212	WORD97= 000302
T\$FSB2= 000010	WORD13= 000032	WORD41= 000122	WORD7= 000016	WORD98= 000304
T\$IB= 000026	WORD14= 000034	WORD42= 000124	WORD70= 000214	WORD99= 000306
T\$IBAR= 000024	WORD15= 000036	WORD43= 000126	WORD71= 000216	WOW= ***** GX
T\$IBE= 020000	WORD16= 000040	WORD44= 000130	WORD72= 000220	WRDVAL= 000310
T\$IBF= 040000	WORD17= 000042	WORD45= 000132	WORD73= 000222	XO ID= ***** GX
T\$ICD= 000040	WORD18= 000044	WORD46= 000134	WORD74= 000224	XTREAD= 001000
T\$MODE= 004000	WORD19= 000046	WORD47= 000136	WORD75= 000226	XTWRITE= 000400
T\$OB= 000036	WORD2= 000004	WORD48= 000140	WORD76= 000230	\$MUL= ***** GX
T\$OBE= 004000	WORD20= 000050	WORD49= 000142	WORD77= 000232	

. ABS. 000000 000
 000000 001
 SDLBLD 000720 002.
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3250 WORDS (13 PAGES)
 DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
 ELAPSED TIME: 00:00:44
 SDLBLD, SBLBLD /-SP=C 20, 1 JIM, C 20, 1 JSDLBLD

```

1
2 000000 .TITLE: HQMAP
3 .PSECT: HQMAP
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 000000 HMAPQ::
26 000000   SAVE  R0,R1,R3,R4
27 000010   SAVE  GFLAG
28 000014 012700 000000G MOV  #0LB,R0 ;START OF QLB STRUCTURE
29 000020 005067 000000G CLR  WQA ;INITIALIZE TO START OF HARWARE QLB
30 000024 012767 000002 000000G MOV  #2,BFIRST ;LOW BIT ADDRESS AVAILABLE FOR QLB MAPPING
31 000032 012767 000015 000000G MOV  #13,,BLAST ;HIGH BIT ADDRESS AVAILABLE FOR QLB MAPPING
32 000040 012767 000012 000000G MOV  #10,,BSHIFT ;SHIFT COUNT TO REACH BIT ADDRESS FIELD
33 000046 042767 000000G 000000G BIC  #SDMAP,GFLAG ;RESET FLAG - QLB MAPPING
34 000054 000425 BR   HMAPA ;BRANCH TO COMMON CODE
35
36
37
38 000056 HMAPS::
39 000056   SAVE  R0,R1,R3,R4
40 000066   SAVE  GFLAG
41 000072 012700 000000G MOV  #SDLB,R0 ;START OF SDLB STRUCTURE
42 000076 005067 000000G CLR  WQA ;INITIALIZE TO START OF HARWARE SDLB
43 000102 005067 000000G CLR  BFIRST ;LOW BIT ADDRESS AVAILABLE FOR SDLB MAPPING
44 000106 012767 000012 000000G MOV  #10,,BLAST ;HIGH BIT ADDRESS AVAILABLE FOR QLB MAPPING
45 000114 012767 000010 000000G MOV  #8,,BSHIFT ;SHIFT COUNT TO REACH BIT ADDRESS FIELD
46 000122 052767 000000G 000000G BIS  #SDMAP,GFLAG ;SET SDLB MAPPING FLAG

```

```

48
49
50
51 000130
52 000130
53 000144 005067 000000G
54 000150
55 000150 042767 000000G 000000G
56 000156 016767 000000G 000000G
57 000164 112001
58 000166 010003
59 000170 042701 177400
60 000174 060103
61 000176 005302
62 000200 032767 000000G 000000G
63 000206
64 000210 112367 000000G
65 000214 042767 177400 000000G
66 000222 000406
67 000224
68 000224 032703 000001
69 000230
70 000232 005203
71 000234
72 000234 012367 000000G
73 000240
74 000240 010367 000000G
75 000244
76 000244 026767 000000G 000000G
77 000252 001440
78 000254 032767 000000G 000000G
79 000262
80 000264 032767 000000G 000000G
81 000272
82 000274 016767 000000G 000000G
83 000302 000424
84 000304
85 000304 042767 000000G 000000G
86 000312 032767 000000G 000000G
87 000320
88 000322 016704 000000G
89 000326 132764 000000G 000000G
90 000334
91 000336 005267 000000G
92 000342
93 000342 005267 000000G
94 000346 005267 000000G
95 000352 000734

```

COMMON ADDRESS MAPPING CODE

HMAPA:

SAVE LXQID,XQID,NPSQLB LXQID ; POINTER IN BLOCK LAST PROCESSED

CLR LXQID

HMAPB:

BIC #LMN,GFLAG ; CLEAR PREVIOUS ELEMENT MULTINOT FLAG

MOV BFIRST,BITADD ; REINITIALIZE BIT ADDRESS

MOVB (R0)+,R1 ; GET COUNT OF NUMBER OF ENTRIES

MOV R0,R3 ; ALSO SAVE IN TEMPORARY

BIC #177400,R1 ; CLEAR HIGH ORDER BYTE

ADD R1,R3 ; SKIP OVER COUNT FIELD

DEC R2 ; DECREMENT SOFTWARE OFFSET

BIT *SDMAP,GFLAG ; IS THIS SDLB?

BON 10\$; BRANCH IF YES

MOVB (R3)+,XQID ; GET POINTER IN QLB BLOCK TO BE PROCESSED

BIC #177400,XQID ; CLEAR HIGH ORDER BYTE

BR 20\$

10\$:

BIT #BIT0,R3 ; IS ADDRESS ON WORD BOUNDARY?

BOFF 15\$; BRANCH IF YES

INC R3 ; SKIP PAD BYTE

15\$:

MOV (R3)+,XQID ; GET POINTER IN SDLB BLOCK TO BE PROCESSED

20\$:

MOV R3,NPSQLB ; SAVE ADDRESS OF NEXT BLOCK

HMAPC:

CMPL LXQID,XQID ; PREVIOUS POINTER EQUAL CURRENT

BEQ HMAPL ; BRANCH IF YES

BIT #QLB,GFLAG ; WAS PREVIOUS BLOCK PROCESSED

BON 10\$; BRANCH IF YES

BIT *SDMAP,GFLAG ; IS THIS SUBELEMENT MAPPING

BOFF 20\$; BRANCH IF NO

MOV XQID,LXQID ; NO DUMMY ENTRIES IN HARDWARE

BR HMAPL

10\$:

BIC #POLB,GFLAG ; CLEAR BLOCK PROCESSED FLAG

BIT *SDMAP,GFLAG ; SUBDOC ELEMENT MAPPING?

BOFF 20\$; BRANCH IF NOT

MOV LXQID,R4 ; GET LAST QLB POINTER

BITB #BEQ0,QLB(R4) ; IS IT END OF A QUERY BLOCK?

BOFF 20\$; BRANCH IF NOT

INC WQA ; ALLOW FOR SIDMEM ADDRESS WORD

20\$:

INC WQA ; ALLOW FOR DUMMY OR EQ0/EOB WORD

INC LXQID ; BUMP LAST BLOCK PROCESSED POINTER

BR HMAPC


```

97 000354          HMAPL:
98 000354 132710 000000G BITB  #B$FMN.(R0)      ;FIRST MULTI-NOT?
99 000360          BOFF 20$          ;BRANCH IF NO
100 000362 032767 000000G 000000G BIT  #LMN.GFLAG      ;WAS PREVIOUS ELEMENT A MULTI-NOT?
101 000370          BON 50$          ;BRANCH IF YES
102 000372 052767 000000G 000000G BIS  #LMN.GFLAG      ;SET CURRENT ELEMENT MULTI-NOT FLAG
103 000400 026767 000000G 000000G CMP  BITADD,BFIRST ;ANY ACTIVITY FOR CURRENT WORD?
104 000406 003013          BGT 50$          ;BRANCH IF YES
105 000410 000417          BR   JOIN         ;DONE WITH CURRENT ELEMENT
106 000412          ;
107 000412 132710 000000G 20$:
108 000416          BITB #B$MUL.(R0)    ;IS THIS ELEMENT A MULTI-NOT?
109 000420 032767 000000G 000000G BON  JOIN         ;BRANCH IF YES (LMN ALREADY SET)
110 000426          BIT  #LMN.GFLAG      ;WAS PREVIOUS ELEMENT A MULTI-NOT?
111 000430 042767 000000G 000000G BOFF JOIN         ;BRANCH IF NOT
112 000436          BIC  #LMN.GFLAG      ;CLEAR LOCAL MULTI-NOT FLAG
113 000436 005267 000000G 50$:
114 000442 016767 000000G 000000G INC  WQA          ;BUMP TO NEXT HARDWARE ADDRESS
115 000450          MOV  BFIRST,BITADD ;REINITIALIZE BIT ADDRESS
116 000450 005702          JOIN:
117 000452 003433          TST  R2          ;HAS TARGET ELEMENT BEEN PROCESSED?
118 000454 005200          BLE  JOINX      ;BRANCH IF SO
119 000456 005302          INC  R0          ;POINT TO NEXT ELEMENT IN BLOCK
120 000460 005301          DEC  R2          ;DECREMENT TARGET OFFSET
121 000462 003414          DEC  R1          ;DECREMENT ELEMENT COUNT
122 000464 005267 000000G 000000G BLE  50$        ;BRANCH IF BLOCK DONE
123 000470 026767 000000G 000000G INC  BITADD      ;SET BIT ADDRESS FOR NEXT ELEMENT
124 000476 003726          CMP  BITADD,BLAST ;CHECK FOR LIMIT IN CURRENT WORD
125 000500 005267 000000G          BLE  HMAPL      ;IF OK, GO DO NEXT ELEMENT
126 000504 016767 000000G 000000G INC  WQA          ;BUMP HARDWARE ADDRESS
127 000512 000720          MOV  BFIRST,BITADD ;REINIT BIT ADDRESS
128 000514          BR   HMAPL      ;GO DO NEXT ELEMENT
129 000514 160003          50$:
130 000516 016700          SUB  R0,R3      ;CALCULATE NUMBER OF PAD + POINTER BYTES
131 000522 160302          MOV  NPSQLB,R0 ;GET ADDRESS OF NEXT BLOCK
132 000524 005267 000000G          SUB  R3,R2      ;ADJUST TARGET OFFSET
133 000530 052767 000000G 000000G INC  WQA          ;NEW BLOCK - BUMP HARDWARE ADDRESS
134 000536 000167 177406          BIS  #POLB.GFLAG ;SET BLOCK PROCESSED FLAG
135          JMP  HMAPB   ;PROCESS NEXT BLOCK
136          ;
137          ;
138 000542          ; FINISHED COMPUTATION OF HARDWARE ADDRESS
139 000542 016701 000000G JOINX:
140 000546 016702 000000G MOV  BSHIFT,R1  ;NUMBER OF SHIFTS
141 000552          MOV  BITADD,R2  ;LOAD BIT ADDRESS
142 000552 006302          1$:
143 000554 005301          ASL  R2          ;
144 000556 001375          DEC  R1          ;
145 000560 066702 000000G          BNE  1$         ;
146 000564 016767 000000G 000000G ADD  WQA,R2     ;ADD IN HARDWARE WORD ADDRESS
147          MOV  XQID,IQID ;ALSO RETURN QUERY ID (HMAPQ)
148 000572          ;
149 000606          RESTOR LXDID,XQID,NPSQLB
150 000612          RESTOR GFLAG
151 000622          RESTOR R0,R1,R3,R4
152          RETURN
153          ;
153 000001          .END

```

ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	MMLEFT = 000002	Q\$MSET = 000004
ALUDE = 004000	BYTE4 = 000004	BYTE91 = 000133	MMOE = 000004	Q\$MSP = 100000
A01 = 010000	BYTE40 = 000050	BYTE92 = 000134	MMWRTE = 000010	Q\$NCLK = 176000
BFIRST = ***** GX	BYTE41 = 000051	BYTE93 = 000135	MNOBRE = 100000	Q\$PP = 000100
BITADD = ***** GX	BYTE42 = 000052	BYTE94 = 000136	MREN1 = 000001	Q\$PPSW = 000320
BITVAL = 000000	BYTE43 = 000053	BYTE95 = 000137	MREN2 = 020000	Q\$PP2 = 000300
BIT0 = 000001	BYTE44 = 000054	BYTE96 = 000140	MSYN = 000040	Q\$QHLT = 000013
BIT1 = 000002	BYTE45 = 000055	BYTE97 = 000141	N = 000144	Q\$QL = 000043
BIT10 = 002000	BYTE46 = 000056	BYTE98 = 000142	NPSQLB = ***** GX	Q\$QLA = 000053
BIT11 = 004000	BYTE47 = 000057	BYTE99 = 000143	PLB = 000010	Q\$QLB = 000054
BIT12 = 010000	BYTE48 = 000060	BYTVAL = 000144	PLC = 000020	Q\$QLR = 000001
BIT13 = 020000	BYTE49 = 000061	B\$EQ = ***** GX	PLD = 000030	Q\$QLW = 000042
BIT14 = 040000	BYTE5 = 000005	B\$FMN = ***** GX	PLRWR = 000200	Q\$RDCD = 000005
BIT15 = 100000	BYTE50 = 000062	B\$MUL = ***** GX	PLR.EN = 000200	Q\$RDMND = 000006
BIT2 = 000004	BYTE51 = 000063	CBKALL = 001000	POLB = ***** GX	Q\$REBK = 001000
BIT3 = 000010	BYTE52 = 000064	CBKCLK = 000400	QLB = ***** GX	Q\$RNC = 000000
BIT4 = 000020	BYTE53 = 000065	CNOBRE = 100000	QR\$CR1 = 176420	Q\$RSC = 004000
BIT5 = 000040	BYTE54 = 000066	CPCCEN = 010000	QR\$CR2 = 176422	Q\$RSEC = 000010
BIT6 = 000100	BYTE55 = 000067	CPREAD = 040000	QR\$LBR = 176424	Q\$SM = 100000
BIT7 = 000200	BYTE56 = 000070	CPURTE = 020000	Q\$ATTN = 000100	Q\$SP = 000120
BIT8 = 000400	BYTE57 = 000071	CSADRD = 000004	Q\$BCL = 000001	Q\$SP2 = 000340
BIT9 = 001000	BYTE58 = 000072	CSEQC1 = 100000	Q\$CCCP = 000040	RGQ.EN = 000200
BLAST = ***** GX	BYTE59 = 000073	CSOE = 000040	Q\$CHB = 000400	RGQ.VA = 020000
BSHIFT = ***** GX	BYTE6 = 000006	CSURTE = 000100	Q\$CHRL = 000200	SDLB = ***** GX
BYTE0 = 000000	BYTE60 = 000074	DBR.RD = 000001	Q\$CLR = 000040	SDMAP = ***** GX
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$CNC = 030000	SEQ.CI = 000010
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$CPC = 000060	S\$CLR = 000000
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$CPCO = 000010	S\$LA = 000001
BYTE12 = 000014	BYTE64 = 000100	DISPSS = 100000	Q\$CP2 = 000260	S\$OB = 000005
BYTE13 = 000015	BYTE65 = 000101	DMAAR = 000005	Q\$CSC = 010000	S\$QR = 000006
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$CSEL = 000360	S\$QX = 000004
BYTE15 = 000017	BYTE67 = 000103	DMARWR = 000004	Q\$CSET = 000002	S\$SR = 000007
BYTE16 = 000020	BYTE68 = 000104	ENBR = 010000	Q\$CSP = 020000	S\$S1 = 000010
BYTE17 = 000021	BYTE69 = 000105	GFLAG = ***** GX	Q\$DMA = 000001	S\$S2 = 000014
BYTE18 = 000022	BYTE7 = 000007	HMAPA = 000130R	002.Q\$ENBK = 040000	TD\$CTR = 176370
BYTE19 = 000023	BYTE70 = 000106	HMAPB = 000150R	002.Q\$ENOP = 020000	TD\$CTL = 176360
BYTE2 = 000002	BYTE71 = 000107	HMAPC = 000244R	002.Q\$FAL = 004000	TD\$INL = 004000
BYTE20 = 000024	BYTE72 = 000110	HMAPL = 000354R	002.Q\$FC = 000045	TD\$MEM = 000270
BYTE21 = 000025	BYTE73 = 000111	HMAPQ = 000000RG	002.Q\$FO = 000044	TD\$OAR = 176344
BYTE22 = 000026	BYTE74 = 000112	HMAPS = 000056RG	002.Q\$FP = 000046	TD\$QTR = 176346
BYTE23 = 000027	BYTE75 = 000113	IQID = ***** GX	Q\$HBF = 000002	TD\$QRD = 000274
BYTE24 = 000030	BYTE76 = 000114	JOIN = 000450R	002.Q\$ICP = 000006	TD\$SWJ = 176376
BYTE25 = 000031	BYTE77 = 000115	JOINX = 000542R	002.Q\$IHB = 000003	TD\$STAR = 176372
BYTE26 = 000032	BYTE78 = 000116	LMN = ***** GX	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE27 = 000033	BYTE79 = 000117	LOC.EN = 000100	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE28 = 000034	BYTE8 = 000010	LOC.WA = 040000	Q\$LBD = 001000	TD\$TDJW = 176364
BYTE29 = 000035	BYTE80 = 000120	LOC.WB = 100000	Q\$LBDP = 001001	T\$AD = 000020
BYTE3 = 000003	BYTE81 = 000121	LXQID = ***** GX	Q\$LBP = 000001	T\$BA = 000002
BYTE30 = 000036	BYTE82 = 000122	MAREN1 = 000001	Q\$LCD = 000003	T\$BD = 000010
BYTE31 = 000037	BYTE83 = 000123	MAREN2 = 004000	Q\$LDID = 000004	T\$BSO = 100000
BYTE32 = 000040	BYTE84 = 000124	MARLOD = 010000	Q\$LDPP = 002000	T\$BT = 000020
BYTE33 = 000041	BYTE85 = 000125	MAROUT = 000002	Q\$LHP = 010000	T\$BTAR = 000030
BYTE34 = 000042	BYTE86 = 000126	MAR.LO = 002000	Q\$MNC = 140000	T\$BDT = 002000
BYTE35 = 000043	BYTE87 = 000127	MAR.DU = 000040	Q\$MR = 000052	T\$CLK = 002000
BYTE36 = 000044	BYTE88 = 000130	MBKALL = 001000	Q\$MRP = 000040	T\$DLSK = 000200
BYTE37 = 000045	BYTE89 = 000131	MBKCLK = 000400	Q\$MRP2 = 000240	T\$DRD = 000004
BYTE38 = 000046	BYTE9 = 000011	MHADR = 000100	Q\$MSC = 040000	

T\$EMEM = 010000	T\$1CLK = 000400	WORD30 = 000074	WORD55 = 000156	WORD8 = 000020
T\$FSA = 000000	T\$BEN = 000020	WORD31 = 000076	WORD56 = 000160	WORD9 = 000024
T\$FSAB = 000004	UBD.IN = 000020	WORD32 = 000100	WORD57 = 000162	WORD10 = 000024
T\$FSAC = 000014	WORD0 = 000000	WORD33 = 000102	WORD58 = 000164	WORD11 = 000024
T\$FSB2 = 000010	WORD1 = 000002	WORD34 = 000104	WORD59 = 000166	WORD12 = 000024
T\$IB = 000026	WORD10 = 000024	WORD35 = 000106	WORD6 = 000014	WORD13 = 000024
T\$IBAR = 000024	WORD11 = 000026	WORD36 = 000110	WORD60 = 000170	WORD14 = 000026
T\$IBE = 020000	WORD12 = 000030	WORD37 = 000112	WORD61 = 000172	WORD15 = 000026
T\$IBF = 040000	WORD13 = 000032	WORD38 = 000114	WORD62 = 000174	WORD16 = 000026
T\$ICD = 000040	WORD14 = 000034	WORD39 = 000116	WORD63 = 000176	WORD17 = 000026
T\$MODE = 004000	WORD15 = 000036	WORD4 = 000010	WORD64 = 000200	WORD18 = 000026
T\$OB = 000036	WORD16 = 000040	WORD40 = 000120	WORD65 = 000202	WORD19 = 000026
T\$OBE = 004000	WORD17 = 000042	WORD41 = 000122	WORD66 = 000204	WORD20 = 000026
T\$OBF = 010000	WORD18 = 000044	WORD42 = 000124	WORD67 = 000206	WORD21 = 000026
T\$OBRA = 000034	WORD19 = 000046	WORD43 = 000126	WORD68 = 000210	WORD22 = 000026
T\$OBWA = 000032	WORD2 = 000004	WORD44 = 000130	WORD69 = 000212	WORD23 = 000026
T\$OUTA = 100000	WORD20 = 000050	WORD45 = 000132	WORD7 = 000016	WORD24 = 000026
T\$RBD0 = 000200	WORD21 = 000052	WORD46 = 000134	WORD70 = 000214	WORD25 = 000026
T\$RNB = 000040	WORD22 = 000054	WORD47 = 000136	WORD71 = 000216	WORD26 = 000026
T\$RSET = 040000	WORD23 = 000056	WORD48 = 000140	WORD72 = 000220	WORD27 = 000026
T\$SC = 000022	WORD24 = 000060	WORD49 = 000142	WORD73 = 000222	WORD28 = 000026
T\$SCLK = 020000	WORD25 = 000062	WORD5 = 000012	WORD74 = 000224	WORD29 = 000026
T\$SEG1 = 000000	WORD26 = 000064	WORD50 = 000144	WORD75 = 000226	WORD30 = 000026
T\$SEG2 = 000001	WORD27 = 000066	WORD51 = 000146	WORD76 = 000230	WORD31 = 000026
T\$SEG3 = 000002	WORD28 = 000070	WORD52 = 000150	WORD77 = 000232	WORD32 = 000026
T\$SO = 000001	WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	WORD33 = 000026
T\$UBUS = 100000	WORD3 = 000006	WORD54 = 000154	WORD79 = 000236	WORD34 = 000026

. ABS: 000000 000
000000 001
HQMAP: 000624 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3152 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:44
HQMAP, HQMAP /-SP=C20.1JIM, C20.1JHQMAP

```

1
2 000000 .TITLE: LOADQX.
3 .PSECT: LOADQX.
4
5
6 SUBROUTINE TO LOAD QEX MEMORY VIA CP MICROCODE.
7
8 .MCALL WTSE$,CLEF$.
9
10 LOADQX:
11 CLR: -(SP) ;CLEAR NOTHING IN CSR1
12 MOV: #Q$NCLK,-(SP) ;SET NO-CLOCKS.
13 CALL: CSR1
14 CLR: OR$CR2. ;SET LOAD MODE.
15
16 SELECT MEMORY TO BE LOADED (WINDOW OR LOCATION).
17 START THE MICROCODE THAT DOES THE LOADING OF THE QEX.
18
19 MOV: CODE,-(SP) ;SELECT WINDOW OR LOCATION MEMORY.
20 CALL: PPCR
21 MOV: #1760,-(SP) ;START CP MICROCODE AT X'3F0'
22 CALL: SEQCS.
23
24 CLR: -(SP) ;RESET BR INHIBIT.
25 CALL: PPCR
26 MOV: #377,-(SP) ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
27 CALL: SEQMM.
28 CLR: -(SP) ;RESET BR INHIBIT.
29 CALL: MRPCR.
30 MOV: #Q$REBK,OR$CR2 ;RE-ARM INTERRUPTS.
31 MOV: #<Q$LDPP+Q$ENOP>,OR$CR2 ;SET LOAD PPS MODE + ENABLE INTERRUPTS.
32 MOV: #Q$CSEL,-(SP) ;CLEAR ALL SELECTIONS.
33 BIS: #<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE.
34 BIS: #Q$NCLK,(SP) ;CLEAR ALL NO-CLOCKS
35 CLR: -(SP) ;SET NOTHING.
36 CALL: CSR1
37
38 WAIT FOR INTERRUPT FROM CP.
39 USE DMA INTERRUPT (SEE CP MICROCODE FOR QEX)
40
41 WTSE$: #EFN.3
42
43 CLEF$: #EFN.3
44
45 RE-ARM INTERRUPTS.
46
47 MOV: #<Q$SM+Q$CHB>,OR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
48 MOV: #<Q$SM+Q$REBK>,OR$CR2 ;RE-ARM.
49 MOV: #<Q$SM+Q$ENBK+Q$ENOP>,OR$CR2 ;ENABLE.
50
51 CLR: -(SP) ;CLEAR NOTHING IN CSR1
52 MOV: #Q$NCLK,-(SP) ;SET NO-CLOCKS.
53 CALL: CSR1
54 CLR: OR$CR2. ;SET LOAD MODE.
55 RETURN.
56 .END.

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PPCR = ***** GX	Q#RNC = 006000
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR#CR1 = 176420	Q#RSC = 004000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR#CR2 = 176422	Q#RSET = 000010
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q#LBR = 176424	Q#SM = 100000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q#ATTN = 000100	Q#SP = 000120
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q#BCL = 000001	Q#SP2 = 000340
BIT10 = 002000	BYTE48 = 000060	BYTVL = 000144	Q#CCCP = 000040	RGD.EN = 000200
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#CHB = 000400	RGD.VA = 020000
BIT12 = 010000	BYTES = 000005	CBKCLK = 000400	Q#CHRL = 000200	SEQCS = ***** GX
BIT13 = 020000	BYTE50 = 000062	CHOBRE = 100000	Q#CLR = 000040	SEQRM = ***** GX
BIT14 = 040000	BYTE51 = 000063	CODE = ***** GX	Q#CNC = 030000	SEQ.CI = 000010
BIT15 = 100000	BYTE52 = 000064	CPCCEN = 010000	Q#CP = 000060	S#CLR = 000000
BIT2 = 000004	BYTE53 = 000065	CPCR = ***** GX	Q#CPCC = 000010	S#LA = 000001
BIT3 = 000010	BYTE54 = 000066	CPREAD = 040000	Q#CP2 = 000260	S#QB = 000005
BIT4 = 000020	BYTE55 = 000067	CPWRTE = 020000	Q#CSC = 010000	S#QR = 000006
BIT5 = 000040	BYTE56 = 000070	CSADRD = 000004	Q#CSEL = 000360	S#QX = 000004
BIT6 = 000100	BYTE57 = 000071	CSEQC1 = 100000	Q#CSET = 000002	S#SR = 000007
BIT7 = 000200	BYTE58 = 000072	CSDC = 000040	Q#CSP = 020000	S#S1 = 000010
BIT8 = 000400	BYTE59 = 000073	CSR1 = ***** GX	Q#DMA = 000001	S#S2 = 000014
BIT9 = 001000	BYTE60 = 000006	CSWRTE = 000100	Q#ENBK = 040000	TD#CTR = 176370
BYTE0 = 000000	BYTE61 = 000074	DBR.RD = 000001	Q#ENOP = 020000	TD#CTW = 176360
BYTE1 = 000001	BYTE62 = 000075	DB#CPP = 001457	Q#FAL = 004000	TD#CNL = 004000
BYTE10 = 000012	BYTE63 = 000076	DB#SPT = 000026	Q#FC = 000045	TD#MEM = 000270
BYTE11 = 000013	BYTE64 = 000077	DB#TPC = 000023	Q#FD = 000044	TD#OAR = 176344
BYTE12 = 000014	BYTE65 = 000100	DISPGS = 100000	Q#FP = 000046	TD#QDR = 176346
BYTE13 = 000015	BYTE66 = 000101	DMARUR = 000005	Q#HBF = 000002	TD#QRD = 000224
BYTE14 = 000016	BYTE67 = 000102	DMARUR = 000003	Q#ICP = 000006	TD#SU = 176376
BYTE15 = 000017	BYTE68 = 000103	EFN.3 = ***** GX	Q#IHB = 000003	TD#TAR = 176372
BYTE16 = 000020	BYTE69 = 000104	ENBR = 010000	Q#IHL = 000002	TD#TAU = 176362
BYTE17 = 000021	BYTE70 = 000105	LOADQX: 000000RG	Q#IMRP = 000007	TD#TDR = 176374
BYTE18 = 000022	BYTE71 = 000106	LOC.EN = 000100	Q#LBD = 001000	TD#TDW = 176364
BYTE19 = 000023	BYTE72 = 000107	LOC.WA = 040000	Q#LBDP = 001001	T#AD = 000020
BYTE2 = 000002	BYTE73 = 000110	LOC.WB = 100000	Q#LBP = 000001	T#BA = 000002
BYTE20 = 000024	BYTE74 = 000111	MAREN1 = 000001	Q#LDCD = 000003	T#BD = 000010
BYTE21 = 000025	BYTE75 = 000112	MAREN2 = 004000	Q#LDMD = 000004	T#BSC = 100000
BYTE22 = 000026	BYTE76 = 000113	MARLDB = 010000	Q#LDPP = 002000	T#BT = 000020
BYTE23 = 000027	BYTE77 = 000114	MAROUT = 000002	Q#LHP = 010000	T#BTAR = 000030
BYTE24 = 000030	BYTE78 = 000115	MAR.LO = 002000	Q#MNC = 140000	T#BTD = 002000
BYTE25 = 000031	BYTE79 = 000116	MAR.OU = 000040	Q#MR = 000052	T#CD = 000100
BYTE26 = 000032	BYTE80 = 000117	MBKALL = 001000	Q#MRP = 000040	T#CLK = 002000
BYTE27 = 000033	BYTE81 = 000120	MBKCLK = 000400	Q#MRP2 = 000240	T#D ISK = 000200
BYTE28 = 000034	BYTE82 = 000121	MHADRD = 000100	Q#MSC = 040000	T#DDR = 000004
BYTE29 = 000035	BYTE83 = 000122	MHLEFT = 000002	Q#MSET = 000004	T#ENEM = 010000
BYTE3 = 000003	BYTE84 = 000123	MHOE = 000004	Q#MSP = 100000	T#FSAA = 000000
BYTE30 = 000036	BYTE85 = 000124	MHURTE = 000010	Q#NCLK = 176000	T#FSAB = 000004
BYTE31 = 000037	BYTE86 = 000125	MNOBRE = 100000	Q#PP = 000100	T#FSAC = 000014
BYTE32 = 000040	BYTE87 = 000126	MREN1 = 000001	Q#PPSW = 000320	T#FSB2 = 000010
BYTE33 = 000041	BYTE88 = 000127	MREN2 = 020000	Q#PP2 = 000300	T#IB = 000026
BYTE34 = 000042	BYTE89 = 000130	MRPCR = ***** GX	Q#QHLT = 000013	T#IBAR = 000024
BYTE35 = 000043	BYTE90 = 000131	MSYN = 000040	Q#QL = 000043	T#IBE = 020000
BYTE36 = 000044	BYTE91 = 000132	N = 000144	Q#QLA = 000053	T#IBF = 040000
BYTE37 = 000045	BYTE92 = 000133	PLB = 000010	Q#QLB = 000054	T#ICD = 000040
BYTE38 = 000046	BYTE93 = 000135	PLC = 000020	Q#QLR = 000001	T#MODE = 004000
BYTE39 = 000047		PLD = 000030	Q#QW = 000042	
BYTE4 = 000004		PLRWR = 000200	Q#RDICD = 000005	
BYTE40 = 000050		PLR.EN = 000200	Q#RDMD = 000006	
BYTE41 = 000051			Q#REBK = 001000	

T#0BWA= 000032	WORD17= 000042	WORD39= 000116	WORD60= 000170	WORD81= 000242
T#0UTA= 100000	WORD18= 000044	WORD4 = 000010	WORD61= 000172	WORD82= 000244
T#RBDQ= 000200	WORD19= 000046	WORD40= 000120	WORD62= 000174	WORD83= 000246
T#RNB = 000040	WORD2 = 000004	WORD41= 000122	WORD63= 000176	WORD84= 000250
T#RSET= 000000	WORD20= 000050	WORD42= 000124	WORD64= 000200	WORD85= 000252
T#SC = 000022	WORD21= 000052	WORD43= 000126	WORD65= 000202	WORD86= 000254
T#SCLK= 020000	WORD22= 000054	WORD44= 000130	WORD66= 000204	WORD87= 000256
T#SEG1= 000000	WORD23= 000056	WORD45= 000132	WORD67= 000206	WORD88= 000260
T#SEG2= 000001	WORD24= 000060	WORD46= 000134	WORD68= 000210	WORD89= 000262
T#SEG3= 000002	WORD25= 000062	WORD47= 000136	WORD69= 000212	WORD9 = 000022
T#SO = 000001	WORD26= 000064	WORD48= 000140	WORD7 = 000016	WORD90= 000264
T#UBUS= 100000	WORD27= 000066	WORD49= 000142	WORD70= 000214	WORD91= 000266
T#1CLK= 000400	WORD28= 000070	WORD5 = 000012	WORD71= 000216	WORD92= 000270
T#8BEN= 000020	WORD29= 000072	WORD50= 000144	WORD72= 000220	WORD93= 000272
USD: IN= 000020	WORD3 = 000006	WORD51= 000146	WORD73= 000222	WORD94= 000274
WORD0 = 000000	WORD30= 000074	WORD52= 000150	WORD74= 000224	WORD95= 000276
WORD1 = 000002	WORD31= 000076	WORD53= 000152	WORD75= 000226	WORD96= 000300
WORD10= 000024	WORD32= 000100	WORD54= 000154	WORD76= 000230	WORD97= 000302
WORD11= 000026	WORD33= 000102	WORD55= 000156	WORD77= 000232	WORD98= 000304
WORD12= 000030	WORD34= 000104	WORD56= 000160	WORD78= 000234	WORD99= 000306
WORD13= 000032	WORD35= 000106	WORD57= 000162	WORD79= 000236	WORDVAL= 000310
WORD14= 000034	WORD36= 000110	WORD58= 000164	WORD8 = 000020	XTREAD= 001000
WORD15= 000036	WORD37= 000112	WORD59= 000166	WORD80= 000240	XTWRITE= 000400
WORD16= 000040	WORD38= 000114	WORD6 = 000014		

. ABS. 000000 000
000000 001
LOADQX: 000206 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3258 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
LOADQX, LOADQX/-SP=C 20, 1 JIM, C 20, 1 JLOADQX

```

1
2
3
4
5
6
7
8
9
10      000003
11      000115
12      020000
13
14
15 000000 032272 023753
16 000004 000000
17 000006 000000
18 000010
19 000020 000000
20 000022 000000
21 000024 000000
22 000026 000000
23 000030 000000
24 000032 000000
25 000034
26 001034
27

      .TITLE- -HQRFOS-
      ;
      ;
      EXTRACT-FLU-OCCURRENCE SUMMARY FROM HQR
      ;
      ;
      .MCALL- EXIT$S,RDAF$S,CLEF$S,RQST#C
      ;
      ;
      EFN.3 = 3
      OCLBUF = 115 ;=X'4D' IN CP-DATA-MEMORY
      HRLMRK = BIT13 ;HRL-MARK IN-FAL-POINTER
      ;
      .NLIST- BEX-
      MYSELF: .RAD50 /HQRFOS/
      TSKTCB: .WORD- 0 ;TCB-OF-MY-TASK
      OLDVEC: .WORD- 0 ;OLD-VECTOR-AT-274
      EFBUF: .BLKW- 4
      APLACE: .WORD- 0
      IQID: .WORD- 0 ;CURRENT-QID
      PCOM: .WORD- FOS ;POINTER-TO-FOS
      PFLUCT: .WORD- 0 ;POINTER-TO-FLU-COUNT-WORD-FOR-CURRENT-QID
      FLUCT: .WORD- 0 ;FLU-COUNT-FOR-CURRENT-QID
      DATA1: .WORD- 0
      CDOCL: .BLKW- 256 ;OCL-FROM-CP-DATA-MEMORY
      TEMP: .BLKW- 256 ;TEMPORARY-FOR-TESTING
      .LIST- BEX-

```

```

29.      ;
30.      ;
31.      ;      ENTER HERE.
32.      ;
33.      ;
34.      ;      READ QCL FROM CP DATA MEMORY.
35.      ;
36. 002034      START:
37.      ;
38.      ;
39.      ;      PROVIDE FOR INTERRUPTS FROM HQR.
40. 002034 016767 000000G 175742      MOV.   $TKTCB,TSKTCB      ;SAVE MY TCB.
41. 002042 013767 000274 175736      MOV.   @#274,OLDVEC      ;SAVE ADDRESS AT 274
42. 002050 012737 002570 000274      MOV.   #BPTISR,@#274    ;MOVE IN MY ISR ADDRESS.
43.      ;
44. 002056 005046      CLR.   -(SP)              ;START MICROCODE AT 0
45. 002060      CALL.  SEQCS.
46. 002064 005046      CLR.   -(SP)              ;REINHIBIT BR.
47. 002066      CALL.  CPCR
48. 002072 012767 001000 176422      MOV.   #Q$REBK,QR$CR2   ;RE-ARM INTERRUPTS.
49. 002100 012767 120000 176422      MOV.   #<Q$SM+Q$ENOP>,QR$CR2 ;SET SEARCH MODE + ENABLE INTERRUPTS.
50. 002106 012746 000360      MOV.   #Q$CSEL,-(SP)    ;CLEAR ALL SELECTIONS.
51. 002112 052716 001001      BIS.   #<Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE.
52. 002116 052716 030000      BIS.   #Q$CHC,(SP)     ;CLEAR CP NO-CLOCK.
53. 002122 005046      CLR.   -(SP)              ;SET NOTHING.
54. 002124      CALL.  CSR1
55.      ;
56. 002130 012767 000005 176424      MOV.   #Q$RDCD,QR$LBR   ;MOVE ATTN CODE TO LOD BUS REG
57. 002136 012767 120100 176422      MOV.   #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY.
58. 002144 016701 176422      15$: MOV.   QR$CR2,R1         ;READ CSR2
59. 002150 032701 000100      BIT.   #Q$ATTN,R1       ;ATTN CLEAR.
60. 002154 001373      BNE.   1$              ;NO, READ AGAIN.
61.      ;
62. 002156 012767 000115 176424      MOV.   #QCLBUF,QR$LBR   ;CD MEMORY START ADDRESS.
63. 002164 012767 120040 176422      MOV.   #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP.
64. 002172 016701 176422      2$: MOV.   QR$CR2,R1         ;READ CSR2
65. 002176 032701 000040      BIT.   #Q$CCCP,R1       ;IS CC TO CP CLEAR.
66. 002202 001373      BNE.   2$              ;NO, READ AGAIN.
67.      ;
68. 002204 016767 000002G 176424      MOV.   FOSCTL+2,QR$LBR  ;TRANSFER COUNT = NO. OF QUERIES.
69. 002212 012767 120040 176422      MOV.   #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP.
70. 002220 016701 176422      3$: MOV.   QR$CR2,R1         ;READ CSR2
71. 002224 032701 000040      BIT.   #Q$CCCP,R1       ;IS CC TO CP CLEAR.
72. 002230 001373      BNE.   3$              ;NO, READ AGAIN.
73.      ;
74. 002232 012767 000034 176424      MOV.   #CDQCL,QR$LBR    ;CC MEMORY DATA BUFFER.
75. 002240 012767 120040 176422      MOV.   #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP.
76.      ;
77.      ;
78.      ;      WAIT FOR INTERRUPT FROM CP.
79. 002246      4$: RDAF$. #EFBUF      ;READ EVENT FLAGS.
80. 002260 032767 000004 175522      BIT.   #BIT2,EFBUF      ;INTERRUPT ?
81. 002266 001767      BEQ.   4$              ;NO, READ AGAIN.
82.      ;
83. 002270      CLEF$. #EFN.3
84.      ;
85.      ;      RE-ARM INTERRUPTS.

```



```

86
87 002302 012767 100400 176422 MOV #<Q$SM+Q$CHB>,QR$CR2 ;CLEAR INTERRUPT (USE HIT-BUFFER-INT)
88 002310 012767 101000 176422 MOV #<Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
89 002316 012767 160000 176422 MOV #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
90
91 002324 005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
92 002326 012746 176000 MOV #Q$NCLK,-(SP) ;SET NO-CLOCKS
93 002332 CALL CSR1
94 002336 005067 176422 CLR QR$CR2 ;SET LOAD MODE
95
96 002342 016701 175454 MOV IOID,R1 ;LOAD OCL INDEX
97 002346 006301 ASL R1 ;SHIFT FOR WORD OFFSET
98 002350 016101 000034 MOV CDQCL(R1),R1 ;GET OCL ENTRY
99 002354 000410 BR THREE
100
101
102
103
104 PROCESS A OCL ENTRY
105 IF OCL ENTRY = 0, STOP FOS EXTRACTION
106
107
108 002356 TWO:
109 002356 016701 175440 MOV IOID,R1 ;LOAD OCL INDEX
110 002362 006301 ASL R1 ;SHIFT FOR WORD OFFSET
111 002364 016101 000034 MOV CDQCL(R1),R1 ;GET OCL ENTRY
112 002370 001002 BNE THREE ;NON-ZERO, PROCESS
113 002372 000167 000140 JMP TWELVE ;FINISHED WITH OCL
114
115 MOVE QID INTO COMMON
116 SAVE THE ADDRESS OF THE NEXT WORD IN COMMON FOR FLU COUNT
117
118 002376 016700 175422 THREE: MOV PCOM,R0 ;LOAD POINTER TO COMMON
119 002402 016720 175414 MOV IOID,(R0)+ ;MOVE CURRENT QID TO COMMON
120 002406 010067 175414 MOV R0,PFLUCT ;SAVE ADDR OF NEXT WORD IN COMMON
121 002412 062700 000002 ADD #2,R0 ;BUMP TO NEXT ADDR IN COMMON
122
123 PROCESS FAL BLOCK
124 READ THE FAL POINTER WORD
125 LOOK FOR HRL MARK = END OF FAL BLOCK
126 R0 -> COMMON
127 R1 = FAL ADDRESS
128
129 002416 FIVE:
130 002416 012746 000046 MOV #Q$FP,-(SP) ;SET CODE FOR FAL POINTER MEMORY
131 002422 010146 MOV R1,-(SP) ;SET FAL POINTER MEMORY ADDRESS
132 002424 CALL RDFAL ;READ FROM FP MEMORY
133 002430 032767 020000 175374 BIT #HRLMRK,DATA1 ;HRL MARK SET? (LAST FP IN BLOCK)
134 002436 001402 BEQ 2$ ;NO
135 002440 000167 000044 JMP ELEVEN ;FINISHED WITH FAL BLOCK
136
137 READ FAL OVERFLOW MEMORY
138 PUT THE OVERFLOW WORD INTO COMMON
139
140 002444 012746 000044 2$: MOV #Q$FO,-(SP) ;SELECT FAL OVERFLOW MEMORY
141 002450 010146 MOV R1,-(SP) ;FO ADDR IS THE SAME AS FP ADDRESS
142 002452 CALL RDFAL ;READ FO MEMORY

```

```

143 002456 016720 175350      MOV.   DATA1,(R0)+          ;PUT.F0.INTO.COMMON.
144                               ;
145                               ;
146                               ;
147                               ;
148 002462 005201              INC.   R1                    ;BUMP.FP.ADDRESS.
149 002464 012746 000045      MOV.   #0$FC,-(SP)          ;SELECT.FAL.COUNTER.MEMORY.
150 002470 010146              MOV.   R1,-(SP)             ;FC.ADDRESS.
151 002472                      CALL.  RDFAL                 ;READ.FC.
152 002476 016720 175330      MOV.   DATA1,(R0)+          ;MOVE.COUNTER.TO.COMMON.
153 002502 005267 175322      INC.   FLUCT                ;INC.FLU.COUNT.
154 002506 000743              BR     FIVE                 ;PROCESS.NEXT.FP.IN.BLOCK.
155                               ;
156                               ;
157                               ;
158                               ;
159                               ;
160                               ;
161                               ;
162                               ;
163                               ;
164                               ;
165 002510                      ELEVEN:
166 002510 016777 175314 175310 MOV.   FLUCT,@PFLUCT         ;PUT.FLU.COUNT.INTO.COMMON.
167 002516 005067 175306      CLR.   FLUCT                ;CLEAR.FOR.NEXT.TIME.
168 002522 005267 175274      INC.   IQID                 ;BUMP.CURRENT.QID.
169 002526 010067 175272      MOV.   R0,PCOM              ;SAVE.POINTER.TO.COMMON.
170 002532 000167 175200      JMP.   TWO                  ;PROCESS.NEXT.OCL.ENTRY.
171                               ;
172                               ;
173                               ;
174                               ;
175                               ;
176 002536                      TWELVE:
177 002536 012746 000040      MOV.   #0$CLR,-(SP)         ;REINHIBIT.FAL.PROCESOR.
178 002542                      CALL.  PPCR                  ;
179 002546 016737 175234 000274 MOV.   OLDVEC,@#274         ;RESTORE.SYSTEM.VECTOR.ADDRESS.
180 002554                      ROST$C. FOSGET              ;
181 002562                      EXIT$S.                      ;

```

```

183      ;
184      ;
185      ;      INTERRUPT SERVICE ROUTINE
186      ;      TRAP INTERRUPTS FROM HQR THROUGH VECTOR ADDRESS 274
187      ;      SET EVENT FLAG 3
188      ;      CP DEBUGGING ROUTINES WILL READ CSR #2 AND DECODE THE INTERRUPT
189      ;
190      ;
191 002570      BPTISR::
192 002570      ;      SAVE R0,R1,R2,R3,R4,R5
193      ;
194 002604      016705      175174      MOV      TSKTCB,R5      ;LOAD MY TCB
195 002610      012700      000003      MOV      #EFN.3,R0      ;EVENT FLAG TO BE SET
196 002614      CALL      $CEFI
197 002620      050011      BIS      R0,(R1)      ;SET LOCAL FLAG
198 002622      CALL      $DRDSE      ;DECLARE SIGNIFICANT EVENT
199      ;
200 002626      RESTOR R0,R1,R2,R3,R4,R5
201 002642      000002      RTI

```

```

203      ;
204      ;
205      ;      READ FROM FAL MEMORY.
206      ;
207      ;
208      ;      2(SP) = ADDRESS.
209      ;      4(SP) = MEMORY SELECT CODE.
210      ;
211 002644      ;
212 002644 012746 000040      RDFAL:
213 002650      MOV.   #0$CLR,-(SP)      ;CLEAR PPS
214 002654 016646 000002      CALL.  PPCR
215 002660      MOV.   2(SP),-(SP)      ;LOAD ADDR INTO OCL POINTER.
216 002664 016646 000004      CALL.  STOP
217 002670      MOV.   4(SP),-(SP)      ;SELECT MEMORY.
218 002674      CALL.  PPCR      ;WRITE SELECTION TO CONTROL REG.
219 002700 012667 175126      CALL.  PPLB      ;DO PPS TO LOD BUS.
220 002704 012746 000040      MOV.   (SP)+,DATA1      ;GET WORD FROM FAL.
221 002710      MOV.   #0$CLR,-(SP)      ;CLEAR PPS
222 002714 012746 077777      CALL.  PPCR
223 002720      MOV.   #077777,-(SP)      ;VALUE FOR OCL POINTER.
224      ;      CALL.  STOP      ;LOAD OCL POINTER.
225 002724 011666 000004      MOV.   (SP),4(SP)      ;MOVE RETURN ADDRESS DOWN STACK.
226 002730 062706 000004      ADD.   #4,SP      ;POINT TO RETURN ADDRESS.
227 002734      RETURN.
228      ;
229      ;
230      ;      LOAD OCL POINTER.
231      ;
232      ;
233 002736      ;
234 002736 016667 000002 176424      STOP:
235 002744 012746 001001      MOV.   2(SP),QR$LBR      ;MOVE POINTER WORD TO LOD BUS REG.
236 002750 052716 000360      MOV.   #<0$LBD+0$LBP>,-(SP)      ;CLEAR DRIVE AND PULSE.
237 002754 012746 176000      BIS.   #0$CSEL,(SP)      ;CLEAR SELECTION BITS.
238 002760      MOV.   #0$INCL,-(SP)      ;SET NO-CLOCKS.
239      ;      CALL.  CSR1
240 002764 005046      CLR.   -(SP)      ;CLEAR NOTHING.
241 002766 012746 001300      MOV.   #<0$PP2+0$LBD>,-(SP)      ;SELECT PPS AND SET DRIVE.
242 002772      CALL.  CSR1
243      ;
244      ;      SET FAL LOAD
245      ;
246 002776 012767 004000 176422      MOV.   #0$FAL,QR$CR2      ;SET FAL LOAD.
247      ;
248      ;      EXTRA CLOCK FOR PPS.
249      ;
250 003004 012746 000001      MOV.   #0$LBP,-(SP)      ;CLEAR PULSE.
251 003010 052716 006000      BIS.   #0$RNC,(SP)      ;CLEAR PPS NO-CLOCK.
252 003014 005046      CLR.   -(SP)      ;SET NOTHING.
253 003016      CALL.  CSR1
254      ;
255      ;      TURN OFF FAL LOAD.
256      ;
257 003022 005067 176422      CLR.   QR$CR2.
258      ;
259      ;      DE-SELECTION

```

260					
261	003026	012746	001001	MOV	#\$LBD+\$LBP, -(SP) ; CLEAR DRIVE AND PULSE
262	003032	052716	000360	BIS	#\$CSEL, (SP) ; CLEAR SELECTION BITS
263	003036	012746	176000	MOV	#\$NCLK, -(SP) ; SET NO-CLOCKS
264	003042			CALL	CSR1
265					
266	003046	011666	000002	MOV	(SP), 2(SP) ; MOVE RETURN ADDRESS DOWN STACK
267	003052	005726		TST	(SP)+ ; POINT TO RETURN ADDRESS
268	003054			RETURN	

```

270                                     :
271                                     :
272                                     :
273                                     :
274                                     :
275                                     :
276                                     :
277                                     :
278                                     :
279                                     :
280                                     :
281                                     :
282                                     :
283                                     :
284                                     :
285                                     :
286 003056                               CSR1::
287 003056 016767 176420 174734      MOV.   QR#CR1,APLACE.       ;GET THE CURRENT VALUE.
288 003064 046667 000004 174726      BIC.   4(SP),APLACE.       ;CLEAR FIRST.
289 003072 056667 000002 174720      BIS.   2(SP),APLACE.       ;THEN SET.
290 003100 016767 174714 176420      MOV.   APLACE,QR#CR1      ;NOW RETURN IT.
291 003106 011666 000004              MOV.   (SP),4(SP)         ;MOVE RETURN ADDR TO TOP OF STACK.
292 003112 022626                      CMP.   (SP)+,(SP)+        ;BUMP STACK POINTER PAST ARGS.
293 003114                              RETURN.                   ;SPLIT.
294                                     :
295 002034'                               .END.  START.

```

ALUCKE = 040000
ALUDE = 004000
APLACE = 000020R
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BPTISR = 002570RG
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000010
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000011
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CDCL = 000034R
CNOBRE = 100000
CPCCEN = 010000
CPCR = ***** GX
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSEDCI = 100000
CSOE = 000040
CSR1 = 003056RG
CSURTE = 000100
DATA1 = 000032RG
DBR.RD = 000001
DBCPP = 001457
DBSPT = 000026
DBTPC = 000023
DISPGS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
EFBU = 000010R
EFN.3 = 000003
ELEVEN = 002510R
ENBR = 010000
FIVE = 002416R
FLUCT = 000030R
FOS = ***** GX
FOSCTL = ***** GX
HRLMRK = 020000
IQID = 000022R
LOC.EN = 000100
LOC.WR = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MBKALL = 001000
MBKCLK = 000400
MHADRK = 000100
MHLEFT = 000002
MHDE = 000004
MHWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
MYSELF = 000000R
N = 000144
OLDVEC = 000006R
PCOM = 000024R
PFLUCT = 000026R
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
PPCR = ***** GX
PPLB = ***** GX
QCLBUF = 000115
QRSCR1 = 176420
QRSCR2 = 176422
QRSLBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RDFAL = 002644R
RGD.EN = 000200
RGD.VA = 020000
R.OSGC = 000015
R.OSPC = 000014
R.OSPN = 000006
R.OSPR = 000012
R.OSTN = 000002
SEDCS = ***** GX
SEQ.CI = 000010
START = 002034R
STOP = 002736R
S\$CLR = 000000
S\$LA = 000001
S\$QB = 000005
S\$QR = 000006
S\$QX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$NAM = 000270
TD\$NEP = 176344
TD\$OTR = 176346
TD\$ORD = 000274
TD\$SU = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDW = 176364
TEMP = 001034R
THREE = 002376R
TSKTCB = 000004R
TWELVE = 002536R

TWO... 002356R
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTB = 002000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DRD = 000004
T\$EMEM = 010000
T\$FSAB = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBE = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036
T\$OBE = 004000
T\$OBF = 010000
T\$OBRA = 000034
T\$OBWA = 000032
T\$OUTA = 100000
T\$RBD0 = 000200
T\$RNB = 000040

T\$RSET = 000000
T\$SC = 000022
T\$SCLK = 020000
T\$SEG1 = 000000
T\$SEG2 = 000001
T\$SEG3 = 000002
T\$SO = 000001
T\$UBUS = 100000
T\$ICLK = 000400
T\$BEN = 000020
UBD.IN = 000020
WORD0 = 000000
WORD1 = 000002
WORD10 = 000024
WORD11 = 000026
WORD12 = 000030
WORD13 = 000032
WORD14 = 000034
WORD15 = 000036
WORD16 = 000040
WORD17 = 000042
WORD18 = 000044
WORD19 = 000046
WORD2 = 000004
WORD20 = 000050
WORD21 = 000052
WORD22 = 000054
WORD23 = 000056
WORD24 = 000060
WORD25 = 000062

WORD26 = 000064
WORD27 = 000066
WORD28 = 000070
WORD29 = 000072
WORD3 = 000006
WORD30 = 000074
WORD31 = 000076
WORD32 = 000100
WORD33 = 000102
WORD34 = 000104
WORD35 = 000106
WORD36 = 000110
WORD37 = 000112
WORD38 = 000114
WORD39 = 000116
WORD4 = 000010
WORD40 = 000120
WORD41 = 000122
WORD42 = 000124
WORD43 = 000126
WORD44 = 000130
WORD45 = 000132
WORD46 = 000134
WORD47 = 000136
WORD48 = 000140
WORD49 = 000142
WORD5 = 000012
WORD50 = 000144
WORD51 = 000146
WORD52 = 000150

WORD53 = 000152
WORD54 = 000154
WORD55 = 000156
WORD56 = 000160
WORD57 = 000162
WORD58 = 000164
WORD59 = 000166
WORD6 = 000014
WORD60 = 000170
WORD61 = 000172
WORD62 = 000174
WORD63 = 000176
WORD64 = 000200
WORD65 = 000202
WORD66 = 000204
WORD67 = 000206
WORD68 = 000210
WORD69 = 000212
WORD7 = 000016
WORD70 = 000214
WORD71 = 000216
WORD72 = 000220
WORD73 = 000222
WORD74 = 000224
WORD75 = 000226
WORD76 = 000230
WORD77 = 000232
WORD78 = 000234
WORD79 = 000236
WORD8 = 000020

WORD80 = 000240
WORD81 = 000242
WORD82 = 000244
WORD83 = 000246
WORD84 = 000250
WORD85 = 000252
WORD86 = 000254
WORD87 = 000256
WORD88 = 000260
WORD89 = 000262
WORD9 = 000022
WORD90 = 000264
WORD91 = 000266
WORD92 = 000270
WORD93 = 000272
WORD94 = 000274
WORD95 = 000276
WORD96 = 000300
WORD97 = 000302
WORD98 = 000304
WORD99 = 000306
WORDVAL = 000310
XTREAD = 001000
XTWRITE = 000400
\$CEFI = ***** GX
\$DRDSE = ***** GX
\$KTCCB = ***** GX
\$\$\$ = 000000R
\$\$\$OST = 000016
\$\$T1 = 000006

002

. ABS. 000000 000
003116 001
\$DPB\$\$ 000016 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3999 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:49
HORFOS, HORFOS /-SP=C20, IJM, C20, IJHORFOS

FOSGET PROGRAM MACRO M1110 27-MAR-80 14:32
TABLE OF CONTENTS

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

9-	13	FILE INITIALIZATION
10-	40	FOSGET CODE
12-	98	ERROR HANDLING ROUTINE

1
2
3
4
5
6
7
8
9
10
11

```
      .TITLE - FOSGET-PROGRAM  
      ;  
      ; WRITE-FLU-OCCURRENCE-SUMMARY-FILE  
      ;  
      ; THIS-PROGRAM-EXTRACTS-THE-FLU-OCCURRENCE-COUNTERS-FOR-A-GIVEN-SEARCH-UNIT  
      ; FROM-COMMON-AND-WRITES-THEM-TO-FILE-[7,4]FOS.DAT. IT-THEN-SENDS-THE  
      ; FILE-DESCRIPTOR-TO-CCOUT, WHICH, IN-TURN, TRANSMITS-THE-FILE-TO-THE-  
      ; MASTER-COMPUTER.  
      ;  
      .MCALL - FDBDF$, FDRC$, FDBK$, FDDP$, FRSZ$, OFNB$, FDAT$, CLOSE$,  
      .MCALL - WRITE$, WAIT$, FINIT$, EXIT$, SETF$, DECL$, SDAT$
```

```
13 .SBTTL FILE INITIALIZATION
14 ;
15 ; FLU OCCURRENCE SUMMARY FILE
16 ;
17 000000 FDB: FDBDF$
18 000140 FDRCS$A FD,RWM
19 000140 FDBK$A FOSHD,2*N,BUFU,,1,IOST
20 000140 FDRP$A 1
21 000140 FRSZ$ 0
22 .EVEN
23 ;
24 ; BUFFERS AND EQUATES
25 ;
26 000140 002 RCVB: .BYTE 2 ;PROGRAM ID
27 000141 000 .BYTE 0 ;TRANSMIT TO MASTER
28 000142 FDISC: .BLKW 4 ;FILE DESCRIPTOR AREA
29 ;
30 000152 117 106 FOSHD: .ASCII /OF/ ;FILE ID
31 000154 000000 .WORD 0 ;PAD
32 000156 000000 .WORD 0 ;BATCH NO
33 000160 000000 FORYN0: .WORD 0 ;# QUERIES IN FOS
34 000162 FOSBLK: .BLKW N,BUFU-4 ;FOS COUNTER AREA
35 ;
36 004152 IOST: .WORD 0
37 004152 000000 PAR1: .WORD 0
38 004154 000000 PAR2: .WORD 0
```

```

40 .SBTTL FOSGET CODE
41 004156 FOSGET:
42 004156 FINIT$
43
44 004162 012700 000000 MOV #FDB,R0 ;FDB ADDRESS
45 004166 012701 000020 MOV #FN,FSD,R1 ;FILE NUMBER
46 004172 CALL BLDNFL ;BUILD FILE NAME BLOCK IN FDB
47
48 ;
49 ; ALLOCATE AND OPEN FILE
50
51 004176 FDATA# #FDB, ..., #-4.
52 004210 OFNB$# #FDB
53 004226 103006 BCC 10$
54 004230 116001 000052 MOV# F,ERR(R0),R1
55 004234 CALL FCSERR
56 004240 000167 000162 JMP FOSXIT
57 004244 016767 000020 173706 10$: MOV FOSCTL+2, FORYNO ;MOVE QUERY COUNT FROM COMMON
58 004252 016701 000000 MOV FOSCTL,R1 ;GET WORD SIZE OF FOS
59 004256 012702 000000 MOV #FOS,R2 ;LOAD ADDRESS OF FOS
60 004262 012703 000162 MOV #FOSBLK,R3 ;LOAD ADDRESS OF FILE OUTPUT BUFFER
61 004266 15$:
62 004266 011223 MOV @R2,(R3)+ ;MOVE WORD FROM FOS TO BUFFER
63 004270 005022 CLR (R2)+ ;RESET WORD IN FOS
64 004272 SSOB R1,15$
65
66 ;
67 ; WRITE FILE AND SET UP FILE DESCRIPTOR
68
69 004276 WRITE# #FDB
70 004306 WAIT# #FDB
71 004316 103006 BCC 20$
72 004320 116001 000052 MOV# F,ERR(R0),R1
73 004324 CALL FCSERR
74 004330 000167 000072 JMP FOSXIT
75 004334 20$:
76 004342 016767 173542 000000 MOV FDB+F,FNB+N,FID,FDSC+FD,FID
77 004350 016767 173536 000000 MOV FDB+F,FNB+N,FID+2,FDSC+FD,FID+2
78 004356 016767 173544 000000 MOV FDB+F,FNB+N,FVER,FDSC+FD,FVR
79 004356 012767 000020 000000 MOV #FN,FSD,FDSC+FD,FNB
80 004364 CLOSE# #FDB
```

FOSGET PROGRAM MACRO M1110
FOSGET CODE

27-MAR-88 14:32 PAGE 11

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
82.      : SHIP FILE INFO TO CCOUT.  
83.      :  
84 004374      :  
85 004402 103003      : SDAT$C CCOUT.RCVB  
86 004404      : BCC 30$  
87 004410 000406      : CALL DIRERR  
88 004412      : BR FOSXIT  
89 004412      : 30$: SETF$C CF.COT  
90 004420      : DECL$S  
91.      :  
92.      : EXIT  
93.      :  
94 004426      : FOSXIT:  
95.      : MOUT$S #MSG0  
96 004426      : EXIT$S
```

```

98          .SBTTL  ERROR HANDLING ROUTINE
99          .NLIST  BEX
100         :
101         : DIRECTIVE ERROR
102         :
103 004434 011667 177512 DIRERR: MOV   (SP),PAR1
104 004440 116700 0000000 MOVB  $DSW,R0
105 004444 010067 177504 MOV   R0,PAR2
106 004450          MOUT$S #MSG1,#PAR1
107 004470 000207          RETURN
108         :
109         : FCS ERROR
110         :
111 004472 011667 177454 FCSERR: MOV   (SP),PAR1
112 004476 010167 177452 MOV   R1,PAR2
113 004502          MOUT$S #MSG2,#PAR1
114 004522 000207          RETURN
115         :
116         : STRING DESCRIPTORS
117         :
118 004524 000040 MSG1:  .WORD  LN1E-LN1
119 004526 004540' .WORD  LN1
120 004530 000032' MSG2:  .WORD  LN2E-LN2
121 004532 004500' .WORD  LN2
122 004534 000024 MSG0:  .WORD  LN0E-LN0
123 004536 004632' .WORD  LN0
124         :
125         : FORMAT STRINGS
126         :
127 004540 120 103 040 LN1:  .ASCIZ  /PC= %10, DIRECTIVE ERROR= %1D/
128 004600          LN1E:
129 004600 120 103 040 LN2:  .ASCIZ  /PC= %10, FCS ERROR= %1D/
130 004632          LN2E:
131 004632 106 117 123 LN0:  .ASCIZ  /FOSGET PROGRAM EXIT/
132 004656          LN0E:
133 004156' .END  FOSGET
    
```

BITVAL = 000000	FN.FSD 000020	004 F.RTYP = 000000	SR.LEN 000122	002 T\$EMEM = 010000
BIT0 = 000001	FN.MHR 000010	004 F.SEQN = 000100	SR.LIN 000066	002 T\$FSAA = 000000
BIT1 = 000002	FN.NMB 000022	004 F.SPDV = 000072	SR.LIP 000062	002 T\$FSAB = 000004
BIT10 = 002000	FN.QLS 000006	004 F.SPUN = 000074	SR.MDN 000006	002 T\$FSAC = 000014
BIT11 = 004000	FN.RDC 000014	004 F.STBK = 000036	SR.NDC 000042	002 T\$FSB2 = 000010
BIT12 = 010000	FN.UPD 000012	004 F.UNIT = 000136	SR.NDS 000036	002 T\$B = 000026
BIT13 = 020000	FOS = ***** GX	F.URBD = 000020	SR.NIN 000030	002 T\$IBAR = 000024
BIT14 = 040000	FOSBLK 000162R	F.VBN = 000064	SR.NIP 000022	002 T\$IBE = 020000
BIT15 = 100000	FOSCTL = ***** GX	F.VBSZ = 000060	SR.SDB 000032	002 T\$IBF = 040000
BIT2 = 000004	FOSGET 004156R	IDST = 004152R	SR.SRC 000002	002 T\$ICD = 000040
BIT3 = 000010	FOSH D 000152R	LN0 = 004632R	SR.SUN 000000	002 T\$MODE = 004000
BIT4 = 000020	FOSKIT 004426R	LN0E = 004656R	SR.TWS 000056	002 T\$OB = 000036
BIT5 = 000040	FD,URT = ***** GX	LN1 = 004540R	SR.WSL 000052	002 T\$OBE = 004000
BIT6 = 000100	FD,RYNO 000160R	LN1E = 004600R	SR.YR = 000004	002 T\$OBF = 010000
BIT7 = 000200	F.ACTL = 000076	LN2 = 004600R	SR.IIN 000024	002 T\$OBRA = 000034
BIT8 = 000400	F.ALCC = 000040	LN2E = 004632R	SR.IIP 000016	002 T\$OBWA = 000032
BIT9 = 001000	F.BBFS = 000062	MSGOUT = ***** GX	S.BFHD = 000020	T\$OUTA = 100000
BLDNFL = ***** GX	F.BDE = 000070	MSG0 004534R	S.DABA = 000006	T\$RBD0 = 000200
BYTE0 = 000000	F.BGBC = 000057	MSG1 004524R	S.DAEF = 000010	T\$RNB = 000040
BYTE1 = 000001	F.BKDN = 000026	MSG2 004530R	S.DATN = 000002	T\$RSET = 040000
BYTE2 = 000002	F.BKDS = 000020	MS.DGN = 010000	S.ETEF = 000002	T\$SC = 000022
BYTE3 = 000003	F.BKEF = 000050	N = 000012	S.FATT = 000016	T\$SCLK = 020000
BYTE4 = 000004	F.BKP1 = 000051	N.BFAC = 000004	S.FDB = 000140	T\$SEG1 = 000000
BYTE5 = 000005	F.BKST = 000024	N.BHGH = 000006	S.FNAM = 000006	T\$SEG2 = 000001
BYTE6 = 000006	F.BKVB = 000064	N.BTCH = 000004	S.FNB = 000036	T\$SEG3 = 000002
BYTE7 = 000007	F.CHR = 000075	N.BUFB = 004000	S.FNBW = 000017	T\$SO = 000001
BYTE8 = 000010	F.CHTG = 000034	N.BUFW = 002000	S.FNTY = 000004	T\$UBUS = 100000
BYTE9 = 000011	F.DFNB = 000046	N.DID = 000024	S.FITYP = 000002	T\$ICLK = 000400
BYTVAL = 000012	F.DSPT = 000044	N.DVNM = 000032	S.NFEN = 000020	T\$BEN = 000020
CF.COT = 000041	F.DVHM = 000134	N.FID = 000000	TD\$CTR = 176370	WORD0 = 000000
CF.DGN = 000046	F.EFBK = 000010	N.FNAM = 000006	TD\$CTW = 176360	WORD1 = 000002
CF.DHR = 000042	F.EFN = 000050	N.FOS = 000764	TD\$INL = 004000	WORD2 = 000004
CF.DMC = 000047	F.EOBB = 000032	N.FITYP = 000014	TD\$MEM = 000270	WORD3 = 000006
CF.HBR = 000045	F.ERR = 000052	N.FVER = 000016	TD\$OAR = 176344	WORD4 = 000010
CF.HRL = 000044	F.FACC = 000043	N.NEXT = 000022	TD\$OTR = 176346	WORD5 = 000012
CF.UPD = 000043	F.FBY = 000014	N.QUERY = 000031	TD\$ORD = 000274	WORD6 = 000014
DBSLEN = 000116	F.FNAM = 000110	N.STAT = 000020	TD\$RST = 176366	WORD7 = 000016
DG.ERR = 001000	F.FNB = 000102	N.SUNT = 000002	TD\$SW = 176376	WORD8 = 000020
DG.SDF = 002000	F.FITYP = 000116	N.UNIT = 000034	TD\$STAR = 176372	WORD9 = 000022
DG.TDF = 004000	F.FVER = 000120	PAR\$\$\$ = 000027	TD\$TAR = 176362	WRDVAL = 000024
DIRERR = 004434R	F.HIBK = 000004	PAR1 004152R	TD\$TDR = 176374	\$D\$W = ***** GX
FCSERR = 004472R	F.LUN = 000042	PAR2 004154R	TD\$TDW = 176364	\$\$\$ = 000012R
FDB = 0000000R	F.MBCT = 000054	RCVB 000140R	T\$AD = 000020	\$\$\$OST = 000004
FDSC = 000142R	F.MBC1 = 000055	SR.ARE = 000114	T\$BD = 000002	\$\$\$T1 = 000005
FD.FID = 000000	003 F.MBFG = 000056	SR.ARS = 000106	002 T\$BD = 000010	.CLOSE = ***** G
FD.FNB = 000006	003 F.NRBD = 000024	SR.DAY = 000010	002 T\$BSO = 100000	.FINIT = ***** G
FD.FVR = 000004	003 F.NREC = 000030	SR.DLT = 000014	002 T\$BT = 000020	.FSRCB = ***** G
FD.LEN = 000010	003 F.OVBS = 000030	SR.ECB = 000047	002 T\$BTAR = 000030	.OPFN7 = ***** G
FD.RHM = ***** GX	F.RACC = 000016	SR.ECH = 000046	002 T\$BT D = 002000	.WAIT = ***** G
FN.ACK = 000016	004 F.RATT = 000001	SR.ECL = 000050	002 T\$CD = 000100	.WRITE = ***** G
FN.FSA = 000000	004 F.RCHM = 000034	SR.FIB = 000012	002 T\$CLK = 002000	...PC1 = 000000R
FN.FSB = 000002	004 F.RCTL = 000017	SR.GRE = 000100	002 T\$D1SK = 000200	...PC2 = 000140R
FN.FSC = 000004	004 F.RSIZ = 000002	SR.GRS = 000072	002 T\$DRD = 000004	...TPC = 000020

. ABS = 000000 000
 004656 001
 SRCOFF = 000122 002

FOSGET, PRO... MACRO-H1110 27-MAR-66 14:33 PAGE 13
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

FOSCOF 000010 003
FNOFFS 000022 004
\$FSR1 000000 005
\$DPB\$ 000016 006
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 5339 WORDS (21 PAGES)
DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
ELAPSED TIME: 00:00:37
FOSGET, FOSGET /-SP=C20.1JP.C, FOSGET

DATA BASE
UPDATE

10-	1	GCET
11-	1	TASK NAME: DBPROC - DATA BASE PROCESSOR
11-	2	
11-	3	
11-	4	
11-	5	
11-	6	
11-	7	CREATION DATE:
11-	8	MODIFICATIONS:
12-	10	TASK OVERVIEW:
14-	52	TASK IMMEDIATES:
15-	143	TASK VARIABLES AND BUFFERS:
16-	186	FDB DEFINITIONS:
17-	202	DBPROC - MAIN CONTROL:
18-	369	ADDOC - ADD NEW DOCUMENT
19-	506	NXTCHR:
19-	578	CLRBUF - CLEAR BUFFER:
20-	598	CMP3WD - COMPARE THREE WORD ITEMS:
21-	654	COBHDR - CREATE OUTPUT BUFFER HEADER:
22-	766	--- COMPUT:
23-	810	CRDSS - CREATE DOCUMENT START SEQUENCE:
24-	874	CRSDSS - CREATE SUB-DOCUMENT START SEQUENCE:
25-	1003	ERRSEQ - ERROR SEQUENCE:
26-	1052	EXIT - EXIT FOR FATAL ERROR ROUTINE:
27-	1059	MINCMD - CREATE MINIMAL COMMAND RECORD:
28-	1089	--- OPEN:
29-	1134	PACK6 - PACK SIX BIT CHARACTERS:
30-	1219	PRDOC - PURGE OR READ DOCUMENTS:
31-	1312	PREPEX - PREPROCESS EXCHANGE REQUEST:
32-	1354	--- READ:
33-	1411	--- SEND:
34-	1458	SORTID - SORT IDS IN ASCENDING ORDER SEQUENCE:
35-	1508	VALIDID - VALIDATE DOCUMENT ID:
36-	1545	SNDEND - SEND END-OF-UPDATE EXCHANGE:
37-	1579	ERROR MESSAGES

```
1          .SBTTL . CCET
2 000000    .PSECT . CCET
3          ; CHARACTER CODE EQUATE TABLE (CCET)
4          ;
5          ; ASCII TO 6-BIT CONVERSION TABLE
6          ;
7 000000    200    CET6B:: .BYTE 200    ;NULL-----0
8 000001    074    .BYTE 74      ;SOH
9 000002    074    .BYTE 74      ;STX
10 000003    074   .BYTE 74      ;ETX
11 000004    074   .BYTE 74      ;EOT
12 000005    074   .BYTE 74      ;END
13 000006    074   .BYTE 74      ;ACK
14 000007    074   .BYTE 74      ;BEL
15 000010    074   .BYTE 74      ;BS-----10
16 000011    074   .BYTE 74      ;HT
17 000012    005   .BYTE 5      ;LF
18 000013    074   .BYTE 74      ;VT
19 000014    074   .BYTE 74      ;FF
20 000015    005   .BYTE 5      ;CR
21 000016    074   .BYTE 74      ;SO
22 000017    074   .BYTE 74      ;SI
23 000020    074   .BYTE 74      ;DLE-----20
24 000021    201   .BYTE 201     ;DC1 - PARAGRAPH
25 000022    202   .BYTE 202     ;DC2 - ZONE
26 000023    203   .BYTE 203     ;DC3 - SUBZONE
27 000024    204   .BYTE 204     ;DC4 - SENTENCE
28 000025    074   .BYTE 74      ;NAK
29 000026    074   .BYTE 74      ;SYN
30 000027    074   .BYTE 74      ;ETB
31 000030    074   .BYTE 74      ;CAN-----30
32 000031    074   .BYTE 74      ;EM
33 000032    074   .BYTE 74      ;SUB
34 000033    074   .BYTE 74      ;ESC
35 000034    074   .BYTE 74      ;FS
36 000035    074   .BYTE 74      ;GS
37 000036    074   .BYTE 74      ;RS
38 000037    074   .BYTE 74      ;US
39 000040    005   .BYTE 5      ;SPACE----40
40 000041    006   .BYTE 6      ;!
41 000042    007   .BYTE 7      ;"
42 000043    010   .BYTE 10     ;#
43 000044    074   .BYTE 74      ;$
44 000045    011   .BYTE 11     ;%
45 000046    012   .BYTE 12     ;&
46 000047    013   .BYTE 13     ;'
47 000050    014   .BYTE 14     ;(-----50
48 000051    015   .BYTE 15     ;)
49 000052    016   .BYTE 16     ;*
50 000053    017   .BYTE 17     ;+
51 000054    020   .BYTE 20     ;,
52 000055    021   .BYTE 21     ;-
53          000056 .DECPT=, .CET6B
54 000056    022   .BYTE 22     ;.
55 000057    023   .BYTE 23     ;/
56 000060    024   .BYTE 24     ;0-----60
57 000061    025   .BYTE 25     ;1
```

58 000062	026	.BYTE	26	:2	
59 000063	027	.BYTE	27	:3	
60 000064	030	.BYTE	30	:4	
61 000065	031	.BYTE	31	:5	
62 000066	032	.BYTE	32	:6	
63 000067	033	.BYTE	33	:7	
64 000070	034	.BYTE	34	:8	-----70
65 000071	035	.BYTE	35	:9	
66 000072	075	.BYTE	75	::	
67 000073	076	.BYTE	76	::	
68 000074	074	.BYTE	74	:<	
69 000075	036	.BYTE	36	:=	
70 000076	074	.BYTE	74	:>	
71 000077	077	.BYTE	77	:?	
72 000100	074	.BYTE	74	:@	-----100
73 000101	037	.BYTE	37	:A	
74 000102	040	.BYTE	40	:B	
75 000103	041	.BYTE	41	:C	
76 000104	042	.BYTE	42	:D	
77 000105	043	.BYTE	43	:E	
78 000106	044	.BYTE	44	:F	
79 000107	045	.BYTE	45	:G	
80 000110	046	.BYTE	46	:H	-----110
81 000111	047	.BYTE	47	:I	
82 000112	050	.BYTE	50	:J	
83 000113	051	.BYTE	51	:K	
84 000114	052	.BYTE	52	:L	
85 000115	053	.BYTE	53	:M	
86 000116	054	.BYTE	54	:N	
87 000117	055	.BYTE	55	:O	
88 000120	056	.BYTE	56	:P	-----120
89 000121	057	.BYTE	57	:Q	
90 000122	060	.BYTE	60	:R	
91 000123	061	.BYTE	61	:S	
92 000124	062	.BYTE	62	:T	
93 000125	063	.BYTE	63	:U	
94 000126	064	.BYTE	64	:V	
95 000127	065	.BYTE	65	:W	
96 000130	066	.BYTE	66	:X	-----130
97 000131	067	.BYTE	67	:Y	
98 000132	070	.BYTE	70	:Z	
99 000133	071	.BYTE	71	:_	
100 000134	072	.BYTE	72	:`	(DEC. POINT)
101 000135	073	.BYTE	73	:!	
102 000136	074	.BYTE	74	:"	
103 000137	074	.BYTE	74	:#	
104 000140	074	.BYTE	74	:@	-----140
105 000141	037	.BYTE	37	:A	
106 000142	040	.BYTE	40	:B	
107 000143	041	.BYTE	41	:C	
108 000144	042	.BYTE	42	:D	
109 000145	043	.BYTE	43	:E	
110 000146	044	.BYTE	44	:F	
111 000147	045	.BYTE	45	:G	
112 000150	046	.BYTE	46	:H	-----150
113 000151	047	.BYTE	47	:I	
114 000152	050	.BYTE	50	:J	

DBPROC - DATA BASE PROCESSOR
CCET

MACRO M1110 27-MAR-80 15:39 PAGE 10-2
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

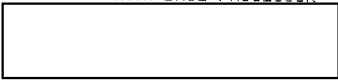
115 000153	051	.BYTE	51	:K
116 000154	052	.BYTE	52	:L
117 000155	053	.BYTE	53	:M
118 000156	054	.BYTE	54	:N
119 000157	055	.BYTE	55	:O
120 000160	056	.BYTE	56	:P-----160
121 000161	057	.BYTE	57	:Q
122 000162	060	.BYTE	60	:R
123 000163	061	.BYTE	61	:S
124 000164	062	.BYTE	62	:T
125 000165	063	.BYTE	63	:U
126 000166	064	.BYTE	64	:V
127 000167	065	.BYTE	65	:W
128 000170	066	.BYTE	66	:X-----170
129 000171	067	.BYTE	67	:Y
130 000172	070	.BYTE	70	:Z
131 000173	074	.BYTE	74	:[
132 000174	074	.BYTE	74	:\ :
133 000175	074	.BYTE	74	:] :
134 000176	074	.BYTE	74	:^ :
135 000177	074	.BYTE	74	:DEL-----177 :
136				
137				
138				

END OF ASCII TO 6-BIT CONVERSION TABLE

DBPROC - DATA-BASE-PROCESSOR. MACRO M1110 27 MAR 80 17:30 R
TASK-NAME: DBPROC - DATA-BASE-PROCESSOR. Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1
0
7
0
6
0
4
0
0
0
1

.SBTTL TASK-NAME: DBPROC - DATA-BASE-PROCESSOR.
.SBTTL
.SBTTL
.SBTTL
.SBTTL PHONE:
.SBTTL
.SBTTL CREATION-DATE:
.SBTTL MODIFICATIONS:



DBPROC - DATA BASE PROCESSOR
TASK OVERVIEW

MACRO M1110 27-MAR-80 13:30 PAGE 12

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
10 .SBTTL - TASK OVERVIEW
11 ;
12 ; DBPROC IS NOTIFIED BY MSCHED, VIA THE RCVD, TO PROCESS THE
13 ; [7,4]DBUPD.SPL FILES, LOWEST VERSION NUMBER FIRST.
14 ;
15 ; DBPROC PERFORMS PRELIMINARY ERROR CHECKING ON THE UPDATE REQUESTS
16 ; BEFORE PROCEEDING TO UPDATE PROCESSING. THE FIRST ERROR CHECK IS FOR
17 ; LEGAL REQUEST TYPE. THEN A PRELIMINARY CHECK IS MADE FOR VALID DOCUMENT
18 ; ID FOR THE APPROPRIATE SEARCH UNIT. IF ANY ERROR IS FOUND, A MINIMAL
19 ; COMMAND RECORD IS CREATED AND SENT ACROSS THE LINK TO THE
20 ; APPROPRIATE SEARCH UNIT SO AS TO BE RECORDED IN THE DATA BASE UPDATE
21 ; ACKNOWLEDGEMENT EXCHANGE THAT IS SENT BACK TO THE HOST COMPUTER.
22 ;
23 ; IF NO ERRORS ARE FOUND, VARIOUS PROCESSING IS PERFORMED DEPENDING ON
24 ; THE REQUEST TYPE INVOLVED. THERE ARE FIVE UPDATE REQUESTS: ADD DOCUMENT,
25 ; PURGE DOCUMENT, MODIFY-REPLACE DOCUMENT, MODIFY-OVERLAY DOCUMENT CHARACTERS,
26 ; AND READ DOCUMENTS.
27 ;
28 ; THE FOLLOWING PROCESSING IS PERFORMED FOR EACH OF THE FOLLOWING
29 ; UPDATE REQUESTS:
30 ;
31 ; ADD DOCUMENT: PACK DOCUMENT AND ITS CORRESPONDING WHITE SPACE
32 ;
33 ; PURGE DOCUMENTS: ORDER DOCUMENT IDS IN ASCENDING ORDER
34 ;
35 ; MODIFY-REPLACE DOCUMENT: PACK THE DOCUMENT
36 ;
37 ; MODIFY-OVERLAY DOCUMENT CHARACTERS: PACK THE CHARACTERS
38 ;
39 ; READ DOCUMENTS: ORDER DOCUMENT IDS IN ASCENDING ORDER
40 ;
41 ;
42 ; THE OUTPUT OF DBPROC IS QUEUED TO THE APPROPRIATE DMC FOR
43 ; TRANSMISSION TO A SEARCH UNIT, ONE BLOCK (2048 BYTES PER BLOCK)
44 ; AT A TIME.
45 ;
```

DBPROC - DATA BASE PROCESSOR
TASK OVERVIEW

MACRO-M1110 27-MAR-88 13:38 P. 13

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

47
48 000000
49
50

.TITLE : DBPROC - DATA BASE PROCESSOR
.PSECT :
.ENABL : AMA
.NLIST : MEB


```
52 .          .SBTTL  TASK IMMEDIATES.  
53 .          .MCALL  RCVMX$$,EXIT$$,SDAT$$,RSUM$$,QIOW$,MRKT$$,WTSE$$  
54 .          .MCALL  FDBDF$,FDRCA$,FDBK$,FDOA$,FDAT$,FSRSZ$  
55 .          .MCALL  FINIT$  
56 .          .MCALL  OFNB$,OFNB$,READ$,WAIT$,CLOSE$,DELET$,WRITE$  
57 .          .MCALL  OPNT$,ABRT$  
58 .          ;  
59          000001 REF  = 1 ; READ EVENT FLAG  
60          000003 SND.EF = 3 ; DMC SEND EVENT FLAG  
61          000002 WAT.EF = 2 ; MASS UPDATE WAIT EVENT FLAG  
62          000003 RD.LUN = 3 ; READ FROM FILE LUN  
63          000001 SD.LUN = 1 ; LUN OF XMO  
64          ;  
65          ; CLSUP = 37 ; CLOSE UP MARKER  
66          000021 PRGH = 21 ; PARAGRAPH MARKER  
67          000000 SDMK = 00 ; SUB-DOCUMENT MARKER  
68          000024 SNTC = 24 ; SENTENCE MARKER  
69          000023 SZMK = 23 ; SUB-ZONE MARKER  
70          000022 ZMK = 22 ; ZONE MARKER  
71          ;  
72          000000 000000 000000 IOSTAT: .WORD 0.0 ; I/O STATUS BLOCK FOR SEND  
73          000004 000000 000000 RIOST: .WORD 0.0 ; READ STATUS WORD  
74          ;  
75          ; OFFSET DEFINITIONS FOR RECEIVE QUEUE DATA  
76          ;  
77          000000 .PSECT  RCVOFF,ABS  
78          000000 RQ.DUM: .BLKW 2 ; PAD FILL  
79          000004 RQ.CB: .BLKB 1 ; COMMAND BYTE  
80          000005 RQ.CS: .BLKB 1 ; COMMAND SOURCE  
81          000006 RQ.BN: .BLKW 1 ; BATCH NUMBER  
82          ;  
83          ; OFFSET DEFINITIONS FOR DOCUMENT UPDATE REQUESTS  
84          ;  
85          ; ADD DOCUMENT REQUEST EXCHANGE  
86          ;  
87          000000 .PSECT  ADDOFF,ABS  
88          000000 DU.TYP: .BLKW 1 ; UPDATE REQUEST TYPE  
89          000002 DU.WSC: .BLKW 1 ; WHITE SPACE CHARACTER COUNT  
90          000004 DU.IDH: .BLKW 1 ; HIGH ORDER PORTION OF DOC. ID  
91          000006 DU.IDM: .BLKW 1 ; MIDDLE PORTION OF DOC. ID  
92          000010 DU.IDL: .BLKW 1 ; LOW ORDER PORTION OF DOC. ID  
93          000012 DU.TCC: .BLKW 2 ; TEXT CHARACTER COUNT  
94          000016 DU.DT: .BLKB 1 ; DOCUMENT TYPE  
95          000017 DU.SDT: .BLKB 1 ; START OF DOCUMENT TEXT  
96          ;  
97          ; DOCUMENT MODIFY-OVERLAY DOCUMENT CHARACTERS  
98          ;  
99          000016 DU.OCD = DU.TCC+WORD2 ; OFFSET FOR CHARACTER OVERLAY  
100         000022 DU.SCT = DU.OCD+WORD2 ; START OF CHARACTER TEXT  
101         000000 .PSECT  PUROFF,ABS  
102         ;  
103         ; PURGE AND READ DOCUMENT REQUESTS  
104         ;  
105         000000 PR.TYP: .BLKW 1 ; UPDATE REQUEST TYPE  
106         000002 PR.PAD: .BLKW 1 ; PAD FILL  
107         000004 PR.DIC: .BLKW 1 ; DOCUMENT ID COUNT  
108         000006 PR.EDI: .BLKW 1 ; FIRST DOCUMENT ID
```

DBPROC - DB BASE PROCESSOR
TASK IMMEDIATES

MACRO M1110 27-MAR-80 17:20 P. 14-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
109 000006 N:BPRES = WORD3 ; # OF BYTES/ENTRY FOR READ & PURGE REQUESTS
110 000020 N:BDSD = WORD8 ; NO. OF BYTES/ENTRY FOR DELETE SUB-DOC.
111 ;
112 ; OFFSET DEFINITIONS FOR PROCESSED COMMAND REQUESTS
113 ;
114 000000 .PSECT: PCROFF,ABS
115 000000 PC:DBT: .BLKW 1 ; DATA BASE EXCHANGE TYPE
116 000002 PC:PAD: .BLKW 1 ; PAD FILL
117 000004 PC:TYP: .BLKW 1 ; REQUEST TYPE
118 000006 PC:STW: .BLKW 1 ; STATUS WORD FOR ACTION REQUESTED
119 000010 PC:IDH: .BLKW 1 ; HIGH ORDER DOCUMENT ID WORD
120 000012 PC:IDM: .BLKW 1 ; MIDDLE ORDER DOCUMENT ID WORD
121 000014 PC:IDL: .BLKW 1 ; LOW ORDER DOCUMENT ID WORD
122 000016 PC:TCC: .BLKW 2 ; TEXT CHARACTER COUNT
123 000022 PC:USC: .BLKW 1 ; REQUIRED WHITE SPACE COUNT
124 000024 PC:UWS: .BLKW 1 ; UNUSED WHITE SPACE COUNT
125 000026 PC:PWC: .BLKW 2 ; PACKED WORD COUNT (TEXT & WHITE SPACE)
126 000032 PC:PTC: .BLKW 2 ; PACKED TEXT COUNT ONLY (WORDS)
127 000036 PC:SPD: .BLKW 1 ; START OF PACKED DATA
128 ;
129 000010 RP:NID: = PC:IDH ; # OF DOC. IDS FOR READ AND PURGE REQUESTS
130 000012 RP:FID: = PC:IDM ; OFFSET FOR FIRST DOC. ID IN READ & PURGE REQUESTS
131 000010 .PSECT:
132 ;
133 ; BIT DEFINITIONS FOR ACTION STATUS WORDS
134 ;
135 000013 AS:EOF: = 11 ; END-OF-FILE FOUND BEFORE END OF DOCUMENT
136 000011 AS:IVI: = 9 ; INVALID ID
137 000012 AS:IVR: = 10 ; INVALID REQUEST
138 000014 AS:FAT: = 12 ; FATAL ERROR IN DBPROC
139 000001 AS:SUP: = 1 ; SUCCESSFUL UPDATE
140 ;
141 ;
```

```
143          .SBTTL - TASK VARIABLES AND BUFFERS
144          ;
145 000010      INPUT:  .BLKW  N.BUFW      ; INPUT BUFFER
146 004010      OUTPUT: .BLKW  N.BUFW    ; OUTPUT BUFFER
147 010010      OUTPUT2: .BLKW N.BUFW   ; 2ND OUTPUT BUFFER FOR READ & PURGE REQUESTS
148 014010      PRB:    .BLKW  15      ; PARAMETER REQUEST BLOCK TO RECEIVE FILE ID
149 014046 000000 000000      CHARCT: .WORD  0,0 ; TEXT CHARACTER COUNT
150 014052 000000      CMPST:  .WORD  0 ; COMPARE STATUS WORD AND VALIDITY INDICATOR
151 014054 000002      NO.SU:  .WORD  2 ; NUMBER OF SEARCH UNITS
152 014056 000001      CSU:    .WORD  1 ; CURRENT SU LUN FOR CURRENT UPDATE
153          ;
154 014060 000000      EXTYPE: .WORD  0 ; REQUEST LUN 1 = XM0, LUN 2 = XM1
155          ;
156          ;
157          ;
157 014062 000000      FILNBR: .WORD  0 ; FILE NUMBER FOR RESIDENT FCS
158 014064 000000      MASUPD: .WORD  0 ; MASS UPDATE FLAG
159 014066 000000      FLAG:   .WORD  0 ; NO. OF FILES PROCESSED IN CURRENT UPDATE BATCH
160 014070 052073 031314      MSCHED: .RAD50 /MSCHED/ ; RAD50 MASTER SCHEDULER TASK NAME
161 014074 000000      NBLK.I: .WORD  0 ; NUMBER OF BLOCKS TO INPUT
162 014076 000000      NBLK.O: .WORD  0 ; NUMBER OF BLOCKS TO OUTPUT
163 014100 000000      NBRD:   .WORD  0 ; # OF BYTES READ IN I/O BLOCK READ
164 014102 000000 000000      PARI:  .WORD  0,0 ; PARAMETER FOR ERROR MESSAGES
165 014106 000000 000000      PTCW:  .WORD  0,0 ; PACKED TEXT COUNT ONLY IN WORDS
166 014112 000000 000000      PWCT:  .WORD  0,0 ; PACKED WORD COUNT (TEXT & WHITE SPACE)
167 014116 000000      RDRSLT: .WORD  0 ; READ RESULT INDICATOR
168 014120 000000      REDO.F: .WORD  0 ; REDO SPOOL FILE FLAG
169 014122 000000 000000      R8:    .WORD  0,0 ; ADDITIONAL REGISTER
170 014126 000000      R9:    .WORD  0 ; ADDITIONAL REGISTER
171 014130 000000 000000      SAVE:  .WORD  0,0 ; FOR COMMUNICATION TO DBLOAD
172 014134 000000      STATWD: .WORD  0 ; STATUS WORD FOR PROCESSED EXCHANGE
173 014136 000000 000001      VBLKN: .WORD  0,1 ; VIRTUAL BLOCK NUMBER
174 014142 000000      WSPCNT: .WORD  0 ; WHITE SPACE COUNT
175          ;
176          ;
177          ;
177          ;
178 014144 010 000      AK1:  .BYTE  10,0 ; AK1 - UPDATE DONE, DONE EXCHANGE TRANSMISSION
179          ;
180 014146 000000      .WORD  0 ; IN ERROR
181 014150 000000      .WORD  0 ; SEARCH UNIT NUMBER
182 014152 010 000      AK2:  .BYTE  10,0 ; BATCH NUMBER
183 014154 000000      .WORD  0 ; AK2 - NO UPDATES FOR THIS SEARCH UNIT
184 014156 000000      .WORD  0 ; SEARCH UNIT NUMBER
184 014156 000000      .WORD  0 ; BATCH NUMBER
```

DBPROC - DATABASE PROCESSOR
FDB DEFINITIONS

MACRO M1110 27-MAR-80 13:30 P. 16
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

186
187
188
189
190 014160
191 014320
192 014320
193 014320
194
195
196 014320
197
198
199 014320
200

```
.SBTTL FDB DEFINITIONS...  
:  
: DBUPD,SPL FDB  
:  
: SPLFDB: FDBDF$  
:         FDRCA$  FD,RUM  
:         FDBKA$  INPUT,N,BUFB,,REF,RIOST  
:         FDDPA$  RD,LUN  
:  
:  
:         FRSZ$  0  
:  
:-----STATIC DIRECTIVES-----  
DMC:  QIOW$  IO,WLB,SD,LUN,SND,EF,,IOSTAT,,<OUTPUT,N,BUFB>  
      .EVEN
```

```
.SBTTL DBPROC-- MAIN CONTROL
;
; DBPROC:
; FINIT$
DBP001:
RCVX$S ., *PRB . ; RECEIVE COMMAND FROM MSCHED
BCC . 1$ ; DIRECTIVE SUCCESSFUL
MOVB . $DSJ,R1 ; ERROR EXISTS
CALL . ERROR
JMP . EXIT
;
; DETERMINE NUMBER OF SEARCH UNITS ON LINE (1 OR 2)
;
1$: MOV . SRECPT+2,R0 ; R0->SU1'S SREC
TST . SR,YR(R0) ; IS SU1 ON LINE?
BNE . DUPDT ; BRANCH IF YES
MOV . #1,NO,SU ; SINGLE SU CONFIGURATION
;
; DETERMINE UPDATE TYPE
;
DUPDT:
CLR . MASUPD ; CLEAR MASS UPDATE FLAG
TSTB . PRB+R0,CB ; TEST COMMAND BYTE FROM MSCHED
BEQ . NXTFIL ; CONTINUOUS UPDATE
INC . MASUPD ; MASS UPDATE
NXTFIL:
TRYAG:
MOV . #PCK1,PACKS ; MAKE SURE PACKING STARTS ON
; 3-WORD BOUNDARY
CLR . REDO,F ; CLEAR REDO FLAG
MOV . #FN,DBS,R1 ; UPDATE SPOOL FILE NUMBER
;
; BUILD FILE IDENTIFICATION BLOCK
;
MOV . #SPLFDB,R0 ; R0->FILE DEFINITION BLOCK
CALL . BLDNFL ; BUILD FILE ID BLOCK
;
; OPEN FILE
;
MOV . #-1,SPLFDB+F,FNB+N,FVER ; REQUEST LOWEST VERSION NO
CALL . OPEN ; OPEN THE FILE
BCC . CONT ; CONTINUE
TRY:1: TST . FLAG ; WERE THERE ANY FILES?
BEQ . 1$ ; BRANCH IF NO
CALL . SNDEND ; SEND END-OF-UPDATE EXCHANGE TO ALL SU'S
BR . DBP001
;
; THERE WAS NOTHING TO DO SEND AK2 TO MSCHED
;
1$: CALL . GETFRE ; GET PACKET
MOVB . #*DBPROC,2(R2) ; COMMAND SOURCE
MOVB . #0,3(R2) ; COMMAND BYTE
MOV . PRB+R0,BN,4(R2) ; BATCH NUMBER
CALL . PUTSSQ ; PUT PACKET TO SCHED
BR . DBP001
;
CONT:
INC . FLAG ; INDICATE THAT AT LEAST ONE FILE WAS FOUND
```

DBPROC - BASE PROCESSOR
DBPROC - MAIN CONTROL

MACRO M1110 27-MAR-80 17:30 P. 17-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
259 014564          CALL    COMPUT          ; DETERMINE THE NUMBER OF N.BUFW BLOCKS IN THE
260                                     ; FILE (IN R2)
261 014570 103747    BCS     TRY.1           ; BRANCH IF FILE NOT CLOSED PROPERLY
262 014572 010237 014074' MOV     R2,NBLK.I      ; # OF BLOCKS TO READ
263                                     ;
264                                     ; READ THE FIRST BLOCK AND DETERMINE THE EXCHANGE TYPE
265                                     ;
266 014576          CALL    READ
267 014602 005737 014116' TST     RDRSLT          ; GOOD READ?
268 014606 001413    BEQ     BGPROC          ; YES, BEGIN PROCESSING
269 014610 032737 000010 014116' BIT     #BIT3,RDRSLT   ; END-OF-FILE ERROR?
270 014616          BDN     1$              ; BRANCH IF YES (ERRSEQ HAS BEEN CALLED)
271 014620 012737 000014 014134' MOV     #AS.FAT,STATWD ; INDICATE FATAL ERROR FOR EXCHANGE
272 014626          CALL    MINCMD          ; TELL DBUPD TO NACK THIS EXCHANGE
273 014632 000137 015164' 1$:    JMP     DBP120        ; DELETE BAD FILE AND GO TO NEXT ONE
274                                     ;
275                                     ; BEGIN PROCESSING
276                                     ;
277 014636          BGPROC:
278 014636 005737 014064' TST     MASUPD          ; IS THIS A MASS UPDATE?
279 014642 001426    BEQ     55$             ; NO
280 014644 022737 046505 000000C CMP     #*EM.INPUT+DU.TYP ; END OF MASS UPDATE COMMAND?
281 014652 001022    BNE     55$             ; NO
282 014654 004737 022116' JSR     PC,SENDEND     ; YES, SEND END OF UPDATE COMMAND TO DBUPD
283 014660          DELET$ #SPLFDB          ; DELETE INPUT SPOOL FILE
284 014670          BCC     5$              ;
285 014672 105760 000052 TSTB   F.ERR(R0)       ; ERROR?
286 014676 003006    BGT     5$              ; BRANCH IF NO
287 014700 116001 000052 MOVB   F.ERR(R0),R1    ; R1 = ERROR CODE
288 014704          CALL    FCSERR          ;
289 014710 000137 017606' JMP     EXIT
290 014714          5$:
291 014714 000137 014354' JMP     DBP001          ;
292 014720 005037 014060' CLR     EXTYPE          ; CLEAR TYPE OF EXCHANGE INDICATOR
293 014724 004737 020766' JSR     PC,PREPEX      ; PREPROCESS THE EXCHANGE
294 014730 023727 014060' 000001 CMP     EXTYPE,#1      ; ADD OR MODIFY DOCUMENT EXCHANGE?
295 014736 003402    BLE     DBP030          ; YES
296 014740 000137 015134' JMP     DBP070          ; NO, PURGE, READ OR DELETE SUB-DOC EXCHANGE
297                                     ;
298 014744          DBP030:
299 014744          SAVE   #INPUT+DU.IDH      ; ADDR OF ID ON STACK
300 014750 004737 021742' JSR     PC,VALIDID    ;
301 014754 005737 014052' TST     CMPST          ; VALID ID?
302 014760 001410    BEQ     42$             ; YES
303 014762 012737 000011 014134' MOV     #AS.IVI,STATWD ; SET INVALID ID ERROR CODE
304 014770 004737 017622' JSR     PC,MINCMD      ; CREATE MINIMUM COMMAND RECORD
305 014774 103465    BCS     DBP100          ; TRY TO REDO UPDATE - PROBLEM SENDING IT
306 014776 000137 015164' JMP     DBP120          ;
307 015002          42$:
308 015002 022737 041504 000000C CMP     #*DC.INPUT+DU.TYP ; ADD NEW DOCUMENT?
309 015010 001006    BNE     DBP045          ; NO
310 015012 004737 015224' JSR     PC,ADDOC       ; YES, PROCESS THE EXCHANGE
311 015016 005737 014120' TST     REDO.F          ; DO UPDATE AGAIN?
312 015022 001052    BNE     DBP100          ; YES
313 015024 000457    BR     DBP120          ;
314 015026          DBP045:
315 015026 022737 050125 000000C CMP     #*UP.INPUT+DU.TYP ; MODIFY/REPLACE DOCUMENT?
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBPROC - DATA BASE PROCESSOR
DBPROC - MAIN CONTROL

MACRO: M1110 27 MAR 80 17:20 PAGE 17-2
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
316 015034 001012.          BNE  DBP055          ; NO
317                          ;
318                          ; INDICATE THAT REQUIRED WHITE SPACE IS ZERO.
319                          ;
320 015036 005037 000000C.    CLR  DU.WSC+INPUT.
321 015042 005037 000000C.    CLR  DU.WSC+2+INPUT
322 015046 004737 015224'    JSR  PC.ADDOC.      ; PROCESS THE EXCHANGE.
323 015052 005737 014120'    TST  REDO.F.       ; DO UPDATE AGAIN?
324 015056 001034          BNE  DBP100        ; YES
325 015060 000441          BR   DBP120        ; WRAP UP.
326 015062.
327 015062 022737 051501 000000C. DBP055: CMP  #AS.INPUT+DU.TYP. ; ADD SUB-DOCUMENT?
328 015070 001012.          BNE  1$           ; BRANCH IF NO
329 015072 005037 000000C.    CLR  DU.WSC+INPUT. ; INDICATE THAT REQUIRED WHITE SPACE
330 015076 005037 000000C.    CLR  DU.WSC+2+INPUT ; --- COUNT IS ZERO.
331 015102 004737 015224'    JSR  PC.ADDOC.
332 015106 005737 014120'    TST  REDO.F.       ; DO UPDATE AGAIN?
333 015112 001016          BNE  DBP100        ; YES
334 015114 000423          BR   DBP120
335                          ;
336                          ; INVALID REQUEST.
337                          ;
338 015116 012737 000012 014134' 1$: MOV  #AS.IVR,STATUSD ; MOVE INVALID REQUEST CODE IN STATUS WORD.
339 015124 004737 017622'    JSR  PC.MINCMD.   ; CREATE MINIMUM COMMAND RECORD.
340 015130 103407          BCS  DBP100        ; TRY TO REDO UPDATE - PROBLEM SENDING IT.
341 015132 000414          BR   DBP120
342                          ;
343                          ; PURGE OR READ DOCUMENTS OR DELETE SUB-DOCUMENTS.
344                          ;
345 015134          DBP070:
346 015134 004737 020360'    JSR  PC.PRDOC.   ; PROCESS PURGE OR READ EXCHANGE.
347 015140 005737 014120'    TST  REDO.F.       ; DO UPDATE AGAIN?
348 015144 001001          BNE  DBP100        ; YES
349 015146 000406          BR   DBP120        ; DELETE INPUT FILE.
350                          ;
351                          ; CLOSE UPDATE FILE AND START OVER AGAIN ON SAME UPDATE.
352                          ;
353 015150          DBP100:
354 015150          CLOSE$ #SPLFDB. ; CLOSE SPOOL FILE
355 015160 000137 014446'    JMP  TRYAG.      ;
356                          ;
357                          ; DELETE SPOOL FILE.
358                          ;
359 015164          DBP120: DELET$ #SPLFDB. ; DELETE THE INPUT FILE.
360 015174 103011          BCC  20$         ; DELETE SUCCESSFUL.
361 015176 105760 000052          TSTB F.ERR(R0)
362 015202 003006          BGT  20$
363 015204 116001 000052          MOVB F.ERR(R0),R1
364 015210          CALL FCERR.
365 015214 000137 017606'    JMP  EXIT
366 015220          20$:
367 015220 000137 014446'    JMP  NXTFIL.    ; CHECK FOR NEXT UPDATE.
```

DBPROC- DB BASE PROCESSOR
ADDOC- ADD NEW DOCUMENT

MACRO-M1110 27-MAR-88 13:30 P. 18
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
369 .SBTTL-ADDOC- ADD NEW DOCUMENT
370 ;
371 ;
372 ;
373 ;
374 ;
375 ;
376 ;
377 ;
378 015224 .ADDOC:
379 015224 005037 014126' CLR R9 ; CLEAR OUTPUT BUFFER SEND COUNTER.
380 015230 004737 016134' JSR PC,COBHDR ; CREATE OUTPUT BUFFER HEADER.
381 015234 012701 000000C MOV #<OUTPUT+PC.SPD>,R1 ;R1->START OF PACKED DATA.
382 015240 005037 014112' CLR PWCT ; INITIALIZE 2-WORD PACKED WORD COUNT.
383 015244 005037 014114' CLR PWCT+WORD1 ;
384 015250 022737 051501 000000C CMP #*AS,INPUT+DU,TYP ;ADD SUB-DOCUMENT?.
385 015256 001402 BEQ 1$ ;BRANCH IF YES.
386 015260 004737 016714' JSR PC,CRDSS ; CREATE DOCUMENT START SEQUENCE.
387 015264 162737 000017 014100' 1$: SUB #DU,SDT,NBREAD ; # OF TEXT CHAR. IN 1ST INPUT BUFFER.
388 015272 013737 014046' 014122' MOV CHARCT,R8 ; TEXT CHARACTER COUNT.
389 015300 013737 014050' 014124' MOV CHARCT+WORD1,R8+WORD1 ;
390 015306 012704 000000C MOV #INPUT+DU,SDT,R4 ; ADDR OF 1ST TEXT CHARACTER IN BUFFER.
391 ;
392 ;
393 ;
394 015312 112402 .ADDC010: MOVB (R4)+,R2 ; CONVERT 8-BIT ASCII--
395 015314 042702 177600 BIC #177600,R2 ; --CHARACTER TO 6 BIT--
396 015320 116202 000000' MOVB CET6B(R2),R2 ; --CHARACTER.
397 015324 100041 BPL 4$ ; NOT A SPECIAL MARKER.
398 015326 042702 177600 BIC #177600,R2 ; CLEAR SPECIAL MARKER INDICATOR.
399 015332 122702 000000 CMPB #SDMK,R2 ; SUB-DOCUMENT MARKER?.
400 015336 001007 BNE 2$ ; NO.
401 ;
402 ;
403 ;
404 015340 004737 017130' JSR PC,CRSDSS ; YES, CREATE SUB-DOCUMENT START SEQUENCE.
405 015344 103362 BCC ADC010 ;BRANCH IF OK.
406 015346 005737 014116' TST RDRSLT ;IRREGULAR EOF OR ALL CHAR'S PACKED?.
407 015352 001456 BEQ ADC020 ;BRANCH IF ALL CHAR'S ARE PACKED.
408 015354 000436 BR 10$ ;BRANCH IF HARD ERROR OR EOF ERROR.
409 ;
410 ;
411 ;
412 015356 004777 002512 2$: JSR PC,@PACK6 ; PACK THE SPECIAL MARKER.
413 015362 103447 BCS ADC017 ; RECOVERABLE ERROR DETECTED, REDO UPDATE.
414 ;
415 ;
416 ;
417 015364 122764 000024 177777 CMPB #SNTC,-1(R4) ; SENTENCE MARKER?
418 015372 001421 BEQ 5$ ;BRANCH IF YES.
419 015374 122764 000021 177777 CMPB #PRGH,-1(R4) ;PARAGRAPH MARKER?
420 015402 001415 BEQ 5$ ;BRANCH IF YES.
421 015404 CALL NXTCHR ; IS NEXT CHARACTER AVAILABLE?
422 015410 103004 BCC 3$ ;BRANCH IF YES.
423 ;
424 ;
425 ;
ALL TEXT PACKED, OR READ ERROR.
```


DBPROC - DATA BASE PROCESSOR
ADDCC - ADD NEW DOCUMENT

MACRO: M1110 27-MAR-88 13:30 PAGE 18-1

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
426 015412 005737 014116' TST RDRSLT
427 015416 001434 BEQ ADC020 ; BRANCH IF ALL TEXT PACKED
428 015420 000414 BR 10$ ; BRANCH IF READ ERROR
429
430 ;
431 ; PACK NEXT CHARACTER
432 015422 112402 3$: MOVB (R4)+,R2 ; GET NEXT 8-BIT ASCII CHARACTER
433 015424 042702 177700 BIC #177700,R2 ; PACK THIS CHARACTER UNTRANSLATED
434 015430 004777 002440 4$: JSR PC,@PACK6 ; PACK THE CHARACTER
435 015434 103422 BCS ADC017 ; RECOVERABLE ERROR DETECTED, REDO UPDATE
436
437 ;
438 ; GET NEXT CHARACTER
439 015436 5$: CALL NXTCHR ; IS THERE ANOTHER CHAR AVAILABLE?
440 015442 103323 BCC ADC010 ; BRANCH IF YES
441 015444 005737 014116' TST RDRSLT ; ALL TEXT PACKED OR READ ERROR
442 015450 001417 BEQ ADC020 ; BRANCH IF ALL TEXT PACKED
443
444 ;
445 ; READ ERROR
446 015452 032737 000010 014116' 10$: BIT #BIT3,RDRSLT ; END OF FILE ERROR?
447 015460 B0H 15$ ; BRANCH IF YES
448
449 ;
450 ; HARD ERROR
451 015462 012737 000014 014134' MOV #AS,FAT,STATWD ; FATAL ERROR
452 015470 CALL ERRSE0 ; NOTIFY DBUPD TO NACK THIS EXCHANGE
453
454 ;
455 ; EOF ERROR
456 015474 005037 014120' 15$: CLR REDO,F ; SKIP THIS UPDATE
457 015500 000207 RTS PC
458
459 ;
460 ; SET FLAG TO REDO THIS UPDATE - RECOVERABLE ERROR DETECTED
461 015502 005237 014120' ADC017: INC REDO,F ; SET REDO FLAG
462 015506 000207 RTS PC ; RETURN
463
464 ;
465 ; ALL TEXT CHARACTERS PACKED
466 015510 005737 014142' ADC020: TST WSPCNT ; ANY REQUIRED WHITE SPACE?
467 015514 001412 BEQ ADC030 ; NO
468 015516 013705 014142' MOV WSPCNT,R5 ; REQUIRED WHITE SPACE COUNT
469
470 ;
471 ; PACK REQUIRED AMOUNT OF WHITE SPACE
472 015522 112702 000040 ADC025: MOVB #40,R2 ; WHITE SPACE FILL CHARACTER
473 015526 116202 000000' MOVB CET6B(R2),R2
474 015532 004777 002336 JSR PC,@PACK6 ; PACK WHITE SPACE
475 015536 103761 BCS ADC017 ; RECOVERABLE ERROR DETECTED, REDO UPDATE
476 015540 077510 SOB R5,ADC025 ; CONTINUE FOR ALL WHITE SPACE
477
478 ;
479 ; MAKE SURE PACKED TEXT IS ALIGNED ON 3-WORD BOUNDARY
480 015542 022737 020076' 020074' ADC030: CMP #PK1,PACK6 ; MUST HAVE 3 WORD INTEGRITY
481 015550 001412 BEQ ADC035 ; GOT IT
482 015552 112702 000040 MOVB #40,R2 ; CONTINUE PACKING (1-7 BYTES)
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBPROC- DB BASE PROCESSOR
ADDOC- ADD NEW DOCUMENT

MACRO M1118 27-MAR-90 13:30 P 18-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
483 015556 116202 000000'      MOVB  CET6B(R2),R2
484 015562 005237 014142'      INC   WSPCNT          ;
485 015566 004777 002302      JSR   PC,@PACK6     ; PACK ADD'L WHITE SPACE
486 015572 103743          BCS   ADC017        ; RECOVERABLE ERROR DETECTED, REDO UPDATE
487 015574 000762          BR    ADC030        ;
488
489      ; SEND LAST BUFFER TO DBUPD
490      ;
491 015576 012701 004010' ADC035: MOV  #OUTPUT,R1 ; R1 -> OUTPUT BUFFER
492 015602 004737 021320'      JSR   PC,SEND      ; SEND LAST BUFFER
493 015606 103003          BCC   35$          ; NO ERROR ON SEND
494 015610 005237 014120'      INC  REDO.F       ; REDO UPDATE FLAG
495 015614 000207          RTS   PC           ; YES
496
497      ; FOLLOWING IS TEMPORARY DEBUG TO BE REMOVED ON DELIVERY
498      ;
499 015616 023737 014126' 014076' 35$: CMP  R9,NBLK.0     ; BLOCKS SENT - BLOCKS CALCULATED TO SEND?
500 015624 001416          BEQ  40$          ; YES
501 015626 013737 014126' 014102' MOV  R9,PAR1       ; NO, PRINT ERROR MSG WITH VALUES
502 015634 013737 014076' 014104' MOV  NBLK.0,PAR1+2 ;
503 015642          MOVS  #MSG1,#PAR1 ;
504 015662 000207          RTS   PC           ; RETURN
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
506 .SBTTL - NXTCHR  
507 ;  
508 ; SUBROUTINE TO EXAMINE WHETHER  
509 ; 1) ALL CHARACTERS OF TEXT HAVE BEEN PACKED. IF YES,  
510 ; CC IS SET ON RETURN AND RDRSLT WILL BE = 0. OR,  
511 ; 2) ALL CHARACTERS IN INPUT BUFFER HAVE BEEN PACKED,  
512 ; IF YES, NEXT BUFFER IS READ AND R4 AND NBREAD ARE RESET.  
513 ;  
514 ;  
515 ; ON ENTRY,  
516 ; R4 -> NEXT TEXT CHARACTER IN INPUT BUFFER.  
517 ; NBREAD = NUMBER OF CHARACTERS LEFT IN INPUT BUFFER + 1  
518 ; R8 = NUMBER OF UNPACKED TEXT CHARACTERS + 1  
519 ;  
520 ; ON RETURN,  
521 ; IF CC IS CLEAR,  
522 ; R4 -> NEXT CHARACTER IN INPUT BUFFER.  
523 ; NBREAD = NUMBER OF UNPACKED CHARACTERS IN INPUT BUFFER.  
524 ; R8 = NUMBER OF UNPACKED TEXT CHARACTERS.  
525 ;  
526 ; IF CC IS SET,  
527 ; AND RDRSLT = 0, ALL TEXT IS PACKED.  
528 ;  
529 ; IF RDRSLT = BIT3, HARD ERROR, CLOSE FILE AND EXIT.  
530 ; IF RDRSLT = BIT3, IRREGULAR EOF, "READ" SUBROUTINE ALREADY TOOK  
531 ; CARE OF THINGS. CURRENT FILE SHOULD BE DELETED AND TASK  
532 ; SHOULD DO NEXT DBUPD.SPL  
533 ;  
533 015664 162737 000001 014124* NXTCHR: SUB #1,R8+WORD1 ;DEC. LOW ORDER TEXT CHAR COUNT.  
534 015672 005637 014122* SBC R8 ;ADJUST FOR CARRY.  
535 015676 003003 BGT 1$ ;BRANCH IF ALL TEXT IS NOT PACKED.  
536 015700 005737 014124* TST R8+WORD1 ;LOW ORDER ZERO?  
537 015704 001414 BEQ 2$ ;BRANCH IF ALL TEXT IS PACKED.  
538 ;  
539 ; ALL TEXT IS NOT PACKED.  
540 ;  
541 015706 005337 014100* 1$: DEC NBREAD ;NBREAD=NO. OF UNPACKED CHAR'S IN  
542 ; INPUT BUFFER.  
543 015712 003007 BGT 3$ ;BRANCH IF SOME LEFT  
544 ;  
545 ; INPUT BUFFER ALL PACKED. READ ANOTHER BUFFER.  
546 ;  
547 015714 CALL READ ;READ NEXT TEXT BLOCK.  
548 015720 005737 014116* TST RDRSLT ;OK?  
549 015724 001015 BNE 4$ ;BRANCH IF NOT OK.  
550 015726 012704 000010* MOV #INPUT,R4 ;R4->NEXT CHAR. TO BE PACKED.  
551 ;  
552 ; RETURN  
553 ;  
554 015732 000241 3$: CLC  
555 015734 000207 RTS PC  
556 ;  
557 ; ALL TEXT PACKED. LOOK FOR AA(HEX) AS THE TERMINATING CHARACTER.  
558 ;  
559 015736 122714 000252 2$: CMPB #252,(R4) ;HEX AA?  
560 015742 001404 BEQ 5$ ;BRANCH IF YES.  
561 ;  
562 ; HARD ERROR!! TEXT INCORRECTLY TERMINATED
```

DBPROC - DB BASE PROCESSOR
NXTCHR.

MACRO: M1110 27-MAR-88 17:30 P. 10-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
563 ;  
564 015744 012737 000001 014116'      MOV.  #BIT0,RDRSLT. ; INDICATE ERROR.  
565 015752 000402.                    BR    4$  
566 ;  
567 ; TEXT OK.  
568 ;  
569 015754 005037 014116'      5$: CLR.  RDRSLT.  
570 ;  
571 ; ERROR ON READ.  
572 ;  
573 015760 000261                    4$: SEC.  
574 015762 000207.                    RTS.  PC. ; RETURN.  
575 ;  
576 ;  
577 ;  
578 ; SBTTL CLRBUF- CLEAR BUFFER.  
579 ;  
580 ;  
581 ; STACK CONTAINS:      START ADDRESS OF BUFFER TO BE CLEARED.  
582 ;                      LENGTH OF BUFFER IN WORDS.  
583 ;  
584 ;  
585 ;  
586 015764 ;  
587 015764 CLRBUF:  
588 015770 016602 000006      SAVE.  R1,R2. ; SAVE REGISTERS.  
589 015774 016601 000010      MOV.  WORD3(SP),R2. ; BUFFER LENGTH IN WORDS.  
590 016000      MOV.  WORD4(SP),R1 ; BUFFER START ADDRESS.  
591 016000      10$:  
592 016002 077202.          CLR.  (R1)+ ; CLEAR BUFFER.  
593 016004          SOB.  R2,10$ ; TILL DONE.  
594 016010 011666 000004      RESTOR. R1,R2. ; RESTORE REGISTERS.  
595 016014 062706 000004      MOV.  (SP),4(SP) ; SET UP RETURN ADDRESS.  
596 016020 000207          ADD.  #4,SP. ; RESTORE STACK POINTER.  
                    RTS.  PC. ;
```

```

598 .SBTTL - CMP3WD - COMPARE THREE WORD ITEMS
599 ;
600 ; INPUTS:
601 ; R1 - CONTAINS THE ADDRESS OF THE VALUE TO BE COMPARED
602 ; R2 - CONTAINS THE ADDRESS OF THE RANGE START
603 ; R3 - CONTAINS THE ADDRESS OF THE RANGE END
604 ;
605 ; CMP3WD COMPARES THE VALUE IN R1 TO BOTH THE RANGE START AND THE
606 ; RANGE END. IF THE VALUE IN R1 SATISFIES THE GIVEN RANGE, THE COMPARE
607 ; STATUS WORD IS CLEARED. IF THE VALUE FALLS BELOW THE RANGE, THE STATUS
608 ; WORD IS SET TO A NEGATIVE ONE AND THE VALUE IS CONSIDERED INVALID.
609 ; IF THE VALUE FALLS ABOVE THE RANGE, THE STATUS WORD IS SET TO A POSITIVE
610 ; ONE AND THE VALUE MAY BE VALID OR INVALID. IF ANOTHER RANGE EXISTS, THE
611 ; VALUE IS CONSIDERED VALID UNTIL THE NEXT SET OF RANGES IS CHECKED. IF NO
612 ; NO OTHER RANGE EXISTS TO BE CHECKED, THE VALUE IS INVALID.
613 ;
614 ; OUTPUTS:
615 ; CMPST - -1 IF VALUE FALLS BELOW RANGE
616 ; 0 IF VALUE FALLS WITHIN RANGE
617 ; 1 IF VALUE FALLS ABOVE RANGE
618 ;
619 ; COMPARE TO THE RANGE START
620 ;
621 CMP3WD: CMP (R1), (R2) ; COMPARE HIGH ORDER WORDS
622 BLO CMP020 ; INVALID FOR THIS RANGE
623 BHI CMP010 ; CHECK RANGE END VALUES
624 CMP WORD1(R1), WORD1(R2) ; COMPARE MIDDLE WORDS
625 BLO CMP020 ; INVALID FOR THIS RANGE
626 BHI CMP010 ; CHECK RANGE END VALUES
627 CMP WORD2(R1), WORD2(R2) ; COMPARE LOW ORDER WORDS
628 BLO CMP020 ; INVALID FOR THIS RANGE
629 BEQ CMP030 ; VALUE VALID FOR THIS RANGE
630 ;
631 ; COMPARE TO THE RANGE END
632 ;
633 CMP010: CMP (R1), (R3) ; COMPARE HIGH ORDER WORDS
634 BLO CMP030 ; VALID FOR THIS RANGE
635 BHI CMP040 ; GREATER THAN THIS RANGE
636 CMP WORD1(R1), WORD1(R3) ; COMPARE MIDDLE WORDS
637 BLO CMP030 ; VALID FOR THIS RANGE
638 BHI CMP040 ; GREATER THAN THIS RANGE
639 CMP WORD2(R1), WORD2(R3) ; COMPARE LOW ORDER WORDS
640 BLO CMP040 ; GREATER THAN THIS RANGE
641 BR CMP030 ;
642 ;
643 CMP020: MOV #-1, CMPST ; INVALID VALUE FOR THIS RANGE
644 BR CMP050 ;
645 ;
646 CMP030: CLR CMPST ; VALID VALUE FOR THIS RANGE
647 BR CMP050 ;
648 ;
649 CMP040: MOV #1, CMPST ; VALUE GREATER THAN THIS RANGE
650 BR CMP050 ;
651 ;
652 RTS PC
    
```

```

654          .SBTTL--COBHDR-- CREATE-OUTPUT-BUFFER-HEADER.
655          ;
656          ; THIS ROUTINE CREATES THE HEADER FOR:
657          ;   - ADD DOCUMENT.
658          ;   - ADD SUB-DOCUMENT AND
659          ;   - REPLACE DOCUMENT.
660          ; PROCESSED OUTPUT FILES. UPON EXITING THIS ROUTINE, THE FOLLOWING
661          ; REGISTERS WILL CONTAIN:
662          ;
663          ; R1 - ADDRESS OF THE OUTPUT BUFFER
664          ; R4 - ADDRESS OF THE INPUT BUFFER.
665          ;
666          COBHDR:
667          MOV     #OUTPUT,R5          ; BASE ADDR OF OUTPUT BUFFER
668          SAVE   RS,#N,BUFJ         ; PLACE ADDR AND LENGTH OF BUFFER ON STACK
669          JSR   PC,CLRBUF           ; CLEAR BUFFER
670          MOV   #INPUT,R4          ; BASE ADDR ON INPUT BUFFER
671          MOV   DU,TCC(R4),CHARCT   ; TEXT CHARACTER COUNT
672          MOV   DU,TCC+WORD1(R4),CHARCT+WORD1 ;
673          MOV   DU,WSC(R4),WSPCNT   ; REQUIRED WHITE SPACE COUNT
674          MOV   #DB,PC,DBT(R5)     ; PLACE EXCHANGE TYPE IN PROCESSED OUTPUT BUFFER
675          CLR   PC,PAD(R5)         ;
676          MOV   DU,TYP(R4),PC,TYP(R5) ; UPDATE REQUEST TYPE TO OUTPUT BUFFER
677          MOV   DU,IDH(R4),PC,IDH(R5) ; MOVE BINARY DOCUMENT ID--
678          MOV   DU,IDM(R4),PC,IDM(R5) ; --TO PROCESSED UPDATE REQUEST
679          MOV   DU,IDL(R4),PC,IDL(R5) ; --OUTPUT BUFFER
680          MOV   #AS,SUP,PC,STW(R5)  ; SET SUCCESSFUL BIT IN STATUS WORD
681          MOV   DU,TCC(R4),PC,TCC(R5) ; PLACE UNPACKED TEXT CHARACTER COUNT--
682          MOV   DU,TCC+WORD1(R4),PC,TCC+WORD1(R5) ; --IN OUTPUT BUFFER
683          MOV   DU,WSC(R4),PC,WSC(R5) ; UNPACKED WHITE SPACE COUNT
684          ;
685          ; CALCULATE NUMBER OF PACKED 6-BIT WORDS FOR TEXT CHARACTERS ONLY.
686          ;
687          MOV   CHARCT,R2          ; R2 = # OF TEXT CHARACTERS (HI)
688          MOV   CHARCT+WORD1,R3    ; R3 = # OF TEXT CHARACTERS (LO)
689          ADD  #13,,R3            ; ADD # OF BYTES IN DOC, START SEQUENCE
690          ADC  R2,,R2             ;
691          ;
692          ; CALCULATE PACKED WORD COUNT FOR TEXT CHARACTERS ONLY.
693          ; MAKING SURE THAT THE TEXT IS ALIGNED ON A 3-WORD BOUNDARY.
694          ;
695          CGT   SAVE   R2,R3        ; SAVE TOTAL
696          MOV   #3,,R0            ; MULTIPLIER = R0
697          JSR   PC,$DMUL          ; ON RETURN, R0 = HI, R1 = LO
698          MOV   R1,R2            ; R2 = LO
699          MOV   R0,R1            ; R1 = HI
700          MOV   #8,,R0           ; DIVISOR > R0
701          JSR   PC,$DDIV         ; ON RETURN, R2=LO, R1=HI, R0=REM
702          TST  R0               ; ANY REMAINDER?
703          BEQ  C1               ; NO
704          RESTOR R2,R3
705          ADD  #1,R3             ; ADD 1 BYTE OF TEXT (BLANK) UNTIL
706          ADC  R2,,R2            ; 3-WORD INTEGRITY IS OBTAINED
707          ADD  #1,WSPCNT         ; ADJUST WHITE SPACE COUNT TO
708          ; REFLECT ALIGNMENT
709          BR   C6
710          C1:
    
```

```

711 016366 010165 000032      MOV.   R1,PC,PTC(R5) ; # OF PACKED WORDS FOR TEXT CHARS. ONLY.
712 016372 010265 000034      MOV.   R2,PC,PTC+WORD1(R5) ;
713 016376 016565 000022 000024      MOV.   PC,USC(R5),PC,UWS(R5) ; INITIALIZE UNUSED WHITE SPACE TO REQUIRED W.S.
714 016404 001421      BEQ.   C3 ; BRANCH IF NO REQUIRED WHITE SPACE.
715 016406      RESTOR. R2,R3 ; RESTORE PREVIOUS TOTAL.
716 ;
717 ;
718 ; R2 = NUMBER OF TEXT CHARACTERS (HI) (UNPACKED)
719 ; R3 = NUMBER OF TEXT CHARACTERS (LO)
720 016412 066503 000022      ADD.   PC,USC(R5),R3 ; ADD AM'T OF REQUIRED WHITE SPACE TO TOTAL.
721 016416 005502      ADC.   R2, ;
722 ;
723 ; CALCULATE TOTAL NUMBER OF PACKED 16-BIT WORDS - TEXT & WHITE SPACE.
724 ;
725 016420      C2:
726 016420      SAVE.  R2,R3 ; SAVE NEW TOTAL.
727 016424 012700 000003      MOV.   #3,R0 ; MULTIPLIER.
728 016430 004737 000000G.      JSR.   PC,$DMUL. ; ON RETURN, R0 = HI, R1 = LO
729 016434 010102      MOV.   R1,R2 ; R2 = LO.
730 016436 010001      MOV.   R0,R1 ; R1 = HI.
731 016440 012700 000010      MOV.   #8,R0 ; DIVISOR.
732 016444 004737 000000G.      JSR.   PC,$DDIV. ; ON RETURN, R2=LO, R1=HI, R0=REM.
733 ;
734 ; INSURE THREE WORD INTEGRITY BOUNDARY.
735 ;
736 016450      C3:
737 016450 005700      TST.   R0 ; REMAINDER?.
738 016452 001410      BEQ.   C4 ; NO.
739 016454      RESTOR. R2,R3 ; RESTORE PREVIOUS TOTALS.
740 016460 062703 000001      ADD.   #1,R3 ; ADD 1 BYTE OF WHITE SPACE TILL INTEGRITY OBTAINED.
741 016464 005502      ADC.   R2, ;
742 016466 005265 000024      INC.  PC,UWS(R5) ; INCREMENT UNUSED WHITE SPACE COUNT.
743 016472 000752      BR.   C2 ;
744 016474      C4:
745 016474 010165 000026      MOV.   R1,PC,PWC(R5) ; TOTAL WORD COUNT OF PACKED TEXT.
746 ;
747 016500 010265 000030      MOV.   R2,PC,PWC+WORD1(R5) ; AND WHITE SPACE (ALIGNED ON 3-WORD BOUND.)
748 016504 062706 000004      ADD.   #4,SP ; RESTORE STACK POINTER.
749 ;
750 ; CALCULATE NUMBER OF BLOCKS TO TRANSMIT
751 ; R2 = TOTAL PACKED WORD COUNT (LO)
752 ; R1 = " " " " (HI)
753 ;
754 016510 062702 000017      ADD.   #PC,SPD/2,R2 ; ADD TO NUMBER OF PACKED WORDS.
755 016514 005501      ADC.   R1, ;
756 016516 012700 002000      MOV.   #N,BUFW,R0 ; # OF WORDS IN A TRANSMISSION BLOCK.
757 016522 004737 000000G.      JSR.   PC,$DDIV. ; DOUBLE PRECISION DIVIDE.
758 016526 005700      TST.   R0 ; REMAINDER?.
759 016530 001402      BEQ.   C5 ; NO.
760 016532 062702 000001      ADD.   #1,R2 ; YES, ADD ONE MORE BLOCK COUNT.
761 016536      C5:
762 016536 010265 000002      MOV.   R2,PC,PAD(R5) ; # OF BLOCK TO TRANSMIT > OUTPUT HEADER.
763 016542 010237 014076*      MOV.   R2,NBLK,0 ; STORE # OF BLOCKS TO OUTPUT.
764 016546 000207      RTS.  PC ;
    
```

DBPROC--BASE PROCESSOR
--- COMPUT

MACRO M1110 27-MAR-80 13:30 P. 22
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
766 .SBTTL .---- COMPUT.----
767 ;*****
768 ;
769 ; G-O-M-P-U-T S-U-B-R-O-U-T-I-N-E.
770 ;
771 ; THIS ROUTINE CONVERTS THE NUMBER OF FCS FILE BLOCKS TO HSTS RECORD BLOCKS.
772 ;
773 ; INPUTS:
774 ; R0->FDB
775 ; OUTPUTS:
776 ; R2=NO. OF HSTS FILE BLOCKS.
777 ;
778 ;*****
779 ;
780 COMPUT:
781 SAVE R3
782 MOV F,EFBK(R0),R2 ;GET HIGH ORDER FCS BLOCK COUNT.
783 MOV F,EFBK+2(R0),R3 ;GET LOW ORDER FCS BLOCK COUNT.
784 SUB #1,R3 ;MAKE IT ZERO RELATIVE.
785 SBC R2.
786 DIV #N,BFAC,R2. ;CONVERT TO HSTS BLOCK COUNT
787 BEQ 3$ ;THERE CANNOT BE ZERO BLOCKS
788 TST R3 ;THERE MUST BE EVEN MULTIPLES.
789 BEQ 2$ ;THERE IS EVEN MULTIPLES.
790 1$:
791 MOV #ERR101,PAR1+2 ;THERE IS SOMETHING WRONG WITH THE FCS BLOCK NO.
792 CALL ERROR.
793 JMP EXIT
794 2$:
795 RESTOR R3
796 CLC
797 RTS PC
798 3$:
799 RESTOR R3
800 MOV #ERR104,PAR1+2 ;ZERO LENGTH FILE OPENED.
801 CALL ERROR.
802 ;
803 MRKT$S #WAT.EF,#22,.,#2 ;WAIT FOR 22 SEC.
804 UTSE$S #WAT.EF.
805 ;
806 CLOSE$ #SPLFDB ;IF ZERO LENGTH FILE, CLOSE IT.
807 SEC ; AND GET OUT.
808 RTS PC
```



```

810 .SBTTL CRDSS+ CREATE DOCUMENT START SEQUENCE
811 ; *****
812 ;
813 ; GIVEN A 36-BIT BINARY DOCUMENT ID AND THE DOCUMENT TYPE, CRDSS WILL CREATE
814 ; THE 13 BYTE DOCUMENT START SEQUENCE.
815 ;
816 ; INPUT:
817 ; 36-BIT BINARY DOCUMENT ID
818 ; R1 CONTAINS POINTER TO OUTPUT BUFFER.
819 ;
820 ; REGISTERS DESTROYED:
821 ;
822 ; *****
823 ;
824 CRDSS:
825 SAVE R4,R5 ;
826 CLR R2 ; DOCUMENT START MARKER
827 JSR PC,@PACK6 ; PACK THE START MARKER
828 BCS CSS040 ; ERROR
829 CLR R2 ; FLAG BYTE INDICATING DOCUMENT START
830 JSR PC,@PACK6 ; PACK THE FLAG
831 BCS CSS040 ; ERROR RETURN
832 ; PACK THE DOCUMENT ID IN 9 CONSECUTIVE BYTES
833 MOV INPUT+DU,IDL,R2 ; MOST SIGNIFICANT 4 BITS
834 BIC #177760,R2 ; REMOVE EXTRANEOUS MATTER
835 JSR PC,@PACK6 ; PACK THE BYTE
836 BCS CSS040 ; ERROR RETURN
837 CLR R8 ; 2ND TIME ROUND FLAG
838 MOV INPUT+DU,IDL,R4 ; GET NEXT MOST SIGNIFICANT 16 BITS
839 CSS010:
840 ASHC #12,R4 ; GET NEXT MOST SIGNIFICANT 4 BITS
841 MOV R4,R2 ; PASS TO PACKING ROUTINE
842 BIC #177760,R2 ; INSURANCE
843 JSR PC,@PACK6 ; PACK BYTE
844 BCS CSS040 ; ERROR RETURN
845 MOV #3,R8 ; # OF 4-BIT COMBINATIONS REMAINING
846 CSS020:
847 ASHC #4,R4 ; GET NEXT MOST SIGNIFICANT 4 BITS
848 MOV R4,R2 ; PASS TO PACKING ROUTINE
849 BIC #177760,R2 ; INSURANCE
850 JSR PC,@PACK6 ; PACK BYTE
851 BCS CSS040 ; ERROR RETURN
852 SOB R8,CSS020 ; TWICE MORE
853 TST R8 ; LOW ORDER DONE?
854 BGT CSS030 ; YES
855 MOV INPUT+DU,IDL,R4 ; GET LEAST SIGNIFICANT 16 BITS
856 INC R8 ; SET FLAG
857 JMP CSS010 ; DO SAME ONCE MORE FOR LOWER 16 BITS
858 CSS030:
859 MOVB INPUT+DU,DT,R2 ; DOCUMENT TYPE
860 BIC #177760,R2 ;
861 JSR PC,@PACK6 ; PACK DOCUMENT TYPE
862 BCS CSS040 ; ERROR RETURN
863 MOVB #?T,R2 ; ASCII FILL CHARACTER FOR START SEQUENCE
864 MOVB CET6B(R2),R2 ; TRANSLATE TO 6 BIT BYTE
865 JSR PC,@PACK6 ; PACK FILL CHARACTER
866 BCS CSS040 ; ERROR RETURN
    
```

DBPROC - BASE-PROCESSOR MACRO-M1110 27-MAR-90 17:30 P 23-1
CRDSS - CREATE-DOCUMENT-START-SEQUENCE Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

867 017112	RESTOR R4,R5	:
868 017116 000207	RTS PC	:
869 017120	CSS040:	
870 017120	RESTOR R4,R5	:
871 017124 000261	SEC	: SET-ERROR-FLAG
872 017126 000207	RTS PC	:

DBPROC - DATA BASE PROCESSOR MACRO-M1110 27-MAR-80 17:39 PAGE 34
CRSDSS - CREATE SUB-DOCUMENT START SEQUENCE

```

874          .SBTTL CRSDSS- CREATE SUB-DOCUMENT START SEQUENCE.
875          ;*****
876          ;
877          ; GIVEN A 36-BIT BINARY SUBDOCUMENT ID AND THE DOCUMENT TYPE, CRSDSS WILL CREATE
878          ; THE 13 BYTE SUBDOCUMENT START SEQUENCE.
879          ;
880          ; ON ENTRY,
881          ;   R4->5-BYTE SUBDOCUMENT ID SEQUENCE.
882          ;   R1 CONTAINS POINTER TO OUTPUT BUFFER.
883          ;
884          ; ON RETURN,
885          ;   R4-> NEXT BYTE IN INPUT BUFFER.
886          ;   R1-> NEXT BYTE IN OUTPUT BUFFER.
887          ;
888          ; IF CC IS CLEAR, ALL IS OK.
889          ;
890          ; IF CC IS SET, AND
891          ;   RDRSLT = 0, ALL TEXT CHARACTERS ARE PACKED.
892          ;   RDRSLT = BIT3, IRREGULAR EOF.
893          ;
894          ;*****
895          ;
896 017130      CRSDSS:
897 017130      SAVE   R0,R5
898          ;
899          ;   R4-> MOST SIGNIFICANT ID BYTE.
900          ;
901 017134      CALL   NXTCHR      ;CAN THIS BYTE BE MOVED?
902 017140      BCC   11$        ;BRANCH IF YES
903 017142      BIT   #BIT3,RDRSLT ;EOF ERROR?
904 017150      BON   20$        ;BRANCH IF YES
905          ;
906          ;   HARD ERROR.
907          ;
908 017152      MOV   #AS.FAT,STATWD ;FATAL ERROR
909 017160      CALL  ERRESD      ;NOTIFY DBUPD TO NACK THIS EXCHANGE
910 017164      CLR   REDD.F     ;SKIP THIS EXCHANGE
911 017170      JMP   1$        ;RETURN
912          ;
913          ;   MOVE THE 5 BYTE DOC ID INTO TEMPORARY BUFFER.
914          ;
915 017174      MOV   #SDOCB,R2   ;R2-> TEMPORARY BUFFER
916 017200      MOVB  (R4)+,(R2)  ;MOVE MOST SIGNIFICANT BYTE
917 017202      ADD   #2,R2
918 017206      SAVE  R3
919 017210      MOV   #2,R3
920 017214      MOV   #2,R5
921 017220      10$: CALL  NXTCHR      ;CAN NEXT CHARACTER BE MOVED?
922 017224      BCC   12$        ;BRANCH IF YES
923 017226      BIT   #BIT3,RDRSLT ;EOF ERROR?
924 017234      BON   20$        ;BRANCH IF YES
925 017236      RESTOR R3
926 017240      SAVE  R4
927 017242      BR   22$        ;BRANCH IF HARD ERROR
928 017244      12$: MOVB  (R4)+,(R2)+
929 017246      SOB  R3,10$     ;DO NEXT TWO BYTES
930 017250      SWAB -2(R2)
930 017250      177776

```

```

931 017254 012703 000002      MOV.    #2,R3
932 017260 077521              SOB.    R5,10$      ;DO LAST TWO BYTES
933                               ;
934                               ; STEP AROUND THE 7 PAD CHARACTERS IN SUB-DOC START SEQUENCE.
935                               ;
936 017262 012703 000007      MOV.    #7,R3
937 017266                    14$: CALL.  NXTCHR.
938 017272 112402            MOV.    (R4)+,R2.
939 017274 077304            SOB.    R3,14$
940 017276                    CALL.  NXTCHR.      ;ACCOUNT FOR SUB-DOC START BYTE.
941 017302                    RESTOR. R3
942                               ;
943                               ; NOW CREATE THE SUBDOCUMENT START SEQUENCE.
944                               ;
945 017304                    13$: SAVE.  R4      ;R4->NEXT BYTE IN INPUT BUFFER.
946 017306 005002            CLR.   R2      ; DOCUMENT START MARKER.
947 017310 004777 000560    JSR.   PC,@PACK6 ; PACK THE START MARKER.
948 017314 103467            BCS.   1$      ; ERROR.
949 017316 012702 000001    MOV.   #BIT0,R2. ;FLAG BYTE INDICATING SUBDOC START.
950 017322 004777 000546    JSR.   PC,@PACK6 ; PACK THE FLAG.
951 017326 103462            BCS.   1$      ; ERROR RETURN.
952                               ; PACK THE DOCUMENT ID IN 9 CONSECUTIVE BYTES.
953 017330 013702 017506*    MOV.   SDOCB,R2. ; MOST SIGNIFICANT 4 BITS.
954 017334 042702 177760    BIC.   #17760,R2. ; REMOVE EXTRANEIOUS MATTER.
955 017340 004777 000530    JSR.   PC,@PACK6 ; PACK THE BYTE.
956 017344 103453            BCS.   1$      ; ERROR RETURN.
957 017346 012705 000002    MOV.   #2,R5      ;R5 = LOOP COUNTER
958 017352 012704 017510*    MOV.   #SDOCB+2,R4 ;R4-> NEXT MOST SIGNIFICANT 16 BITS.
959                               ;
960                               ; PACK 16 BIT DOC ID INTO FOUR 6-BIT BYTES.
961                               ;
962 017356 012700 177764      3$: MOV.  #-12,,R0 ;R0 = RIGHT SHIFT COUNT.
963 017362 011402 177760      2$: MOV.  (R4),R2.  ;R2 = 16-BITS OF DOC ID.
964 017364 072200            ASH.   R0,R2.    ;SHIFT RIGHT BY AMOUNT IN R0
965 017366 042702 177760    BIC.   #17760,R2. ;LEAVE 4 BITS.
966 017372 004777 000476    JSR.   PC,@PACK6 ;PACK BYTE.
967 017376 103436            BCS.   1$      ;BRANCH IF PROBLEM
968 017400 062700 000004    ADD.   #4,R0
969 017404 003766            BLE.   2$      ;LOOP IF NOT GREATER THAN ZERO.
970                               ;
971                               ; DO NEXT 16-BIT WORD.
972                               ;
973 017406 062704 000002      ADD.   #2,R4      ;R4-> NEXT 16-BIT DOC ID (LESS SIG.)
974 017412 077517            SOB.    R5,3$
975                               ;
976                               ; NOW PACK THE DOC. TYPE AND THE PAD BYTE, UNLESS IT IS A "DELETE.
977                               ; SUB-DOCUMENT" REQUEST.
978                               ;
979 017414 023727 000000C-051504 CMP.   INPUT+PR.TYP,#"DS. ;DELETE SUB-DOCUMENTS?
980 017422 001416            BEQ.   21$      ;BRANCH IF YES.
981 017424 113702 000000C- MOV.    INPUT+DU.DT,R2 ; DOCUMENT TYPE.
982 017430 042702 177700    BIC.   #17700,R2.  ;
983 017434 004777 000434    JSR.   PC,@PACK6 ; PACK DOCUMENT TYPE.
984 017440 103415            BCS.   1$      ; ERROR RETURN.
985 017442 112702 000124    MOV.   #'T,R2.    ; ASCII FILL CHARACTER FOR START SEQUENCE.
986 017446 116202 000000*    MOV.   CET6B(R2),R2. ; TRANSLATE TO 6 BIT BYTE.
987 017452 004777 000416    JSR.   PC,@PACK6 ; PACK FILL CHARACTER.
    
```

```
988 017456 103406
989 017460
990 017466 005037 014116'
991 017472 000207
992 017474
993 017474
994 017502 000261
995 017504 000207
996
997
998
999 017506 000000
1000 017510 000000
1001 017512 000000

21$: BCS... 1$ ; ERROR-RETURN.
      RESTOR R0,R5,R4 ;
      CLR RDRSLT.
      RTS PC. ;
1$: RESTOR R0,R5,R4 ;
     SEC ; SET-ERROR-FLAG.
     RTS PC. ;
;
; SUBDOCUMENT-ID-BUFFER.
;
SDOCB: .WORD 0 ;HI.
        .WORD 0 ;MID.
        .WORD 0 ;LO.
```

DBPROC- DATABASE PROCESSOR
ERRSEQ- ERROR SEQUENCE

MACRO M1110 27-MAR-80 13:30 BY 25
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL .ERRSEQ- ERROR SEQUENCE.
;
; ERRSEQ IS CALLED WHENEVER DBPROC DISCOVERS AN ERROR WITH EITHER THE
;   - ADD DOCUMENT,
;   - ADD SUB-DOCUMENT OR
;   - REPLACE DOCUMENT.
; EXCHANGES. ERRSEQ CLEARS THE OUTPUT
; BUFFER, PLACES AN ERROR SEQUENCE CONSISTING OF FOUR WORDS OF ZERO FOLLOWED
; BY A ONE IN THE OUTPUT BUFFER AND THEN PLACES THE ERROR CODE OF THE ERROR
; DBPROC DISCOVERED FOLLOWING THE ERROR SEQUENCE IN THE BUFFER. THEN ERRSEQ
; SENDS THE OUTPUT BUFFER ACROSS THE DMC LINK AS MANY TIMES AS NECESSARY
; UNTIL ALL THE REQUIRED OUTPUT BLOCKS ARE SENT.
;
; *****
;
; ERRSEQ:
;
; CALCULATE NUMBER OF BLOCKS LEFT TO BE SENT.
;
;   MOV.   NBLK,0,R5      ; REQUIRED NUMBER OF BLOCKS TO BE SENT
;   CMP.   #1,R5         ; ONLY ONE BLOCK?
;   BNE.   1$            ; BRANCH IF NO.
;
; SINGLE BLOCK UPDATE REQUEST.
;
;   CALL.  MINCMD        ; SEND MINIMAL COMMAND RECORD TO NOTIFY
;                               ; DBUPD TO HACK THIS EXCHANGE
;   BR     3$
;
; MULTI-BLOCK EXCHANGE: ERROR OCCURED AFTER FIRST BLOCK HAS BEEN
; TRANSMITTED TO DBUPD.
;
; 1$:   SUB.   R9,R5      ; REMAINING NUMBER OF BLOCKS TO BE SENT.
;       SAVE. #OUTPUT,#N.BUFW ;
;       JSR.   PC,CLRBUF  ; CLEAR OUTPUT BUFFER.
;
; PLACE A ONE IN FIFTH WORD OF OUTPUT BUFFER.
;
;   MOV.   #1,OUTPUT+8.  ; INDICATING ERROR FOUND BY DBPROC.
;   MOV.   STATWD,OUTPUT+10. ; PLACE ERROR CODE IN OUTPUT BUFFER.
;
; SEND REMAINING BLOCKS WITH ERROR SEQUENCE & ERROR CODE IN FIRST BLOCK.
;
; 2$:   MOV.   #OUTPUT,R1  ; R1->OUTPUT BUFFER
;       JSR.   PC,SEND    ; SEND OUTPUT BUFFER.
;       BCS.   3$        ; CCIN GOT ERROR.
;       SOB.   R5,2$     ; SEND REMAINING BLOCKS.
; 3$:   RTS.   PC
```

DBPROC- DATA BASE PROCESSOR MACRO M1110 27-MAR-88 17:39 PAGE 06
EXIT- EXIT FOR FATAL ERROR ROUTINE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1052 .SBTTL EXIT- EXIT FOR FATAL ERROR ROUTINE  
1053  
1054 017606 EXIT:  
1055 017606 012700 014160' MOV #SPLFDB,R0 ;  
1056 017612 CLOSE$  
1057 017616 000137 014354' JMP DBP001
```

DBPROC- DATA-BASE-PROCESSOR
MINCMD- CREATE-MINIMAL-COMMAND-RECORD

MACRO-M1110 27-MAR-66 13:30 PM 37

```

1059                                     .SBTTL- MINCMD- CREATE MINIMAL-COMMAND-RECORD.
1060                                     :
1061                                     ; *****
1062                                     :
1063                                     ; MINCMD- IS CALLED- WHEN- AN- ERROR- IS- DISCOVERED- BEFORE- THE- FIRST- BLOCK- IS-
1064                                     ; SENT- MINCMD- CREATES- A- ONE-BLOCK- LENGTH- COMMAND- CONTAINING- THE- EXCHANGE-
1065                                     ; TYPE, THE- DOCUMENT- ID- AND- THE- ERROR- CODE- IN- THE- STATUS- WORD, THIS- IS-
1066                                     ; THEN- SENT- ACROSS- THE- DMC- LINK- AND- A- RETURN- IS- MADE.
1067                                     :
1068                                     ; *****
1069                                     :
1070 017622-                                MINCMD:
1071 017622- 012704 000010*                   MOV.   #INPUT,R4           ; INPUT-BASE-ADDR.
1072 017626- 012705 004010*                   MOV.   #OUTPUT,R5        ; OUTPUT-BASE-ADDR
1073 017632-                                SAVE.  #OUTPUT,#N,BUFW.    ;
1074 017642- 004737 015764*                   JSR.   PC,CLRBUF.        ; CLEAR-OUTPUT-BUFFER.
1075 017646- 012765 041104 000000                MOV.   #DB,PC,DBT(R5)    ; DATA-BASE-UPDATE-EXCHANGE.
1076 017654- 012765 000001 000002                MOV.   #1,PC,PAD(R5)    ; #-OF-BLOCKS-IN-EXCHANGE.
1077 017662- 016465 000000 000004                MOV.   DU,TYP(R4),PC,TYP(R5) ; UPDATE-EXCHANGE-TYPE.
1078 017670- 016465 000004 000010                MOV.   DU,IDH(R4),PC,IDH(R5) ; HIGH-ORDER-DOC- ID
1079 017676- 016465 000006 000012                MOV.   DU,IDM(R4),PC,IDM(R5) ; MIDDLE-ORDER-DOC- ID
1080 017704- 016465 000010 000014                MOV.   DU,IDL(R4),PC,IDL(R5) ; LOW-ORDER-DOC- ID
1081 017712- 013765 014134* 000006                MOV.   STATWD,PC,STW(R5) ; ERROR-CODE.
1082 017720- 012701 004010*                   MOV.   #OUTPUT,R1       ;R1-> OUTPUT-BUFFER.
1083 017724- 004737 021320*                   JSR.   PC,SEND.         ; SEND-BLOCK- TO-CONTROL-COMPUTER.
1084 017730- 103001                                BCC.   1$               ; GOOD-SEND.
1085 017732- 000251                                SEC.                                ;
1086 017734                                1$:
1087 017734 000207                                RTS.   PC.

```



```
1089 .SBTTL----OPEN...
1090 ;*****
1091 ;
1092 ; O.P.E.N. S.U.B.R.O.U.T.I.N.E.
1093 ;
1094 ; THIS SUBROUTINE OPENS AN FCS FILE. IF NO FILE WAS FOUND AND IT IS A
1095 ; MASS UPDATE, DBPROC WILL ISSUE A 22 SECOND WAIT BEFORE TRYING TO OPEN THE
1096 ; LOWEST VERSIONED FILE AGAIN. IF NO FILE WAS FOUND AND IT IS A CONTINUOUS
1097 ; UPDATE, THE C-BIT WILL BE SET AND A RETURN WILL BE MADE. IF A FILE WAS
1098 ; FOUND, THE C-BIT IS CLEARED AND A RETURN IS TAKEN.
1099 ; ANY OTHER ERROR WILL CAUSE DBPROC TO CLOSE ALL FILES AND EXIT.
1100 ;
1101 ; INPUTS:
1102 ; RO->FDB.
1103 ; OUTPUTS:
1104 ; C BIT, SET=FILE NOT FOUND, NOT SET = FILE FOUND.
1105 ;
1106 ;*****
1107 ;
1108 OPEN:
1109 OFNB$R. #SPLFDB. ; OPEN THE FILE.
1110 TSTB. F.ERR(R0) ; WAS THERE AN ERROR?
1111 BGT. 1$ ; NO.
1112 CMPB. F.ERR(R0),#1E.WAC. ; FILE STILL OPEN BY HITSK?
1113 BEQ. 2$ ; BRANCH IF YES.
1114 CMPB. F.ERR(R0),#1E.NSF. ; WAS THERE NO FILE?
1115 BEQ. 2$ ; BRANCH IF YES.
1116 MOVB. F.ERR(R0),R1 ; ANOTHER ERROR PREPARE TO REPORT.
1117 CALL. FCSERR.
1118 JMP. EXIT
1119 1$:
1120 CLC.
1121 RTS. PC.
1122 ;
1123 ; IF MASS UPDATE, WAIT FOR 22 SEC. AND TRY AGAIN. IF CONTINUOUS UPDATE, SET C-BIT AND RETURN.
1124 ;
1125 2$: TST. MASUPD. ; IS IT A MASS UPDATE?
1126 BEQ. 3$ ; NO.
1127 4$: MKRT$S. #WAT.EF,#22.,#2. ;
1128 WTSE$S. #WAT.EF. ; WAIT.
1129 BR. OPEN. ; TRY AGAIN.
1130 3$:
1131 SEC.
1132 RTS. PC.
```

```

1134 .SBTTL- PACK6- PACK SIX-BIT CHARACTERS.
1135 ;*****
1136 ;
1137 ; PACK SIX-BIT CHARACTERS INTO A WORD-BUFFER. ONCE ALL SIXTEEN BITS OF THE
1138 ; WORD ARE USED, PLACE THE WORD INTO THE OUTPUT BUFFER.
1139 ;
1140 ; INPUT: R2 HAS THE SIX-BIT CHARACTER IN THE LOW-ORDER BYTE.
1141 ; R1 ORIGINALLY INITIALIZED TO ADDRESS OF OUTPUT BUFFER.
1142 ;
1143 ; CALLING SEQUENCE: CALL @PACK6
1144 ;
1145 ; REGISTERS DESTROYED: R2.
1146 ;
1147 ; OUTPUT: R3 CONTAINS PACKED DATA.
1148 ; R1 CONTAINS POINTER TO NEXT FREE WORD OF OUTPUT BUFFER.
1149 ;
1150 ;*****
1151 ;
1152 020074 020076' PACK6: .WORD PCK1
1153 ;
1154 ;
1155 020076 PCK1:
1156 020076 012737 020112' 020074' MOV #PCK2.PACK6 ; FIRST 6-BIT CHARACTER.
1157 020104 073227 177772 ASHC #-6,R2 ; PLACE 1ST 6-BIT CHARACTER INTO R3.
1158 020110 000521 BR PCK10
1159 ;
1160 020112 PCK2:
1161 020112 012737 020126' 020074' MOV #PCK3.PACK6 ; SECOND SIX-BIT CHARACTER.
1162 020120 073227 177772 ASHC #-6,R2 ; R3 NOW CONTAINS 12-BITS OF PACKED DATA.
1163 020124 000513 BR PCK10
1164 ;
1165 020126 PCK3:
1166 020126 012737 020172' 020074' MOV #PCK4.PACK6 ; THIRD SIX-BIT CHARACTER.
1167 020134 073227 177774 ASHC #-4,R2 ; R3 NOW COMPLETELY PACKED.
1168 020140 010321 MOV R3,(R1)+ ; MOVE R3 TO OUTPUT-BUFFER.
1169 020142 062737 000001 014114' ADD #1,PWCT+WORD1 ; INCREMENT PACKED WORD-COUNT.
1170 020150 005537 014112' ADC PWCT ; ACCOUNT FOR CARRY.
1171 020154 073227 177776 ASHC #-2,R2 ; MOVE REMAINING 2 BITS OF CHAR. TO R3.
1172 020160 020127 010010' CMP R1,#OUTPUT+N.BUFB ; OUTPUT-BUFFER FULL?
1173 020164 103473 BLD PCK10 ; NO, CONTINUE.
1174 020166 000137 020336' JMP PCK9 ; YES, SEND THE OUTPUT-BUFFER.
1175 ;
1176 020172 PCK4:
1177 020172 012737 020206' 020074' MOV #PCK5.PACK6 ; FOURTH 6-BIT CHARACTER.
1178 020200 073227 177772 ASHC #-6,R2 ; MOVE 6 BITS INTO R3 - R3 NOW HAS 8 BITS.
1179 020204 000463 BR PCK10
1180 ;
1181 020206 PCK5:
1182 020206 012737 020222' 020074' MOV #PCK6.PACK6 ; FIFTH SIX-BIT CHARACTER.
1183 020214 073227 177772 ASHC #-6,R2 ; R3 NOW HAS 14 BITS OF PACKED DATA.
1184 020220 000455 BR PCK10
1185 ;
1186 020222 PCK6:
1187 020222 012737 020266' 020074' MOV #PCK7.PACK6 ; SIXTH 6-BIT CHARACTER.
1188 020230 073227 177776 ASHC #-2,R2 ; R3 NOW COMPLETELY PACKED.
1189 020234 010321 MOV R3,(R1)+ ; PLACE PACKED WORD INTO OUTPUT-BUFFER.
1190 020236 062737 000001 014114' ADD #1,PWCT+WORD1 ; INCREMENT PACKED WORD-COUNT.
    
```

```

1191 020244 005537 014112' ADC PWCT ; ACCOUNT-FOR-CARRY.
1192 020250 073227 177774 ASHC #-4,R2 ; REMAINING-4-BITS-INTO-R3
1193 020254 020127 010010' CMP R1,#OUTPUT+N,BUFB ; OUTPUT-BUFFER-FULL?
1194 020260 103435 BLD PCK10 ; NO, CONTINUE.
1195 020262 000137 020336' JMP PCK9 ; YES, SEND-THE-OUTPUT-BUFFER.
1196
1197 020266 ; PCK7:
1198 020266 012737 020302' 020074' MOV #PCK8,PACK6 ; SEVENTH-6-BIT-CHARACTER.
1199 020274 073227 177772 ASHC #-6,R2 ; R3-NOW-HAS-10-BITS-OF-PACKED-DATA.
1200 020300 000425 BR PCK10
1201
1202 020302 ; PCK8:
1203 020302 012737 020076' 020074' MOV #PCK1,PACK6 ; EIGHTH-6-BIT-CHARACTER.
1204 020310 073227 177772 ASHC #-6,R2 ; R3-NOW-COMPLETLY-PACKED.
1205 020314 010321 MOV R3,(R1)+ ; MOVE-PACKED-WORD-INTO-OUTPUT-BUFFER.
1206 020316 062737 000001 014114' ADD #1,PWCT+WORD1 ; INCREMENT-PACKED-WORD-COUNT.
1207 020324 005537 014112' ADC PWCT ; ACCOUNT-FOR-CARRY.
1208 020330 020127 010010' CMP R1,#OUTPUT+N,BUFB ; OUTPUT-BUFFER-FULL?
1209 020334 103407 BLD PCK10 ; NO, CONTINUE.
1210 020336 012701 004010' PCK9: MOV #OUTPUT,R1 ; R1-> OUTPUT-BUFFER.
1211 020342 004737 021320' JSR PC,SEND ; YES, SEND-THE-OUTPUT-BUFFER.
1212 020346 103002 BCC PCK10 ;
1213 020350 000261 SEC ;
1214 020352 000401 BR PCK11 ;
1215 020354 PCK10: CLC ;
1216 020354 000241 PCK11: RTS PC ;
1217 020356 000207 ;
    
```

```

1219 .SBTTL PRDOC- PURGE-OR-READ-DOCUMENTS.
1220 ;
1221 ;
1222 ; *****
1223 ; PRDOC-PREPROCESSES BOTH PURGE AND READ DOCUMENT EXCHANGES. ONE
1224 ; OR TWO OUTPUT BATCH FILES WILL BE CREATED.
1225 ;
1226 ; *****
1227 020360 PRDOC:
1228 020360 SAVE #OUTPUT,#N,BUFW ; ADDRESS AND LENGTH OF BUFFER ON STACK.
1229 020370 004737 015764' JSR PC,CLRBUF ; CLEAR 1ST OUTPUT BUFFER.
1230 020374 SAVE #OUTPT2,#N,BUFW ; ADDRESS AND LENGTH OF 2ND BUFFER ON STACK.
1231 020404 004737 015764' JSR PC,CLRBUF ; CLEAR 2ND OUTPUT BUFFER.
1232 020410 012701 000010' MOV #INPUT,R1 ; ADDR OF INPUT BUFFER.
1233 020414 012702 004010' MOV #OUTPUT,R2 ; 1ST OUTPUT BUFFER ADDRESS.
1234 020420 012703 010010' MOV #OUTPT2,R3 ; 2ND OUTPUT BUFFER ADDRESS.
1235 020424 012762 041104 000000 MOV #DB,PC,DBT(R2) ; SET DATA BASE TYPE FOR CCIN.
1236 020432 016162 000000 000004 MOV PR,TYP(R1),PC,TYP(R2) ; MOVE TYPE OF EXCHANGE TO 1ST OUTPUT BUFFER.
1237 020440 012762 000001 000006 MOV #AS,SUP,PC,STW(R2) ; LOAD REQUEST STATUS WORD.
1238 020446 012763 041104 000000 MOV #DB,PC,DBT(R3) ; SET DATA BASE TYPE FOR CCIN.
1239 020454 016163 000000 000004 MOV PR,TYP(R1),PC,TYP(R3) ; MOVE TYPE OF EXCHANGE TO 2ND OUTPUT BUFFER.
1240 020462 012763 000001 000006 MOV #AS,SUP,PC,STW(R3)
1241 020470 004737 021514' JSR PC,SORTID ; SORT IDS INTO ASCENDING ORDER.
1242 020474 016104 000004 MOV PR,DIC(R1),R4 ; * OF DOC. IDS.
1243 020500 062701 000006 ADD #PR,FDI,R1 ; ADDR OF 1ST DOC. ID IN INPUT BUFFER.
1244 020504 062702 000012 ADD #RP,FID,R2 ; ADD OFFSET FOR FIRST ID ADDR IN 1ST OUTPUT BUFFER.
1245 020510 062703 000012 ADD #RP,FID,R3 ; ADD OFFSET FOR FIRST ID ADDR IN 2ND OUTPUT BUFFER.
1246 020514 PRD000:
1247 020514 SAVE R1 ; PLACE ADDR OF DOC. ID ON STACK.
1248 020516 004737 021742' JSR PC,VALDID ; DETERMINE THE SU
1249 020522 012705 000003 MOV #3,R5 ; R5=WORD LENGTH OF DOC ID.
1250 020526 022737 000001 014056' CMP #1,CSU ; SU #0 OR SU #1?
1251 020534 002430 BLT 30$ ; BRANCH IF SU #1 (CSU = 2)
1252 ;
1253 ; IF "DELETE-SUB-DOC" EXCHANGE, FIRST STORE PACKED SUB-DOC ID IN
1254 ; OUTPUT BUFFER, THEN THE DOC ID.
1255 ;
1256 020536 023727 000000C 051504 10$: CMP INPUT+PR,TYP,#"DS ; DELETE-SUB-DOC?
1257 020544 001017 BNE 11$ ; BRANCH IF NO.
1258 020546 SAVE R1,R4
1259 020552 010104 MOV R1,R4 ; R4-> DOC ID.
1260 020554 062704 000006 ADD #6,R4 ; R4-> SUB-DOC ID.
1261 020560 000314 SUBAB (R4) ; MOVE MOST SIG. BYTE OF ID INTO HI-BYTE.
1262 020562 062704 000001 ADD #1,R4 ; R4->MOST SIG. BYTE OF 5-BYTE SUB-DOC ID.
1263 020566 010201 MOV R2,R1 ; R1-> OUTPUT BUFFER.
1264 020570 062701 000005 ADD #5,R1
1265 020574 CALL CRSDSS ; PACK SUB-DOC ID.
1266 020600 RESTOR R1,R4
1267 020604 012122 11$: MOV (R1)+,(R2)+ ; STORE ID IN 1ST OUTPUT BUFFER.
1268 020606 077502 SOB R5,11$ ;
1269 020610 005237 000000C INC OUTPUT+RP,IND ; INCREMENT DOCUMENT ID COUNT FOR BUFFER.
1270 020614 000427 BR PRD040 ;
1271 ;
1272 020616 023727 000000C 051504 30$: CMP INPUT+PR,TYP,#"DS ; DELETE-SUB-DOC?
1273 020624 001017 BNE 31$ ; BRANCH IF NO.
1274 020626 SAVE R1,R4
1275 020632 010104 MOV R1,R4 ; R4-> DOC ID.
    
```

DBPROC- DATA-BASE-PROCESSOR.
PRDOC- PURGE-OR-READ-DOCUMENTS.

MACRO M1110 27 MAR 80 13 20 PAGE 78-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1276 020634 062704 000006      ADD      #5,R4          ;R4-> SUB-DOC-ID.  
1277 020640 000314      SWAB     (R4)          ;MOVE MOST-SIG. BYTE OF ID INTO HI-BYTE.  
1278 020642 062704 000001      ADD      #1,R4          ;R4->MOST-SIG. BYTE OF 5-BYTE SUB-DOC-ID.  
1279 020646 010301      MOV      R3,R1         ;R1-> OUTPUT-BUFFER.  
1280 020650 062701 000005      ADD      #5,R1  
1281 020654      CALL    CRSDSS         ;PACK-SUB-DOC-ID.  
1282 020660      RESTOR  R1,R4  
1283 020664 012123      31#: MOV      (R1)+,(R3)+ ; STORE-ID-IN-2ND-OUTPUT-BUFFER.  
1284 020666 077502      SOB     R5,31#         ;  
1285 020670 005237 000000C.    INC     OUTPT2+RP,NID. ; INCREMENT-DOCUMENT-ID-COUNT.  
1286 020674      PRD040:  
1287 020674 077471      SOB     R4,PRD000     ; MORE-IDS-TO-PROCESS?  
1288 020676 005737 000000C.    TST    OUTPUT+RP,NID. ; ANY-IDS-FOR-SU-#0?  
1289 020702 001413      BEQ    PRD050         ; NO.  
1290  
1291      ;  
1292      ; SEND-BUFFER-TO-SU0  
1293      ;  
1293 020704 012737 000001 014056'    MOV     #1,CSU.       ;SU-0  
1294 020712 012701 004010'    MOV     #OUTPUT,R1    ;R1-> OUTPUT-BUFFER.  
1295 020716      CALL    SEND          ;SEND-BUFFER-TO-SU-0  
1296 020722 103003      BCC    PRD060        ;BRANCH-IF-OK.  
1297 020724 005237 014120'    INC     REDO.F.       ;SET-REDO-FLAG.  
1298 020730 000415      BR     PRD080        ;RETURN.  
1299  
1300      ;  
1301      ; SEND-BUFFER-TO-SU-1 (IF-BUFFER-HAS-SOMETHING-IN-IT).  
1302      ;  
1302 020732 012701 010010'    PRD060: MOV     #OUTPT2,R1 ;R1-> OUTPUT-BUFFER.  
1303 020736 005761 000010      TST    RP,NID(R1)    ;ANY-IDS-FOR-SU-1 ?  
1304 020742 001410      BEQ    PRD080        ;BRANCH-IF-NO.  
1305 020744 012737 000002 014056'    MOV     #2,CSU.       ;SU-1  
1306 020752      CALL    SEND          ;SEND-BUFFER-TO-SU-1  
1307 020756 103002      BCC    PRD080        ;BRANCH-IF-OK.  
1308 020760 005237 014120'    INC     REDO.F.       ;SET-REDO-FLAG.  
1309 020764      PRD080:  
1310 020764 000207      RTS     PC            ;
```

```
.SBTTL- PREPEX- PREPROCESS EXCHANGE REQUEST.  
; *****  
; ;  
; PREPEX MESSAGES CERTAIN EXCHANGES FOR EASE OF HANDLING. ALL EXCHANGES THAT  
; COME ACROSS THE LINK IN BYTES HAVE THEIR HIGH AND LOW ORDER BYTES IN REVERSE  
; ORDER. PREPEX DETERMINES WHICH EXCHANGES NEED THIS SWAPPING OF BYTES. DOES  
; THE SWAPPING AND SETS A FLAG, EXTYPE, TO INDICATE THE TYPE OF EXCHANGE.  
; THE BYTE EXCHANGES ARE ADD DOCUMENT, MODIFY-REPLACE DOCUMENT, AND MODIFY-  
; OVERLAY DOCUMENT CHARACTERS AND ARE REPRESENTED BY EXTYPE =1.  
; THE WORD EXCHANGES ARE PURGE AND READ DOCUMENTS AND DELETE SUB-DOCUMENTS  
; AND ARE REPRESENTED BY EXTYPE=2.  
; *****  
; ;  
PREPEX:  
SAVE R3 ;  
MOV *INPUT,R3 ; BASE ADDR OF INPUT BUFFER.  
MOV *2,EXTYPE ; INDICATES PURGE OR READ EXCHANGE.  
CMP #"RP,PR,TYP(R3) ; PURGE REQUEST?  
BEQ 3$ ; YES  
CMP #"DR,PR,TYP(R3) ; READ DOCUMENT REQUEST?  
BEQ 3$ ; YES  
CMP #"SD,PR,TYP(R3) ; DELETE SUB-DOC?  
BEQ 3$ ; BRANCH IF YES  
DEC EXTYPE ; INDICATES ADD OR MODIFY DOC. EXCHANGES.  
CMP #"DC,DU,TYP(R3) ; ADD NEW DOCUMENT REQUEST?  
BNE 1$ ; NO  
SWAB DU,WSC(R3) ; SWAP BYTES CONTAINING WHITE SPACE COUNT.  
1$:  
SWAB DU,IDH(R3) ; SWAP BYTES CONTAINING HIGH ORDER ID.  
SWAB DU,IDM(R3) ; SWAP BYTES CONTAINING MIDDLE ORDER ID.  
SWAB DU,IDL(R3) ; SWAP BYTES CONTAINING LOW ORDER ID.  
SWAB DU,TCC(R3) ; SWAP BYTES CONTAINING TEXT CHARACTER COUNT.  
SWAB DU,TCC+WORD1(R3) ;  
CMP #"OV,DU,TYP(R3) ; MODIFY-OVERLAY REQUEST?  
BNE 3$ ; NO  
SWAB DU,OCO(R3) ; SWAP BYTES CONTAINING OFFSET FOR OVERLAY--  
SWAB DU,OCO+WORD1(R3) ; --CHARACTERS  
3$:  
RESTOR R3 ;  
RTS PC ;
```

DBPROC--DB BASE PROCESSOR
--- SEND:

MACRO-M1110 27-MAR-88 13:38 P. 33

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1411 .SBTTL-----SEND-----
1412 ;
1413 ; SEND S.E.N.D. S.U.B.R.O.U.T.I.N.E.
1414 ;
1415 ; SEND AN OUTPUT BUFFER TO THE SEARCH UNIT VIA THE DMC LINK. IF AN ERROR IS
1416 ; DETECTED THAT IS RECOVERABLE, DBPROC WILL CLOSE THE INPUT FILE, REOPEN IT AND
1417 ; REPROCESS IT. IF THE ERROR IS UNRECOVERABLE, AN ERROR MESSAGE IS PRINTED
1418 ; AND DBPROC WILL EXIT.
1419 ;
1420 ; INPUTS:
1421 ; R1-> OUTPUT BUFFER
1422 ; OUTPUTS:
1423 ; R1->OUTPUT BUFFER
1424 ;
1425 ;
1426 021320 SEND:
1427 021320 SAVE R1
1428 021322 010137 014334' MOV R1,D,I0PL+DMC ;LOAD BUFFER ADDRESS INTO Q10 DPB
1429 021326 013737 014056' 014324' MOV CSU,DMC+0,I0LU ;SET UP LUN
1430 021334 DIR# #DMC ;SEND THE BUFFER
1431 021342 105737 000000' TSTB I0STAT ;WAS THERE AN ERROR?
1432 021346 003050 BGT 1$ ;NO, RETURN TO CALLER
1433 021350 123727 000000' 000000G CMPB I0STAT,#IE,DNR ;DEVICE NOT READY?
1434 021356 001427 BEQ 2$ ;YES, CLOSE INPUT FILE
1435 021360 123727 000000' 000000G CMPB I0STAT,#IE,CNR ;CONNECTION REJECTED?
1436 021366 001423 BEQ 2$ ;YES, CLOSE INPUT FILE
1437 021370 123727 000000' 000000G CMPB I0STAT,#IE,ABO ;REQUEST ABORTED?
1438 021376 001417 BEQ 2$ ;YES, CLOSE INPUT FILE
1439 021400 123727 000000' 000000G CMPB I0STAT,#IE,TMO ;TIME OUT?
1440 021406 001413 BEQ 2$ ;YES
1441 021410 113737 000000' 014104' MOVB @#I0STAT,PAR1+2 ;YES, CONVERT ERROR NO TO WORD VALUE
1442 021416 CALL ERROR
1443 021422 CLOSE# #SPLFDB
1444 021432 000137 017606' JMP EXIT
1445 021436 2$:
1446 021436 CLOSE# #SPLFDB
1447 021446 MOUT#$ #RETRY
1448 021464 000261 SEC
1449 021466 000410 BR 3$
1450 021470 1$:
1451 021470 005237 014126' INC R9 ; KEEP TRACK OF # OF BLOCKS SENT
1452 021474 SAVE R1,#N,BUFW ;
1453 021502 004737 015764' JSR PC,CLRBUF ; CLEAR OUTPUT BUFFER
1454 021506 000241 CLC
1455 021510 3$: RESTOR R1
1456 021512 000207 RTS PC
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1354 .SBTTL - --- READ  
1355 ;  
1356 ;  
1357 ; R E A D S U B R O U T I N E  
1358 ;  
1359 ; THE READ SUBROUTINE READS A BLOCK FROM A DATA BASE SPOOL FILE IF THERE  
1360 ; ARE ANY REMAINING INPUT BLOCKS TO READ. IF THERE ARE NO MORE INPUT BLOCKS  
1361 ; TO READ, AN ERROR CODE IS SET IN THE STATUS WORD, A RECOVERY ROUTINE IS  
1362 ; CALLED TO SEND THE REMAINING OUTPUT BLOCKS, AND A FLAG IS SET TO INDICATE  
1363 ; THE ERROR TO THE CALLING ROUTINE. IF AN END-OF-FILE IS FOUND WITH THE  
1364 ; READ, THE SAME PROCEDURE IS FOLLOWED. IF ANY OTHER ERROR IS DETECTED,  
1365 ; A FLAG IS SET AND A RETURN IS MADE. IF THE READ IS SUCCESSFUL, THE NUMBER  
1366 ; OF BLOCKS TO INPUT IS DECREMENTED.  
1367 ;  
1368 ; INPUT:  
1369 ; R0 -> SPLFDB  
1370 ; OUTPUT:  
1371 ; INPUT CONTAINS THE BLOCK READ FROM THE FILE.  
1372 ; NBREAD - NO. OF BYTES READ INTO INPUT BUFFER.  
1373 ; RDRSLT - 1 FOR HARD ERRORS  
1374 ; - 10 FOR END-OF-FILE DETECTION.  
1375 ;  
1376 ;  
1377 ;  
1378 ;  
1378 021122 READ:  
1379 021122 CLR RDRSLT ;  
1380 021126 TST NBLK.I ; MORE BLOCKS TO INPUT?  
1381 021132 BGT 1$ ; YES  
1382 021134 MOV #ERR103,PAR1+2 ; NO MORE INPUT BLOCKS  
1383 021142 CALL ERROR  
1384 021146 MOV #AS.EOF,STATWD ; SET END-OF-FILE ERROR CODE IN STATUS WORD  
1385 021154 CALL ERRSEQ ; CALL ERROR SEQUENCE SEND ROUTINE  
1386 021160 BIS #BIT3,RDRSLT ; SET ERROR FLAG  
1387 021166 RTS PC ;  
1388 021170  
1389 021170 1$:  
1390 021200 READ$ #SPLFDB ; ISSUE READ  
1391 021210 WAIT$ #SPLFDB ; WAIT FOR THE READ TO COMPLETE  
1392 021214 TSTB RI0ST ; WAS THERE AN ERROR?  
1393 021216 BGT 10$ ; NO RETURN  
1394 021224 CMPB #IE.EOF,RI0ST ; END-OF-FILE DETECTED?  
1395 021226 BNE 2$ ; NO  
1396 021234 MOV #ERR100,PAR1+2  
1397 021240 CALL ERROR  
1398 021246 BIS #BIT3,RDRSLT ; SET END-OF-FILE INDICATOR  
1399 021254 MOV #AS.EOF,STATWD ; SET END-OF-FILE ERROR CODE IN STATUS WORD  
1400 021260 CALL ERRSEQ ; CALL ERROR SEQUENCE SEND ROUTINE  
1401 021262 RTS PC ;  
1402 021262 2$:  
1403 021270 BIS #BIT0,RDRSLT ; OTHER ERROR  
1404 021276 MOVB @#RI0ST,PAR1+2 ; YES, CONVERT BYTE ERROR TO WORD ERROR  
1405 021302 CALL ERROR  
1406 021304 RTS PC ;  
1407 021304 10$:  
1408 021312 MOV NBREAD ; NO. OF BYTES READ INTO INPUT BUFFER  
1409 021316 DEC NBLK.I ; DECREMENT # OF BLOCKS TO INPUT  
1409 021316 RTS PC ;
```


DBPROC-DB BASE PROCESSOR
VALIDID- VALIDATE DOCUMENT ID

MACRO-M110 27-MAR-80 13:30 P. 35

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1508 ; SBTTL- VALDID- VALIDATE DOCUMENT ID
1509 ;
1510 ; VALDID CHECKS THE DOCUMENT ID AGAINST THE RANGE OF DOCUMENT IDS FOR EACH
1511 ; SEARCH UNIT TO DETERMINE WHICH SEARCH UNIT IS TO RECEIVE THE UPDATE REQUEST
1512 ; IF THE DOCUMENT ID DOES NOT FALL WITHIN EITHER RANGE AN ERROR FLAG IS
1513 ; RETURNED IN CMPST AND THE DOCUMENT IS MARKED TO BE SENT TO SUB
1514 ;
1515 021742 VALDID:
1516 021742 SAVE R0,R1,R2,R3,R4,R5
1517 021756 016601 000016 MOV WORD7(SP),R1 ;R1-> ID TO VALIDATE
1518 021762 013737 014054' 014122' MOV NO,SU,R0 ; NUMBER OF SEARCH UNITS IN CONFIGURATION
1519 021770 005037 014052' CLR CMPST ; CLEAR COMPARE STATUS WORD
1520 021774 012737 000001 014056' MOV #1,CSU ; INITIALIZE CURRENT SEARCH UNIT TO #0
1521 022002 005005 CLR R5 ; INITIALIZE INDEX INTO TABLES TO SU #0
1522 022004 VAL010:
1523 022004 016502 0000006 MOV SRECP(R5),R2
1524 022010 010203 MOV R2,R3
1525 022012 062702 000072 ADD #SR,GRS,R2 ;R2->DOC. ID RANGE START FOR SU
1526 022016 062703 000100 ADD #SR,GRE,R3 ;R3->DOC. ID RANGE END FOR SU
1527 022022 004737 016022' JSR PC,CMP3WD ; ID FALL WITHIN RANGE FOR SU *(R5)
1528 022026 005737 014052' TST CMPST ; RESULTS?
1529 022032 002412 BLT VAL020 ; INVALID ID
1530 022034 001415 BEQ VAL030 ; VALID FOR THIS SU
1531 022036 005337 014122' DEC R0 ; ALL SU'S CHECKED?
1532 022042 001406 BEQ VAL020 ; YES, AND ID IS INVALID FOR ALL SU'S
1533 022044 062705 000002 ADD #2,R5 ; GET NEXT SU POINTER
1534 022050 005237 014056' INC CSU ; CURRENT SU NUMBER
1535 022054 000137 022004' JMP VAL010 ; CHECK NEXT SET OF RANGES
1536 022060 VAL020:
1537 022060 005005 CLR R5 ; DEFAULT SU FOR UPDATE IS #0
1538 022062 012737 000001 014056' MOV #1,CSU ; DEFAULT SU # FOR INVALID ID
1539 022070 VAL030:
1540 022070 RESTOR R0,R1,R2,R3,R4,R5
1541 022104 000002 MOV (SP),WORD1(SP) ;SET UP RETURN ADDRESS
1542 022110 062706 000002 ADD #2,SP ;ADJUST STACK POINTER
1543 022114 000207 RTS PC ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

1458 ; .SBTTL SORTID- SORT IDS IN ASCENDING ORDER SEQUENCE
1459 ;
1460 ; SORTID SORTS ONE TO EIGHT-FOUR (04) DOCUMENT IDS IN ASCENDING ORDER
1461 ; IN ONE 1024 WORD INPUT BUFFER.
1462 ;
1463 ; SORTID:
1463 021514 SAVE R1,R2,R3 ;
1464 021514 MOV INPUT+PR.DIC,R0 ; # OF DOCUMENT IDS
1465 021522 013737 000000C 014122* CMP R0,#1 ; ONLY 1 ID?
1466 021530 023727 014122* 000001 BGT 10$ ; NO, MORE
1467 021536 003004 RESTOR R1,R2,R3 ; POP STACK
1468 021540 RTS PC ; YES, RETURN
1469 021546 000207 DEC R0 ; ONE LESS FOR COMPARISONS
1470 021550 005337 014122* 10$: MOV #INPUT+PR.FDI,R1 ; GET POINTER TO 1ST ENTRY
1471 021554 012701 000000C 1$: MOV R1,R2 ;
1472 021560 010102 CMP INPUT+PR.TYP,#"DS ; DELETE SUB-DOC. EXCHANGE?
1473 021562 023727 000000C 051504 BNE 4$ ; BRANCH IF NO
1474 021570 001004 MOV #N.BDSD,R9 ; R9 = BYTE LENGTH OF ENTRIES
1475 021572 012737 000020 014126* BR 5$
1476 021600 000403 MOV #N.BPRE,R9 ;
1477 021602 012737 000006 014126* 4$: ADD R9,R2 ; R2-> SECOND ENTRY
1478 021610 063702 014126* 5$: MOV R9,R2 ;
1479 021614 013703 014122* MOV R0,R3 ; COMPARISON COUNTER
1480 021620 2$:
1481 021620 021112 CMP (R1),(R2) ; COMPARE 1ST WORDS OF ID
1482 021622 101012 BHI 40$ ; SWAP THE ENTRIES - 1ST > 2ND
1483 021624 001034 BNE 3$ ; BRANCH IF IN ASCENDING ORDER
1484 021626 026162 000002 000002 CMP WORD1(R1),WORD1(R2) ; COMPARE THE 2ND WORDS OF ID
1485 021634 101005 BHI 40$ ; SWAP THE ENTRIES - 1ST > 2ND
1486 021636 001027 BNE 3$ ; BRANCH IF IN ASCENDING ORDER
1487 021640 026162 000004 000004 CMP WORD2(R1),WORD2(R2) ; COMPARE THE 3RD WORDS OF ID
1488 021646 101423 BLOS 3$ ; IN ASCENDING ORDER, CHECK NEXT SET
1489 021650 40$:
1490 021650 011246 MOV (R2),-(SP) ; PUSH LOWER ID ON STACK
1491 021652 016246 000002 MOV WORD1(R2),-(SP) ;
1492 021656 016246 000004 MOV WORD2(R2),-(SP) ;
1493 021662 011112 MOV (R1),(R2) ; PLACE HIGHER ID IN HIGHER ENTRY
1494 021664 016162 000002 000002 MOV WORD1(R1),WORD1(R2) ;
1495 021672 016162 000004 000004 MOV WORD2(R1),WORD2(R2) ;
1496 021700 012661 000004 MOV (SP)+,WORD2(R1) ; PLACE LOWER ID IN LOWER ENTRY
1497 021704 012661 000002 MOV (SP)+,WORD1(R1) ;
1498 021710 012611 MOV (SP)+,(R1) ;
1499 021712 000137 021554* JMP 1$ ;
1500 021716 3$:
1501 021716 063701 014126* ADD R9,R1 ; GET NEXT ENTRY
1502 021722 063702 014126* ADD R9,R2 ; GET NEXT ENTRY
1503 021726 005303 DEC R3 ; DECREMENT COMPARISON COUNTER
1504 021730 003333 BGT 2$ ; MORE TO COMPARE
1505 021732 RESTOR R1,R2,R3 ;
1506 021740 000207 RTS PC ;
    
```

DBPROC - DB BASE PROCESSOR
ERROR MESSAGES

MACRO: M1110 27-MAR-80 13:30 P. 137
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL .ERROR MESSAGES
;
;
;
1583 022220 011637 014102' ERROR: MOV. (SP),PAR1 ;PAR1 = PC OF CALLING ROUTINE.
1584 022224 MOUT$S. #MSG3,#PAR1
1585 022244 000207 RTS. PC.
1586
;
1587 ; DIRECTIVE ERROR.
1588
;
1589 022246 011637 014102' DIRERR: MOV. (SP),PAR1
1590 022252 010137 014104' MOV. R1,PAR1+2.
1591 022256 MOUT$S. #MSG1,#PAR1
1592 022276 000207 RTS. PC.
1593
;
1594 ; FCS ERROR.
1595
;
1596 022300 011637 014102' FCSERR: MOV. (SP),PAR1
1597 022304 010137 014104' MOV. R1,PAR1+2.
1598 022310 MOUT$S. #MSG2,#PAR1
1599 022330 000207 RTS. PC.
1600
;
1601 ; ERROR CODE DEFINITIONS.
1602
;
1603 177634 ERR100 = -100. ;FOUND END OF FILE BEFORE CHAR. COUNT. = 0
1604 177633 ERR101 = -101. ;SOMETHING WRONG WITH FCS BLOCK NUMBER.
1605 177632 ERR102 = -102. ;HARD ERROR.
1606 177631 ERR103 = -103. ;NO MORE INPUT BLOCKS.
1607 177630 ERR104 = -104. ;ZERO LENGTH FILE OPENED.
1608
;
1609
;
1610
;
1611 .NLIST BEX.
1612
;
1613
;
1614 022332 000046 MSG1: .WORD LN0E-LN0
1615 022334 022356' .WORD LN0
1616 022336 000064 EMSG1: .WORD LN1E-LN1
1617 022340 022424' .WORD LN1
1618 022342 000042 MSG2: .WORD LN3E-LN3
1619 022344 022510' .WORD LN3
1620 022346 000026 MSG3: .WORD LN0E-LN0
1621 022350 022552' .WORD LN0
1622 022352 000006 RETRY: .WORD LN11E-LN11
1623 022354 022600' .WORD LN11
1624
;
1625 022356 104 111 122 LN0: .ASCIZ /DIRECTIVE ERROR: PC = %10,DSW = %1D/.
1626 022424 LN0E:
;
1627
;
1628 022424 123 105 116 LN1: .ASCIZ /SENT %1D BLOCKS, CALCULATED %1D BLOCKS TO SEND./.
1629 022510 LN1E:
;
1630
;
1631 022510 106 103 123 LN3: .ASCIZ /FCS ERROR: PC = %10,ERROR = %1D/.
1632 022552 LN3E:
;
1633
;
1634 022552 120 103 040 LN0: .ASCIZ /PC = %10,ERROR = %1D/.
1635 022600 LN0E:
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

1545 .SBTTL- SNDEND- SEND- END-OF-UPDATE- EXCHANGE-
1546 ;*****
1547 ;
1548 ; BUILD- "END-OF-UPDATE- EXCHANGE- AND- TRANSMIT- IT- TO- ALL- SEARCH- UNITS-
1549 ;
1550 ;*****
1551 ;
1552-022116 .SNDEND:
1553-022116 ;
1554-022122-005037-014066' 3$: SAVE- R1,R4
1555-022126-012701-004010' CLR- FLAG ;ZERO-OUT- NO. OF-UPDATES- PERFORMED-
MOV- #OUTPUT,R1 ;R1-> OUTPUT-BUFFER-
1556-022132- SAVE- #OUTPUT,#N, BUFW-
1557-022142- CALL- CLRBUF-
1558-022146-013704-014054' MOV- NO,SU,R4 ;R4 =- NUMBER- OF- SEARCH- UNITS
1559-022152-012737-000001-014056' MOV- #1,CSU ;START- WITH- SU0
1560-022160-012761-042105-000000 2$: MOV- #ED,PC,DBT(R1) ;END- OF- UPDATE- EXCHANGE-
1561-022166-012761-000001-000002- MOV- #1,PC,PAD(R1) ;NUMBER- OF- BLOCKS- IN- EXCHANGE-
1562-022174-004737-021320' JSR- PC,SEND- ;SEND- EXCHANGE-
1563-022200-103001 BCC- 1$ ;BRANCH- IF- OK-
1564 ;
1565 ; EXCHANGE- WAS- NOT TRANSMITTED- TO- SU. RETRY.
1566 ;
1567-022202-000747 BR 3$
1568 ;
1569 ;
1570 ;
1571-022204-005237-014056' 1$: INC- CSU-
1572-022210-077415 SOB- R4,2$
1573 ;
1574 ; RETURN
1575 ;
1576-022212- RESTOR- R1,R4
1577-022216-000207 RTS- PC-
    
```

DBPROC - DATA BASE PROCESSOR
ERROR MESSAGES

MACRO: M1110 27-MAR-68 13:38 PAGE 37-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1636				:	
1637	022600	122	105	124	LN11: .ASCIZ /RETRY/
1638	022606				LN11E:
1639				:	
1640					.EVEN:
1641	014350				.END: DBPROC

DBPROC - DATA BASE PROCESSOR
SYMBOL TABLE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

N.PKSZ= 000020	PRD080 020764R	SEND 021320R	ST.BTC 000000	006 WORD0 = 000000
N.PKTS= 000043	PREPEX 020766R	SNDEND 022116R	ST.CSZ 000030	006 WORD1 = 000002
N.QUERY= 000031	PRGH = 000021	SND.EF = 000003	ST.HRL 000010	006 WORD2 = 000004
N.STAT= 000020	PR.DIC 000004	017 SNTC = 000024	ST.LEN 000044	006 WORD3 = 000006
N.SUNT= 000002	PR.FDI 000006	017 SORTID 021514R	ST.ORY 000002	006 WORD4 = 000010
N.UNIT= 000034	PR.PAD 000002	017 SPLFDB 014160R	ST.OSZ 000034	006 WORD5 = 000012
OPEN 017736R	PR.TYP 000000	017 SRECPT = ***** GX	ST.SCH 000040	006 WORD6 = 000014
OUTPT2 010010R	PTCW 014106R	SR.ARE 000114	002 ST.UHL 000004	006 WORD7 = 000016
OUTPUT 004010R	PUTSSQ = ***** GX	SR.ARS 000106	002 ST.XLT 000014	006 WORD8 = 000020
PACK6 020074R	PWCT 014112R	SR.DAY 000010	002 SU.DBU 000004	006 WORD9 = 000022
PAR***= 000027	QE.RD1 = 000144	SR.DLT 000014	002 SU.DON = 000006	WORDVAL = 000024
PAR1 014102R	Q.FDSC 000004	007 SR.ECB 000047	002 SU.IDL = 000000	WSPCNT 014142R
PCK1 020076R	Q.I0AE = 000012	SR.ECH 000046	002 SU.LDD = 000001	XBATCH = 000013
PCK10 0203354R	Q.I0EF = 000006	SR.ECL 000050	002 SU.SRC = 000002	XDBLOA = 000004
PCK11 0203356R	Q.I0FN = 000002	SR.FIB 000012	002 SU.SRR = 000005	XDBPRO = 000012
PCK2 020112R	Q.I0LU = 000004	SR.GRE 000100	002 SU.XPD = 000003	XDMCIN = 000006
PCK3 020126R	Q.I0PL = 000014	SR.GRS 000072	002 SZMK = 000023	XFOSMR = 000007
PCK4 020172R	Q.I0PR = 000007	SR.LEN 000122	002 S.BFHD = 000020	XGTSRE = 000014
PCK5 020206R	Q.I0SB = 000010	SR.LIN 000066	002 S.FATT = 000016	XHITSK = 000011
PCK6 020222R	Q.NQBK 000000	007 SR.LIP 000062	002 S.FDB = 000140	XHLMER = 000002
PCK7 020266R	Q.NUHL 000002	007 SR.MON 000006	002 S.FNAM = 000006	XHOTSK = 000010
PCK8 020302R	Q.SIZE 000014	007 SR.NDC 000042	002 S.FNB = 000036	XNSCHE = 000000
PCK9 020336R	RDRSLT 014116R	SR.NDS 000036	002 S.FNBW = 000017	XQTS = 000003
PC.DBT 000000	020 RD.LUN = 000003	SR.NIN 000030	002 S.FNTY = 000004	XQT0 = 000001
PC.IDH 000010	020 READ 021122R	SR.NIP 000022	002 S.FTYP = 000002	XSULOA = 000005
PC.IDL 000014	020 REDO.F 014120R	SR.SDB 000032	002 S.HRL = 000240	ZMK = 000022
PC.IDM 000012	020 REF = 000001	SR.SRC 000002	002 S.NFEN = 000020	\$DDIV = ***** GX
PC.PAD 000002	020 RETRY 022352R	SR.SUN 000000	002 TRYAG 014446R	\$DMUL = ***** GX
PC.PTC 000032	020 RI0ST 000004R	SR.TWS 000056	002 TRY.1 014510R	\$DSW = ***** GX
PC.PWC 000026	020 RP.FID = 000012	020 SR.WSL 000052	002 VALDID 021742R	\$\$\$ARG = 000004
PC.SPD 000036	020 RP.NID = 000010	020 SR.YR 000004	002 VAL010 022004R	\$\$\$OST = 000014
PC.STW 000006	020 RD.BH 000006	015 SR.1IH 000024	002 VAL020 022006R	.CLOSE = ***** G
PC.TCC 000016	020 RD.CB 000004	015 SR.1IP 000016	002 VAL030 022070R	.DELETE = ***** G
PC.TYP 000004	020 RD.CS 000005	015 SS.FID 000002	004 VBLKN 014136R	.FINIT = ***** G
PC.UJS 000024	020 RD.DUM 000000	015 SS.FNB 000010	004 WAT.EF = 000002	.FSRCB = ***** G
PC.WSC 000022	020 R8 014122R	SS.FVR 000006	004 WN.NTP 000004	012 .OPFNB = ***** G
PRB 014010R	R9 014126R	SS.LEN 000012	004 WN.NXT 000006	012 .READ = ***** G
PRDOC 020360R	SAVE 014130R	SS.STT 000000	004 WN.ROT 000002	012 .WAIT = ***** G
PRD080 020514R	SBMK = 000000	STATWD 014134R	WN.SIZ 000010	012 ...PC1 = 014160R
PRD040 020674R	SDOCB 017506R	ST.ASZ 000020	006 WN.SRC 000000	012 ...PC2 = 014320R
PRD060 020732R	SD.LUN = 000001	ST.BSZ 000024	006 WN.TYP 000001	012 ...TPC = 000020
. ABS 000000 000				
022606 001				
SRCOFF 000122 002				
FDSCOF 000010 003				
SUSOFF 000012 004				
DHROFF 000012 005				
STTOFF 000044 006				
QSPLOF 000014 007				
BSTOFF 000772 010				
FNOFFS 000044 011				
WNODOF 000010 012				
DNODOF 000010 013				
CCET 000200 014				
RCVOFF 000010 015				
ADOFF 000020 016				

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBPROC - BASE PROCESSOR
SYMBOL TABLE

MACRO-M11 10 27-MAR-80 13:30 P. 37-4
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

PUR0FF- 000010 017
PCROFF- 000040 020
\$\$FSR1 000000 021
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 7827 WORDS (31 PAGES)
DYNAMIC MEMORY: 9140 WORDS (35 PAGES)
ELAPSED TIME: 00:01:37
DBPROC, BBPROC / -SP=C20, 1JP, M, CCET, DBPROC

.MAIN. MA M1110 27-MAR-80 14:29
TABLE OF CONTENTS:

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	1	TASK NAME:	DBUPD-	DATA-BASE-UPDATE.
8-	2			
8-	3			
8-	4			
8-	5			
8-	6			
8-	7	CREATION DATE:		
8-	8	MODIFICATIONS DATE:		
9-	30	TASK IMMEDIATES:		
10-	182	DIRECTIVE PARAMETER BLOCKS:		
10-	189	TASK VARIABLES AND BUFFERS:		
11-	259	TRANSA- TRANSFORMATION TO ASCII TABLE:		
12-	273	INPUT BUFFER CONTROL BLOCK:		
12-	286	FDB DEFINITIONS:		
13-	310	MAIN CONTROL:		
14-	382	DSPTCH:		
15-	426	ADDDOC- ADD DOCUMENT TO DATA BASE:		
16-	571	PREAD- PROCESS READ REQUEST:		
17-	729	PURGED- PURGE DOCUMENT ROUTINE:		
18-	811	RPLDOC OR ADSDOC:		
19-	972	DESDOC - DELETE SUB-DOCUMENT:		
20-	1180	WRAP- WRAP UP AND EXIT:		
20-	1196	OPEACK:		
21-	1222	XMTACK:		
22-	1276	WRITCOM- WRITE COMMON AREA TO DISK:		
24-	1315	ERROR MESSAGES:		

```
1 .SBTTL TASK-NAME: DBUPD- DATA-BASE-UPDATE.  
2 .SBTTL.  
3 .SBTTL.  
4 .SBTTL.  
5 .SBTTL.  
6 .SBTTL.  
7 .SBTTL CREATION-DATE:  
8 .SBTTL MODIFICATIONS: DATE:  
9 .TITLE DBUPD- DATA-BASE-UPDATE  
10 000000 .PSECT:  
11 .ENABL AMA  
12 .NLIST MEB  
13 .MCALL FINIT$,FSRSZ$,FDBDF$,FDAT$,FDRC$,FDBK$,FDOP$A  
14 .MCALL OFNB$,OFNB$,READ$,WAIT$,WRITE$,CLOSE$,DELET$  
15 .MCALL RCVD$,QIOW$,DIR$,EXIT$,WTSE$,CLEF$,SDAT$S  
16 .MCALL RSUM$,SETF$,DECL$,SPL  
17 ;  
18 ;  
19 ; MACRO TO SET THE PURGE BIT IN A SUB-DOCUMENT ID FLAG BYTE.  
20 ;  
21 .MACRO STRTHK,A,?L1  
22 BIC #177700,R2 ;TAKE LOW ORDER 6 BITS  
23 TST R2 ;SUB-DOC START MARK ?  
24 BNE L1 ;BRANCH IF NO  
25 BIS A,(R3) ;SET SUB-DOC PURGE BIT  
26 JMP DES040  
27 LI:  
28 .ENDM
```

DBUPD--.DATA-BASE-UPDATE.
TASK-IMMEDIATES.

MACRO:M1110 27-MAR-88 14:29 BY:SS
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

.SBTTL--TASK IMMEDIATES.

```
30
31
32 000004  ; DPLUN. == 4 ; DATA-BASE-DISK-LUN.
33 000001  ; ILUN. == 1 ; INPUT-LUN.
34 000002  ; OLUN1. == 2 ; OUTPUT-LUN-FOR-RETRIEVED-DOC.
35 000003  ; OLUN2. == 3 ; OUTPUT-LUN-FOR-DATA-BASE-ACK.
36 000004  ; EFLG. == 4 ; EVENT-FLAG-FOR-DATA-BASE-DISK-QIO.
37 000001  ; REF. == 1 ; READ-EVENT-FLAG.
38 000002  ; WEF1. == 2 ; FIRST-WRITE-EVENT-FLAG.
39 000003  ; WEF2. == 3 ; SECOND-WRITE-EVENT-FLAG.
40
41 000001  ; PRGH. == 1 ; PARAGRAPH-MARKER
42 000000  ; SDMK. == 00 ; SUB-DOCUMENT-MARKER.
43 000004  ; SNTC. == 4 ; SENTENCE-MARKER.
44 000003  ; SZMK. == 3 ; SUBZONE-MARKER.
45 000002  ; ZMK. == 2 ; ZONE-MARKER.
46
47 000015  ; DHDRL. == 13. ; DOCUMENT-HEADER-LENGTH.
48 000006  ; LDBAKE. == 6. ; LENGTH-OF-DATA-BASE-ACKNOWLEDGEMENT-ENTRY.
49 002000  ; LNSBUF. == 1024. ; IPR-&-INDEX-BUFFER-LENGTH-IN-BYTES.
50 000044  ; MECIND. == 36. ; MAXIMUM-ENTRY-COUNT-FOR-INDEX-RECORDS.
51 000121  ; MECIPR. == 81. ; MAXIMUM-ENTRY-COUNT-FOR-IPRS.
52 000002  ; NSBUF. == 2. ; #-OF-SECTORS-IN-BUFFER-(***MUST-EVENLY-
53 ; DIVIDE-INTO-DELTA-***)
54 000006  ; N.BIPE. == WORD3 ; #-OF-BYTES/IPR-ENTRY.
55 000016  ; N.BXE. == WORD7 ; #-OF-BYTES/INDEX-ENTRY.
56 000010  ; OFFTS. == WORD4 ; OFFSET-FOR-TEXT-START-IN-PACKED-DOCUMENT.
57 000200  ; R#IPR. == BIT7 ; IPR-RECORD-TYPE.
58 000100  ; R#IND. == BIT6 ; INDEX-RECORD-TYPE.
59 001000  ; SLB. == 512. ; SECTOR-LENGTH-IN-BYTES.
60 000400  ; SLW. == 256. ; SECTOR-LENGTH-IN-WORDS.
61
62 ; PARAMETER-RECEIVE-BLOCK-OFFSETS.
63 ;
64 000000  ; .PSECT. RQOFF,ABS.
65 000000  ; RQ.TNM: .BLKW. 2. ; SENDING-TASK'S-FILE-NAME.
66 000004  ; RQ.CB: .BLKB. 1 ; COMMAND-BYTE.
67 000005  ; RQ.CS: .BLKB. 1 ; COMMAND-SOURCE-BYTE.
68 000006  ; RQ.FID: .BLKW. 2. ; FILE-ID-FOR-DATA-BASE-UPDATE-REQUEST.
69 000012  ; RQ.FVR: .BLKW. 1 ; FILE-VERSION-NUMBER.
70 000014  ; RQ.FNB: .BLKW. 1 ; FILE-NUMBER.
71
72 ; PROCESSED-COMMAND-REQUESTS-OFFSETS.
73 ;
74 000000  ; .PSECT. PCROFF,ABS.
75 000000  ; PC.DBT: .BLKW. 1 ; DATA-BASE-EXCHANGE-TYPE.
76 000002  ; PC.PAD: .BLKW. 1 ; PAD-FILL.
77 000004  ; PC.TYP: .BLKW. 1 ; REQUEST-TYPE.
78 000006  ; PC.STW: .BLKW. 1 ; STATUS-WORD-FOR-ACTION-REQUESTED.
79 000010  ; PC.IDH: .BLKW. 1 ; HIGH-ORDER-WORD-OF-DOC-ID.
80 000012  ; PC.IDM: .BLKW. 1 ; MIDDLE-PORTION-OF-DOC-ID.
81 000014  ; PC.IDL: .BLKW. 1 ; LOW-ORDER-OF-DOC-ID.
82 000016  ; PC.TCC: .BLKW. 2. ; TEXT-CHARACTER-COUNT.
83 000022  ; PC.WSC: .BLKW. 1 ; REQUIRED-WHITE-SPACE-COUNT.
84 000024  ; PC.UUS: .BLKW. 1 ; UNUSED-WHITE-SPACE-COUNT.
85 000026  ; PC.PWC: .BLKW. 2. ; PACKED-WORD-COUNT-(TEXT-&-WHITE-SPACE)
86 000032  ; PC.PTC: .BLKW. 2. ; PACKED-CHARACTER-ONLY-(IN-WORDS)
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA BASE UPDATE
TASK IMMEDIATES

MACRO M1110-22-000-00-0000 PAGE 14
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

87 000036          PC:SPD:: .BLKW 1          ; START OF PACKED DATA
88
89
90
91
92          000010          PR:NID ==          PC.IDH: ;NUMBER OF DOC. IDS IN REQUEST
93          000012          PR:FE  ==          PC.IDM: ;POINTER TO FIRST ENTRY ITEM
94
95
96
97 000000          ; INDEX POINTER RECORD OFFSET DEFINITIONS
98 000000          .PSECT  IPROFF,ABS
99 000001          IP:SAN:: .BLKB 1          ; HIGH ORDER SECTOR ADDRESS # OF IPR
100 000002          IP:TYP:: .BLKB 1          ; RECORD TYPE
101 000004          IP:SAL:: .BLKW 1          ; LOW ORDER SECTOR ADDRESS # OF IPR
102          IP:DLT:: .BLKW 1          ; DELTA FOR # OF INDEX RECORDS REPRESENTED
103          ; -- IN EACH IPR ENTRY
104          IP:CEC:: .BLKW 1          ; CURRENT ENTRY COUNT
105          IP:SRL:: .BLKW 1          ; # OF SECTORS REPRESENTED IN LAST ENTRY
106          IP:HID:: .BLKW 3          ; HIGHEST DOCUMENT ID REPRESENTED
107          IP:BPE:: .BLKW 1          ; BYTES PER IPR ENTRY
108          IP:LIA:: .BLKW 2          ; LAST INDEX RECORD ADDR REPRESENTED
109          IP:PAD:: .BLKW 1          ; PAD FILL
110          IP:FE::          .BLKW 1          ; FIRST ENTRY INDICATOR
111
112          ; INDEX RECORD OFFSET DEFINITIONS
113 000000          .PSECT  INDOFF,ABS
114 000000          IN:SAN:: .BLKB 1          ; HIGH ORDER ADDRESS OF INDEX RECORD
115 000001          IN:TYP:: .BLKB 1          ; RECORD TYPE
116 000002          IN:SAL:: .BLKW 1          ; LOW ORDER ADDRESS OF INDEX SECTOR
117 000004          IN:CEC:: .BLKW 1          ; CURRENT ENTRY COUNT
118 000006          IN:MEC:: .BLKW 1          ; MAXIMUM ENTRY COUNT
119 000010          IN:FE::          .BLKW 1          ; FIRST ENTRY
120
121          ; INDEX RECORD ENTRY OFFSET DEFINITIONS
122          .PSECT  INEOFF,ABS
123 000000          I:BDH:: .BLKB 1          ; BINARY DOCUMENT ID HIGH ORDER
124 000000          I:PGI:: .BLKB 1          ; BYTE CONTAINING PURGE INDICATOR
125 000001          I:BDM:: .BLKW 1          ; BINARY DOCUMENT ID MIDDLE ORDER
126 000002          I:BDL:: .BLKW 1          ; BINARY DOCUMENT ID LOW ORDER
127 000004          I:SANH:: .BLKB 1          ; HIGH ORDER SECTOR ADDRESS NUMBER
128 000006          I:BC::          .BLKB 1          ; WORD COUNT WITHIN SECTOR
129 000007          I:SANL:: .BLKW 1          ; LOW ORDER SECTOR ADDRESS NUMBER
130 000010          I:RWS:: .BLKW 1          ; REQUIRED WHITE SPACE
131 000012          I:UWS::          .BLKW 1          ; UNUSED WHITE SPACE
132 000014
133
134          ; DATA BASE ACKNOWLEDGEMENT RECORD OFFSET DEFINITIONS
135          .PSECT  ACKOFF,ABS
136 000000          AK:TYP:: .BLKW 1          ; EXCHANGE TYPE
137 000000          AK:PAD:: .BLKW 1          ; PAD WORD
138 000002          AK:CNT:: .BLKW 1          ; WORD COUNT OF FOLLOWING DATA
139 000004          AK:FE::          .BLKW 1          ; FIRST ENTRY
140 000006
141
142          ; RETRIEVED DOCUMENT RETURN RECORD OFFSET DEFINITIONS
143
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD--BASE UPDATE
TASK IMMEDIATE

MACRO:M1110 27-MAR-88 14:29 P. 9-2
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
144 000000 .PSECT . RETOFF,ABS.
145 000000 RT,TYP:: .BLKW 1 ; EXCHANGE TYPE.
146 000002 RT,PAD:: .BLKW 1 ; PAD FILL.
147 000004 RT,WSC:: .BLKW 1 ; # OF WHITE SPACE CHARACTERS (BYTES SWAPPED)
148 000006 RT,IDH:: .BLKW 1 ; HIGH ORDER DOC. ID (BYTES SWAPPED)
149 000010 RT,IDM:: .BLKW 1 ; MIDDLE ORDER DOC. ID (BYTES SWAPPED)
150 000012 RT,IDL:: .BLKW 1 ; LOW ORDER DOC. ID (BYTES SWAPPED)
151 000014 RT,TCH:: .BLKW 1 ; TEXT CHARACTER COUNT-HIGH ORDER (BYTES SWAPPED)
152 000016 RT,TCL:: .BLKW 1 ; TEXT CHARACTER COUNT-LOW ORDER (BYTES SWAPPED)
153 000020 RT,FC:: .BLKW 1 ; FIRST CHARACTER OF DOCUMENT TEXT.
154 000000 .PSECT.
155 ;
156 ; BIT DEFINITIONS FOR ACTION REQUESTED WORDS.
157 ;
158 000001 AR,RDC == 1 ; READ DOCUMENT.
159 000002 AR,PDC == 2 ; PURGE DOCUMENT.
160 000003 AR,DSD == 3 ; DELETE SUB-DOCUMENT.
161 000004 AR,ODC == 4 ; OVERLAY DOCUMENT CHARACTERS.
162 000005 AR,RPD == 5 ; REPLACE EXISTING DOCUMENT.
163 000006 AR,ADD == 6 ; ADD NEW DOCUMENT.
164 000007 AR,ASD == 7 ; ADD NEW SUB-DOCUMENT.
165 ;
166 ; BIT DEFINITIONS FOR ACTION STATUS WORDS.
167 ;
168 000002 AS,URD == 2 ; UNABLE TO READ DISK.
169 000003 AS,UWD == 3 ; UNABLE TO WRITE TO DISK.
170 000004 AS,COO == 4 ; CHARACTER OVERLAY OVERFLOW.
171 000005 AS,IWS == 5 ; INSUFFICIENT WHITE SPACE.
172 000006 AS,NRD == 6 ; NO ROOM ON DISK TO DO UPDATE.
173 000007 AS,SDN == 7 ; SUB-DOCUMENT NON-EXISTENT.
174 000010 AS,DNE == 8 ; DOCUMENT NON-EXISTENT.
175 000011 AS,IVI == 9 ; INVALID ID.
176 000012 AS,IVR == 10 ; INVALID REQUEST.
177 000014 AS,FAT == 12 ; FATAL ERROR.
178 000001 AS,SUP == 1 ; SUCCESSFUL UPDATE.
179 ;
180 000000 .PSECT.
```

DBUPD-- DATA-BASE-UPDATE
DIRECTIVE-PARAMETER-BLOCKS

MACRO M1110 27-MAR-80 14:29 PAGE 10
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
182 .SBTTL DIRECTIVE-PARAMETER-BLOCKS
183 ;
184 ; QIO-PARAMETER-BLOCK-TO-DATA-BASE-DISK
185 ;
186 000000 DQIO:: QIOW# ,DPLUN,EFLG,,IOST,<,<,<,>>
187 ;
188 ;
189 .SBTTL TASK-VARIABLES-AND-BUFFERS
190 ;
191 000030 000000 000000 IOST:: .WORD 0.0 ; I/O-STATUS-WORD-FOR-QIOS-TO-DB-DISK
192 000034 000000 000000 RIOST:: .WORD 0.0 ; READ-I/O-STATUS-WORD
193 000040 000000 000000 WIOST:: .WORD 0.0 ; WRITE-I/O-STATUS
194 000044 000000 000000 WIOST1:: .WORD 0.0 ; WRITE-I/O-STATUS-WORD-FOR-RET. DOC.
195 000050 000000 000000 WIOST2:: .WORD 0.0 ; WRITE-I/O-STATUS-WORD-FOR-ACK
196 ;
197 000054 000000 ACKECT:: .WORD 0 ; ACK-WORD-ENTRY-COUNT
198 000056 000000 ACKPTR:: .WORD 0 ; ACK-RECORD-POINTER-TO-NEXT-AVAILABLE-ENTRY
199 000060 000000 ACKSCT:: .WORD 0 ; ACK-SECTOR-COUNT (NUMBER-OF-ACK-SECTORS-WRITTEN)
200 000062 000000 ADDRID:: .WORD 0 ; ADDR-OF-DOC. ID- IN- INPUT-BUFFER
201 000064 000000 BUFADR:: .WORD 0 ; INPUT-BUFFER-START-ADDRESS
202 000066 011507 103140 CCOUT:: RAD50 /CCOUT/ ; RAD50-NAME-FOR-CONTROL-COMPUTER-OUT-ROUTINE
203 000072 000000 CMPST:: .WORD 0 ; RANGE-COMPARE-STATUS-WORD
204 000074 000116 020000 C002:: .WORD 16.020000 ; 2% DISK-CAPACITY-LEFT-IN-BYTES
205 000100 000012 DELTA:: .WORD 10 ; DELTA-REPRESENTED-BY-IPR-ENTRY
206 000102 000000 000000 DOCADR:: .WORD 0.0 ; ADDRESS-OF-CURRENT-DOCUMENT
207 000106 000000 000000 000000 DOCID:: .WORD 0.0.0 ; DOCUMENT-ID
208 000114 000000 DRSTAT:: .WORD 0 ; DIRECTIVE-RESULT-STATUS
209 000116 000000 ENDBFA:: .WORD 0 ; END-OF-BUFFER-ADDR
210 000120 001000 EOC:: .WORD 001000 ; END-OF-COLLECTION-MARKER
211 000122 000000 ETRYCT:: .WORD 0 ; IPR-ENTRY-COUNTER
212 000124 000000 IBFADR:: .WORD 0 ; INDEX-BUFFER-ADDRESS
213 000126 000000 165140 IBLK:: .WORD 0,165140 ; INITIALIZATION-BLOCK-ADDRESS
214 000132 000000 000000 IDXADR:: .WORD 0.0 ; BEGINNING-ADDR-OF-INDEX-DELTA-BLOCK
215 000136 000000 INCOUT:: .WORD 0 ; INPUT-BUFFERS-TO-BE-READ-COUNT
216 000140 000000 000000 NDCADR:: .WORD 0.0 ; ADDRESS-OF-NEXT-DOCUMENT-FROM-CURRENT-DOC.
217 000144 000000 NDOC:: .WORD 0 ; NUMBER-OF-DOCUMENTS
218 000146 000000 NCHNVT:: .WORD 0 ; NO-CONVERT-FLAG-FOR-UNPACKING-ROUTINE
219 000150 000000 HSREAD:: .WORD 0 ; #-OF-SECTORS-READ-INTO-BUFFER
220 000152 000000 NS2RD:: .WORD 0 ; NUMBER-OF-SECTORS-TO-READ
221 000154 000000 000000 PAR1:: .WORD 0.0 ; PARAMETER-STRING-LIST-FOR-MOUT
222 000160 000000 000000 PCCT:: .WORD 0.0 ; PACKED-WORD-COUNT (TEXT-ONLY)
223 000164 000000 000000 PWCT:: .WORD 0.0 ; PACKED-WORD-COUNT (TEXT-AND-WHITE-SPACE)
224 000170 000000 R0STAT:: .WORD 0 ; READ-RESULT-STATUS-WORD
225 ;
226 ; (-1 - ERROR-SEQ. FROM-DBPROC)
227 ; (-2 - ERROR-SEQ. FROM-CCIN)
228 ; (-3 - ERROR-INTERNAL-TO-DBUPD)
228 000172 000000 R0WS:: .WORD 0 ; REQUIRED-WHITE-SPACE
229 000174 000000 UNWS:: .WORD 0 ; UNUSED-WHITE-SPACE (PACKED-WORDS)
230 000176 000000 R3:: .WORD 0 ; ADDITIONAL-WORKING-VARIABLE
231 000200 000000 R3:: .WORD 0 ; ADDITIONAL-WORKING-VARIABLE
232 000202 000000 SAVEPT:: .WORD 0 ; SAVE-AREA
233 000204 000000 000000 EDT:: .WORD 0.0 ; DISK-ADDRESS-OF-END-OF-TEXT
234 000210 000000 000000 TEMP:: .WORD 0.0 ; TEMPORARY-2-WORD-WORKING-AREA
235 000214 000000 XTEMP:: .WORD 0 ; SAVE-AREA-FOR-DOCS-INDEX-ENTRY-ADDR
236 000216 000000 000000 UPCCT:: .WORD 0.0 ; UNPACKED-CHAR. COUNT-EXPECTED-FROM-DOC.
237 000222 000000 WDOFF:: .WORD 0 ; WORD-OFFSET-FROM-BEGINNING-OF-DOC. SECTOR
238 000224 000000 WDOFND:: .WORD 0 ; WORD-OFFSET-FOR-NEXT-DOCUMENT
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD--DB BASE UPDATE
TASK VARIABLES AND BUFFERS

MACRO: M1110 27-MAR-80 14:29 P 10-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

239 000226 000000
240
241
242 000230
243 001230
244 003230
245
246
247
248 003266 000000
249 003270 000000
250 003272 000000
251 003274 000000
252 003276 000000
253 003300 000000
254
255 003302
256 004302
257 005302

WRTST: .WORD 0 ; WRITE DATA BASE STATUS RESULTS
;
;
DBACKB: .BLKW 256 ; DATA BASE ACKNOWLEDGEMENT BUFFER
IPRBUF: .BLKW 512 ; INDEX POINTER AND INDEX RECORD BUFFERS
PRB: .BLKW 15 ; PARAMETER RECEIVE BLOCK
;
; TEMPORARY BUFFER FOR 6-WORD ACK ENTRY
;
ACKBUF: .WORD 0 ; DOC ID (HI)
.WORD 0 ; DOC ID (MID)
.WORD 0 ; DOC ID (LO)
.WORD 0 ; PAD WORD
RQSTWD: .WORD 0 ; ACTION REQUESTED
STATWD: .WORD 0 ; STATUS WORD
;
RDBUF: .BLKW 256 ; DATA BASE DISK READ BUFFER
RTDBUF: .BLKW 256 ; RETRIEVED DOCUMENT OUTPUT BUFFER
WBUF: .BLKW 256 ; WORKING BUFFER

DBUPD-- DATA BASE UPDATE
TRANSA- TRANSFORMATION TO ASCII TABLE

MACRO M1110 27 MAR 88 14 00 PAGE 11

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

259                                     .SBTTL TRANSFORMATION TO ASCII TABLE
260                                     ;
261                                     ; 6 BIT PACKED CHARACTER TO 7 OR 8 BIT ASCII
262                                     ;
263 006302 TRANSA:
264 006302 000 021 022 .BYTE 0,21,22,23,24,40,41,42 ; 00-07
    006305 023 024 040
    006310 041 042
265 006312 043 045 046 .BYTE 43,45,46,47,50,51,52,53 ; 10-17
    006315 047 050 051
    006320 052 053
266 006322 054 055 056 .BYTE 54,55,56,57,60,61,62,63 ; 20-27
    006325 057 060 061
    006330 062 063
267 006332 064 065 066 .BYTE 64,65,66,67,70,71,75,101 ; 30-37
    006335 067 070 071
    006340 075 101
268 006342 102 103 104 .BYTE 102,103,104,105,106,107,110,111 ; 40-47
    006345 105 106 107
    006350 110 111
269 006352 112 113 114 .BYTE 112,113,114,115,116,117,120,121 ; 50-57
    006355 115 116 117
    006360 120 121
270 006362 122 123 124 .BYTE 122,123,124,125,126,127,130,131 ; 60-67
    006365 125 126 127
    006370 130 131
271 006372 132 133 134 .BYTE 132,133,134,135,136,72,73,77 ; 70-77
    006375 135 136 072
    006400 073 077
```


DBUPD-- DATA-BASE UPDATE
INPUT-BUFFER CONTROL BLOCK

MACRO M1110 27-MAR-88 14:29 P. 12
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

273
274
275
276 006402
277 006402 000000G
278 006404 000002
279 006406 006410
280 006410 000000G
281 006412 000004
282 006414 006402
283
284
285
286
287
288
289 006416
290
291
292
293
294 006416
295 006416
296 006556
297 006556
298 006556
299
300
301
302
303 006556
304 006556
305 006716
306 006716
307 006716
308

```
.SBTTL INPUT-BUFFER-CONTROL-BLOCK  
:  
:  
BITTST: ; FIRST-DOUBLE-BUFFER-CONTROL-BLOCK  
    .WORD HRL0  
    .WORD BIT1  
    .WORD .+2  
    .WORD DHR0  
    .WORD BIT2  
    .WORD BITTST  
:  
:  
:  
.SBTTL FDB-DEFINITIONS  
.NLIST MEB  
:  
FIRS2$ 0 ;  
:  
:  
RETRIEVED-DOCUMENT-SPOOL-FILE-FDB  
:  
RTDFDB:  
    FDBDF$  
    FDRC$A FD,RWM  
    FDBK$A RTDBUF,SLB,.WEF1,WI0ST1  
    FDOP$A OLUN1  
:  
:  
:  
DATA-BASE-ACKNOWLEDGEMENT-SPOOL-FILE-FDB  
:  
ACKFDB:  
    FDBDF$  
    FDRC$A FD,RWM  
    FDBK$A DBACKB,SLB,.WEF2,WI0ST2  
    FDOP$A OLUN2  
:  
:  
:
```

DBUPD--DATA BASE UPDATE
MAIN CONTROL

MACRO: M1110 27-MAR-80 14:29 PAGE 13
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

          .SBTTL--MAIN CONTROL
310
311
312 006716          ; DBUPD:
313 006716          FINIT$          ; INITIALIZE FCS
314 006722          CALL OPEACK          ; INITIALIZE ACK SPOOL FILE
315
316
317          ; RECEIVE GLOBAL BUFFER IDENTIFICATION FROM CCIN
318 006726          DBU005:
319 006726          RCVDS$ ,#PRB          ; RECEIVE PRB
320 006744 103032    BCC 5$          ; SUCCESSFUL RECEIVE
321 006746 123727 000000G 000000G    CMPB #DSW,#IE,ITS          ; NO DATA QUEUED?
322 006754 001006    BNE 55$          ; BRANCH IF OTHER ERROR
323
324          ; WAIT FOR CCIN TO SIGNAL THAT ANOTHER RCVD PACKET (PRB) IS QUEUED
325          ; FOR DBUPD
326
327 006756          WTSE$# ,#CF,UPD          ; WAIT FOR SINGLE EVENT FLAG
328 006770 103011    BCC 6$          ; GOOD
329 006772          55$:
330 006772 013737 000000G 000154'    MOV #DSW,PAR1          ;
331 007000 012701 014424'    MOV #EMSG1,R1          ;
332 007004          CALL ERROR          ;
333 007010          CALL EXIT          ;
334
335          ; CCIN HAS SENT SOMETHING
336
337 007014          6$:
338 007014          CLEF$# ,#CF,UPD          ; CLEAR GLOBAL EVENT FLAG
339 007026 103337    BCC DBU005          ; RECEIVE DATA NOW
340 007030 000760    BR 55$          ;
341
342          ; GOT RCVD PACKET
343
344 007032          5$:
345 007032 005037 003276'    CLR RQSTWD          ; CLEAR ACTION REQUESTED WORD
346 007036 023727 000000C 000001    CMP PRB+RQ,CB,#1          ; END OF BATCH?
347 007044 001020    BNE 10$          ; NO
348
349          ; END OF UPDATE BATCH
350
351 007046 004737 013644'    JSR PC.WRAP          ; YES, CLOSE ACK & SEND IT
352 007052 052737 020000 000000G    BIS #BIT13,MSQ          ; NOTIFY CCOUT THAT UPDATE IS DONE
353 007060          SETF$# ,#CF,COT          ; WAKE UP CCOUT
354 007072          DECL$#          ;
355 007100          EXIT$#          ; EXIT
356
357          ; READ PROCESSED REQUEST
358
359 007106          10$:
360 007106 004737 000000G    JSR PC.READ          ; READ THE PROCESSED REQUEST
361 007112 005737 000170'    TST RDSTAT          ; ANY PROBLEMS WITH READ?
362 007116 001406    BEQ 4$          ; NO
363 007120 012701 014450'    MOV #EMSG15,R1          ; ERROR FATAL TO DBUPD
364 007124          CALL ERROR          ;
365 007130          CALL EXIT          ;
366 007134 022762 000001 000006 4$:    CMP #AS,SUP,PC,STW(C2)          ; MINIMAL COMMAND RECORD?

```

DBUPD-- DATABASE UPDATE
MAIN CONTROL

MACRO M1110 27 MAR 88 11 29 PM '87 17-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
367 007142 001406          BEQ 7#          ;BRANCH IF NO
368                          ;
369                          ; MINIMAL COMMAND RECORD HAS BEEN RECEIVED FROM DBPROC INDICATING
370                          ; THAT REQUESTED UPDATE CANNOT BE PROCESSED BY DBPROC CREATE
371                          ; NACK ENTRY AND GO TO NEXT UPDATE REQUEST
372                          ;
373 007144          CALL SDATA          ;STORE AND SAVE NEEDED DATA
374 007150          CALL CACKE          ;CREATE NACK ENTRY
375 007154 000137 006726'    JMP DBU005          ;DO NEXT EXCHANGE
376                          ;
377                          ; PROCESS THE REQUEST
378                          ;
379 007160 004737 007170'    7#: JSR PC,DSPTCH          ;DISPATCH THE EXCHANGE
380 007164 000137 006726'    JMP DBU005          ;DO NEXT EXCHANGE
```

DBUPD-- DATA BASE UPDATE
DSPTCH:

MACRO.M1110 27-MAR-80 14:29 PAGE 14
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
382 .          .SBTTL- DSPTCH-
383          ;
384          ; DISPATCH ROUTINE
385          ;
386 007170 012701 007244* DSPTCH: MOV #EXCHTP,R1 ;R1->DISPATCH TABLE
387 007174 012700 000006 MOV #TABLSZ,R0 ;R0=TABLE SIZE
388          ;
389 007200 022162 000004 1$: CMP (R1)+,PC,TYP(R2) ;CHECK FOR TYPE MATCH
390 007204 001003 BNE 2$ ;BRANCH IF NO MATCH
391 007206 012137 003276* MOV (R1)+,R0STWD ;SAVE ACTION REQUEST CODE
392 007212 000131 JMP @ (R1)+ ;GOT MATCH - GO TO ROUTINE
393          ;
394 007214 062701 000004 2$: ADD #4,R1 ;R1->NEXT ENTRY
395 007220 SSQB R0,1$ ;LOOP
396          ;
397          ; INVALID REQUEST IS DETECTED -- SET UP FOR CALL TO CREATE ACK ENTRY
398          ;
399 007224 012737 000012 003300* MOV #AS,IVR,STATWD ;SET INVALID REQUEST BIT
400 007232 005037 003276* CLR R0STWD ;NO KEY FOR INVALID REQUEST
401 007236 004737 000000G JSR PC,CACKE ;CREATE ACK ENTRY
402 007242 000207 RTS PC ;RETURN
403          ;
404          ; DB UPDATE REQUEST DISPATCH TABLE
405          ;
406 007244 104 103 EXCHTP: .ASCII /DC/ ;ADD NEW DOCUMENT
407 007246 000006 .WORD AR,ADD ;ACTION REQUEST CODE
408 007250 007310* .WORD ADDDOC ;POINTER TO SUBROUTINE
409 007252 104 122 .ASCII /DR/ ;READ DOC
410 007254 000001 .WORD AR,RDC
411 007256 010112* .WORD PREAD
412 007260 122 120 .ASCII /RP/ ;PURGE DOC
413 007262 000002 .WORD AR,PDC
414 007264 011210* .WORD PURGED
415 007266 125 120 .ASCII /UP/ ;MODIFY-REPLACE
416 007270 000005 .WORD AR,RPD
417 007272 011616* .WORD RPLDOC
418 007274 101 123 .ASCII /AS/ ;ADD SUB-DOCUMENT
419 007276 000007 .WORD AR,ASD
420 007300 011616* .WORD ADSDOC
421 007302 104 123 .ASCII /DS/ ;DELETE SUB-DOCUMENT
422 007304 000003 .WORD AR,DSB
423 007306 012420* .WORD DESDOC
424 000006 TABLSZ=<-EXCHTP>/6
```

```

426                                     .SBTTL- ADDDOC- ADD DOCUMENT TO DATA-BASE.
427                                     ;
428 007310                               ADDDOC:
429 007310                               CALL  RDEOC          ;READ LAST BLOCK OF DATA-BASE.
430 007314 103005                         BCC   1$
431 007316 012737 000002 003300'        MOV   #AS.URD,STATWD ;SET ERROR CODE.
432 007324 000137 010104'                JMP   11$            ;CREATE ACK.
433 007330 013737 000000C-000102'      1$:  MOV   SREC+SR,ECH,DOCADR ;ADDRESS OF CURRENT DOC. (HI)
434 007336 013737 000000C-000104'      MOV   SREC+SR,ECL,DOCADR+WORD1 ; " " " (LO)
435                                     ;
436                                     ; SAVE DATA RECEIVED FROM DBPROC IN UPDATE EXCHANGE.
437                                     ;
438 007344 004737 000000G.                JSR   PC,SDATA      ; STORE & SAVE NEEDED DATA.
439                                     ;
440                                     ; VALIDATE ADDRESS OF NEW DOCUMENT.
441                                     ;
442 007350 012701 000106'                MOV   #DOCID,R1     ; R1-> NEW DOCUMENT ID.
443 007354 012702 000000C.                MOV   #SREC+SR,ARS,R2 ; R2-> ACTUAL RANGE START.
444 007360 012703 000000C.                MOV   #SREC+SR,ARE,R3 ; R3-> ACTUAL RANGE END FOR DOC IDS ON DISK.
445 007364 004737 000000G.                JSR   PC,CMP3WD     ; ID OF NEW DOC. MUST BE > LAST ID ON DISK.
446 007370 005737 000072'                TST  CMPST         ; IS IT?
447 007374 003005                         BGT  20$           ; YES.
448 007376 012737 000011 003300'        MOV   #AS.IVI,STATWD ; SET ERROR CODE.
449 007404 000137 010104'                JMP   11$           ; CREATE ACK ENTRY & RETURN.
450                                     ;
451                                     ; CHECK FOR AVAILABLE WHITE SPACE.
452                                     ;
453 007410 013737 000000C-000210'      20$: MOV   SREC+SR,WSL,TEMP ; DETERMINE IF THERE IS ENOUGH WHITE SPACE.
454 007416 013737 000000C-000212'        MOV   SREC+SR,WSL+WORD1,TEMP+WORD1 ; TO ADD ANOTHER DOCUMENT.
455 007424 163737 000166' 000212'        SUB  PWCT+WORD1,TEMP+WORD1 ; SUBTRACT PACKED WORD COUNT.
456 007432 005637 000210'                SBC  TEMP          ;
457 007436 163737 000164' 000210'        SUB  PWCT,TEMP     ;
458 007444 003017                         BGT  4$            ; BRANCH IF UPPER WORD IS POSITIVE.
459                                     ;
460                                     ; NO. CHECK LOWER WORD.
461                                     ;
462 007446 005737 000212'                TST  TEMP+WORD1    ; NO. CHECK LOWER WORD.
463 007452 101014                         BHI  4$            ; YES. THERE IS ROOM TO ADD ANOTHER.
464 007454                                     ;
465 007472 012737 000006 003300'      3$:  MOUT$S. #MSG11     ; DISK FULL - DOCUMENTS NOT BEING ADDED.
466 007500 000137 010104'                MOV   #AS.NRD,STATWD ; SET ERROR CODE.
467                                     ;
468                                     ; GET WORD OFFSET OF EOC.
469                                     ;
470 007504 113701 000000C-                4$:  MOVB  SREC+SR,ECB,R1 ; R1=WORD OFFSET.
471 007510 042701 177400                   BIC  #177400,R1    ; R1 = WORD OFFSET OF EOC.
472 007514 006301                         ASL  R1            ; CONVERT WORD OFFSET TO BYTES.
473 007516 062701 005302'                ADD  #WBUF,R1      ; ADDR TO START ADDING NEW DOC.
474 007522 013702 000064'                MOV  BUFADR,R2    ; R2-> INPUT BUFFER.
475 007526 062702 000036                   ADD  #PC,SPD,R2   ; R2-> START OF PACKED DATA
476                                     ;
477                                     ; PUT PACKED DATA ON DISK.
478                                     ;
479 007532 004737 000000G.                JSR  PC,PPDATA     ; PUT PACKED DATA ON DISK.
480 007536 022737 000001 003300'        CMP  #AS.SUP,STATWD ; CONTINUE?.
481 007544 001423                         BEQ  6$            ; YES.
482                                     ;

```

```

483 ; ERROR, RESET END OF COLLECTION.
484 ;
485 007546 5$: CALL RSTEOC.
486 ;
487 ; TEST NATURE OF ERROR.
488 ;
489 007552 022737 000002 000170* CMP #2,RDSTAT ;ERROR SEQ. FROM CCIN?.
490 007560 001414 BEQ 12$ ;BRANCH IF YES.
491 ;
492 ; ERROR SEQUENCE FROM DBPROC, OR INTERNAL ERROR, CREATE NACK ENTRY.
493 ;
494 007562 004737 000000* JSR PC,CACKE ; CREATE ACK ENTRY
495 ;
496 ; IF ERROR SEQUENCE IS FROM DBPROC, CONTINUE READING BUFFERS FROM
497 ; CCIN UNTIL ALL REMAINING BUFFERS ARE READ.
498 ;
499 007566 022737 000001 000170* CMP #1,RDSTAT ;ERR SEQ. FROM DBPROC?.
500 007574 001006 BNE 12$ ;BRANCH IF NO.
501 007576 005737 000136* 13$: TST INCOU ;MORE BUFFERS?.
502 007602 003403 BLE 12$ ;BRANCH IF NO.
503 007604 CALL READ ;READ ANOTHER BUFFER.
504 007610 000772 BR 13$
505 007612 000207 12$: RTS PC ; RETURN
506 ;
507 ; CREATE INDEX ENTRY FOR NEW DOC.
508 ;
509 007614 004737 000000* 6$: JSR PC,CINDXE ; CREATE INDEX ENTRY FOR NEW DOC.
510 007620 022737 000001 003300* CMP #AS,SUP,STATWD ; ANY PROBLEMS?.
511 007626 001347 BNE 5$ ; YES, CREATE ACK ENTRY.
512 007630 004737 000000* JSR PC,CIPRE ; NO, CREATE IPR ENTRY IF NEEDED.
513 007634 022737 000001 003300* CMP #AS,SUP,STATWD ; ANY PROBLEMS?.
514 007642 001341 BNE 5$ ; YES, CREATE ACK ENTRY.
515 ;
516 007644 SAVE R0,R1,R2,R3
517 007654 013702 000160* MOV PCCT,R2 ;R2,R3 = WORD LENGTH OF PACKED TEXT.
518 007660 013703 000162* MOV PCCT+2,R3
519 007664 012700 000002 MOV #2,R0 ;R0 = MULTIPLIER.
520 007670 CALL $DNUL ;R0,R1 = BYTE LENGTH OF PACKED TEXT.
521 007674 160137 000000* SUB R1,SREC+SR,TWS+2 ;SUBTRACT FROM BYTE LENGTH OF
522 007700 005637 000000* SBC SREC+SR,TWS ; TOTAL W.S.
523 007704 160037 000000* SUB R0,SREC+SR,TWS
524 ;
525 007710 013702 000164* MOV PWCT,R2 ;R2,R3 = WORD LENGTH OF PACKED TEXT+
526 007714 013703 000166* MOV PWCT+2,R3 ; W.S.
527 007720 012700 000002 MOV #2,R0 ;R0 = MULTIPLIER.
528 007724 CALL $DNUL ;R0,R1 = BYTE LENGTH OF PACKED TEXT
529 ; AND W.S.
530 007730 160137 000000* SUB R1,SREC+SR,WSL+2 ;SUBTRACT FROM W.S. AFTER EOC.
531 007734 005637 000000* SBC SREC+SR,WSL
532 007740 160037 000000* SUB R0,SREC+SR,WSL
533 007744 RESTOR R0,R1,R2,R3
534 ;
535 007754 023737 000000*000074* CMP SREC+SR,WSL,C002 ; GETTING LOW ON DISK SPACE?.
536 007762 003006 BGT 7$ ; NO.
537 007764 023737 000000*000076* CMP SREC+SR,WSL+WORD1,C002+WORD1 ; POSSIBLY CHECK LOWER WORDS.
538 007772 101002 BHI 7$ ; NO.
539 007774 000137 007454* JMP 7$ ; DISK FULL - DOCUMENTS NOT BEING ADDED

```

```
540 ;  
541 ; DOCUMENT ADDED TO DISK SUCCESSFULLY.  
542 ;  
543 010000 005737 000000C. 7$: TST SREC+SR,NDC+WORD1 ; ANY DOCUMENTS ON DISK?  
544 010004 001014 BNE 40$ ; YES  
545 010006 005737 000000C. TST SREC+SR,NDC ; MAYBE, CHECK LOWER WORD TO BE SURE  
546 010012 001011 BNE 40$ ; YES, THERE ARE DOCUMENTS.  
547 ;  
548 ; PLACE FIRST DOCUMENT ID IN ACTUAL RANGE START.  
549 ;  
550 010014 013737 000106' 000000C- MOV DOCID,SREC+SR,ARS  
551 010022 013737 000110' 000000C- MOV DOCID+WORD1,SREC+SR,ARS+WORD1 ;  
552 010030 013737 000112' 000000C- MOV DOCID+WORD2,SREC+SR,ARS+WORD2 ;  
553 ;  
554 ; PLACE DOC ID IN ACTUAL RANGE END.  
555 ;  
556 010036 013737 000106' 000000C-40$: MOV DOCID,SREC+SR,ARE  
557 010044 013737 000110' 000000C- MOV DOCID+WORD1,SREC+SR,ARE+WORD1 ;  
558 010052 013737 000112' 000000C- MOV DOCID+WORD2,SREC+SR,ARE+WORD2 ;  
559 010060 062737 000001 000000C- ADD #1,SREC+SR,NDC+WORD1 ; INCREMENT # OF DOCUMENTS COUNT  
560 010066 005537 000000C- ADC SREC+SR,NDC ; COUNT FOR CARRY  
561 010072 10$: JSR PC,CACKE ; CREATE ACK ENTRY  
562 010072 004737 000000G JSR PC,WRTCOM ; WRITE COMMON AREA TO INITIALIZATION BLOCK OF DISK  
563 010076 004737 014262' RTS PC ; RETURN  
564 010102 000207  
565 ;  
566 ; CREATE ACK ENTRY FOR ERROR CONDITION  
567 ;  
568 010104 11$: CALL CACKE  
569 010110 000207 RTS PC
```

DBUPD- DATA BASE UPDATE
PREAD- PROCESS READ REQUEST

MACRO: M1110, 27 MAR 68, 14:25 PAGE 16
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
571 .SBTTL- PREAD- PROCESS READ REQUEST.  
572 ;  
573 ; PREAD WILL READ THE NUMBER OF DOCUMENTS REQUESTED. THESE DOCUMENTS WILL  
574 ; BE READ, UNPACKED AND WRITTEN TO A RETRIV.DOC FILE ONE DOCUMENT AT A TIME.  
575 ; AS EACH DOCUMENT IS COMPLETELY READ AND UNPACKED, THE FILE ID WILL BE SENT  
576 ; TO CCOUT. AFTER THE LAST DOCUMENT IS DONE, CONTROL RETURNS TO THE MAIN  
577 ; PROGRAM FOR THE NEXT UPDATE REQUEST.  
578 ;  
579 ;  
580 010112. PREAD:  
581 010112. 012737 000001 003300' MOV. #AS,SUP,STATWD ;SHOW SUCCESS IN STATUS WORD  
582 010120 013702 000064' MOV. BUFADR,R2 ; BUFFER START ADDR.  
583 010124 016237 000010 000144' MOV. PR,NID(R2),NDOC ; # OF DOCUMENTS TO READ.  
584 010132 010237 000062' MOV. R2,ADDRID ;  
585 010136 062737 000012 000062' ADD. #PR,FE,ADDRID ; ADDR OF FIRST DOC. ID ENTRY.  
586 010144  
587 010144 013701 000062' PRD010:  
588 010150 011137 003266' MOV. ADDRID,R1 ; ADDR OF DOC. ID.  
589 010154 016137 000002 003270' MOV. (R1),ACKBUF ;SAVE DOC ID FOR ACK.  
590 010162 016137 000004 003272' MOV. 2(R1),ACKBUF+2  
591 010170 012702 000000C MOV. 4(R1),ACKBUF+4  
592 010174 012703 000000C MOV. #SREC+SR,ARS,R2 ; ADDR OF ACTUAL DOC. ID RANGE START.  
593 010200 004737 000000G MOV. #SREC+SR,ARE,R3 ; ADDR OF ACTUAL DOC. ID RANGE END.  
594 010204 005737 000072' JSR. PC,CMP3WD ; DOC ID WITHIN RANGE FOR THIS SU?  
595 010210 001405 TST. CMPST ;  
596 010212 012737 000011 003300' BEQ. PRD015 ; YES.  
597 010220 000137 011140' MOV. #AS,IVI,STATWD ; NO, SET STATUS WORD FOR ACK ENTRY.  
598 010224 JNP. PRD045 ;  
599 010224 004737 000000G PRD015:  
600 010230 022737 000001 003300' JSR. PC,LIPRE ; LOCATE APPROPRIATE IPR ENTRY.  
601 010236 001402C CMP. #AS,SUP,STATWD ;EVERYTHING OK?  
602 010240 000137 011140' BEQ. PRD020 ;BRANCH IF YES.  
603 010244 JNP. PRD045 ; CREATE ACK ENTRY AND GET NEXT ID.  
604 010244 004737 000000G PRD020:  
605 010250 022737 000001 003300' JSR. PC,CDSTAT ; CALCULATE DOCUMENT STATISTICS.  
606 010256 001402C CMP. #AS,SUP,STATWD ;EVERYTHING OK?  
607 010260 000137 011140' BEQ. PRD022 ;BRANCH IF YES.  
608 010264 JNP. PRD045 ;  
609 010264 012700 006416' PRD022:  
610 010270 012701 000014 MOV. #RTDFDB,R0 ; FDB ADDRESS FOR RETRIEVED DOC. FILE.  
611 010274 004737 000000G MOV. #FN,RDC,R1 ; RETRIEVED DOC. FILE NUMBER  
612 010300 JSR. PC,BLDNFL ; CREATE RETRIEVED DOC. OUTPUT FILE.  
613 010320 FDATSR. #RTDFDB,....#-4, #-4, ;  
614 010336 OFNB$W. #RTDFDB ;  
615 010340 103012 BCC. PRD025 ; SUCCESSFUL OPEN.  
616 010344 010437 000154' MOV. F,ERR(R0),R4 ; ERROR.  
617 010350 012701 014420' MOV. R4,PAR1 ;  
618 010354 CALL. #MSG0,R1 ;  
619 010360 000137 010526' JNP. PRD020 ; DELETE OUTPUT FILE & GET NEXT ID.  
620 010364  
621 010364 012700 004302' PRD025:  
622 010370 012700 051124 000000 MOV. #RTDBUF,R0 ; ADDR OF RETRIEVED DOCUMENT BUFFER.  
623 010376 013700 000174' 000004 MOV. #*TR,RT,TYP(R0) ; EXCHANGE TYPE.  
624 010404 000360 000004 MOV. UNUS,RT,WSC(R0) ; UNUSED WHITE SPACE FOR THIS DOCUMENT  
625 010410 013701 000062' SWAB. RT,WSC(R0) ; SWAP BYTES  
626 010414 011160 000006 MOV. ADDRID,R1 ; ADDRESS OF DOCUMENT ID  
627 010420 000360 000006 MOV. (R1),RT,IDH(R0) ; HIGH ORDER DOCUMENT ID  
SWAB. RT,IDH(R0) ; SWAP BYTES FOR ACC LINK INTERFACE.
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD--DBASE UPDATE
PREAD- PROCESS READ REQUEST

MACRO-M1110 27-MAR-80 14:29 P. 16-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
628 010424 016160 000002 000010      MOV. 2(R1),RT,IDL(R0) ; MIDDLE ORDER OF DOCUMENT ID.
629 010432 000360 000010      SWAB. RT,IDL(R0) ;
630 010436 016160 000004 000012      MOV. 4(R1),RT,IDL(R0) ; LOW ORDER OF DOCUMENT ID.
631 010444 000360 000012      SWAB. RT,IDL(R0) ; SWAP BYTES.
632 010450 012737 00000006 000002* PRD029: MOV. #IO,RLB,DQIO+0,IOFN ; READ FUNCTION CODE.
633 010456 012737 0003302' 000014*      MOV. #RDBUF,DQIO+0,IOPL ; READ BUFFER.
634 010464 012737 001000 000016*      MOV. #SLB,DQIO+0,IOPL+2 ; BUFFER LENGTH.
635 010472 013737 000102' 000022*      MOV. DQADR,DQIO+0,IOPL+6 ; ADDR OF DOCUMENT BEING READ.
636 010500 013737 000104' 000024*      MOV. DQADR+WORD1,DQIO+0,IOPL+8 ;
637 010506 004737 00000006      JSR. PC,ISSDIR ; ISSUE READ.
638 010512 005737 000114*      TST. DRSTAT ; GOOD READ?
639 010516 001411      BEQ. PRD030 ; YES.
640 010520 012737 000002 003300*      MOV. #AS.URD,STATWD ; SET READ ERROR BIT.
641 010526
642 010526 012700 006416*      PRD028: MOV. #RTDFDB,R0 ; FDB ADDR.
643 010532 004737 00000006      JSR. PC,DLFNB ; DELETE OUTPUT FILE.
644 010536 000137 011140*      JMP. PRD045 ; GET NEXT ID.
645 010542
646 010542 123727 000030' 00000006      PRD030: CMPB. IOST,#IS,SUC ; SUCCESSFUL READ?
647 010550 001414      BEQ. PRD035 ; YES.
648 010552 013737 000030' 000154*      MOV. @#IOST,PAR1 ; READ ERROR.
649 010560 012701 014430*      MOV. #EMSG2,R1
650 010564      CALL. ERROR.
651 010570 012737 000002 003300*      MOV. #AS.URD,STATWD ; SET ERROR BIT.
652 010576 000137 010526*      JMP. PRD028 ; DELETE OUTPUT FILE & GET NEXT ID.
653 010602
654 010602 005337 000152*      PRD035: DEC. NS2RD ; DECREMENT # OF SECTORS TO READ COUNT.
655 010606 012703 003302*      MOV. #RDBUF,R3 ; INITIALIZE INPUT REGISTER.
656 010612 013702 000222*      MOV. WDOFF,R2 ; WORD OFFSET OF DOC. START IN 1ST SECTOR.
657 010616 006302      ASL. R2 ; CONVERT TO BYTES.
658 010620 002023      ADD. R2,R3 ; ADDR OF PACKED DOC. START.
659 010622 002703 000010      ADD. #OFFTS,R3 ; ADDR OF 1ST PACKED TEXT CHARACTER.
660
661
662
663 010626 020327 004302*      ; IF R3 POINTS BEYOND END-OF-BUFFER, READ IN NEXT BLOCK.
664 010632 002413      CMP. R3,#CRDBUF+512,>; DOES R3 -> BEYOND BUFFER.?
665 010634 002737 000001 000104*      BLT. 1$ ; BRANCH IF NO.
666 010642 005537 000102*      ADD. #1,DOCADR+2 ; POINT TO NEXT SECTOR ON DISK.
667 010646 162703 004302*      ADC. DQADR.
668 010652 006203      SUB. #CRDBUF+512,>,R3
669 010654 010337 000222*      ASR. R3
670 010660 000673      MOV. R3,WDOFF ;
671      BR. PRD029
672 010662 012704 00000006      1$: MOV. #RTDBUF+RT.FC,R4 ; INITIALIZE OUTPUT BUFFER REGISTER POINTER.
673 010666 004737 00000006      JSR. PC,UNPACK ;
674 010672 022737 000001 003300*      CMP. #AS.SUP,STATWD ; EVERYTHING OK.?
675 010700 001405      BEQ. 2$ ; BRANCH IF YES.
676 010702      DELET$ #RTDFDB ; DELETE RETRIEVED DOC SPOOL FILE.
677 010712 000512      BR. PRD045 ; NACK.
678
679      ; READ FIRST BLOCK, INSERT CHARACTER COUNT, AND REWRITE THE BLOCK.
680
681 010714 012700 006416*      2$: MOV. #RTDFDB,R0 ; ADDR OF FDB.
682 010720 004737 00000006      JSR. PC,READF ; READ THE FIRST BLOCK OF FDB.
683 010724 013737 000210' 00000006      MOV. TEMP,RTDBUF+RT.TCH ; HIGH ORDER TEXT CHARACTER COUNT.
684 010732 000337 00000006      SWAB. RTDBUF+RT.TCH ; SWAP BYTES FOR ACC LINK TRANSFER.
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA-BASE-UPDATE
PREAD-- PROCESS-READ-REQUEST

MACRO M1110 27-MAR-68 14:29 PAGE 16-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
685 010736 013737 000212' 0000000- MOV. TEMP+WORD1,RTDBUF+RT.TCL ; LOW-ORDER-TEXT-CHARACTER-COUNT-
686 010744 000337 0000000- SWAB. RTDBUF+RT.TCL ; SWAP-BYTES-FOR-ACC-LINK-TRANSFER-
687 010750 005060 000064 CLR. F,BKVB(R0) ; ADDR-OF-FIRST-VIRTUAL-BLOCK-
688 010754 012760 000001 000066 MOV. #1,F,BKVB+2(R0) ;
689 010762 004737 000000G JSR. PC,WRTRTD ; WRITE-FIRST-BLOCK-AGAIN-
690 010766 005037 000200' CLR. R9 ;
691 ;
692 ; BUILD-PRB-FOR-CCOUT-
693 ;
694 010772 012700 003230' MOV. #PRB,R0 ;R0->PRB-
695 010776 112760 000001 000005 MOVB. #1,R0,CS(R0) ;COMMAND-SOURCE-
696 011004 112760 000000 000004 MOVB. #0,R0,CB(R0) ;COMMAND-BYTE-
697 011012 013760 005520' 000006 MOV. RTDFDB+F,FNB+N,FID,R0,FID(R0) ;FID-
698 011020 013760 005522' 000010 MOV. RTDFDB+F,FNB+N,FID+2,R0,FID+2(R0)
699 011026 013760 005536' 000012 MOV. RTDFDB+F,FNB+N,FVER,R0,FVR(R0) ;FILE-VERSION-NUMBER-
700 011034 012760 000014 000014 MOV. #FN,RDC,R0,FNB(R0) ;FILE-NUMBER-
701 ;
702 ; CLOSE-OUTPUT-FILE-
703 ;
704 011042. CLOSE$ #RTDFDB-
705 ;
706 ; NOTIFY-CCOUT-
707 ;
708 011052. SDAT$S. #CCOUT,#PRB+4,#CF.COT-
709 011114 103011 BCC. PRD045
710 011116 013737 000000G 000154' MOV. #DSW,PAR1
711 011124 012701 014424' MOV. #EMSG1,R1
712 011130 CALL. ERROR-
713 011134 CALL. EXIT
714 ;
715 ; CALL-ROUTINE-TO-CREATE-ACK-ENTRY-
716 ;
717 011140 005337 000144' PRD045: DEC. NDOC ;DECREMENT-NO. OF-DOC'S-TO-READ-COUNT-
718 011144 004737 000000G JSR. PC,CACKE ; CREATE-ACK-ENTRY
719 011150 005737 000144' TST. NDOC ;MORE-DOC'S-TO-READ-?
720 011154 003410 BLE. PRD050 ;BRANCH-IF-NO-
721 011156 062737 000006 000062' ADD. #WORD3,ADDRID ; GET-ADDR-OF-NEXT-DOC- ID-
722 011164 012737 000001 003300' MOV. #AS,SUP,STATWD ;SET-UP-STATUS-WORD-FOR-NEXT-DOC-
723 011172 000137 010144' JMP. PRD010 ; CONTINUE-
724 011176 PRD050:
725 011176 005037 003276' CLR. RQSTWD ; FOR-NEXT-UPDATE-REQUEST-
726 011202 005037 003300' CLR. STATWD ; FOR-NEXT-UPDATE-REQUEST-
727 011206 000207 RTS. PC ; RETURN
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD--DA...BASE UPDATE.
PURGED- PURGE DOCUMENT ROUTINE

MACRO M1110 27-MAR-80 14:29 P. 17
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL PURGED- PURGE DOCUMENT ROUTINE.
;
; PURGED ROUTINE PROCESSES ONE 1024 WORD BUFFER CONTAINING ONE OR MORE
; DOCUMENT IDS. THE DOCUMENT IS LOCATED, THE FIRST SECTOR OF THE DOCU-
; MENT IS READ, THE PURGE BIT IS SET IN THE DOCUMENT START SEQUENCE AND
; THE FIRST SECTOR IS RE-WRITTEN TO DISK, THE ENTRY IN THE INDEX RECORD
; IS ALSO MODIFIED TO INDICATE THE DOCUMENT HAS BEEN PURGED.
;
729.
730.
731.
732.
733.
734.
735.
736.
737 011210 PURGED:
738 011210 012737 000001 003300* MOV. #AS,SUP,STATWD ; INITIALIZE ACK STATUS WORD.
739 011216 013702 000064* MOV. BUFADR,R2 ;
740 011222 010237 000062* MOV. R2,ADDRID ;
741 011226 062737 000012 000062* ADD. #PR,FE,ADDRID ; POINT TO 1ST DOCUMENT ID IN INPUT BUFFER.
742 011234 016237 000010 000144* MOV. PR,NID(R2),NDOC ; # OF DOCUMENTS TO PURGE.
743 011242. PRG010:
744 011242. 013701 000062* MOV. ADDRID,R1 ; ADDR OF DOCUMENT ID.
745 011246 011137 003266* MOV. (R1),ACKBUF ; SAVE DOC ID FOR ACK.
746 011252 016137 000002 003270* MOV. 2(R1),ACKBUF+2 ;
747 011260 016137 000004 003272* MOV. 4(R1),ACKBUF+4 ;
748 011266 012702 000000C MOV. #SREC+SR,ARS,R2 ; ADDR OF ACTUAL DOC. ID RANGE START.
749 011272 012703 000000C MOV. #SREC+SR,ARE,R3 ; ADDR OF ACTUAL DOC. ID RANGE END.
750 011276 004737 000000G JSR. PC,CMP3WD ; COMPARE DOC. ID TO ID RANGE FOR THIS SU.
751 011302 005737 000072* TST. CMPST ; DOC. ID WITHIN RANGE FOR SU?
752 011306 001405 BEQ. 15$ ; YES.
753 011310 012737 000011 003300* MOV. #AS,IVI,STATWD ; SET DOC. ID OUT-OF-RANGE ERROR CODE.
754 011316 000137 011570* JMP. PRG050 ; CREATE ACK ENTRY & CONTINUE.
755 011322. 15$:
756 011322. 004737 000000G JSR. PC,LIPR ; LOCATE IPR & INDEX RECORD ENTRY.
757 011326 022737 000001 003300* CMP. #AS,SUP,STATWD ; ALL OK?
758 011334 001402 BEQ. 20$ ; BRANCH IF YES.
759 011336 000137 011570* JMP. PRG050 ; YES, CREATE ACK ENTRY.
760.
761.
762.
763.
; AS SOON AS RETURNING FROM LOCATING THE IPR BRACKET AND THE INDEX ENTRY,
; R1 WILL CONTAIN THE ADDRESS OF THE ENTRY WITHIN THE INDEX RECORD.
;
764 011342. 20$:
765 011342. 010137 000214* MOV. R1,XTEMP ; SAVE ADDRESS OF INDEX ENTRY.
766 011346 012737 000000G 000002* MOV. #IO,RLB,DQIO+Q,IOFN ; READ FUNCTION CODE
767 011354 012737 003302* 000014* MOV. #RDBUF,DQIO+Q,IOPL ; DOCUMENT READ BUFFER.
768 011362 012737 001000 000016* MOV. #SLB,DQIO+Q,IOPL+2 ; READ BUFFER LENGTH
769 011370 013737 000102* 000022* MOV. DQADR,DQIO+Q,IOPL+6 ; ADDR OF DOC. BEING PURGED.
770 011376 013737 000104* 000024* MOV. DQADR+WORD1,DQIO+Q,IOPL+10 ;
771 011404 004737 000000G JSR. PC,ISSDIR ; ISSUE THE READ.
772 011410 005737 000114* TST. DRSTAT ; GOOD READ?
773 011414 001404 BEQ. 25$ ; YES.
774 011416 012737 000002 003300* MOV. #AS,URD,STATWD ; NO, SET UNABLE TO READ DISK ERROR CODE.
775 011424 000461 BR. PRG050 ; CREATE ACK ENTRY
776 011426. 25$:
777 011426 012701 003302* MOV. #RDBUF,R1 ; ADDRESS OF DOCUMENT BUFFER
778 011432 013702 000222* MOV. WDOFF,R2 ; WORD OFFSET FOR DOC. START SEQUENCE.
779 011436 006302. ASL. R2 ; CONVERT TO BYTES
780 011440 000201. ADD. R2,R1 ; ADDR OF 1ST WORD OF DOCUMENT START SEQUENCE.
781 011442 052711 000400. BIS. #BIT0,(R1) ; SET PURGE BIT IN FLAG BYTE OF DOC. START SEQUENCE.
782 011446 012737 000000G 000002* MOV. #IO,WLB,DQIO+Q,IOFN ; WRITE FUNCTION CODE.
783 011454 004737 000000G JSR. PC,ISSDIR ; ISSUE RE-WRITE OF DOC. 1'S 1ST BLOCK.
784 011460 005737 000114* TST. DRSTAT ; GOOD WRITE?
785 011464 001404 BEQ. 30$ ; YES.
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA BASE UPDATE.
PURGED- PURGE DOCUMENT ROUTINE

```

786 011466 012737 000003 003300'      MOV  #AS,UWD,STATWD ; SET ERROR CODE
787 011474 000435                      BR   PRG050         ; CREATE ACK ENTRY
788 011476                                30$:
789 011476 013702 000214'      MOV  XTEMP,R2      ; ADDR OF INDEX ENTRY
790 011502 152762 000200 000001      BISB #BIT7,I,PGI(R2) ; SET PURGE INDICATOR IN INDEX ENTRY
791 011510 013737 000124' 000014'      MOV  IBFADR,DQ10+0,IOPL ; BUFFER ADDR OF INDEX RECORD
792 011516 012737 001000 000016'      MOV  #SLB,DQ10+0,IOPL+2 ; BUFFER LENGTH
793 011524 013737 000132' 000022'      MOV  IDXADR,DQ10+0,IOPL+6 ; ADDR OF INDEX RECORD ON DISK
794 011532 013737 000134' 000024'      MOV  IDXADR+WORD1,DQ10+0,IOPL+10 ;
795 011540 004737 000000G      JSR  PC,ISSDIR    ; ISSUE WRITE
796 011544 005737 000114'      TST  DRSTAT      ; GOOD WRITE?
797 011550 001404                      BEQ  40$         ; YES, SET SUCCESSFUL ACK CODE
798 011552 012737 000003 003300'      MOV  #AS,UWD,STATWD ; NO, SET ERROR CODE
799 011560 000403                      BR   PRG050         ;
800 011562                                40$:
801 011562 012737 000001 003300'      MOV  #AS,SUP,STATWD ; SUCCESSFUL ACK CODE
802 011570                                PRG050:
803 011570 004737 000000G      JSR  PC,CACKE    ; CREATE ACK ENTRY
804 011574 062737 000006 000062'      ADD  #WORD3,ADDRID ; GET ADDR OF NEXT ID
805 011602 005337 000144'      DEC  NDOC        ; DECREMENT DOC ID COUNT
806 011606 001402                      BEQ  60$         ; ALL DOC PURGED
807 011610 000137 011242'      JMP  PRG010     ; PURGE NEXT ONE
808 011614                                60$:
809 011614 000207      RTS  PC ;

```

DBUPD-- DATA-BASE-UPDATE.
RPLDOC-OR-ADSDOC.

MACRO M1110 27-MAR-80 14:28 P. 18
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
811 .SBTTL- RPLDOC-OR-ADSDOC.
812 ;
813 ; SUBROUTINE TO PROCESS THE "MODIFY-REPLACE DOCUMENT" OR THE "ADD
814 ; SUB-DOCUMENT" REQUEST. IT LOCATES THE INDEX ENTRY OF THE INDICATED
815 ; DOCUMENT AND DETERMINES WHETHER THE NEW COPY OR THE NEW SUB-DOC
816 ; WILL FIT THE AVAILABLE SPACE. IF YES, THE PACKED TEXT RECEIVED FROM
817 ; DBPROC IS
818 ; - WRITTEN OVER THE CURRENT TEXT (MODIFY-REPLACE), OR
819 ; - ADDED TO THE EXISTING DOCUMENT TEXT (ADD SUB-DOC).
820 ; IT THEN UPDATES THE UNUSED WHITE SPACE FIELD (I,UWS) OF THE
821 ; DOCUMENT'S INDEX ENTRY.
822 ; FOR A REPLACE-DOC REQUEST, IF THE NEW TEXT IS SHORTER THAN THE
823 ; THE OLD, THE EXTRA TEXT IS OVERWRITTEN BY BLANKS.
824 ;
825 011616 ADSDOC:
826 011616 004737 000000G RPLDOC: JSR PC,SDATA ;SAVE-NEEDED-DATA.
827 011622 012737 000001 003300* MOV #AS,SUP,STATWD ;SHOW-SUCCESS-IN-STATUS-WORD
828 011630 012737 000106* 000062* MOV #DOCID,ADDRID ;ADDRID->3-WORD-DOCUMENT-ID
829 ;
830 ;
831 ; LOCATE DOCUMENT'S INDEX ENTRY.
832 011636 004737 000000G JSR PC,LIPRE ;LOCATE-APPROPRIATE-IPR-ENTRY.
833 011642 032737 000001 003300* BIT #AS,SUP,STATWD ;EVERYTHING-OK?.
834 011650 BON 1$ ;BRANCH-IF-YES.
835 011652 000137 012412* JMP 99$ ;NO: CREATE-NACK-ENTRY.
836 ;
837 ;
838 ; CALCULATE STATISTICS OF DOCUMENT ON DATA-BASE.
839 ; R1-> INDEX-ENTRY.
840 011656 010137 000214* 1$: MOV R1,XTEMP ;SAVE-ADDRESS-OF-INDEX-ENTRY
841 011662 004737 000000G JSR PC,CDSTAT ;CALCULATE-DOCUMENT-STATISTICS.
842 011666 032737 000001 003300* BIT #AS,SUP,STATWD ;EVERYTHING-OK?.
843 011674 BON 2$ ;BRANCH-IF-YES.
844 011676 000137 012412* JMP 99$ ;NO: CREATE-NACK-ENTRY.
845 ;
846 ;
847 ; DETERMINE IF NEW COPY OF DOC WILL FIT INTO AVAILABLE SPACE.
848 ; TOTAL AVAILABLE SPACE FOR DOC = PWCT.PWCT+2 (16 BIT WORDS)
849 ; LENGTH OF PACKED TEXT OF NEW COPY = PC.PTC, PC.PTC+2 (16 BIT WORDS)
850 011702 013702 000064* 2$: MOV BUFADR,R2 ;R2->NEW-DOC'S-BUFFER.
851 011706 023727 003276* 000007 CMP ROSTWD,#AR,ASD ;ADD-SUB-DOC-?.
852 011714 001006 BNE 10$ ;BRANCH-IF-NO.
853 011716 005037 000176* CLR R8
854 011722 013737 000174* 000200* MOV UNWS,R9 ;R9=SPACE-AVAIL. FOR-SUB-DOC
855 011730 000406 BR 11$
856 ;
857 011732 013737 000164* 000176* 10$: MOV PWCT,R8 ;R8=SPACE-AVAIL. FOR-DOC. (HI)
858 011740 013737 000166* 000200* MOV PWCT+2,R9 ;R9= " " " " (LO)
859 011746 166237 000034 000200* 11$: SUB PC,PTC+2(R2),R9 ;R9 = UNUSED-WHITE SPACE (PACKED)
860 011754 005637 000176* SBC R9
861 011760 166237 000032 000176* SUB PC,PTC(R2),R8
862 011766 001003 BNE 3$ ;BRANCH-IF-R8-NOT- = 0 (ERROR)
863 011770 005737 000200* TST R9 ;TEST-UNUSED-WHITE SPACE.
864 011774 002005 BGE 4$ ;BRANCH-IF- > ZERO.
865 ;
866 ; LENGTH OF NEW COPY GREATER THAN AVAILABLE SPACE. RETURN ERROR TO HOST.
867 ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA-BASE-UPDATE.
RPLDOC-OR-ADSDOC.

MACRO M1110 27 MAR 80 14420 PAGE 10 1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
868 011776 012737 000005 003300 3$: MOV #AS,IWS,STATWD ;ERROR=INSUFFICIENT WHITE SPACE.  
869 012004 000137 012412' JMP 99$ ;BUILD-NACK.  
870 ;  
871 ; SUFFICIENT SPACE FOR MODIFIED DOCUMENT TO REPLACE ORIGINAL, OR  
872 ; FOR NEW SUB-DOCUMENT TO BE ADDED TO EXISTING DOCUMENT.  
873 ;  
874 012010 016237 000026 000164' 4$: MOV PC,PWC(R2),PWCT ;PWC = PACKED WORD COUNT OF  
875 012016 016237 000030 000166' MOV PC,PWC+2(R2),PWCT+2 ; NEW TEXT (TEXT + W.S.)  
876 012024 062702 000036 ADD #PC,SPD,R2 ;R2-> START OF PACKED DATA  
877 012030 012737 000000G 000002' MOV #IO,RLB,DQ10+0,IOFN ; READ FUNCTION CODE  
878 012036 012737 005302' 000014' MOV #WBUF,DQ10+0,IOPL ; WORK BUFFER  
879 012044 012737 001000 000016' MOV #SLB,DQ10+0,IOPL+2 ; BUFFER LENGTH  
880 012052 023727 003276' 000007' CMP RQSTWD,#AR,ASD ;ADD-SUB-DOC?  
881 012060 001017 BNE 15$ ;BRANCH IF NO  
882 ;  
883 ; ADD-SUB-DOCUMENT REQUEST  
884 ;  
885 012062 005037 000222' CLR WDOFF  
886 012066 113737 000205' 000222' MOV# EOT+1,WDOFF ;WDOFF=WORD OFFSET OF END-OF-TEXT  
887 012074 013737 000204' 000222' MOV EOT,DQ10+0,IOPL+6  
888 012102 042737 177400 000022' BIC #177400,DQ10+0,IOPL+6 ;SECTOR ADDRESS OF EOT (HI)  
889 012110 013737 000206' 000024' MOV EOT+2,DQ10+0,IOPL+10 ; " " " (LO)  
890 012116 000413 BR 14$  
891 ;  
892 ; MODIFY-REPLACE REQUEST  
893 ;  
894 012120 013701 000214' 15$: MOV XTEMP,R1 ;R1->INDEX-ENTRY  
895 012124 016137 000006 000022' MOV I,SANH(R1),DQ10+0,IOPL+6 ; ADDR OF DOCUMENT BEING READ  
896 012132 042737 177400 000022' BIC #177400,DQ10+0,IOPL+6 ;SECTOR ADDRESS (HI)  
897 012140 016137 000010 000024' MOV I,SANL(R1),DQ10+0,IOPL+10 ;  
898 ;  
899 012146 004737 000000G 14$: JSR PC,ISSDIR ; ISSUE READ  
900 012152 005737 000114' TST DRSTAT ; GOOD-READ?  
901 012156 001405 BEQ 5$ ; YES  
902 012160 012737 000002 003300' MOV #AS,URD,STATWD ; SET READ ERROR BIT  
903 012166 000137 012412' JMP 99$  
904 012172 123727 000030' 5$: CMPB IOST,#IS,SUC ; SUCCESSFUL READ?  
905 012200 001414 BEQ 6$ ; YES  
906 012202 013737 000030' 000154' MOV @#IOST,PAR1 ; READ ERROR  
907 012210 012701 014430' MOV #EMSG2,R1  
908 012214 CALL ERROR  
909 012220 012737 000002 003300' MOV #AS,URD,STATWD ; SET ERROR BIT  
910 012226 000137 012412' JMP 99$  
911 ;  
912 ; GET WORD OFFSET OF START OF EXISTING DOCUMENT ON DATA-BASE  
913 ;  
914 012232 013701 000222' 6$: MOV WDOFF,R1 ;R1=WORD OFFSET OF DOC START IN 1ST SECTOR  
915 012236 006301 ASL R1 ; CONVERT TO BYTES  
916 012240 062701 005302' ADD #WBUF,R1 ;R1->WHERE NEW COPY SHOULD START  
917 ; ;R2->START OF PACKED DATA  
918 ;  
919 ; PUT PACKED DATA ON DISK  
920 ;  
921 012244 004737 000000G JSR PC,PPDATA  
922 012250 022737 000001 003300' CMP #AS,SUP,STATWD ; CONTINUE?  
923 012256 001421 BEQ 7$ ; YES  
924
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD--BASE-UPDATE-
RPLDOC-OR-ADSDOC-

MACRO M1118 27-MAR-98 14:29 P. 18-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
925 : TEST-NATURE-OF-ERROR-
926 :
927 012260 022737 000002 000170' CMP #2,RDSTAT ;ERROR-SEQ-FROM-CCIN-?
928 012266 001414 BEQ 12$ ;BRANCH-IF-YES
929 :
930 : ERROR-SEQUENCE-FROM-DBPROC, OR-INTERNAL-ERROR, CREATE-ACK-ENTRY-
931 :
932 012270 004737 000000G JSR PC,CACKE ; CREATE ACK-ENTRY
933 :
934 : IF-ERROR-SEQUENCE-IS-FROM-DBPROC, CONTINUE-READING-BUFFERS-FROM-
935 : CCIN-UNTIL-ALL-REMAINING-BUFFERS-ARE-READ,
936 :
937 012274 022737 000001 000170' CMP #1,RDSTAT ;ERR-SEQ-FROM-DBPROC-?
938 012302 001006 BNE 12$ ;BRANCH-IF-NO
939 012304 005737 000136' 13$: TST INCOU- ;MORE-BUFFERS?
940 012310 003403 BLE 12$ ;BRANCH-IF-NO
941 012312 CALL READ ;READ-ANOTHER-BUFFER
942 012316 000772 BR 13$
943 012320 000207 12$: RTS PC ; RETURN
944 :
945 :
946 : UPDATE-UNUSED-WHITE-SPACE-FIELD-OF-DOC'S-INDEX-ENTRY-
947 : - SECTOR-ADDRESS-OF-INDEX-RECORD-IN-IDXADR, IDXADR+2
948 : - INDEX-RECORD-IN-IPRBUF
949 : - POINTER-TO-INDEX-ENTRY-IN-"XTEMP"
950 : - R9 =-PACKED-WHITE-SPACE-REMAINING-AFTER-NEW-COPY-OF-DOC-
951 : HAS-BEEN-ADDED.
952 012322 013701 000214' 7$: MOV XTEMP,R1 ;R1->DOC'S-INDEX-ENTRY-
953 012326 013761 000200' 000014 MOV R9,I,UWS(R1) ;UPDATE-UNUSED-WHITE-SPACE-FIELD-
954 :
955 : WRITE-UPDATED-INDEX-RECORD
956 :
957 012334 012737 000000G 000002' MOV #IO,WLB,DQIO+Q,IOFN-
958 012342 013737 000124' 000014' MOV IBFADR,DQIO+Q,IOPL ; INDEX-BUFFER'S-ADD'S-
959 012350 012737 001000 000016' MOV #SLB,DQIO+Q,IOPL+2-
960 012356 013737 000132' 000022' MOV IDXADR,DQIO+Q,IOPL+6
961 012364 013737 000134' 000024' MOV IDXADR+2,DQIO+Q,IOPL+10
962 012372 004737 000000G JSR PC,ISSDIR
963 012376 005737 000114' TST DRSTAT
964 012402 001403 BEQ 99$
965 012404 012737 000003 003300' MOV #AS,UWD,STATWD
966 :
967 :
968 : CALL ROUTINE-TO-CREATE-ACK-ENTRY-
969 012412 004737 000000G 99$: JSR PC,CACKE ; CREATE ACK-ENTRY
970 012416 000207 RTS PC ; RETURN
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA BASE UPDATE.
DESDOC-- DELETE SUB-DOCUMENT.

MACRO: M1110 27 MAR 80 1129 PAGE 13
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
          .SBTTL DESDOC-- DELETE SUB-DOCUMENT.
:
:
:   DESDOC PROCESSES THE DELETE SUB-DOCUMENT PROCESSED REQUEST RECEIVED
:   FROM DBPROC.
:
:
:   DESDOC:
977 012420
978 012420 012737 000001 003300'   MOV.   #AS,SUP,STATWD ;SHOW SUCCESS IN STATUS WORD
979 012426 013702 000064'   MOV.   BUFADR,R2      ; BUFFER START ADDR
980 012432 016237 000010 000144'   MOV.   PR,NIID(R2),NDOC ; # OF SUB-DOC'S TO DELETE
981 012440 010237 000062'   MOV.   R2,ADDRID     ;
982 012444 062737 000012 000062'   ADD.   #PR,FE,ADDRID  ; ADDR OF FIRST DOC ID ENTRY
983 012452
:
:   DES010:
984 012452 013701 000062'   MOV.   ADDRID,R1     ; ADDR OF DOC ID
985 012456 011137 003266'   MOV.   (R1),ACKBUF   ;SAVE DOC ID FOR ACK
986 012462 016137 000002 003270'   MOV.   2(R1),ACKBUF+2
987 012470 016137 000004 003272'   MOV.   4(R1),ACKBUF+4
988 012476 012702 000000C   MOV.   #SREC+SR,ARS,R2 ; ADDR OF ACTUAL DOC ID RANGE START
989 012502 012703 000000C   MOV.   #SREC+SR,ARE,R3 ; ADDR OF ACTUAL DOC ID RANGE END
990 012506 004737 000000G   JSR.   PC,CMP$WD     ; DOC ID WITHIN RANGE FOR THIS SUP
991 012512 005737 000072'   TST.   CMP$T        ;
992 012516 001405   BEQ.   DES015       ; YES
993 012520 012737 000011 003300'   MOV.   #AS,IVI,STATWD ; NO, SET STATUS WORD FOR ACK ENTRY
994 012526 000137 013600'   JMP.   DES045       ;
995 012532
:
:   DES015:
996 012532 004737 000000G   JSR.   PC,LIPRE     ; LOCATE APPROPRIATE IPR ENTRY
997 012536 022737 000001 003300'   CMP.   #AS,SUP,STATWD ;EVERYTHING OK?
998 012544 001402   BEQ.   DES020       ;BRANCH IF YES
999 012546 000137 013600'   JMP.   DES045       ; CREATE ACK ENTRY AND GET NEXT ID
1000 012552
:
:   DES020:
1001 012552 004737 000000G   JSR.   PC,CDSTAT    ; CALCULATE DOCUMENT STATISTICS
1002 012556 022737 000001 003300'   CMP.   #AS,SUP,STATWD ;EVERYTHING OK?
1003 012564 001402   BEQ.   DES022       ;BRANCH IF YES
1004 012566 000137 013600'   JMP.   DES045       ;
1005 012572
:
:   DES022:
:
:   READ DOC FROM DATA BASE
:
:
:
:   DES029:
1009 012572 012737 000000G 000002'   MOV.   #IO,RLB,DQIO+0,IOFN ;FUNCTION CODE
1010 012600 012737 003302' 000014'   MOV.   #RBUF,DQIO+0,IOPL ;READ BUFFER ADDS
1011 012606 012737 001000 000015'   MOV.   #SLB,DQIO+0,IOPL+2 ;BUFFER LENGTH
1012 012614 013737 000102' 000022'   MOV.   DOCADR,DQIO+0,IOPL+6 ; ADDR OF DOCUMENT BEING READ
1013 012622 013737 000104' 000024'   MOV.   DOCADR+WORD1,DQIO+0,IOPL+10 ;
1014 012630 004737 000000G   JSR.   PC,ISSDIR    ; ISSUE READ
1015 012634 005737 000114'   TST.   DR$STAT     ; GOOD READ?
1016 012640 001405   BEQ.   DES030       ; YES
1017 012642 012737 000002 003300'   MOV.   #AS,URD,STATWD ; SET READ ERROR BIT
1018 012650 000137 013600'   JMP.   DES045       ; GET NEXT ID
1019 012654
:
:   DES030:
1020 012654 123727 000030' 000000G   CMPB.  IOST,#IS,SUC  ; SUCCESSFUL READ?
1021 012662 001414   BEQ.   DES035       ; YES
1022 012664 013737 000030' 000154'   MOV.   @#IOST,PAR1  ; READ ERROR
1023 012672 012701 014430'   MOV.   #MSG2,R1    ;
1024 012676   CALL.  ERROR        ;
1025 012702 012737 000002 003300'   MOV.   #AS,URD,STATWD ; SET ERROR CODE
1026 012710 000137 013600'   JMP.   DES045       ; NACK
1027 012714
:
:   DES035:
1028 012714 005337 000152'   DEC.   NS$RD       ; DECREMENT # OF SECTORS TO READ COUNT
```


DBUPD--BASE UPDATE
DESDOC--DELETE SUB-DOCUMENT

MACRO M1110 27-MAR-80 14:29 P. 19-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1029 012720 012703 003302'      MOV.  #RDBUF,R3      ; INITIALIZE INPUT REGISTER
1030 012724 013702 000222'      MOV.  WDOFF,R2     ; WORD OFFSET OF DOC; START IN 1ST SECTOR
1031 012730 006302'      ASL.  R2           ; CONVERT TO BYTES
1032 012732 060203'      ADD.  R2,R3       ; ADDR OF PACKED DOC; START
1033 ;
1034 ;
1035 ;
1036 012734 020327 004302'      ; IF R3 POINTS BEYOND END-OF-BUFFER, READ IN NEXT BLOCK
1037 012740 002413'      CMP.  R3,#<RDBUF+512.>; DOES R3 -> BEYOND BUFFER?
1038 012742 062737 000001 000104'  BLT.  1$          ; BRANCH IF NO
1039 012750 005537'      ADD.  #1,DOCADR+2 ; POINT TO NEXT SECTOR ON DISK
1040 012754 162703 004302'      ADC.  DOCADR      ;
1041 012754 162703 004302'      SUB.  *<RDBUF+512.>,R3 ; R3 = BYTE OFFSET OF START OF
1042 012760 006203'      ; TEXT IN NEXT SECTOR
1043 012762 010337 000222'      ASR.  R3          ; R3 = WORD OFFSET
1044 012766 000701'      MOV.  R3,WDOFF   ; SAVE WORD OFFSET
1045 012766 000701'      BR.   DES029    ; GO READ NEXT SECTOR
1046 012770 010337 000000G'      ;
1047 012774 013737 000102' 000002G' 1$: MOV.  R3,UNPTMP   ; SAVE ADDR (WORD OFFSET)
1048 013002 013737'      MOV.  DOCADR,UNPTMP+2 ; DISK ADDRESS (HI)
1049 013002 013737'      MOV.  DOCADR+2,UNPTMP+4 ; " (LO) OF 1ST 3-WORD
1050 013010 162737 000002 000000G'      SUB.  #2,UNPTMP   ; GROUP IN DOC TEXT
1051 013016 023727 000000G 003302'      CMP.  UNPTMP,#RDBUF ; ????.
1052 013024 002010'      BGE.  2$        ; STILL IN BUFFER?
1053 ;
1054 ;
1055 ;
1056 013026 062737 001000 000000G'      ; UNPTMP POINTS TO LAST WORD OF PREVIOUS BUFFER
1057 013034 162737 000001 000004G'      ADD.  #512,UNPTMP ; POINT TO LAST WORD OF BUFFER
1058 013042 005637 000002G'      SUB.  #1,UNPTMP+4 ; SAVE ADDR'S OF PREVIOUS SECTOR
1059 013046 012704 000000G'      SBC.  UNPTMP+2   ;
1060 013052 013705 000062'      2$: MOV.  #RDBUF+RT,FC,R4 ; R4-> OUTPUT BUFFER
1061 013056 062705 000006'      MOV.  ADDRID,R5  ;
1062 013062 004737 000000G'      ADD.  #6,R5      ; R5-> SUB-DOC ID STRING
1063 013066 103405'      JSR.  PC,UNPACK ; LOOK FOR SPECIFIED SUB-DOC ID
1064 013066 103405'      BCS.  3$        ; BRANCH IF ID IS FOUND
1065 ;
1066 ;
1067 013070 012737 000011 003300'      ; SPECIFIED SUB-DOC ID WAS NOT FOUND
1068 013076 000137 013600'      MOV.  #AS,IVI,STATWD ; INVALID ID
1069 013076 000137 013600'      JMP.  DES045    ; NACK
1070 ;
1071 ;
1072 ;
1073 ;
1074 ;
1075 ;
1076 013102 023737 000002G 000102' 3$: CMP.  DSDTMP+2,DOCADR ; COMPARE HI ADDR'S
1077 013110 001004'      BNE.  4$        ; BRANCH IF NOT THE SAME
1078 013112 023737 000004G 000104'      CMP.  DSDTMP+4,DOCADR+2 ; LO ADDR'S
1079 013120 001420'      BEQ.  5$        ; BRANCH IF DESIRED SECTOR IS CURRENTLY
1080 ;
1081 ;
1082 ;
1083 ;
1084 013122 012737 000002 000152' 4$: MOV.  #2,NS2RD   ; NUMBER OF SECTORS TO READ VARIABLE
1085 013130 013737 000002G 000102'      MOV.  DSDTMP+2,DOCADR ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA BASE UPDATE
DESDOC - DELETE SUB-DOCUMENT

MACRO M1110 27 MAR 80 14:29 PAGE 12-3
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1086 013136 013737 000004G 000104' MOV DSDTMP+4,DOCADR+2
1087 013144 162737 000001 000104' SUB #1,DOCADR+2
1088 013152 005637 000102' SBC DOCADR
1089 013156 CALL READN
1090
1091 013162 013703 000000G 5#: MOV DSDTMP,R3 ;R3->1ST PACKED WORD OF 3-WORD GROUP
1092 013166 011302 MOV (R3),R2 ;R2 = 1ST PACKED WORD
1093 013170 STRTMK #BIT8 ;1ST CHARACTER
1094 013210 011302 MOV (R3),R2 ;STILL 1ST WORD OF 3-WORD GROUP
1095 013212 006302 ASL R2 ;MOVE 2ND SIX-BIT CHARACTER--
1096 013214 006302 ASL R2 ; -- TO LOWER 6-BITS OF R2
1097 013216 000302 SWAB R2
1098 013220 STRTMK #BIT14 ;2ND CHARACTER
1099 013240 012301 MOV (R3)+,R1 ;NEED UPPER 4 BITS OF FIRST WORD
1100 013242 020327 004302' CMP R3,#RDBUF+SLB ;END OF INPUT BUFFER ?
1101 013246 103402 BLO 6$ ;NOT YET
1102 013250 004737 000000G JSR PC,READN ;READ NEXT SECTOR OF PACKED DOC
1103 013254
1104 013254 162737 000001 000166' 6$: SUB #1,PWCT+WORD1 ;DECREMENT PACKED WORD COUNT
1105 013262 005637 000164' SBC PWCT
1106 013266 011302 MOV (R3),R2 ;R2 = 2ND WORD OF 3-WORD GROUP
1107 013270 006101 ROL R1 ;4TH BIT TO C-BIT
1108 013272 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1109 013274 006101 ROL R1 ;3RD BIT TO C-BIT
1110 013276 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1111 013300 006101 ROL R1 ;2ND BIT TO C-BIT
1112 013302 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1113 013304 006101 ROL R1 ;1ST BIT TO C-BIT
1114 013306 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1115 013310 STRTMK #BIT4 ;3RD CHARACTER
1116 013330 011302 MOV (R3),R2 ;STILL 2ND WORD OF 3-WORD GROUP
1117 013332 006202 ASR R2 ;MOVE 4TH 6-BIT CHAR. TO
1118 013334 006202 ASR R2 ; -- LOWER 6 BITS
1119 013336 STRTMK #BIT10 ;4TH CHARACTER
1120 013356 011302 MOV (R3),R2 ;STIL 2ND WORD OF 3-WORD BLOCK
1121 013360 000302 SWAB R2 ;GET 5TH CHAR. IN LOWER 6 BITS OF R2
1122 013362 042702 177700 BIC #177700,R2 ;R2 = 5TH CHARACTER
1123 013366 005702 TST R2 ;SUB-DOC START MARK ?
1124 013370 001005 BNE 7$ ;BRANCH IF NO
1125 013372 052763 000001 000002' BIS #BIT0,2(R3) ;SET PURGE BIT
1126 013400 000137 013600' JMP DES045 ;ACK
1127 013404 012301 7$: MOV (R3)+,R1 ;NEED UPPER 2 BITS OF 2ND WORD
1128 013406 020327 004302' CMP R3,#RDBUF+SLB ;END OF INPUT BUFFER ?
1129 013412 103402 BLO 10$ ;BRANCH IF NO
1130 013414 004737 000000G JSR PC,READN ;READ NEXT SECTOR OF PACKED DOC
1131 013420
1132 013420 162737 000001 000166' 10$: SUB #1,PWCT+2 ;DECREMENT PACKED WORD COUNT
1133 013426 005637 000164' SBC PWCT
1134 013432 011302 MOV (R3),R2 ;R2 = 3RD WORD OF 3-WORD GROUP
1135 013434 006101 ROL R1 ;2ND BIT TO C-BIT
1136 013436 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1137 013440 006101 ROL R1 ;1ST BIT TO C-BIT
1138 013442 006102 ROL R2 ;C-BIT TO BIT0 OF R2
1139 013444 STRTMK #BIT6 ;6TH CHARACTER
1140 013464 011302 MOV (R3),R2 ;STILL 3RD WORD OF 3-WORD GROUP
1141 013466 006202 ASR R2 ;GET 7TH CHAR. IN LOWER --
1142 013470 006202 ASR R2 ; -- 6 BITS OF R2
```

DBUPD--BASE UPDATE.
DESDOC--DELETE SUB-DOCUMENT.

MACRO M1110 27-MAR-80 14:29 19-3

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1143 013472 006202 ASR R2
1144 013474 006202 ASR R2
1145 013476 STRTMK #BIT12 ;7TH CHARACTER
1146 013516 012302 MOV (R3)+,R2 ;STILL 3RD WORD OF 3-WORD GROUP
1147 013520 006202 ASR R2
1148 013522 006202 ASR R2
1149 013524 000302 SWAB R2 ;GET 8TH CHAR INTO LOWER 6 BITS
1150 013526 162737 000001 000166' SUB #1,PWCT+2 ;DECREMENT PACKED WORD COUNT
1151 013534 005637 000164' SBC PWCT
1152 ;
1153 ; START MARK WAS NOT FOUND. FLAG BYTE HAD TO BE THE FIRST BYTE.
1154 ;
1155 013540 013703 000000G MOV DSDTMP,R3
1156 013544 052713 000004 BIS #BIT2,(R3) ;SET PURGE BIT
1157 ;
1158 ; WRITE BLOCK BACK TO DATA BASE.
1159 ;
1160 013550 005037 000226' DES040: CLR WRTST
1161 013554 012737 000000G 000002' MOV #10,WLB,DIO+0,IOFN
1162 013562 004737 000000G JSR PC,ISSDIR
1163 013566 005737 000114' TST DRSTAT
1164 013572 001402 BEQ DES045
1165 013574 004737 000000G JSR PC,EXIT
1166 ;
1167 ; CALL ROUTINE TO CREATE ACK ENTRY.
1168 ;
1169 013600 004737 000000G DES045: JSR PC,CACKE ; CREATE ACK ENTRY
1170 013604 005337 000144' DEC NDOC ; DECREMENT # OF DOCUMENTS TO READ COUNT
1171 013610 001410 BEQ DES050 ; DONE
1172 013612 062737 000020 000062' ADD #WORDS,ADDRID ; GET ADDR OF NEXT DOC ID
1173 013620 012737 000001 003300' MOV #AS,SUP,STATWD ; SET UP STATUS WORD FOR NEXT DOC
1174 013626 000137 012452' JMP DES010 ; CONTINUE
1175 013632 DES050:
1176 013632 005037 003276' CLR ROSTWD ; FOR NEXT UPDATE REQUEST
1177 013636 005037 003300' CLR STATWD ; FOR NEXT UPDATE REQUEST
1178 013642 000207 RTS PC ; RETURN
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUPD-- DATA BASE UPDATE
WRAP-- WRAP UP AND EXIT

MACRO M1110 27-MAR-80 14:29 PAGE 30
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1180 .SBTTL WRAP- WRAP UP AND EXIT
1181 ;
1182 ; WRAP WRITES THE LAST BLOCK OF THE ACK RECORD, TRANSMITS THE ACK
1183 ; EXCHANGE TO THE MASTER AND CAUSES DBUPD TO EXIT.
1184 ;
1185 013644 012700 006556* WRAP: MOV #ACKFDB,R0 ;R0-> FDB
1186 013650 004737 000000G JSR PC,WRTRTD ; WRITE LAST BLOCK OF ACK RECORD
1187 ;
1188 ; TRANSMIT ACK RECORD TO MASTER
1189 ;
1190 013654 CALL XMTACK
1191 013660 MOUT$S #MSG14 ; UPDATE BATCH COMPLETED - DBUPD EXITING
1192 013676 000207 RTS PC ; RETURN
1193 ;
1194 ;
1195 ;
1196 .SBTTL OPEACK
1197 ;
1198 ; SUBROUTINE TO CREATE A NEW SPOOL FILE FOR DB UPDATE ACKNOWLEDGEMENTS
1199 ; AND TO INITIALIZE THE FIRST BUFFER OF THE NEW ACK EXCHANGE
1200 ;
1201 013700 OPEACK::SAVE R0,R1
1202 013704 MOV #ACKFDB,R0 ; FDB ADDRESS FOR ACKNOWLEDGEMENT RECORD
1203 013710 MOV #FN,ACK,R1 ; ACK FILE NUMBER
1204 013714 004737 000000G JSR PC,BLDNFL ; CREATE A NEW FILE
1205 013720 FDAT$R #ACKFDB,.,.,#-4.,#-4. ;
1206 013740 OFNB$W #ACKFDB ;
1207 013756 103012 BCC 1$ ; SUCCESSFUL OPEN
1208 013760 116004 000052 MOV$ F,ERR(R0),R4 ; ERROR
1209 013764 010437 000154* MOV R4,PAR1 ;
1210 013770 012701 014420* MOV #MSG0,R1
1211 013774 CALL ERROR
1212 014000 CALL EXIT
1213 ;
1214 ; SET UP EXCHANGE HEADER
1215 ;
1216 014004 012737 042101 000000C-1$: MOV #AD,DBACKB+AK,TYP ; SET EXCHANGE TYPE FOR ACK EXCHANGE
1217 014012 012737 000000C-000056* MOV #DBACKB+AK,FE,ACKPTR ; INITIALIZE ACK POINTER FOR 1ST SECTOR
1218 014020 005037 000054* CLR ACKECT ; ZERO OUT ACK EXCHANGE'S ENTRY COUNT
1219 014024 RESTOR R0,R1
1220 014030 000207 RTS PC
```

DBUPD... BASE-UPDATE
XMTACK:

MACRO-M1110 27-MAR-80 14:29 P. 21
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1222 .SBTTL XMTACK  
1223 ;  
1224 ; SUBROUTINE TO TRANSMIT COMPLETED ACK EXCHANGE TO MASTER  
1225 ; THE SUBROUTINE PERFORMS THE FOLLOWING PROCESSING STEPS:  
1226 ; 1) BUILDS RCVO PACKET FOR CCOUT (CONTAINING FILE ID)  
1227 ; 2) READS FIRST BLOCK OF ACK FILE AND INSERTS TOTAL ENTRY COUNT  
1228 ; 3) WRITES FIRST BLOCK OF ACK FILE BACK TO FILE  
1229 ; 4) CLOSES ACK SPOOL FILE  
1230 ; 5) QUEUES PACKET CONTAINING SPOOL FILE ID TO CCOUT VIA THE RCVO  
1231 ; IF THERE ARE NO ENTRIES IN THE ACK FILE, THE FILE IS NOT SENT.  
1232 ;  
1233 014032 005737 000054' XMTACK: TST ACKECT ; ANY ENTRIES IN ACK FILE ?  
1234 014036 001005 BNE 2$ ; BRANCH IF YES  
1235 014040 DELET$ #ACKFDB ; DELETE EMPTY FILE  
1236 014050 000503 BR 1$  
1237 ;  
1238 ; BUILD PRB  
1239 ;  
1240 014052 112737 000001 000000C-2$ MOV$ #1,PRB+R0,CS ; COMMAND SOURCE  
1241 014060 105037 000000C CLR$ PRB+R0,CB ; COMMAND BYTE  
1242 014064 013737 006660' 000000C MOV$ ACKFDB+F,FNB+N,FID,PRB+R0,FID ; FILE ID  
1243 014072 013737 006662' 000000C MOV$ ACKFDB+F,FNB+N,FID+WORD1,PRB+R0,FID+WORD1 ;  
1244 014100 013737 006676' 000000C MOV$ ACKFDB+F,FNB+N,FVER,PRB+R0,FVER ; FILE VERSION NUMBER  
1245 014106 012737 000016 000000C MOV$ #FN,ACK,PRB+R0,FNB ; FILE NUMBER  
1246 ;  
1247 ; READ FIRST BLOCK OF ACK SPOOL FILE INTO DBACKB  
1248 ;  
1249 014114 012700 006556' MOV$ #ACKFDB,R0 ; R0 -> FDB  
1250 014120 CALL READF ; READ FIRST BLOCK  
1251 014124 013737 000054' 000000C MOV$ ACKECT,DBACKB+AK,CNT ; INSERT ENTRY COUNT  
1252 ;  
1253 ; WRITE FIRST BLOCK BACK TO SPOOL FILE  
1254 ;  
1255 014132 012700 006556' MOV$ #ACKFDB,R0 ; R0 -> FDB  
1256 014136 005060 000064 CLR$ F,BKVB(R0)  
1257 014142 012760 000001 000066 MOV$ #1,F,BKVB+2(R0) ; VIRTUAL BLOCK 1  
1258 014150 CALL WRTRTD  
1259 ;  
1260 ; CLOSE ACK SPOOL FILE  
1261 ;  
1262 014154 012760 000005 000012 MOV$ #5,F,EFBK+2(R0) ; SET END-OF-FILE BLOCK NUMBER  
1263 014162 CLOSE$ #ACKFDB  
1264 ;  
1265 ; SEND PRB CONTAINING FILE ID OF SPOOLED ACK EXCHANGE TO CCOUT  
1266 ;  
1267 014172 SDAT$S #CCOUT,#PRB+WORD2,#CF,COT ; SEND DATA TO CCOUT  
1268 014234 103011 BCC 1$ ; GOOD SEND  
1269 014236 013737 000000G 000154' MOV$ #DSJ,PAR1  
1270 014244 012701 014424' MOV$ #EMSG1,R1  
1271 014250 CALL ERROR  
1272 014254 CALL EXIT  
1273 ;  
1274 014260 000207 1$: RTS PC ; RETURN
```

DBUPD-- DATA-BASE-UPDATE.
WRTCOM- WRITE-COMMON-AREA-TO-DISK.

MAR 60 11 11 P 77 800 000
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1276  
1277  
1278  
1279  
1280  
1281  
1282 014262.  
1283  
1284  
1285  
1286 014262. 012700 005302'  
1287 014266 004737 000000G.  
1288  
1289  
1290  
1291 014272. 012701 000000G.  
1292 014276 012702. 005302'  
1293 014302. 013703 000116  
1294 014306 006203  
1295 014310  
1296 014310 012122.  
1297 014312. 005303  
1298 014314 003375  
1299  
1300  
1301  
1302 014316 012737 000000G. 000002'  
1303 014324 012737 005302' 000014'  
1304 014332. 012737 001000 000016'  
1305 014340 013737 000126' 000022'  
1306 014346 013737 000130' 000024'  
1307 014354 004737 000000G.  
1308 014360 005737 000114'  
1309 014364 001402.  
1310 014366  
1311 014372.  
1312 014372. 000207  
; .SBTTL- WRTCOM- WRITE-COMMON-AREA-TO-DISK.  
; THIS-ROUTINE-PLACES-THE-DATA-BASE-INFORMATION-STORED-IN-COMMON-IN-  
; A-BUFFER-TO-WRITE-THE-INITIALIZATION-BLOCK-OF-THE-DISK-AFTER-EACH-UPDATE-  
; TO-HELP-MAINTAIN-THE-INTEGRITY-OF-THE-DISK.  
; WRTCOM:  
; CLEAR-OUTPUT-BUFFER.  
; MOV- #WBUF,R0 ;  
; JSR- PC,CLEARB. ; CLEAR-WORKING-BUFFER.  
; COPY-DB-INFO-FROM-SREC-TO-OUTPUT-BUFFER.  
; MOV- #SREC,R1 ; ADDRESS-OF-STATUS-RECORD.  
; MOV- #WBUF,R2 ; ADDR-OF-WORKING-BUFFER.  
; MOV- DBSLEN,R3 ; LENGTH-OF-DATA-BASE-DATA-IN-SREC.  
; ASR- R3 ; CONVERT-BYTES-TO-WORDS.  
10$: MOV- (R1)+,(R2)+ ;  
; DEC- R3 ; GOT-ALL-DATA? .  
; BGT- 10$ ; NO.  
; WRITE-DB-INFO-TO-DISK.  
; MOV- #IO,MLB,DQIO+0,IOFN ; WRITE-FUNCTION-CODE.  
; MOV- #WBUF,DQIO+0,IOPL ;  
; MOV- #SLB,DQIO+0,IOPL+2 ; LENGTH.  
; MOV- 1BLK,DQIO+0,IOPL+6 ;  
; MOV- 1BLK+WORD1,DQIO+0,IOPL+10 ;  
; JSR- PC,ISSDIR ; ISSUE-WRITE.  
; TST- DRSTAT ; GOOD-WRITE? .  
; BEQ- 20$ ; YES.  
; CALL- EXIT ;  
20$: RTS- PC ;
```

DBUPD-- DATABASE UPDATE
ERROR MESSAGES

MACRO: M1110 27-MAR-80 14:29 P L24
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL- ERROR MESSAGES
.NLIST- BEX
;
; SUBROUTINE TO OUTPUT ERROR MESSAGES
;
ERROR:: MOV- (SP),PAR1+2- ;PC OF CALLING ROUTINE-
        MOUT$S- R1,#PAR1
        RTS- PC-
;
;
EMSG0:: .WORD- LN0E-LN0
        .WORD- LN0
EMSG1:: .WORD- LN1E-LN1
        .WORD- LN1
EMSG2:: .WORD- LN2E-LN2-
        .WORD- LN2-
        MSG11:: .WORD- LN11E-LN11
        .WORD- LN11
        MSG12:: .WORD- LN12E-LN12-
        .WORD- LN12
        MSG14:: .WORD- LN14E-LN14
        .WORD- LN14
        MSG15:: .WORD- LN15E-LN15
        .WORD- LN15
;
LN0: .ASCIZ- /FCS ERROR- =-%1D, PC- =-%10/
LN0E:
;
LN1: .ASCIZ- /DIRECTIVE ERROR- =-%1D, PC- =-%10/
LN1E:
;
LN2: .ASCIZ- /IO ERROR- =-%1D, PC- =-%10/
LN2E:
;
LN11: .ASCIZ- /DISK FULL- DOCUMENTS NOT BEING ADDED-/
LN11E:
;
LN12: .ASCIZ- /DISK MORE THAN 98% FULL-/
LN12E:
;
LN14: .ASCIZ- /UPDATE BATCH COMPLETED- DBUPD EXITING-/
LN14E:
;
LN15: .ASCIZ- /FATAL ERROR-/
LN15E:
.EVEN-
.END- DBUPD-
```

DBUPD-- DATA BASE UPDATE
SYMBOL TABLE

MACRO M1110 27-MAR-80 14:29 PAGE 24-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

ACKBUF = 003266RG	BYTE7 = 000007	ECC = 000120RG	F.RATT = 000001	LN1 = 014507R
ACKECT = 000054RG	BYTE8 = 000010	EOT = 000204RG	F.RCNM = 000034	LN1E = 014550R
ACKFDB = 000556RG	BYTE9 = 000011	ERROR = 014374RG	F.RCTL = 000017	LN11 = 014602R
ACKPTR = 000056RG	BYTVAL = 000012	ETRYCT = 000122RG	F.RSIZ = 000002	LN11E = 014651R
ACKSCT = 000060RG	CACKE = ***** GX	EXCHTP = 007244R	F.RTYP = 000000	LN12 = 014651R
ADDDOC = 007310R	CCOUT = 000066RG	EXIT = ***** GX	F.SEQN = 000100	LN12E = 014701R
ADDRID = 000062RG	CDSTAT = ***** GX	FD.FID = 000000	003 F.SPDV = 000074	LN14 = 014701R
ADSDOC = 011616R	CF.COT = 000041	FD.FNB = 000006	003 F.SPUN = 000074	LN14E = 014750R
AK.CHT = 000004 G	012 CF.DGN = 000046	FD.FVR = 000004	003 F.STBK = 000036	LN15 = 014750R
AK.FE = 000006 G	012 CF.DHR = 000042	FD.LEN = 000010	003 F.UNIT = 000136	LN15E = 014764R
AK.PAD = 000002 G	012 CF.DMC = 000047	FD.RWM = ***** GX	F.URBD = 000020	LN2 = 014550R
AK.TYP = 000000 G	012 CF.HBR = 000045	FN.ACK = 000016	004 F.VBN = 000064	LN2E = 014602R
AR.ADD = 000006 G	CF.HRL = 000044	FN.FSA = 000000	004 F.VBSZ = 000060	MECIND = 000044 G
AR.ASD = 000007 G	CF.UPD = 000043	FN.FSB = 000002	004 HRL0 = ***** GX	MECIPR = 000121 G
AR.DSD = 000003 G	CINDXE = ***** GX	FN.FSC = 000004	004 IBFADR = 000124RG	MSGOUT = ***** GX
AR.DDC = 000004 G	CIPRE = ***** GX	FN.FSD = 000020	004 IBLK = 000126RG	MSQ = ***** GX
AR.PDC = 000002 G	CLEARB = ***** GX	FN.MHR = 000010	004 IDXADR = 000132RG	MS.DGN = 010000
AR.RDC = 000001 G	CMFST = 000072RG	FN.NMB = 000022	004 IE.ITS = ***** GX	N = 000012
AR.RPD = 000005 G	CMF3WD = ***** GX	FN.QLS = 000006	004 ILUN = 000001	NDCADR = 000140RG
AS.COO = 000004 G	C002 = 000074RG	FN.RDC = 000014	004 INCOUT = 000136RG	NDOC = 000144RG
AS.DNE = 000010 G	DBACKB = 000230RG	FN.UPD = 000012	004 IN.CEC = 000004 G	010 NDCNVT = 000146RG
AS.FAT = 000014 G	DBSLEN = 000116	FD.WRT = ***** GX	IN.FE = 000010 G	010 NSBUF = 000002 G
AS.IVI = 000011 G	DBUPD = 006716R	F.ACTL = 000076	IN.MEC = 000006 G	010 NSREAD = 000150RG
AS.IVR = 000012 G	DBU005 = 006726R	F.ALOC = 000040	IN.SAL = 000002 G	010 NS2RD = 000152RG
AS.IWS = 000005 G	DELTA = 000100RG	F.BBFS = 000062	IN.SAN = 000000 G	010 N.BFAC = 000004
AS.NRD = 000006 G	DESDOC = 012420R	F.BDB = 000070	IN.TYP = 000001 G	010 N.BFCH = 000006
AS.SDH = 000007 G	DESD01 = 012422R	F.BGBC = 000057	IOST = 000030RG	N.BIPE = 000006 G
AS.SUP = 000001 G	DES015 = 012532R	F.BKDN = 000026	IO.RLB = ***** GX	N.BTCH = 000004
AS.URD = 000002 G	DES020 = 012552R	F.BKDS = 000020	IO.WLB = ***** GX	N.BUFB = 004000
AS.UWD = 000003 G	DES022 = 012572R	F.BKEF = 000050	IPRBUF = 001230RG	N.BUFW = 002000
BITTST = 006402RG	DES029 = 012572R	F.BKPT = 000051	IP.BPE = 000020 G	007 N.BXE = 000016 G
BITVAL = 000000	DES030 = 012654R	F.BKST = 000024	IP.CEC = 000006 G	007 N.DID = 000024
BIT0 = 000001	DES035 = 012714R	F.BKVB = 000064	IP.DLT = 000004 G	007 N.DVNM = 000032
BIT1 = 000002	DES040 = 013550R	F.CHR = 000075	IP.FE = 000030 G	007 N.FID = 000000
BIT10 = 0002000	DES045 = 013600R	F.CNTG = 000034	IP.HID = 000012 G	007 N.FNAM = 000006
BIT11 = 0004000	DES050 = 013632R	F.DFNB = 000046	IP.LIA = 000022 G	007 N.FOS = 000764
BIT12 = 0100000	DG.ERR = 001000	F.DSPT = 000044	IP.PAD = 000026 G	007 N.FTYP = 000014
BIT13 = 0200000	DG.SDF = 002000	F.DVNH = 000134	IP.SAL = 000002 G	007 N.FVER = 000016
BIT14 = 0400000	DG.TDF = 004000	F.EFBK = 000010	IP.SAN = 000000 G	007 N.NEXT = 000022
BIT15 = 1000000	DHDLR = 000015 G	F.FENB = 000050	IP.SRL = 000010 G	007 N.QUERY = 000031
BIT2 = 0000004	DHR0 = ***** GX	F.EOBB = 000032	IP.TYP = 000001 G	007 N.STAT = 000020
BIT3 = 0000010	DOCADR = 000102RG	F.ERR = 000052	ISSDIR = ***** GX	N.SUNT = 000002
BIT4 = 0000020	DOCID = 000106RG	F.FACC = 000043	IS.SUC = ***** GX	N.UNIT = 000034
BIT5 = 0000040	DPLUN = 000004	F.FFBY = 000014	I.BC = 000007 G	011 OFFTS = 000010 G
BIT6 = 0001000	DQIO = 000000RG	F.FNAM = 000010	I.BIDH = 000000 G	011 OLUN1 = 000002
BIT7 = 0002000	DRSTAT = 000114RG	F.FNB = 000102	I.BIDL = 000004 G	011 OLUN2 = 000007
BIT8 = 0004000	DSDFTH = ***** GX	F.FTYP = 000116	I.BIDM = 000002 G	011 OPEACK = 013700RG
BIT9 = 0010000	DSPTCH = 007170R	F.FVER = 000120	I.PGI = 000001 G	011 PAR\$\$\$ = 000027
BLDNFL = ***** GX	EFLG = 000004	F.HIBK = 000004	I.RWS = 000012 G	011 PAR1 = 000154RG
BUFADR = 000064RG	EMSG0 = 014420RG	F.LUN = 000042	I.SANH = 000006 G	011 PCCT = 000160RG
BYTE0 = 0000000	EMSG1 = 014424RG	F.MBCT = 000054	I.SANL = 000010 G	011 PC.DBT = 000000 G 006
BYTE1 = 0000001	EMSG11 = 014434R	F.MBC1 = 000055	I.UWS = 000014 G	011 PC.IDH = 000010 G 006
BYTE2 = 0000002	EMSG12 = 014440R	F.MBFG = 000056	LDRAKE = 000006 G	PC.IDL = 000014 G 006
BYTE3 = 0000003	EMSG14 = 014444R	F.NRBD = 000024	LIPE = ***** GX	PC.IDM = 000012 G 006
BYTE4 = 0000004	EMSG15 = 014450RG	F.NRBC = 000030	LNSBUF = 002000 G	PC.PAD = 000002 G 006
BYTE5 = 0000005	EMSG2 = 014430RG	F.OVBS = 000030	LN0 = 014454R	PC.PTC = 000032 G 006
BYTE6 = 0000006	ENDBFA = 000116RG	F.PACC = 000016	LNBE = 014507R	PC.PWC = 000026 G 006

DBUPD... BASE UPDATE
SYMBOL TABLE

MACRO Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
PC SPD 000036 G 006 RQSTWD 003276RG SR.LIN 000066 002 T$AD = 000020 UNWS 000174RG
PC STW 000006 G 006 RQWS 000172RG SR.LIP 000062 002 T$BA = 000002 UPCCT 000216RG
PC TCC 000016 G 006 RQCB 000004 005 SR.MON 000006 002 T$BB = 000010 WBUF 005302RG
PC TYP 000004 G 006 RQCS 000005 005 SR.NDC 000042 002 T$BSO = 100000 WDOFF 000222RG
PC UWS 000024 G 006 RQFID 000006 005 SR.NDS 000036 002 T$BT = 000020 WDOFND 000224RG
PC WSC 000022 G 006 RQFNB 000014 005 SR.NIN 000030 002 T$BTAR = 000030 WEF1 = 000002
PPDATA = ***** GX RQFVR 000012 005 SR.NIP 000022 002 T$BTD = 002000 WEF2 = 000003
PRB 003230RG RQTHH 000000 005 SR.SDB 000032 002 T$CD = 000100 WIOST 000040RG
PRD010 010144R RSTEDC = ***** GX SR.SRC 000002 002 T$CLK = 002000 WIOST1 000044RG
PRD015 010224R RTDBUF 004302RG SR.SUN 000000 002 T$DISK = 000200 WIOST2 000050RG
PRD020 010244R RTDFDB 006416RG SR.TWS 000056 002 T$DRD = 000004 WORD0 = 000000
PRD022 010264R RTFC 000020 G 013 SR.WSL 000052 002 T$EMEM = 010000 WORD1 = 000002
PRD025 010364R RT.IDH 000006 G 013 SR.YR 000004 002 T$FSA = 000000 WORD2 = 000004
PRD028 010526R RT.IDL 000012 G 013 SR.IIN 000024 002 T$FSAB = 000004 WORD3 = 000006
PRD029 010450R RT.IDM 000010 G 013 SR.IIP 000016 002 T$FSAC = 000014 WORD4 = 000010
PRD030 010542R RT.PAD 000002 G 013 STATUD 003300RG T$FSB2 = 000010 WORD5 = 000012
PRD035 010602R RT.TCH 000014 G 013 SZMK = 000003 G T$IB = 000026 WORD6 = 000014
PRD045 011140R RT.TCL 000016 G 013 S.BFHD = 000020 T$IBAR = 000024 WORD7 = 000016
PRD050 011176R RT.TYP 000000 G 013 S.FATT = 000016 T$IBF = 020000 WORD8 = 000020
PREAD 010112R RT.WSC 000004 G 013 S.FDB = 000140 T$IBF = 040000 WORD9 = 000022
PRGH = 000001 G R$IND = 000100 G T$ICD = 000040 WRAP 013644R
PRG010 011242R R$IPR = 000200 G T$NODE = 004000 WRDVAL = 000024
PRG050 011570R R9 000176RG S.FNB = 000036 T$OB = 000036 WRTRTD = ***** GX
PR.FE = 000012 G 006 R9 000200RG S.FNBW = 000017 T$OBE = 004000 WRTST 000226RG
PR.NID = 000010 G 006 SAVEPT 000202RG S.FNTY = 000002 T$OBF = 010000 XMTACK 014032RG
PURGED 011210R SDATA = ***** GX S.NFEN = 000020 T$OBWA = 000032 XTEMP 000214RG
PWCT 000164RG SDMK = 000000 G T$OUTA = 100000 ZMK = 000002 G
Q IOAE = 000012 SLB = 001000 G TD$CTR = 176370 T$RBD0 = 000200 $DMUL = ***** GX
Q IOEF = 000006 SLW = 000400 G TD$CTW = 176360 T$RNB = 000040 $DSU = ***** GX
Q IOFN = 000002 SNTC = 000004 G TD$INL = 004000 T$RSET = 040000 $$$ARG = 000005
Q IOLU = 000004 SREC = ***** GX TD$MEM = 000270 T$SC = 000022 $$$OST = 000014
Q IOPL = 000014 SR.ARE 000114 002 TD$OAR = 176344 T$SCLK = 020000 $$$T2 = 000027
Q IOPR = 000007 SR.ARS 000106 002 TD$OTR = 176346 T$SEG1 = 000000 .CLOSE = ***** G
Q IOSB = 000010 SR.DAY 000010 002 TD$ORD = 000274 T$SEG2 = 000001 .DELETE = ***** GX
RDBUF 003302RG SR.DLT 000014 002 TD$RST = 176366 T$SEG3 = 000002 .DLFNB = ***** GX
RDEOC = ***** GX SR.ECB 000047 002 TD$SLW = 176376 T$SO = 000001 .FINIT = ***** G
RDSTAT 000170RG SR.ECH 000046 002 TD$TAR = 176372 T$UBUS = 100000 .FSRCB = ***** G
READ = ***** GX SR.ECL 000050 002 TD$TAW = 176362 T$1CLK = 000400 .OPFNB = ***** G
READF = ***** GX SR.FIB 000012 002 TD$TAW = 176362 T$BBEN = 000020 ...PC1 = 006556R
READN = ***** GX SR.GRE 000100 002 TD$TDW = 176364 ...PC2 = 006716R
REF = 000001 SR.GRS 000072 002 TENC = 000210RG UNPACK = ***** GX ...PC3 = 000140
RIOST 000034RG SR.LEN 000122 002 TRANSA 006302RG UNPTMP = ***** GX
RPLDOC 011616R
```

. ABS 000000 000
014764 001
SRCOFF 000122 002
FDSCOF 000010 003
FNOFFS 000022 004
ROFF 000016 005
PCROFF 000040 006
IPROFF 000032 007
INDOFF 000012 010
INEOFF 000016 011
ACKOFF 000010 012
RETOFF 000021 013
\$\$FSR1 000000 014

DBUPD--DATA-BASE-UPDATE-
SYMBOL-TABLE-

MACRO-M1110 27-MAR-88 14:29 PAGE-24-7

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

ERRORS-DETECTED: 0

VIRTUAL-MEMORY-USED: 7413 WORDS (29 PAGES)
DYNAMIC-MEMORY: 8084 WORDS (31 PAGES)
ELAPSED-TIME: 00:01:25
DBUPD, DBUPD /-SP/NL:MEB=C20,13P,C, DBUPD-

.MAIN: . MA M1110 27-MAR-80 14:31
TABLE OF CONTENTS:

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	1	TASK NAME: DBUPD- DATA-BASE-UPDATE-
8-	2	
8-	3	
8-	4	
8-	5	
8-	6	
8-	7	CREATION DATE:
8-	8	MODIFICATIONS DATE:
9-	73	CACKE- CREATE ACKNOWLEDGEMENT RECORD-
10-	155	CDSTAT- CALCULATE DOCUMENT STATISTICS-
11-	290	CINDEXE- CREATE INDEX ENTRY-
12-	362	CIPRE- CREATE IPR ENTRY-
13-	456	CLEARB- CLEAR ONE SECTOR BUFFER-
13-	471	CLOSEF- CLOSE FDB FILES-
14-	487	CMP2ID- COMPARE 2 DOCUMENT IDS-
15-	525	CMP3WD- COMPARE THREE WORD ITEMS-
16-	581	EXIT- EXIT FOR FATAL ERROR-
16-	592	ISSDIR- ISSUE DIRECTIVE-
17-	619	LINDEXE- LOCATE INDEX ENTRY FOR A DOCUMENT-
18-	707	LIPRE- LOCATE IPR ENTRY-
19-	819	PPDATA- PUT PACKED DATA-
20-	973	WHTSPC-
21-	992	PACK6 - PACK 6-BIT CHARS-
22-	1044	READ- READ INPUT FILE-
23-	1100	READF- READ FIRST BLOCK OF AN ACTIVE FCS FILE-
23-	1123	READN- READ NEXT SECTOR-
24-	1151	SDATA- SAVE DATA-
25-	1189	RSTEOC-
26-	1220	RDEOC-
27-	1252	STORE- STORE CHARACTER-
28-	1348	UNPACK- UNPACK DATA-BASE DOCUMENT-
29-	1456	WRITDB- WRITE DATA-BASE-
29-	1489	WRTRTD- WRITE OUTPUT BUFFER-

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - DBU SUBROUTINES.
TASK NAME: DBUPD- DATA BASE UPDATE

MACRO M1110 27-MAR-69 14:31 PAGE 8

1
2
3
4
5
6
7
8
9
10 000000
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

```

.SBTTL TASK NAME: DBUPD- DATA BASE UPDATE
.SBTTL
.SBTTL
.SBTTL
.SBTTL
.SBTTL
.SBTTL CREATION DATE:
.SBTTL MODIFICATIONS DATE:
.TITLE DBUSBR - DBU SUBROUTINES
.PSECT
.ENABL AMA
.NLIST MEB
.MCALL FINIT$,FSR52$,FDBDF$,FDAT$,FDRC$,FDBK$,FDOP$A
.MCALL OFNB$,OFNB$,READ$,WAIT$,WRITE$,CLOSE$,DELET$
.MCALL RCVD$,QIOW$,DIR$,EXIT$,WTSE$,CLEF$,SDAT$
.MCALL RSUM$,SETF$,DECL$SPL
;
; LOCAL MACROS
;
; MACRO UNPAK - IT UNPACKS A 6-BIT CHARACTER CONTAINED IN R2
;
.MACRO UNPAK,?L1,?L2
BIC #177700,R2
JSR PC,STORE ;STORE CHARACTER
BCC L2
JMP U,2 ;SUB-DOC TO BE DELETED IS FOUND
L2: INC UNPTMP+6 ;INC PACKED BYTE COUNT
SUB #1,UPCCT+WORD1 ;DECREMENT UNPACKED CHARACTER COUNT
SBC UPCCT
;
BGT L1 ;MORE TO UNPACK
TST UPCCT+WORD1 ;HIGH ORDER 0, CHECK LOW ORDER
BHI L1 ;MORE TO UNPACK
JMP U,15 ;DONE
L1:
.ENDM
;
;
;

```

```

73 .SBTTL CACKE-- CREATE ACKNOWLEDGEMENT RECORD.
74 ;
75 ; CACKE CREATES A 6 WORD ACKNOWLEDGEMENT ENTRY FOR THE DOCUMENT ID
76 ; THAT HAS JUST BEEN PROCESSED. THE ENTRY CONTAINS THE 3-WORD DOCUMENT ID,
77 ; A WORD INDICATING THE ACTION REQUESTED, A WORD INDICATING THE STATUS OF THE
78 ; ATTEMPT TO CARRY OUT THAT ACTION AND A PAD-FILL WORD.
79 ; REQUESTS ADD-DOC, ADD-SUB-DOC AND REPLACE-DOC ALL CONTAIN REQUESTS THAT
80 ; AFFECT A SINGLE DOCUMENT AND HENCE WILL CAUSE ONLY A SINGLE ENTRY IN
81 ; THE ACK-BUFFER. FOR THESE REQUESTS, CACKE FIRST RELEASES THE INPUT
82 ; BUFFER CURRENTLY IN USE, NOTIFIES CCIN, AND IF A MULTI-BLOCK EXCHANGE,
83 ; DEQUEUES THE RECEIVE QUEUE ENTRY (OR WAITS FOR ONE) CONTAINING THE
84 ; IDENTIFIER OF THE GLOBAL COMMON BUFFER WHICH CONTAINS THE NEXT
85 ; BLOCK OF DATA. FOR THE OTHER REQUESTS, WHICH MAY AFFECT MULTIPLE
86 ; DOCUMENTS, CCIN RELEASES THE CURRENT BUFFER ONLY WHEN ALL
87 ; DOCUMENT ID'S HAVE BEEN PROCESSED (NDOC = 0).
88 ;
89 ; ON ENTRY, ACKBUF CONTAINS THE 6-WORD ACK DATA.
90 ;
91 CACKE:: TST NDOC ; IF NDOC IS NOT ZERO, DON'T.
92 BNE CAC010 ; SHOW THAT BUFFER IS FREE
93 MOV SAVEPT,R1 ; R1-> DOUBLE-BUFFER CONTROL BLOCK
94 BIC WORD1(R1),MSQ ; SHOW BUFFER IS NOW FREE
95 RSUM#C CCIN ; RESUME CCIN IF NECESSARY
96 TST INCONT ; ALL BLOCKS FOR THIS MSG INPUT?
97 BLE CAC010 ; YES
98
99 3$:
100 RCVD#S ,#PRB ; GET NEXT BUFFER POINTER
101 BCC 1$ ;
102 CMP #DSW,#IE,ITS ; NO DATA QUEUED?
103 BNE 2$ ; FALSE, DATA AVAILABLE
104 WTSE#S #CF,UPD ; WAIT FOR SINGLE EVENT FLAG
105 CLEF#S #CF,UPD ;
106 BR 3$ ;
107
108 2$:
109 MOV #DSW,PAR1 ;
110 MOV #MSG1,R1 ;
111 CALL ERROR ;
112 CALL EXIT ;
113
114 1$:
115 JSR PC,READ ; GET THE BUFFER
116 BR CACKE ; GO SEE IF ALL INPUT
117
118 CAC010:
119 SAVE R3,R4 ;
120 MOV #ACKBUF,R3 ; R3->6-WORD ACK ENTRY
121 MOV ACKPTR,R5 ; R5->NEXT ENTRY IN ACK BUFFER
122 MOV #6,R4 ; R4 = LOOP COUNTER
123 ;
124 ; MOVE DATA FROM ACKBUF TO ACK-BUFFER ONE WORD AT A TIME
125 ;
126 2$:
127 MOV (R3)+,(R5)+ ; MOVE ONE WORD TO ACK-BUFFER
128 CMP R5,#DBACKB+SLB ; END OF BUFFER REACHED?
129 BLO 1$ ; BRANCH IF NO
130 ;
131 ; BUFFER IS FULL: WRITE IT TO FILE, CLEAR DBACKB AND RESET R5
132 ;
133 MOV #ACKFDB,R0 ; R0-> FDB
134 CALL WRTFDB
    
```

```
130 000202 012705 000000G.      MOV.   #DBACKB,R5      ;R5-> NEXT ENTRY IN DBACKB.
131 000206                      1$:   SSOB.   R4,2$      ;LOOP.
132.                               ;
133.                               ;   ACK LOADED INTO ACK BUFFER
134.                               ;
135 000212 010537 000000G.      MOV.   R5,ACKPTR      ;SAVE ADDRESS OF NEXT ENTRY IN DBACKB.
136 000216 062737 000001 000000G.  ADD.   #1,ACKECT      ;INCREMENT ACK EXCHANGE'S ENTRY COUNT.
137 000224 023727 000000G 000251  CMP.   ACKECT,#<<N:BUFW-3-6>/6> ;ROOM FOR ANOTHER ACK?
138 000232 003410                      BLE.   3$              ;BRANCH IF YES.
139.                               ;
140.                               ;   ACK EXCHANGE FULL. WRITE LAST BLOCK AND TRANSMIT EXCHANGE TO MASTER.
141.                               ;
142 000234 012700 000000G.      MOV.   #ACKFDB,R0
143 000240                      CALL.  WRTRD.
144 000244                      CALL.  XMTACK.
145.                               ;
146.                               ;   OPEN ANOTHER ACK FILE.
147.                               ;
148 000250                      CALL.  OPEACK.
149.                               ;
150.                               ;   RETURN
151.                               ;
152 000254                      3$:   RESTOR. R3,R4
153 000260 000207                      RTS.   PC
```

```

155 ; SBTTL CDSTAT - CALCULATE DOCUMENT STATISTICS.
156 ;
157 ; CDSTAT CALCULATES NUMBER OF SECTORS TO READ FOR THE PACKED DOCUMENT, THE
158 ; NUMBER OF PACKED WORDS IN THE DOCUMENT (INCLUDING WHITE SPACE), AND THE
159 ; NUMBER OF UNPACKED TEXT BYTES TO RESULT FROM UNPACKING THE DOCUMENT.
160 ;
161 ; - NUMBER OF SECTORS TO READ FOR THE PACKED DOC. (NS2RD)
162 ; - START ADDS (OFFSET, DISK ADDS), OF NEXT DOC. (WDQFND)
163 ; (NDCADR)
164 ; (NDCADR+2)
165 ; - PACKED WORD COUNT (TEXT AND WHITE SPACE) (PWCT)
166 ; (PWCT+2)
167 ; - SECTOR ADD'S OF END-OF-TEXT. (EOT)
168 ; (EOT+2)
169 ; - UNPACKED TEXT BYTE COUNT. (UPCCT)
170 ; (UPCCT+2)
171 ; ON ENTRY,
172 ; R2 -> INDEX RECORD.
173 ; R1 -> DOC'S INDEX ENTRY
174 ;
175 000262. CDSTAT:
176 000262. SAVE R3,R4,R5 ;
177 000270 010103 MOV R1,R3 ; R3 -> INDEX ENTRY
178 000272 062703 000000G ADD #N,BXE,R3 ; GET ADDR OF NEXT ENTRY
179 000276 010204 MOV R2,R4 ; GET ADDR OF END OF CURRENT
180 000300 062704 000000G ADD #SLB,R4 ; -- INDEX RECORD
181 000304 020304 CMP R3,R4 ; NEW ADDR WITHIN INDEX RECORD?
182 000306 002003 BGE CDS010 ; NO, BEYOND RECORD
183 000310 010304 MOV R3,R4 ; YES (R4 -> NEXT INDEX ENTRY)
184 000312 000137 000470' JMP CDS035 ;
185 000316 CDS010:
186 000316 020427 000000G CMP R4,#IPRBUF+LNSBUF ; LAST INDEX RECORD IN BUFFER?
187 000322 002004 BGE CDS020 ; YES
188 000324 062704 000000G ADD #IN,FE,R4 ; NO, POINT TO 1ST ENTRY IN NEXT X-RC'D
189 000330 000137 000470' JMP CDS035 ;
190 000334 CDS020:
191 000334 012737 000000G 000000G MOV #IO,RLB,DQIO+0,IOFN ; SET UP FOR READ OF NEXT INDEX RCD
192 000342 012737 000000G 000000G MOV #WBUF,DQIO+0,IOPL ; WORKING STORAGE
193 000350 012737 000000G 000000G MOV #SLB,DQIO+0,IOPL+2 ;
194 000356 013737 000000G 000000G MOV IDXADR,DQIO+0,IOPL+6 ; ADDR OF CURRENT INDEX RECORD
195 000364 013737 000002G 000000G MOV IDXADR+WORD1,DQIO+0,IOPL+10 ;
196 000372 062737 000001 000000G ADD #1,DQIO+0,IOPL+10 ; ADDR OF NEXT INDEX RECORD
197 000400 005537 000000G ADC DQIO+0,IOPL+6 ;
198 000404 004737 002476' JSR PC,ISSDIR ; ISSUE READ FOR NEXT INDEX RECORD
199 000410 005737 000000G TST DRSTAT ; GOOD READ?
200 000414 001404 BEQ CDS025 ; YES
201 000416 052737 000000G 000000G BIS #AS,URD,STATWD ; SET ERROR CODE
202 000424 000207 RTS PC ;
203 000426 CDS025:
204 000426 123727 000000G 000000G CNFB IOST,#IS,SUC ; READ SUCCESSFUL?
205 000434 001413 BEQ CDS030 ; YES
206 000436 113737 000000G 000000G MOVB @#IOST,PAR1 ; READ ERROR
207 000444 012701 000000G MOV #EMSG2,R1
208 000450 CALL ERROR
209 000454 052737 000000G 000000G BIS #AS,URD,STATWD ;
210 000462 000207 RTS PC ;
211 000464 CDS030:

```

```

212 000464 012704 000000C MOV. #WBUF+IN.FE.R4 ; ADDR OF 1ST ENTRY IN NEXT INDEX RECORD.
213 000470 CDS035:
214 000470 016437 000000G 000000G MOV. I.SANH(R4),TEMP ; ADDR OF NEXT DOCUMENT ON DISK
215 000476 042737 177400 000000G BIC. #177400,TEMP ; CLEAR UPPER BYTE
216 000504 013737 000000G 000000G MOV. TEMP,NDCADR ; STORE HIGH ADDR OF NEXT DOCUMENT
217 000512 016437 000000G 000002G MOV. I.SANH(R4),TEMP+WORD1 ; LOW ORDER ADDR OF NEXT DOCUMENT
218 000520 016437 000000G 000002G MOV. I.SANH(R4),NDCADR+WORD1 ; STORE LOW ORDER ADDR
219 000526 016105 000000G MOV. I.SANH(R1),R5 ; HIGH ADDR OF DESIRED DOCUMENT
220 000532 042705 177400 BIC. #177400,R5
221 000536 166137 000000G 000002G SUB. I.SANH(R1),TEMP+WORD1 ; CALCULATE # OF DOCUMENT SECTORS
222 000544 005637 000000G SBC. TEMP
223 000550 160537 000000G SUB. R5,TEMP
224 000554 013737 000002G 000000G MOV. TEMP+WORD1,NS2RD ; # OF SECTORS TO READ
225 000562 016437 000000G 000000G MOV. I.SANH(R4),WDOFND ; GET WORD OFFSET START FOR 2ND DOCUMENT
226 000570 000337 000000G SWAB. WDOFND ; GET OFFSET IN LOWER BYTE
227 000574 042737 177400 000000G BIC. #177400,WDOFND ; OFFSET
228 ;
229 ; CALCULATE WORD COUNT OF PACKED TEXT AND WHITE SPACE IN THE DOCUMENT.
230 ;
231 000602 010137 000000G MOV. R1,XTEMP ; XTEMP-> DOC'S INDEX ENTRY
232 000606 013703 000000G MOV. NS2RD,R3 ; # OF SECTORS TO READ
233 000612 005002 CLR. R2 ; CLEAR UPPER WORD FOR MULTIPLY
234 000614 012700 000000G MOV. #SLW,R0 ; WORD LENGTH OF ONE SECTOR
235 000620 004737 000000G JSR. PC,#DMUL ; TOTAL WORDS IN ALL SECTORS
236 000624 163701 000000G SUB. WDOFF,R1 ; SUBTRACT PREVIOUS DOCUMENT'S WORDS FROM
237 000630 005600 SBC. R0 ; -- TOTAL WORD COUNT
238 000632 005237 000000G INC. NS2RD ; ACCOUNT FOR LAST SECTOR
239 000636 063701 000000G ADD. WDOFND,R1 ; ADD REMAINING WORDS OF DOC. FROM LAST SECTOR
240 000642 005500 ADC. R0 ; ADJUST FOR CARRY
241 000644 010037 000000G MOV. R0,PWCT ; PACKED WORD COUNT (HI)
242 000650 010137 000002G MOV. R1,PWCT+WORD1 ; " " " (LO)
243 ;
244 ; CALCULATE END-OF-TEXT ADDRESS
245 ;
246 000654 SAVE. R0,R1
247 000660 013702 000000G MOV. XTEMP,R2 ; R2-> DOC'S INDEX ENTRY
248 000664 166201 000000G SUB. I,UWS(R2),R1 ; SUBTRACT UNUSED WHITE SPACE
249 000670 005600 SBC. R0
250 000672 063701 000000G ADD. WDOFF,R1 ; ADD WORD OFFSET OF DOC. START
251 000676 005500 ADC. R0
252 000700 010102 MOV. R1,R2 ; R2 = LO DIVIDEND
253 000702 010001 MOV. R0,R1 ; R1 = HI
254 000704 012700 000000G MOV. #SLW,R0 ; R0 = DIVISOR (WORD LENGTH OF A SECTOR)
255 000710 004737 000000G JSR. PC,#DDIV ; ON RETURN, R2=LOW QUOTIENT
256 ; R1=HI
257 ; R0=REMAINDER
258 000714 013737 000000G 000000G MOV. DOCADR,EOT ; EOT = DISK ADDS OF START-OF-TEXT (HI)
259 000722 013737 000002G 000002G MOV. DOCADR+2,EOT+2 ; (LO)
260 000730 060237 000002G ADD. R2,EOT+2 ; EOT = DISK ADDS OF END-OF-TEXT (LO)
261 000734 005537 000000G ADC. EOT
262 000740 060137 000000G ADD. R1,EOT ; (HI)
263 000744 110037 000001G MOV. R0,EOT+1 ; UPPER BYTE=WORD OFFSET OF EOT
264 000750 RESTOR. R0,R1
265 ;
266 ; CALCULATE NUMBER OF UNPACKED BYTES BY MULTIPLYING PACKED BYTES BY 4/3
267 ;
268 000754 010002 MOV. R0,R2

```


DBUSBR - D SUBROUTINES. MACRO M1110 27-MAR-68 14:31 BY 18-2
 CDSTAT - CALCULATE DOCUMENT STATISTICS. Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

269 000756 010103          MOV.   R1,R3          ;
270 000760 012700 000002  MOV.   #2,R0         ;
271 000764 004737 000000G. JSR.   PC,$DMUL.     ; CONVERT TO PACKED BYTES.
272.                                     ;
273 000770 010002.        MOV.   R0,R2.        ;
274 000772 010103        MOV.   R1,R3          ;
275 000774 012700 000004  MOV.   #4,R0         ;
276 001000 004737 000000G. JSR.   PC,$DMUL.     ; MULTIPLY BY 4
277 001004 010102.        MOV.   R1,R2.        ;
278 001006 010001        MOV.   R0,R1         ;
279 001010 012700 000003  MOV.   #3,R0         ;
280 001014 004737 000000G. JSR.   PC,$DDIV.     ; DIVIDE BY 3
281 001020 010137 000000G. MOV.   R1,UPCCT.     ;
282 001024 010237 000002G. MOV.   R2,UPCCT+WORD1 ; TOTAL UNPACKED BYTE COUNT
283 001030 163737 000000G.000002G. SUB.   UNWS,UPCCT+WORD1 ; MINUS UNUSED WHITE SPACE BYTE COUNT
284 001036 005637 000000G. SBC.   UPCCT.        ; TOTAL UNPACKED TEXT BYTE COUNT
285 001042 162737 000015 000002G. SUB.   #13,UPCCT+WORD1 ; SUBTRACT DOCUMENT START SEQUENCE CHAR. COUNT
286 001050 005637 000000G. SBC.   UPCCT.        ; TOTAL TEXT CHARACTER COUNT
287 001054          RESTOR. R3,R4,R5     ;
288 001062 000207          RTS.   PC.           ;
  
```

DBUSBR - DBU SUBROUTINES.
CINDEXE - CREATE INDEX ENTRY.

MACRO M1110 27-MAR-80 14:31 PAGE 11

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL CINDEXE- CREATE INDEX ENTRY.
;
; CREATE THE INDEX ENTRY FOR THE NEW DOCUMENT BEING ADDED. PLACE THE
; ADDRESS OF THE END-OF-COLLECTION MARK IN THE NEXT INDEX ENTRY.
;
295 001064 CINDEXE:
296 001064 012737 000000G-000000C MOV #10,RLB,DQ10+0,IOFN ; READ LATEST INDEX RECORD.
297 001072 012737 000000G-000000C MOV #RDBUF,DQ10+0,IOPL ; READ BUFFER FOR INDEX RECORD
298 001100 012737 000000G-000000C MOV #SLB,DQ10+0,IOPL+2 ; BUFFER SIZE.
299 001106 013737 000000C-000000C MOV SREC+SR,LIN,DQ10+0,IOPL+6 ;
300 001114 013737 000000C-000000C MOV SREC+SR,LIN+WORD1,DQ10+0,IOPL+10 ;
301 001122 004737 002476' JSR PC,ISSDIR ; ISSUE READ.
302 001126 005737 000000G TST DRSTAT ; GOOD READ?
303 001132 001404 BEQ CINDEX0 ; YES.
304 001134 CINDEX05:
305 001134 012737 000000G-000000G MOV #AS,URD,STATWD ; SET ERROR CODE.
306 001142 000207 RTS PC ;
307 001144 CINDEX10:
308 001144 012705 000000C MOV #RDBUF+IN,FE,R5 ; ADDR OF 1ST INDEX ENTRY.
309 001150 013701 000000C MOV RDBUF+IN,CEC,R1 ; # OF ENTRIES IN INDEX RECORD.
310 001154 001405 BEQ 10$ ; NO ENTRIES AS YET.
311 001156 012700 000000G MOV #N,BXE,R0 ; # OF BYTES/INDEX ENTRY.
312 001162 004737 000000G JSR PC,$MUL ; OFFSET TO NEXT AVAILABLE ENTRY.
313 001166 000105 ADD R1,R5 ; ADDR OF NEXT INDEX ENTRY.
314 001170 10$:
315 001170 013765 000000G-000000G MOV DOCID,I,BIDH(R5) ; HIGH ORDER DOC. ID.
316 001176 013765 000002G-000000G MOV DOCID+WORD1,I,BIDM(R5) ; MIDDLE ORDER DOC. ID.
317 001204 013765 000004G-000000G MOV DOCID+WORD2,I,BIDL(R5) ; LOW ORDER DOC. ID.
318 001212 013765 000000G-000000G MOV ROWS,I,RWS(R5) ; REQUIRED WHITE SPACE.
319 001220 013765 000000G-000000G MOV UNWS,I,UWS(R5) ; UNUSED WHITE SPACE.
320 001226 005237 000000C INC RDBUF+IN,CEC ; INCREMENT INDEX ENTRY COUNT.
321 001232 013765 000000G-000000G MOV DOCADR,I,SANH(R5) ; PLACE DOCUMENT ADDR IN INDEX ENTRY
322 001240 013765 000002G-000000G MOV DOCADR+WORD1,I,SANL(R5) ;
323 001246 023727 000000C-000000G CMP RDBUF+IN,CEC,#MECIND ; INDEX RECORD FULL?
324 001254 002004 BGE CINDEX20 ; YES.
325 001256 062705 000000G ADD #N,BXE,R5 ; GET ADDR OF NEXT ADDR IN RECORD.
326 001262 000137 JMP CINDEX0 ;
327 001266 CINDEX20:
328 001266 012737 000000G-000000C MOV #10,ULB,DQ10+0,IOFN ; WRITE FUNCTION CODE.
329 001274 004737 002476' JSR PC,ISSDIR ; WRITE LAST INDEX RECORD.
330 001300 005737 000000G TST DRSTAT ; GOOD WRITE?
331 001304 001404 BEQ CINDEX30 ; YES.
332 001306 CINDEX25:
333 001306 012737 000000G-000000G MOV #AS,UWD,STATWD ; SET ERROR CODE.
334 001314 000207 RTS PC ; RETURN
335 001316 CINDEX30:
336 001316 062737 000001 000000C ADD #1,DQ10+0,IOPL+10 ; GET NEXT INDEX RECORD SECTOR #.
337 001324 005537 000000C ADC DQ10+0,IOPL+6 ; ACCOUNT FOR CARRY.
338 001330 012737 000000G-000000C MOV #10,RLB,DQ10+0,IOFN ; READ NEXT INDEX RECORD.
339 001336 004737 002476' JSR PC,ISSDIR ; ISSUE READ.
340 001342 005737 000000G TST DRSTAT ; GOOD READ?
341 001346 001402 BEQ CINDEX40 ; YES.
342 001350 000137 001134' JMP CINDEX05 ; NO.
343 001354 CINDEX40:
344 001354 123727 000000C-000000G CMPB RDBUF+IN,TYP,#R$IND ; INDEX RECORD?
345 001362 001404 BEQ 40$ ; YES.
346 001364 005037 000000G CLR STATWD ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - D SUBROUTINES.
CINDEXE - CREATE INDEX ENTRY.

MACRO M1110 27-MAR-88 14:31 P. 11-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
347      : >>>> CALL FOR A REORGANIZATION & EXIT.  
348 001370 000137 001371'      JMP      .+1      ;  
349 001374      40$:  
350 001374 012705 000000C      MOV      #RDBUF+IN.FE,R5 ; NEXT INDEX ADDR IN NEW INDEX RECORD.  
351 001400 013737 000000C-000000C      MOV      DQ10+Q,IOPL+6,SREC+SR.LIN ; SET LATEST INDEX RECORD.  
352 001406 013737 000000C-000000C      MOV      DQ10+Q,IOPL+10,SREC+SR.LIN+WORD1 ;  
353 001414      CIN050:  
354 001414 013765 000000C-000000G      MOV      SREC+SR.ECH,I,SANH(R5) ; ADDR FOR START OF NEXT NEW DOC.  
355 001422 013765 000000C-000000G      MOV      SREC+SR.ECL,I,SANL(R5) ;  
356 001430 012737 000000G-000000C      MOV      #IO.WLB,DQ10+Q,IOFN ; WRITE LATEST INDEX RECORD.  
357 001436 004737 002476'      JSR      PC,ISSDIR ; ISSUE READ.  
358 001442 005737 000000G      TST      DRSTAT ; GOOD READ?  
359 001446 001317      BNE      CIN025 ; NO.  
360 001450 000207      RTS      PC ; YES.
```

DBUSBR - DBU-SUBROUTINES
CIPRE- CREATE IPR-ENTRY

MACRO.M1110 27-MAR-80 (4:7) PAGE 12

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL CIPRE- CREATE IPR-ENTRY
:
: CIPRE-CREATES-AN IPR-ENTRY IF THE-APPROPRIATE-NUMBER-OF INDEX-
: RECORDS-HAS-BEEN REPRESENTED- IN THE-IPR.
:
CIPRE::
362.
363.
364.
365.
366.
367 001452.
368 001452. 012737 000000G.000000C. MOV. #IO.RLB,DQIO+Q.IOFN. ; READ-FUNCTION-CODE
369 001460 012737 000000G.000000C. MOV. #IPRBUF,DQIO+Q.IOPL. ; IPR-BUFFER
370 001466 012737 000000G.000000C. MOV. #SLB,DQIO+Q.IOPL+2. ; READ-ONLY-ONE-SECTOR
371 001474 013737 000000C.000000C. MOV. SREC+SR.LIP,DQIO+Q.IOPL+6 ; ADDR-OF-LATEST-INDEX-POINTER-RECORD
372 001502. 013737 000000C.000000C. MOV. SREC+SR.LIP+WORD1,DQIO+Q.IOPL+10 ;
373 001510 004737 002476' JSR. PC,ISSDIR. ; ISSUE-READ
374 001514 005737 000000G. TST. DRSTAT. ; GOOD-READ?
375 001520 001406 BEQ. CIP010 ; YES
376 001522.
377 001522. 005037 000000G. CLR. STATWD. ;
378 001526 052737 000000G.000000G. BIS. #AS.URD,STATWD ; SET-ERROR-CODE
379 001534 000207 RTS. PC. ;
380 001536
CIP005:
381 001536 013737 000000C.000000G. MOV. SREC+SR.LIN,TEMP ; STORE-ADDR-OF-LATEST-INDEX-RECORD
382 001544 013737 000000C.000002G. MOV. SREC+SR.LIN+WORD1,TEMP+WORD1 ;
383 001552. 005737 000000C. TST. RDBUF+IN.CEC. ; LATEST-INDEX-RECORD-CONTAIN-ANY-ENTRIES?
384 001556 003011 BGT. 10$ ; YES
385 001560 023737 000002G.000000C. CMP. TEMP+WORD1,SREC+SR.IIN+WORD1 ; FIRST-INDEX-RECORD?
386 001566 001405 BEQ. 10$ ; YES
387 001570 162737 000001 000002G. SUB. #1,TEMP+WORD1 ; USE-ADDR-OF-PREVIOUS-INDEX-RECORD
388 001576 005637 000000G. SBC. TEMP ; CARRY-ADJUST
389 001602.
390 001602. 023737 000000G.000000C. 10$: CMP. TEMP,IPRBUF+IP.LIA ; COMPARE-LAST-INDEX-RCD-REPRESENTED
391. ; --- TO LATEST-INDEX-RECORD
392 001610 101012. BHI. CIP020 ; NEW-INDEX-RECORD
393 001612. 103407 BLO. CIP015 ; SEVERE-PROBLEM-ENCOUNTERED
394 001614 023737 000002G.000000C. CMP. TEMP+WORD1,IPRBUF+IP.LIA+WORD1 ; CMP-LOW-WORDS
395 001622. 101005 BHI. CIP020 ; NEW-INDEX-RECORD-TO-REPRESENT
396 001624 103402. BLO. CIP015 ; SEVERE-PROBLEM
397 001626 000137 002056' JMP. CIP000 ; SAME-INDEX-RECORD-AS-LAST-TIME
398 001632.
CIP015:
399. ; >>>> LATEST-INDEX-RECORD-IS-LESS-THAN-LAST-INDEX-RECORD-REPRESENTED-IN-IPR.
400 001632. 000137 001633' JMP. +1
401 001636
CIP020:
402 001636 023737 000000C.000000G. CMP. IPRBUF+IP.SRL,DELTA ; TIME-TO-ESTABLISH-NEW-IPR-ENTRY
403 001644 002010 BGE. 20$ ; YES
404 001646 005737 000000C. TST. IPRBUF+IP.LIA+WORD1 ; FIRST-DOCUMENT?
405 001652. 001003 BNE. 2$ ; NO
406 001654 005737 000000C. TST. IPRBUF+IP.LIA. ; CHECK-HIGHER-WORD
407 001658 001402. BEQ. 20$ ; FIRST-DOCUMENT, SET-IPR-AS-IF-IT-IS-TIME-FOR-NEW-ENTRY
408 001662.
2$:
409 001662. 000137 002036' JMP. CIP070 ; NO
410 001666
20$:
411 001666 023727 000000C.000000G. CMP. IPRBUF+IP.CEC,#MECIPR. ; IPR-FULL?
412 001674 002014 BGE. CIP030 ; IPR-IS-FULL, GET-NEXT-ONE
413 001676 013701 000000C. MOV. IPRBUF+IP.CEC,R1 ; #-OF-CURRENT-ENTRIES
414 001702. 012700 000000G. MOV. #N,BIPE.R0 ; #-OF-BYTES/IPR-ENTRY
415 001706 004737 000000G. JSR. PC,#MUL. ; OFFSET-FROM-1ST-ENTRY
416 001712. 062701 000000C. ADD. #IPRBUF+IP.FE,R1 ; ADDR-OF-NEXT-AVAILABLE-ENTRY
417 001716 005037 000000C. CLR. IPRBUF+IP.SRL. ; CLEAR-#-OF-SECTORS-REPRESENTED-IN-LAST-ENTRY
418 001722. 000137 001776' JMP. CIP050 ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - D SUBROUTINES.
CIPRE- CREATE IPR-ENTRY.

MACRO M1110 27-MAR-80 14:31 P 12-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
419 001726          CIP030:
420 001726 062737 000001 000000C.  ADD.    #1,DQ10+Q,IOP1+10 ; GET ADDR OF NEXT IPR SECTOR.
421 001734 005537 000000C.  ADC.    DQ10+Q,IOP1+6      ;
422 001740 004737 002476'  JSR.    PC,ISSDIR.      ; ISSUE READ OF NEXT IPR.
423 001744 005737 0000000C.  TST.    DRSTAT.        ; GOOD READ?
424 001750 001402.  BEQ.    CIP040         ; YES.
425 001752 000137 001522'  JMP.    CIP005         ;
426 001756          CIP040:
427 001756 132737 000000G 000000C.  BITB.   #R#IPR,IPRBUF+IP,TYP. ; IS THIS AN IPR?
428 001764          BON.    CIP050         ; YES.
429          : >>>> CALL FOR AN REORGANIZATION AND EXIT.
430 001766 000137 001767'  JMP.    .+1           ;
431 001772          CIP050:
432 001772 012701 000000C.  MOV.    #IPRBUF+IP,FE,R1 ; FIRST ENTRY ADDR IN NEW IPR.
433 001776          CIP060:
434 001776 013711 000000G.  MOV.    DOCID,(R1)      ; PLACE DOC. ID IN ENTRY.
435 002002 013761 000002G 000002.  MOV.    DOCID+WORD1,WORD1(R1) ;
436 002010 013761 000004G 000004.  MOV.    DOCID+WORD2,WORD2(R1) ;
437 002016 005237 000000C.  INC.    IPRBUF+IP,CEC.  ; INCREMENT CURRENT ENTRY COUNT.
438 002022 013737 000000C 000000C.  MOV.    DQ10+Q,IOP1+6,SREC+SR,LIP. ; UPDATE LATEST IPR SECTOR.
439 002030 013737 000000C 000000C.  MOV.    DQ10+Q,IOP1+10,SREC+SR,LIP+WORD1 ;
440 002036          CIP070:
441 002036 005237 000000C.  INC.    IPRBUF+IP,SRL.  ; INCREMENT # OF SECTORS REPRESENTED.
442 002042 013737 000000G 000000C.  MOV.    TEMP,IPRBUF+IP,LIA. ; PLACE LATEST INDEX RECORD IN LATEST SECTOR REP.
443 002050 013737 000002G 000000C.  MOV.    TEMP+WORD1,IPRBUF+IP,LIA+WORD1 ;
444 002056          CIP080:
445 002056 013737 000000G 000000C.  MOV.    DOCID,IPRBUF+IP,HID. ; PLACE DOC. ID IN HIGHEST DOC. REPRESENTED.
446 002064 013737 000002G 000000C.  MOV.    DOCID+WORD1,IPRBUF+IP,HID+WORD1 ;
447 002072 013737 000004G 000000C.  MOV.    DOCID+WORD2,IPRBUF+IP,HID+WORD2 ;
448 002100 012737 000000G 000000C.  MOV.    #10,ULB,DQ10+Q,IOFN. ; WRITE FUNCTION CODE.
449 002106 004737 002476'  JSR.    PC,ISSDIR.      ; ISSUE WRITE.
450 002112 005737 000000G.  TST.    DRSTAT.        ; GOOD READ?
451 002116 001403.  BEQ.    CIP090         ;
452 002120 012737 000000G 000000C.  MOV.    #AS,UWD,STATWD ; SET ERROR CODE.
453 002126          CIP090:
454 002126 000207          RTS.    PC.            ;
```

```

456 .SBTTL CLEARB- CLEAR ONE SECTOR BUFFER.
457 ;
458 ; THIS ROUTINE CLEARS A BUFFER OF ONE SECTOR LENGTH.
459 ;
460 ; R0: HOLDS ADDRESS OF BUFFER TO BE CLEARED.
461 ;
462 002130 CLEARB::
463 002130 012737 000000G 000000G. MOV. #SLB,R0. ; SECTOR BYTE LENGTH.
464 002136 105020 1$: CLR. (R0)+ ; CLEAR BUFFER.
465 002140 005337 000000G. DEC. R0. ;
466 002144 003374 BGT. 1$ ; CONTINUE.
467 002146 000207 RTS. PC. ; DONE.
468 ;
469 ;
470 ;
471 .SBTTL CLOSEF- CLOSE FDB FILES.
472 ;
473 ;
474 ; R0 CONTAINS THE ADDRESS OF THE FDB TO BE CLOSED.
475 ;
476 ;
477 002150 CLOSEF::
478 002150 CLOSE$ ; CLOSE THE OPEN FILE.
479 002154 103012. BCC. 10$ ; SUCCESSFUL.
480 002156 116002. 000000G. MOVB. F.ERR(R0),R2. ; ERROR CODE.
481 002162 010237 000000G. MOV. R2,PAR1
482 002166 012701 000000G. MOV. #MSG0,R1
483 002172. CALL. ERROR.
484 002176. CALL. EXIT.
485 002202. 000207 10$: RTS. PC. ;
    
```

DBUSBR - DUB SUBROUTINES.
CMP2ID- COMPARE 2 DOCUMENT IDS

MACRO M1110 27-MAR-80 14:31:44
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL - CMP2ID- COMPARE 2 DOCUMENT IDS.  
:  
: CMP2ID WILL COMPARE 2 THREE-WORD DOCUMENT IDS AND INDICATE THE RESULT OF  
: THE COMPARE IN CMPST.  
:  
: CMPST = -1 WHEN 1ST ID IS < 2ND ID ( THEREFORE IN THE GIVEN RECORD )  
: = 0 WHEN THE IDS ARE EQUAL.  
: = +1 WHEN 1ST ID IS > 2ND ID ( THEREFORE NOT IN THE GIVEN RECORD )  
:  
: CMP2ID::  
497 002204          SAVE R1,R2,R3  
498 002204          CLR  CMPST          ; CLEAR COMPARE STATUS FLAG.  
499 002212 005037 0000006 MOV  WORD4(SP),R2      ; ADDR OF 2ND ID.  
500 002216 016602 000010 MOV  WORD5(SP),R1      ; ADDR OF 1ST ID.  
501 002222 016601 000012 MOV  (R2),R3          ; GET 1ST WORD OF SECOND ID.  
502 002226 011203          BIC  #BIT15,R3        ; CLEAR PURGE INDICATOR IF SET.  
503 002230 042703 100000 CMP  (R1),R3          ; COMPARE 1ST WORDS OF IDS.  
504 002234 021103          BHI  1$              ; 1ST ID > 2ND ID.  
505 002236 101014          BLO  2$              ; 1ST ID DEFINITELY < 2ND ID  
506 002240 103416          CMP  WORD1(R1),WORD1(R2) ; 1ST WORDS =, CHECK 2ND WORDS OF ID.  
507 002242 026162 000002 000002 BHI  1$              ; 1ST ID > 2ND ID.  
508 002250 101007          BLO  2$              ; 1ST ID < 2ND ID.  
509 002252 103411          CMP  WORD2(R1),WORD2(R2) ; 2ND WORDS =, CHECK 3RD WORDS OF ID.  
510 002254 026162 000004 000004 BHI  1$              ; 1ST ID DEFINITELY > 2ND ID  
511 002262 101002          BLO  2$              ; 1ST ID DEFINITELY < 2ND ID  
512 002264 103404          BR   3$              ; IDS ARE EQUAL.  
513 002266 000405          1$:  
514 002270          INC  CMPST          ; 1ST ID GREATER THAN 2ND ID  
515 002270 005237 0000006 BR   3$              ;  
516 002274 000402          2$:  
517 002276          DEC  CMPST          ; 1ST ID LESS THAN 2ND ID  
518 002276 005337 0000006 3$:  
519 002302          RESTOR R1,R2,R3      ;  
520 002302          MOV  (SP),WORD2(SP)   ;  
521 002310 011666 000004 ADD  #WORD2,SP      ; CLEAR STACK.  
522 002314 062706 000004 RTS  PC              ;  
523 002320 000207
```

DBUSBR - DBU SUBROUTINES.
CMP3WD - COMPARE THREE WORD ITEMS.

MACRO M1110 32 DEC 88 1 7 PAGE 11
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL CMP3WD COMPARE THREE WORD ITEMS.  
;  
; INPUTS:  
; R1 - CONTAINS THE ADDRESS OF THE VALUE TO BE COMPARED.  
; R2 - CONTAINS THE ADDRESS OF THE RANGE START.  
; R3 - CONTAINS THE ADDRESS OF THE RANGE END.  
;  
; CMP3WD COMPARES THE VALUE IN R1 TO BOTH THE RANGE START AND THE  
; RANGE END. IF THE VALUE IN R1 SATISFIES THE GIVEN RANGE, THE COMPARE  
; STATUS WORD IS CLEARED. IF THE VALUE FALLS BELOW THE RANGE, THE STATUS  
; WORD IS SET TO A NEGATIVE ONE AND THE VALUE IS CONSIDERED INVALID.  
; IF THE VALUE FALLS ABOVE THE RANGE, THE STATUS WORD IS SET TO A POSITIVE  
; ONE AND THE VALUE MAY BE VALID OR INVALID. IF ANOTHER RANGE EXISTS, THE  
; VALUE IS CONSIDERED VALID UNTIL THE NEXT SET OF RANGES IS CHECKED. IF NO  
; NO OTHER RANGE EXISTS TO BE CHECKED, THE VALUE IS INVALID.  
;  
; OUTPUTS:  
; CMPST - -1 IF VALUE FALLS BELOW RANGE.  
; 0 IF VALUE FALLS WITHIN RANGE.  
; 1 IF VALUE FALLS ABOVE RANGE.  
;  
CMP3WD::  
546 002322. CLR CMPST ; INITIALIZE THE STATUS WORD.  
547 002322. 005037 000000G. ;  
548 ;  
549 ; COMPARE TO THE RANGE START.  
550 ;  
551 002326 021112. CMP (R1),(R2) ; COMPARE HIGH ORDER WORDS.  
552 002330 103430 BLO 2# ; INVALID FOR THIS RANGE.  
553 002332 101012. BHI 1# ; CHECK RANGE END VALUES.  
554 002334 026162. 000002 000002. CMP WORD1(R1),WORD1(R2) ; COMPARE MIDDLE WORDS.  
555 002342. 103423 BLO 2# ; INVALID FOR THIS RANGE.  
556 002344 101005 BHI 1# ; CHECK RANGE END VALUES.  
557 002346 026162. 000004 000004. CMP WORD2(R1),WORD2(R2) ; COMPARE LOW ORDER WORDS.  
558 002354 103416 BLO 2# ; INVALID FOR THIS RANGE.  
559 002356 001421 BEQ 3# ; VALUE VALID FOR THIS RANGE.  
560 ;  
561 ; COMPARE TO THE RANGE END.  
562 ;  
563 ;  
564 002360 1#:  
565 002362 103417. CMP (R1),(R3) ; COMPARE HIGH ORDER WORDS.  
566 002364 101021. BLO 3# ; VALID FOR THIS RANGE.  
567 002366 026163. 000002 000002. BHI 4# ; GREATER THAN THIS RANGE.  
568 002374 103412. CMP WORD1(R1),WORD1(R3) ; COMPARE MIDDLE WORDS.  
569 002376 101014. BLO 3# ; VALID FOR THIS RANGE.  
570 002400 026163. 000004 000004. BHI 4# ; GREATER THAN THIS RANGE.  
571 002406 101010. CMP WORD2(R1),WORD2(R3) ; COMPARE LOW ORDER WORDS.  
572 002410 000404. BHI 4# ; GREATER THAN THIS RANGE.  
573 002412. BR 3# ;  
574 002412. 012737 177777 000000G. 2#:  
575 002420 000406. MOV #-1,CMPST ; INVALID VALUE FOR THIS RANGE.  
576 002422 005037 000000G. 3#:  
577 002426 000403. CLR CMPST ; VALID VALUE FOR THIS RANGE.  
578 002430 012737 000001 000000G. 4#:  
579 002436 000207. BR 5# ;  
580 002436 000207. MOV #1,CMPST ; VALUE GREATER THAN THIS RANGE.  
581 002436 000207. RTS PC ;
```


DBUSBR - D SUBROUTINES.
EXIT- EXIT FOR FATAL ERROR.

MACRO.M1110, 27-MAR-80 14:31, P. 16
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
581  
582 002440  
583 002440 012700 000000G  
584 002444 004737 002150'  
585 002450  
586 002466  
587 002474 000207  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600 002476  
601 002476 005037 000000G  
602 002502  
603 002510 103020  
604 002512 013737 000000G 000000G  
605 002520 011637 000002G  
606 002524  
607 002544 005237 000000G  
608 002550 000423  
609 002552  
610 002552 123727 000000G 000000G  
611 002560 001417  
612 002562 113737 000000G 000000G  
613 002570 011637 000002G  
614 002574  
615 002614 005237 000000G  
616 002620  
617 002620 000207
```

```
.SBTTL-EXIT- EXIT FOR FATAL ERROR.  
EXIT::  
MOV- #ACKFDB,R0 ; ADDR OF DATA BASE ACKNOWLEDGEMENT.  
JSR- PC,CLOSEF ; CLOSE THE ACK FILE.  
MOUT#$ #EMSG15 ; ABORTING - IRRECOVERABLE ERROR.  
EXIT#$  
RTS- PC  
;  
;  
;  
;  
.SBTTL- ISSDIR- ISSUE DIRECTIVE.  
;  
;  
THIS ROUTINE ISSUES THE READ OR WRITE DIRECTIVE AFTER ALL VALUES HAVE BEEN  
SET IN THE CALLING ROUTINE.  
;  
;  
IT PERFORMS THE ERROR CHECKS AND RETURNS A NON-ZERO STATUS IF  
A PROBLEM IS FOUND.  
;  
;  
ISSDIR::  
CLR- DRSTAT ; CLEAR READ STATUS WORD.  
DIR$ #D010 ; ISSUE DIRECTIVE.  
BCC- 1$ ; SUCCESSFUL DIRECTIVE.  
MOV- $DSW,PAR1 ; BAD DIRECTIVE.  
MOV- (SP),PAR1+2 ; LOAD CALLING ROUTINE'S ADDS.  
MOUT#$ #EMSG1,#PAR1 ;  
INC- DRSTAT ; SET READ STATUS.  
BR- 2$ ;  
;  
1$:  
CMPB- IOST,#IS,SUC ; READ SUCCESSFUL?  
BEQ- 2$ ; YES.  
MOV- @#IOST,PAR1 ; BAD READ.  
MOV- (SP),PAR1+2.  
MOUT#$ #EMSG2,#PAR1 ;  
INC- DRSTAT ; SET READ STATUS.  
;  
2$:  
RTS- PC ;
```

```

        .SBTTL - LINDX- LOCATE INDEX ENTRY FOR A DOCUMENT.
619      ;
620      ;
621      ; LOCATE THE INDEX ENTRY FOR A SPECIFIED DOCUMENT ID.
622      ;
623      002622.      LINDX:
624      002622. 005037 000000G. CLR.      NSREAD.      ; CLEAR # OF SECTORS TO READ
625      002626 012737 000000G.000000C. MOV.      #IO.RLB,DQ10+0,IOFN. ; READ FUNCTION CODE
626      002634 012737 000000G.000000C. MOV.      #IPRBUF,DQ10+0,IOPL. ; BUFFER ADDRESS
627      002642. 012737 000000G.000000C. MOV.      #LNSBUF,DQ10+0,IOPL+2. ; BUFFER LENGTH
628      002650 013737 000000G.000000C. MOV.      IDXADR,DQ10+0,IOPL+6. ; ADDR OF 1ST SECTOR
629      002656 013737 000002G.000000C. MOV.      IDXADR+WORD1,DQ10+0,IOPL+10 ;
630      002664
631      002664 004737 002476* LIX010: JSR.      PC,ISSDIR. ; ISSUE READ
632      002670 005737 000000G. TST.      DRSTAT. ; GOOD READ?
633      002674 001404 BGT.      15$. ; YES
634      002676 012737 000000G.000000G. MOV.      #AS.URD,STATWD ; SET UNABLE TO READ DISK ERROR CODE
635      002704 000207 RTS.      PC. ;
636      002706
637      002706 062737 000000G.000000G. 15$: ADD.      #NSBUF,NSREAD. ; KEEP TRACK OF # OF SECTORS READ
638      002714 013704 000000G. MOV.      ADDRID,R4 ; ADDR OF DOCUMENT ID DESIRED
639      002720 012702 000000G. MOV.      #IPRBUF,R2. ; POINT TO FIRST INDEX RECORD IN BUFFER
640      002724 012705 000000G. MOV.      #NSBUF,R5 ; STORE # OF INDEX RECORDS READ
641      002730
642      002730 005762. 000000G. LIX020: TST.      IN,CEC(R2) ; ANY ENTRIES IN THIS INDEX RECORD
643      002734 003002. BGT.      25$. ; YES
644      002736 000137 003050* JMP.      LIX035 ; SET ERROR AND RETURN
645      002742.
646      002742. 016201 000000G. 25$: MOV.      IN,CEC(R2),R1 ; # OF ENTRIES
647      002746 005301 DEC.      R1 ; FOR OFFSET TO LAST ENTRY
648      002750 012700 000000G. MOV.      #N,BXE,R0 ; # OF BYTES/INDEX ENTRY
649      002754 004737 000000G. JSR.      PC,#MUL. ; LENGTH OF N-1 ENTRIES IN BYTES
650      002760 060201 ADD.      R2,R1 ; ADD ADDR OF BUFFER TO LENGTH OF ENTRIES
651      002762. 062701 000000G. ADD.      #IN,FE,R1 ; ADD OFFSET TO FIRST ENTRY
652      002766 SAVE.      R4,R1 ; PUT ID ADDR ON STACK
653      002772. 004737 002204* JSR.      PC,CMP2ID. ; COMPARE THE IDS
654      002776 005737 000000G. TST.      CMPST. ; ID IN THIS INDEX RECORD?
655      003002. 003003 BGT.      30$. ; NO, CHECK NEXT INDEX RECORD
656      003004 002425 BLT.      LIX045 ; ID IS IN THIS RECORD, NOW LOCATE ID'S ENTRY
657      003006 000137 003126* JMP.      LIX060 ; FOUND ID'S ENTRY
658      003012.
659      003012. 062702 000000G. 30$: ADD.      #SLB,R2. ; POINT TO NEXT INDEX RECORD IN BUFFER
660      003016 005305 DEC.      R5 ; MORE INDEX RECORDS IN BUFFER?
661      003020 003343 BGT.      LIX020 ; YES
662      003022. 023737 000000G.000000G. CMP.      NSREAD,DELTA. ; READ ALL INDEX RECORDS IN 1 DELTA?
663      003030 002007 BGE.      LIX035 ; YES, DOCUMENT DOESN'T EXIST
664      003032. 062737 000000G.000000C. ADD.      #NSBUF,DQ10+0,IOPL+10 ; GET ADDR OF NEXT SET OF INDEX RCDS TO READ
665      003040 005537 000000C. ADC.      DQ10+0,IOPL+6 ;
666      003044 000137 002664* JMP.      LIX010 ;
667      003050
668      003050 012737 000000G.000000G. LIX035: MOV.      #AS,DNE,STATWD ; SET DOCUMENT DOESN'T EXIST ERROR CODE
669      003056 000207 RTS.      PC. ;
670      003060
671      003060 016203 000000G. LIX045: MOV.      IN,CEC(R2),R3 ; # OF INDEX ENTRIES
672      003064 010201 MOV.      R2,R1 ; ADDR OF INDEX START
673      003066 062701 000000G. ADD.      #IN,FE,R1 ; ADDR OF FIRST ENTRY
674      003072.
675      003072. LIX050: SAVE.      R4,R1 ; ID ADDR ON STACK
    
```

DBUSBR - INDEX SUBROUTINES. MACRO M1110 27 MAR 68 14:31 P. 17-1
 LINDXE- LOCATE INDEX ENTRY FOR A DOCUMENT. Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

676 003076 004737 002204' JSR PC,CMP2ID ; COMPARE THE IDS.
677 003102 005737 000000G TST CMPST ; IS THIS THE ID?
678 003106 002760 BLT LIX035 ; NO, SET ERROR CODE AND RETURN.
679 003110 001406 BEQ LIX060 ; YES, FOUND THE ENTRY FOR THE ID.
680 003112 062701 000000G ADD #N,BXE,R1 ; POINT TO NEXT ENTRY.
681 003116 005303 DEC R3 ; MORE ENTRIES?
682 003120 003364 BGT LIX050 ; YES.
683 003122 000137 003050' JMP LIX035 ; NO.
684 003126 LIX060:
685 003126 022737 000000G 000000G CMP #AR,RPD,R0STUD ; REPLACE DOC?
686 003134 001404 BEQ I$ ; BRANCH IF YES.
687 ;
688 ; SEE F PURGE BIT IS SET IN DOC'S INDEX ENTRY.
689 ;
690 003136 132761 000200 000000G BITB #BIT7,I,PGI(R1) ; DOCUMENT PURGED?
691 003144 BON LIX035 ; BRNACH IF YES.
692 ;
693 003146 016137 000000G 000000G 1$: MOV I,SANH(R1),DOCADR ;
694 003154 042737 177400 000000G BIC #177400,DOCADR ; HIGH ORDER ADDR OF DOCUMENT.
695 003162 016137 000000G 000000G MOV I,SANH(R1),WDOFF ;
696 003170 000337 000000G SWAB WDOFF ;
697 003174 042737 177400 000000G BIC #177400,WDOFF ; WORD OFFSET WITHIN SECTOR OF DOC, START.
698 003202 016137 000000G 000002G MOV I,SANL(R1),DOCADR+WORD1 ; LOW ORDER ADDR OF DOCUMENT.
699 003210 016137 000000G 000000G MOV I,RWS(R1),RQWS ; REQUIRED WHITE SPACE FOR THIS DOCUMENT.
700 003216 016137 000000G 000000G MOV I,UWS(R1),UNWS ; UNUSED WHITE SPACE FOR THIS DOCUMENT.
701 003224 016237 000000G 000000G MOV IN,SAN(R2),IDXADR ; ADDR OF X RCD HAVING SPECIFIED DOCUMENT.
702 003232 042737 177400 000000G BIC #177400,IDXADR ;
703 003240 016237 000000G 000002G MOV IN,SAL(R2),IDXADR+WORD1 ;
704 003246 010237 000000G MOV R2,IBFADR ; STORE STARTING ADDR OF X RCD WITHIN BUFFER.
705 003252 000207 RTS PC ;

```

DBUSBR - DBU SUBROUTINES.
LIPRE - LOCATE IPR ENTRY.

MACRO M1110 27-MAR-80 14:31 PAGE 18
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

; SBTTL LIPRE LOCATE IPR ENTRY
;
; ON ENTRY, ADDRID -> 3-WORD DOCUMENT ID.
;
; ON RETURN,
;   DOCADR - 2-WORD SECTOR ADDRESS OF CURRENT DOCUMENT.
;   WDOFF - WORD OFFSET FROM BEGINNING OF SECTOR TO START
;           OF DOCUMENT.
;   ROWS - REQUIRED WHITE SPACE (UNPACKED COUNT)
;   UNWS - UNUSED WHITE SPACE (PACKED)
;   IDXADR - 2-WORD SECTOR ADDRESS OF INDEX RECORD.
;   IBFADR - R2 - START ADDRESS OF INDEX RECORD WITHIN BUFFER.
;   R1 -> DOCUMENT'S INDEX ENTRY.
;
LIPRE:
722 003254 013737 000000C 000000G MOV SREC+SR,LIP,TEMP ; LAST IPR ADDR.
723 003262 013737 000000C 000002G MOV SREC+SR,LIP+WORD1,TEMP+WORD1 ;
724 003270 163737 000000C 000002G SUB SREC+SR,IIP+WORD1,TEMP+WORD1 ; SUBTRACT ADDR OF 1ST IPR.
725 003276 005637 000000G SBC TEMP ;
726 003302 163737 000000C 000000G SUB SREC+SR,IIP,TEMP ;
727 003310 062737 000001 000002G ADD #1,TEMP+WORD1 ; # OF ACTIVE IPRS TO BE READ.
728 003316 013737 000002G 000000G MOV TEMP+WORD1,NS2RD ; # OF SECTORS TO READ.
729 003324 023727 000000G 000000G CMP NS2RD,*NSBUF ; FEWER SECTORS TO READ THAN ALLOTTED BUF SIZE?
730 003332 003004 10$ ; NO, MORE SECTORS THAN NSBUF.
731 003334 013737 000000G 000000G MOV NS2RD,NSREAD ; YES, READ IN ONLY ACTIVE IPRS.
732 003342 000403 BR 15$ ;
733 003344 10$ ;
734 003344 012737 000000G 000000G MOV *NSBUF,NSREAD ; READ IN ONE COMPLETE BUFFER.
735 003352 15$ ;
736 003352 013737 000000C 000000C MOV SREC+SR,IIP,DOIO+0,IOP1+6 ; ADDR OF FIRST IPR.
737 003360 013737 000000C 000000C MOV SREC+SR,IIP+WORD1,DOIO+0,IOP1+10 ;
738 003366 012737 177777 000000G MOV #+1,ETRYCT ; SET IPR ENTRY COUNTER FOR OFFSET CONDITION.
739 003374 LIP020:
740 003374 012737 000000G 000000C MOV #IO,RLB,DOIO+0,IOPN ; READ FUNCTION CODE
741 003402 012737 000000G 000000C MOV #IPRBUF,DOIO+0,IOP1 ; ADDR OF BUFFER.
742 003410 012701 000000G MOV #SLB,R1 ; SECTOR LENGTH IN BYTES.
743 003414 013700 000000G MOV NSREAD,R0 ; # OF SECTORS TO BE READ.
744 003420 004737 000000G JSR PC,$MUL ; LENGTH OF BUFFER BEING READ.
745 003424 010137 000000C MOV R1,DOIO+0,IOP1+2 ; LENGTH OF BUFFER.
746 003430 004737 002476' JSR PC,ISSDIR ; ISSUE READ.
747 003434 005737 000000G TST DRSTAT ; GOOD READ?
748 003440 001404 BEQ 30$ ; YES.
749 003442 012737 000000G 000000G MOV *AS,URD,STATWD ; SET CAN'T READ ERROR CODE.
750 003450 000207 RTS ;
751 003452 30$ ;
752 003452 012737 000001 000000G MOV #1,R8 ; IPR SECTOR # COUNTER.
753 003460 012701 000000G MOV #IPRBUF,R1 ; ADDR OF 1ST IPR IN BUFFER.
754 003464 -IP040:
755 003464 005761 000000G TST IPR,CEC(R1) ; ANY ENTRIES IN THIS IPR?
756 003470 003002 BGT 45$ ; YES.
757 003472 000137 003630' JMP LIP060 ; NO, DOCUMENT DOESN'T EXIST ERROR.
758 003476 45$ ;
759 003476 066137 000000G 000000G ADD IPR,CEC(R1),ETRYCT ; ADD # OF ENTRIES TO IPR ENTRY COUNT.
760 003504 010103 MOV R1,R3 ; IPR STARTING ADDR.
761 003506 062703 000000G ADD #IP,HID,R3 ; ADDR OF HIGHEST ID IN IPR.
762 003512 SAVE ADDRID,R3 ; ADDR OF 2 IDS TO BE COMPARED ON STACK.
763 003520 004737 002204' JSR PC,CMP2ID ; COMPARE IDS

```

DBUSBR - D SUBROUTINES
LIPRE+ LOCHE IPR ENTRY

```

764 003524 005737 000000G. TST CMPST ; ID IN THIS IPR?
765 003530 003443 BLE LIP080 ; SHOULD BE
766 003532 005237 000000G. INC R0 ; NO, CHECK NEXT IPR IN BUFFER
767 003536 062701 000000G. ADD *SLB,R1 ; ADDR OF NEXT IPR
768 003542 023737 000000G.000000G. CMP R0,NSREAD ; ANOTHER IPR IN BUFFER?
769 003550 003745 BLE LIP040 ; YES
770 003552 163737 000000G.000000G. SUB NSREAD,NS2RD ; MORE IPR SECTORS TO READ?
771 003560 001423 BEQ LIP060 ; NO, DOCUMENT DOESN'T EXIST
772 003562 063737 000000G.000000G. ADD NSREAD,D0I0+0,IOPL+10 ; GET ADDR OF NEXT IPR TO READ
773 003570 005537 000000G.000000G. ADC D0I0+0,IOPL+6 ;
774 003574 023727 000000G.000000G. CMP NS2RD,#NSBUF ; MORE THAN BUFFER SIZE TO READ?
775 003602 003005 BGT 50$ ; YES
776 003604 013737 000000G.000000G. MOV NS2RD,NSREAD ; NO, READ REMAINING SECTORS
777 003612 000137 003374' JMP LIP020 ; CONTINUE CHECK OF NEXT IPRS
778 003616
779 003616 012737 000000G.000000G. MOV #NSBUF,NSREAD ; READ ONE COMPLETE BUFFER
780 003624 000137 003374' JMP LIP020 ;
781 ;
782 ; *** ERROR - DOCUMENT DOES NOT EXIST
783 ;
784 003630
785 003630 012737 000000G.000000G. LIP060: MOV #AS,DNE,STATUD ; SET DOCUMENT DOESN'T EXIST ERROR CODE
786 003636 000207 RTS PC ;
787 003640
788 003640 LIP080: SAVE R1 ; SAVE IPR ADDR
789 003642 016100 000000G. MOV IP,CEC(R1),R0 ; # OF IPR ENTRIES
790 003646 005300 DEC R0 ;
791 003650 012701 000000G. MOV #N,BIPE,R1 ; LENGTH OF EACH ENTRY
792 003654 004737 000000G. JSR PC,$MUL ; TOTAL OFFSET TO LAST ENTRY
793 003660 010102 MOV R1,R2 ; KEEP TOTAL OFFSET
794 003662 RESTOR R1 ; RESTORE IPR ADDR
795 003664 060102 ADD R1,R2 ; ADD START ADDR OF IPR
796 003666 062702 000000G. ADD #IP,FE,R2 ; POINTING TO LAST ENTRY
797 003672 016100 000000G. MOV IP,CEC(R1),R0 ; # OF ENTRIES IN IPR
798 003676
799 003676 LIP085: SAVE ADDRID,R2 ; ADDR OF 2 IDS TO BE COMPARED ON STACK
800 003704 004737 002204' JSR PC,CHP2ID ; IS THIS THE LOWER BRACKET FOR THE ID?
801 003710 005737 000000G. TST CMPST ;
802 003714 002010 BGE LIP090 ; LOCATED THE BRACKET
803 003716 162702 000000G. SUB #N,BIPE,R2 ; POINT TO NEXT LOWER ENTRY
804 003722 005337 000000G. DEC ETRYCT ; DECREMENT TOTAL ENTRY COUNT
805 003726 005300 DEC R0 ; DECREMENT SINGLE IPR ENTRY COUNT
806 003730 003362 BGT LIP085 ; MORE TO CHECK
807 003732 000137 003630' JMP LIP060 ; ERROR, DOCUMENT MUST NOT EXIST
808 003736
809 003736 013737 000000G.000000G. LIP090: MOV SREC+SR,1IN,IDXADR ; ADDR OF 1ST INDEX RECORD
810 003744 013737 000000G.000002G. MOV SREC+SR,1IN+WORD1,IDXADR+WORD1 ;
811 003752 013701 000000G. MOV ETRYCT,R1 ; # OF IPR ENTRIES
812 003756 013700 000000G. MOV DELTA,R0 ; # OF INDEX RECORDS REPRESENTED
813 003762 004737 000000G. PC,$MUL ;
814 003766 060137 000002G. ADD R1,IDXADR+WORD1 ; ADD OFFSET TO START ADDR OF INDEX RCDS FOR 1 DELTA
815 003772 005537 000000G. ADC IDXADR ;
816 003776 004737 002622' JSR PC,LINDXE ; LOCATE INDEX ENTRY FOR DOCUMENT
817 004002 000207 RTS PC ;

```

DBUSBR - DBU SUBROUTINES
PPDATA- PUT PACKED DATA

MACRO M1110 27-MAR-80 14:31 PAGE 19
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
      .SBTTL PPDATA- PUT PACKED DATA
819      ;
820      ;
821      ;
822      ; R1 - HOLDS ADDRESS OF WHERE TO BEGIN STORING PACKED DATA IN WBUF.
823      ; R2 - HOLDS START ADDRESS OF PACKED DATA IN INPUT BUFFER.
824      ;
825      ;
826      004004 005037 004614' PPDATA::CLR RPLFLG
827      004010          SAVE PWCT,PWCT+WORD1 ; PUSH PACKED WORD COUNT FOR USE LATER
828      ;
829      ; MOVE DATA INTO OUTPUT BUFFER.
830      ;
831      004020      PPD005:
832      004020 012221      MOV (R2)+,(R1)+
833      004022 162737 000001 000002G SUB #1,PWCT+WORD1 ; DECREMENT PACKED WORD COUNT.
834      004030 005637 000000G SBC PWCT ; ACCOUNT FOR CARRY.
835      004034 003035 BGT PPD010 ; BRANCH IF MORE TO MOVE.
836      004036 005737 000002G TST PWCT+WORD1 ; FINISHED?
837      004042 101032 BHI PPD010 ; BRANCH IF MORE TO MOVE.
838      004044 023727 000000G 000000G CMP RQSTWD,#AR,RPD ; MODIFY-REPLACE?
839      004052 001402 BEQ 1$ ; BRANCH IF YES.
840      004054 000137 004462' JMP PPD030 ; JUMP IF NO.
841      004060 005737 004614' 1$: TST RPLFLG ; OLD TEXT OVERWRITTEN?
842      004064 001176 BNE PPD030 ; BRANCH IF YES.
843      ;
844      ; FOR A MODIFY-REPLACE REQUEST, IF NEW COPY OF DOCUMENT IS
845      ; SHORTER THAN OLD COPY, WRITE WHITE SPACE OVER EXTRA TEXT.
846      ; R9 = UNUSED WHITE SPACE OF NEW TEXT.
847      ; I,UWS(-) = UNUSED WHITE SPACE OF OLD TEXT.
848      ;
849      004066 013705 000000G MOV R9,R5 ; R5 = UNUSED W.S. OF NEW TEXT.
850      004072 013702 000000G MOV XTEMP,R2 ; R2-> DOC'S INDEX ENTRY.
851      004076 166205 000000G SUB I,UWS(R2),R5 ; R5 = WORD LENGTH OF EXTRA TEXT.
852      004102 003567 BLE PPD030 ; BRANCH IF NEW TEXT IS LONGER
853      ; THAN OLD.
854      ;
855      ; EXTRA TEXT MUST BE OVERWRITTEN WITH WHITE SPACE.
856      ;
857      004104 010537 000002G MOV R5,PWCT+2 ; WORD LENGTH OF TEXT TO BE
858      004110 005037 000000G CLR PWCT ; OVERWRITTEN.
859      004114 005237 004614' INC RPLFLG ; SET FLAG THAT INDICATES THAT
860      ; OVERWRITE IS IN PROGRESS.
861      004120 CALL WHTSPC ; FILL INPUT BUFFER WITH PACKED BLANKS.
862      004124 000137 004020' JMP PPD005 ; JUMP IF FINISHED
863      004130      PPD010:
864      004130 020127 000000G CMP R1,#WBUF+SLB-2 ; END OF OUTPUT BUFFER?
865      004134 003454 BLE 4$ ; NO.
866      004136 004737 007660' JSR PC,WRTDB ; YES, WRITE ADDITIONAL DATA BASE.
867      004142 005737 000000G TST WRTST ; GOOD WRITE?
868      004146 001401 BEQ 10$ ; YES.
869      004150 000560 BR PPD035 ; NO.
870      004152 022737 000000G 000000G 13$: CMP #AR,ADD,RQSTWD ; ADD DOCUMENT?
871      004160 001442 BEQ 4$ ; BRANCH IF YES.
872      ;
873      ; DISK ADDRESS OF NEXT DOC'S START = NDCADR, NDCADR+2.
874      ; NEXT SECTOR ADDRESS TO BE WRITTEN = DQ10+Q, IOPL+6, DQ10+Q, IOPL+10
875      ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - D SUBROUTINES
PPDATA - PUT PACKED DATA

MACRO M1110 27-MAR-80 14:31 P. 19-1
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
876 ; DETERMINE IF NEXT SECTOR TO BE WRITTEN CONTAINS START OF NEXT DOCUMENT.  
877 ;  
878 004162 013737 000000G 000000G MOV NDCADR,TEMP  
879 004170 042737 177400 000000G BIC #177400,TEMP  
880 004176 023737 000000G 000000G CMP TEMP,DQIO+Q,IOP1+6 ;HI ADDRESSES MATCH?  
881 004204 101030 BHI 4$ ;BRANCH IF NO  
882 004206 103405 BLO 6$ ;ERROR  
883 004210 023737 000002G 000000G CMP NDCADR+2,DQIO+Q,IOP1+10 ;LO ADDRESSES MATCH?  
884 004216 101023 BHI 4$ ;BRANCH IF NO  
885 004220 001402 BEQ 7$ ;BRANCH IF YES  
886 004222 000137 004223' 6$: JMP .+1 ;FATAL ERROR  
887 ;  
888 ; LAST SECTOR OF TEXT TO BE ADDED, READ SECTOR CURRENTLY ON DISK  
889 ; TO ASSURE THAT NEXT DOCUMENT, WHICH STARTS IN THIS SECTOR, IS NOT  
890 ; OVERWRITTEN.  
891 ;  
892 004226 012737 000000G 000000G 7$: MOV #10,RLB,DQIO+Q,IOP1  
893 004234 004737 002476' JSR PC,ISSDIR  
894 004240 005737 000000G TST DRSTAT  
895 004244 001406 BEQ 5$  
896 004246 012737 000000G 000000G MOV #AS,URD,STATUD  
897 004254 062706 000004 ADD #WORD2,SP ;CLEAR STACK  
898 004260 000207 RTS ;RETURN  
899 ;  
900 004262 012701 000000G 5$: MOV #WBUF,R1 ;R1->WBUF  
901 004266 020237 000000G 4$: CMP R2,ENDBFA ;END OF INPUT BUFFER?  
902 004272 002652 BLT PPD005 ;NO, CONTINUE  
903 004274 005737 004614' TST RPLFLG ;TEXT OVERWRITE IN PROGRESS?  
904 004300 001404 BEQ 8$ ;BRANCH IF NO  
905 004302 CALL WHTSPC ;FILL INPUT BUFFER WITH W.S.  
906 004306 000137 004020' JMP PPD005 ;CONTINUE  
907 ;  
908 ; GET NEXT INPUT BUFFER FROM CCIN  
909 ;  
910 004312 8$: SAVE R1 ;  
911 004314 013701 000000G MOV SAVEPT,R1 ;R1->DOUBLE BUFFER CONTROL BLOCK  
912 004320 046137 000002 000000G BIC WORD1(R1),MSQ ;SHOW BUFFER IS NOW FREE  
913 004326 RSUM#C CCIN ;RESUME CCIN IF IT IS OUT OF BUFFERS  
914 004334 RESTOR R1 ;  
915 004336 2$: ;  
916 004336 RCVD#$# ,#PRB ;GET ID OF NEXT INPUT BUFFER  
917 004354 103030 BCC 1$ ;BRANCH IF OK  
918 004356 023727 000000G 000000G CMP #DSW,#IE,ITS ;NOTHING INPUT?  
919 004364 001013 BNE 3$ ;NO, ANOTHER ERROR  
920 004366 WTSE#$# #CF,UPD ;WAIT FOR DATA  
921 004400 CLEF#$# #CF,UPD ;DATA INPUT, GO GET IT  
922 004412 000751 BR 2$ ;  
923 004414 3$: ;  
924 004414 013737 000000G 000000G MOV #DSW,PAR1 ;SET UP ERROR MSG  
925 004422 012701 000000G MOV #EMSG1,R1  
926 004426 CALL ERROR  
927 004432 CALL EXIT  
928 004436 1$: ;  
929 004436 004737 005224' JSR PC,READ ;READ NEXT INPUT BUFFER  
930 004442 005737 000000G TST RDSTAT ;READ OK?  
931 004446 001002 BNE 9$ ;BRANCH IF NO  
932 004450 000137 004020' JMP PPD005 ;JMP IF YES
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - DBU. SUBROUTINES.
PPDATA- PUT. PACKED. DATA.

MACRO M1110 27-MAR-68 14:31 PAGE 18-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
933 004454 062706 000004      9#:  ADD. #WORD2,SP. ; CLEAR STACK.
934 004460 000207      RTS. PC. ;
935 ;
936 ; ALL BYTES OF NEW DOCUMENT MOVED INTO OUTPUT BUFFER.
937 ;
938 004462.      PPD030:
939 004462. 022737 000000G.000000G. CMP. #AR,ADD,RQSTWD ;ADD DOCUMENT?
940 004470 001037      BNE. PPD041 ;BRANCH IF NO (REPL.DOC, ADD SUB.DOC)
941 004472. 020127 000000C. CMP. R1,#UBUF+SLB. ; END OF OUTPUT BUFFER?
942.004476 002413      BLT. PPD040 ; NO
943 004500 004737 007650* JSR. PC,VRTDB. ; YES, WRITE FULL BUFFER OF ADD DOCUMENT.
944 004504 005737 000000G. TST. WRTST. ; GOOD WRITE?
945 004510 001406      BEQ. PPD040 ; YES
946 004512.      PPD035:
947 004512. 012737 000000G.000000G. MOV. #AS,UWD,STATWD ; SET ERROR BIT.
948 004520 062706 000004      ADD. #WORD2,SP. ; CLEAR STACK.
949 004524 000207      RTS. PC. ; RETURN
950 ;
951 ; PUT EOC MARKER AT END OF TEXT.
952.
953 004526 013711 000000G. PPD040: MOV. EOC,(R1) ; PLACE END-OF-COLLECTION
954 004532. 013737 000000C.000000C. MOV. DQ10+0,IOPL+6,SREC+SR,ECH ; UPPER ADDR OF EOC.
955 004540 010103      MOV. R1,R3 ; ADDR OF EOC.
956 004542. 162703 000000G. SUB. #UBUF,R3 ; WORD OFFSET * 2.
957 004546 006203      ASR. R3 ; CONVERT BYTES TO WORDS.
958 004550 000303      SWAB. R3 ; GET IT IN UPPER BYTE.
959 004552. 042703 000377      BIC. #377,R3 ; CLEAR LOWER BYTE
960 004556 050337 000000C. BIS. R3,SREC+SR,ECH ; PUT IN ADDR.
961 004562. 013737 000000C.000000C. MOV. DQ10+0,IOPL+10,SREC+SR,ECL ; LOW ADDR OF EOC
962.004570 004737 007650* PPD041: JSR. PC,VRTDB. ;WRITE LAST BLOCK OF DOC.
963 004574 005737 000000G. TST. WRTST.
964 004600 001344      BNE. PPD035
965 004602.      RESTOR. PWCT,PWCT+WORD1 ; POP STACK FOR PACKED WORD COUNT.
966 004612. 000207      RTS. PC. ;
967 ;
968 ; FLAG TO INDICATE THAT TEXT OVERWRITE IS IN PROGRESS (FOR
969 ; MODIFY-REPLACE REQUEST).
970 ;
971 004614 000000      RPLFLG, .WORD. 0
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - D SUBROUTINES
WHTSPC:

MACRO.M1110 27-MAR-80 14:31 P. 30

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
973 .SBTTL - WHTSPC -
974 ;
975 ; SUBROUTINE TO FILL A BUFFER WITH PACKED BLANKS
976 ;
977 004616 WHTSPC: SAVE R1,R3
978 004622 013701 000000G MOV BUFADR,R1
979 004626 062701 000002 ADD #2,R1 ;R1-> BUFFER TO BE LOADED WITH W/S;
980 004632 012705 005250 MOV #2728,R5 ;R5 = NO. OF UNPACKED BLANKS NEEDED TO
981 ; FILL INPUT BUFFER.
982 004636 012737 004700' 004676' MOV #PCK1,PACK6 ;START OUT RIGHT.
983 004644 012702 000005 1$: MOV #5,R2 ;R2 = 6-BIT W/S; FILL CHAR (FROM CCET)
984 004650 004777 000022 JSR PC,@PACK6
985 004654 SSQB R5,1$
986 ;
987 004660 013702 000000G MOV BUFADR,R2.
988 004654 062702 000002 ADD #2,R2.
989 004670 RESTOR R1,R3
990 004674 000207 RTS PC
```

DBUSBR - DBU SUBROUTINES.
PACK6 - PACK 6-BIT CHARS.

MACRO M1110 27-MAR-80 14:31 PAGE 21
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL PACK6 - PACK 6-BIT CHARS.
992 ;
993 ;
994 ; INPUT:
995 ; R2 HAS THE 6-BIT CHAR IN THE LOW ORDER BYTE.
996 ; R1 -> BUFFER TO BE LOADED WITH WHITE SPACE.
997 ;
998 ; REGISTERS DESTROYED: R2, R3
999 ;
1000 ; OUTPUT: R1 -> NEXT FREE WORD OF OUTPUT BUFFER
1001 ;
1002 004676 004700* PACK6: .WORD PCK1
1003 ;
1004 ;
1005 004700 012737 004726* 004676* PCK1: MOV #PCK2.PACK6 ;FIRST 6-BIT CHAR.
1006 004706 ASHCM -6,R2 ;PLACE 1ST 6-BIT CHAR INTO R3
1007 004724 000535 BR PCK10
1008 ;
1009 004726 012737 004754* 004676* PCK2: MOV #PCK3.PACK6 ;SECOND 6-BIT CHAR.
1010 004734 ASHCM -6,R2 ;R3 NOW CONTAINS 12 BITS OF PACKED CHARS.
1011 004752 000522 BR PCK10
1012 ;
1013 004754 012737 005022* 004676* PCK3: MOV #PCK4.PACK6 ;THIRD 6-BIT CHAR.
1014 004762 ASHCM -4,R2 ;R3 NOW COMPLETELY PPACKED.
1015 005000 010321 MOV R3,(R1)+ ;MOVE R3 TO OUTPUT BUFFER.
1016 005002 ASHCM -2,R2 ; MOVE REMAINING 2 BITS OF CHAR. TO R3
1017 005020 000477 BR PCK10 ;CONTINUE.
1018 ;
1019 005022 012737 005050* 004676* PCK4: MOV #PCK5.PACK6 ; FOURTH 6 BIT CHARACTER.
1020 005030 ASHCM -6,R2 ; MOVE 6 BITS INTO R3 - R3 NOW HAS 8 BITS.
1021 005046 000464 BR PCK10
1022 ;
1023 005050 012737 005076* 004676* PCK5: MOV #PCK6.PACK6 ; FIFTH SIX BIT CHARACTER.
1024 005056 ASHCM -6,R2 ; R3 NOW HAS 14 BITS OF PACKED DATA.
1025 005074 000451 BR PCK10
1026 ;
1027 005076 012737 005144* 004676* PCK6: MOV #PCK7.PACK6 ; SIXTH 6 BIT CHARACTER.
1028 005104 ASHCM -2,R2 ; R3 NOW COMPLETELY PACKED.
1029 005122 010321 MOV R3,(R1)+ ; PLACE PACKED WORD INTO OUTPUT BUFFER.
1030 005124 ASHCM -4,R2 ; REMAINING 4 BITS INTO R3
1031 005142 000426 BR PCK10 ;CONTINUE.
1032 ;
1033 005144 012737 005172* 004676* PCK7: MOV #PCK8.PACK6 ; SEVENTH 6 BIT CHARACTER.
1034 005152 ASHCM -6,R2 ; R3 NOW HAS 10 BITS OF PACKED DATA.
1035 005170 000413 BR PCK10
1036 ;
1037 005172 012737 004700* 004676* PCK8: MOV #PCK1.PACK6 ; EIGHTH 6 BIT CHARACTER.
1038 005200 ASHCM -6,R2 ; R3 NOW COMPLETELY PACKED.
1039 005216 010321 MOV R3,(R1)+ ; MOVE PACKED WORD INTO OUTPUT BUFFER.
1040 005220 PCK10:
1041 005220 000241 CLC
1042 005222 000207 RTS PC
```

DBUSBR - D SUBROUTINES.
READ- READ INPUT FILE

MACRO M1110 27-MAR-80 14:31 P 22

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL READ- READ INPUT FILE
;
; CCIN HAS PLACED A DATA BLOCK RECEIVED FROM DBPROC INTO ONE OF
; TWO GLOBAL BUFFERS. RCVD NODE (PRB) CONTAINS IDENTIFIER OF BUFFER.
;
READ::
1044
1045
1046
1047
1048
1049 005224
1050 005224
1051 005230 005037 000000G
1052 005234 012701 000000G
1053 005240 012703 000002
1054 005244
1055 005244 033761 000004G 000002
1056 005252
1057 005254 016101 000004
1058 005260
1059 005264 012737 000003 000000G
1060 005272 012737 000000G 000000G
1061 005300
1062 005304 000207
1063
1064
1065
1066 005306 010137 000000G
1067 005312 011137 000000G
1068 005316 013702 000000G
1069 005322 013737 000000G 000000G
1070 005330 062737 004000 000000G
1071 005336 021227 041104
1072 005342 001003
1073 005344 016237 000000G 000000G
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083 005352 012701 000004
1084 005356
1085 005360 005722
1086 005362 001012
1087 005364
1088
1089
1090
1091 005370
1092 005372 016237 000010 000000G
1093 005400 016237 000012 000000G
1094 005406 000401
1095 005410
1096 005412 005337 000000G
1097 005416
1098 005422 000207

;
; MOVE ADDRESS OF DATA BUFFER IN COMMON TO LOCAL VARIABLE.
;
1$: MOV R1,SAVEPT ; SAVE THE POINTER TO THE DOUBLE BUF CONTROL BLK
MOV (R1),BUFADR ; ADDR OF DATA BUFFER IN COMMON
MOV BUFADR,R2 ; R2->NEW INPUT BUFFER START
MOV BUFADR,ENDBFA ;
ADD #N,BUF,ENDBFA ; END OF BUFFER ADDR
CMP (R2),#"DB" ; START OF AN EXCHANGE?
BNE 4$ ; NO
MOV PC,PAD(R2),INCOUT ; YES, SAVE BLOCK INPUT COUNT
;
; TEST FOR ERROR SEQUENCE (FIRST FOUR WORDS OF BUFFER ARE ZEROS)
; INDICATING THAT EITHER
; 1) DBPROC DISCOVERED AN ERROR (WORD 4 = 1) - DBUPD MUST
; RESTORE DATA BASE TO ITS PREVIOUS STATE AND CREATE A NACK, OR
; 2) CCIN HAS DETECTED A LINK ERROR (WORD 4 = 2) - DBUPD MUST
; RESTORE DATA BASE TO ITS PREVIOUS STATE. NO ACK IS GENERATED
; BECAUSE THIS IS A RECOVERABLE ERROR.
;
4$: MOV #4,R1 ; R1 = LOOP COUNTER
SAVE R2
3$: TST (R2)+ ; ZERO?
BNE 5$ ; BRANCH IF NO
SSOB R1,3$
;
; ERROR SEQUENCE
;
RESTOR R2
MOV 10(R2),RDSTAT ; INDICATE NATURE OF READ ERROR
MOV 12(R2),STATWD
BR 6$
5$: RESTOR R2
6$: DEC INCOUT ; DEC EXPECTED INPUT BLOCK COUNT
RESTOR R1,R3
RTS PC ;
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - DBU SUBROUTINES
 READF - READ FIRST BLOCK OF AN ACTIVE FCS FILE

MACRO M1110 27 MAR 80 1117 PAGE 37

```

1100 .SBTTL READF READ FIRST BLOCK OF AN ACTIVE FCS FILE.
1101 ;
1102 ;
1103 ; READS THE FIRST LOGICAL BLOCK OF EITHER THE DATA BASE ACKNOWLEDGEMENT
1104 ; RECORD OR THE FIRST LOGICAL BLOCK OF A RETRIEVED DOCUMENT FILE.
1105 ;
1106 ; ON ENTRY, R0->FDB.
1107 ;
1108 005424 READF::
1109 005424 005060 000000G CLR F,BKVB(R0) ; CLEAR UPPER WORD OF VIRTUAL BLOCK COUNTER.
1110 005430 012760 000001 000002G MOV #1,F,BKVB+2(R0) ; SPECIFY VIRTUAL BLOCK 1
1111 005436 READ$ ; READ THE FIRST BLOCK
1112 005442 WAIT$ ; FOR I/O COMPLETION
1113 005446 103012 BCC 1$ ; GOOD READ
1114 005450 116001 000000G MOV F,ERR(R0),R1
1115 005454 010137 000000G MOV R1,PAR1
1116 005460 012701 000000G MOV #EMSG0,R1
1117 005464 CALL ERROR
1118 005470 004737 002440' JSR PC,EXIT
1119 005474 000207 1$: RTS PC
1120 ;
1121 ;
1122 ;
1123 .SBTTL READN READ NEXT SECTOR
1124 ;
1125 ;
1126 005476 READN::
1127 005476 005737 000000G TST NS2RD ; MORE SECTORS TO READ?
1128 005502 003002 BGT 1$ ; YES
1129 005504 000137 005614' JMP 3$ ; NO
1130 005510 1$:
1131 005510 062737 000001 000002G ADD #1,DOCADR+WORD1 ; GET ADDRESS OF NEXT SECTOR
1132 005516 005537 000000G ADC DOCADR
1133 005522 012737 000000G 000000C MOV #RDBUF,DQIO+Q,IOPL ; READ BUFFER
1134 005530 012737 000000G 000000C MOV #SLB,DQIO+Q,IOPL+2 ; BUFFER LENGTH
1135 005536 013737 000000G 000000C MOV DOCADR,DQIO+Q,IOPL+6 ; ADDR OF DISK SECTOR
1136 005544 013737 000002G 000000C MOV DOCADR+WORD1,DQIO+Q,IOPL+10 ;
1137 005552 012737 000000G 000000C MOV #IO,RLB,DQIO+Q,IOFN ; READ FUNCTION CODE
1138 005560 004737 002476' JSR PC,ISSDIR ; ISSUE READ
1139 005564 005737 000000G TST DRSTAT ; READ OK?
1140 005570 001404 BEQ 2$ ; YES
1141 005572 012737 000000G 000000C MOV #AS,URD,STATWD ; SET APPROPRIATE BIT IN ACTION STATUS
1142 005600 000207 RTS PC
1143 005602 2$:
1144 005602 012703 000000G MOV #RDBUF,R3 ; REINITIALIZE INPUT REGISTER
1145 005606 005337 000000G DEC NS2RD ; DECREMENT # OF SECTORS TO READ
1146 005612 000207 RTS PC
1147 005614 3$:
1148 005614 000137 005615' JMP ,+1 ; HAVE A PROBLEM - NO MORE SECTORS TO READ
1149 005620 000207 RTS PC
    
```

DBUSBR - D SUBROUTINES
SDATA- SAVE DATA

MACRO M1110 27-MAR-80 14:31 P 24
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
          .SBTTL- SDATA- SAVE DATA
1151      ;
1152      ;
1153      ; THIS ROUTINE EXTRACTS THE NECESSARY INFORMATION FROM THE PROCESSED
1154      ; COMMAND REQUEST TO BE USED AS THE REQUEST IS PROCESSED.
1155      ;
1156      005622.          SDATA::
1157      005622. 013702. 000000G.      MOV.   BUFADR,R2.          ;
1158      005626 016237 000000G.000000G.      MOV.   PC, IDH(R2),DOCID.      ; HIGH ORDER DOC. ID
1159      005634 016237 000000G.000002G.      MOV.   PC, IDM(R2),DOCID+WORD1 ; MIDDLE ORDER DOC. ID
1160      005642. 016237 000000G.000004G.      MOV.   PC, IDL(R2),DOCID+WORD2 ; LOW ORDER DOC. ID
1161      ;
1162      ; SAVE DOC ID FOR ACK ENTRY
1163      ;
1164      005650 013737 000000G.000000G.      MOV.   DOCID,ACKBUF.      ;HI
1165      005656 013737 000002G.000002G.      MOV.   DOCID+2,ACKBUF+2;MID
1166      005664 013737 000004G.000004G.      MOV.   DOCID+4,ACKBUF+4;LO
1167      ;
1168      005672. 016237 000000G.000000G.      MOV.   PC,STW(R2),STATWD.      ; STATUS WORD FOR ACTION REQUESTED
1169      005700 016237 000000G.000000G.      MOV.   PC,TCC(R2),UPCCT.      ; UNPACKED CHARACTER COUNT
1170      005706 016237 000002G.000002G.      MOV.   PC,TCC+WORD1(R2),UPCCT+WORD1 ;
1171      005714 016237 000000G.000000G.      MOV.   PC,PWC(R2),PWCT.      ; PACKED WORD COUNT (TEXT +WHITE SPACE)
1172      005722. 016237 000002G.000002G.      MOV.   PC,PWC+WORD1(R2),PWCT+WORD1 ;
1173      005730 016237 000000G.000000G.      MOV.   PC,PTC(R2),PCCT.      ; PACKED WORD COUNT (TEXT ONLY)
1174      005736 016237 000002G.000002G.      MOV.   PC,PTC+WORD1(R2),PCCT+WORD1 ;
1175      005744 026227 000000G.041504      CMP.   PC,TYP(R2),#"DC.      ; ADD NEW DOCUMENT?
1176      005752. 001026      BNE.   1$      ; NO
1177      005754 013737 000000G.000000G.      MOV.   PWCT,ROWS.      ;HI PACKED WORD CNT (TEXT+WHITE SPACE)
1178      005762. 013737 000002G.000000G.      MOV.   PWCT+2,UNWS.      ;LO " " "
1179      005770 163737 000002G.000000G.      SUB.   PCCT+2,UNWS.      ;UNWS=UNUSED WHITE SPACE (PACKED)
1180      005776 005637 000000G.      SBC.   ROWS
1181      006002. 163737 000000G.000000G.      SUB.   PCCT,ROWS
1182      006010 005737 000000G.      TST.   ROWS
1183      006014 001402.      BEQ.   2$      ;BRANCH IF ROWS ZERO
1184      006016 000137 006017*      JMP.   .+1      ;FATAL ERROR!!!!!!
1185      006022. 016237 000000G.000000G. 2$:      MOV.   PC,USC(R2),ROWS.      ; REQUIRED WHITE SPACE (UNPACKED)
1186      006030      1$:
1187      006030 000207      RTS.   PC.      ;
```

DBUSBR - DBU-SUBROUTINES.
RSTE0C.

MACRO: M1110 - 27-MAR-80 14:31 PAGE 25
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1189          .SBTTL - RSTE0C.
1190          ;
1191          ; SUBROUTINE TO RESTORE EOC MARKER TO WHERE IT WAS WHEN PROCESSING
1192          ; OF CURRENT DB UPDATE (WHICH IS ABORTED DUE TO ERROR) BEGAN.
1193          ;
1194 006032.    RSTE0C::SAVE R1
1195          ;
1196          ; READ BLOCK WHICH USED TO CONTAIN EOC
1197          ;
1198 006034 013737 000000G-000000C-    MOV.  DDCADR,SREC+SR,ECH.    ;DISK ADDR. (HI)
1199 006042- 013737 000002G-000000C-    MOV.  DDCADR+2,SREC+SR,ECL.    ; " " (LO)
1200 006050          CALL.  RDEOC.          ;READ FORMER EOC BLOCK.
1201 006054 103420          BCS.  99#
1202          ;
1203          ; RESTORE EOC MARKER IN BLOCK.
1204          ;
1205 006056 113701 000000C-    MOVB.  SREC+SR,ECB,R1          ;R1 = WORD OFFSET OF EOC.
1206 006062- 042701 177400          BIC.  #177400,R1
1207 006066 006301          ASL.  R1          ;R1 = BYTE OFFSET.
1208 006070 013761 000000G-000000G-    MOV.  EOC,WBUF(R1)          ;WRITE EOC MARKER.
1209          ;
1210          ; WRITE EOC BLOCK BACK TO DISK.
1211          ;
1212 006076 012737 000000G-000000C-    MOV.  #10,WLB,DQIO+0,IOFN.
1213 006104 004737 002476'    JSR.  PC,ISSDIR.
1214 006110 005737 000000G-    TST.  DRSTAT.
1215 006114 001402-    BEQ.  1$
1216 006116 004737 002440'    99$: JSR.  PC,EXIT.          ;GOOD WRITE?
1217 006122-    1$: RESTOR. R1          ;BRANCH IF YES.
1218 006124 000207          RTS.  PC.          ;FATAL ERROR.
```

DBUSBR - D SUBROUTINES
RDEOC

MACRO M1110 27-MAR-88 14:31 P. 26
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL RDEOC
:
: SUBROUTINE TO READ THE EOC BLOCK FROM THE SEARCH DATA BASE, PROVIDED
: THAT THE BLOCK IS NOT ALREADY IN WBUF.
:
RDEOC:: SAVE R1
1225 006126
1226 006130 123737 000000C.000000C. CMPB SREC+SR.ECH,DQIO+Q,IOPL+6 ;EOC BLOCK STILL IN WBUF?
1227 006136 001015 BNE 3$
1228 006140 023737 000000C.000000C. CMP SREC+SR.ECL,DQIO+Q,IOPL+10
1229 006146 001011 BNE 3$ ;BRANCH IF NO
1230 006150 113701 000000C. MOVB SREC+SR.ECB,R1 ;R1 = WORD OFFSET OF EOC
1231 006154 042701 177400 BIC #177400,R1
1232 006160 006301 ASL R1 ;R1 = BYTE OFFSET
1233 006162 023761 000000G.000000G. CMP EOC,WBUF(R1) ;HAS EOC MARKER BEEN MOVED?
1234 006170 001426 BEQ 1$ ;BRANCH IF NO
1235
:
: HAVE TO READ EOC BLOCK
:
1238 006172 012737 000000G.000000C. 3$: MOV #IO,RLB,DQIO+Q,IOFN ;FUNCTION CODE
1239 006200 012737 000000G.000000C. MOV #WBUF,DQIO+Q,IOPL ;INPUT BUFFER
1240 006206 012737 000000G.000000C. MOV #SLB,DQIO+Q,IOPL+2 ;BUFFER'S BYTE LENGTH
1241 006214 113737 000000C.000000C. MOVB SREC+SR.ECH,DQIO+Q,IOPL+6 ;DISK ADDR. (HI)
1242 006222 013737 000000C.000000C. MOV SREC+SR.ECL,DQIO+Q,IOPL+10 ; " (LO)
1243 006230 004737 002476 JSR PC,ISSDIR ;READ BLOCK
1244 006234 005737 000000G. TST DRSTAT ;ERROR?
1245 006240 001402 BEQ 1$ ;BRANCH IF NO
1246 006242 000261 SEC
1247 006244 000401 BR 2$
1248 006246 000241 1$: CLC
1249 006250 2$: RESTOR R1
1250 006252 000207 RTS PC
```

DBUSBR - DBU SUBROUTINES.
STORE - STORE CHARACTER.

MACRO M1110 27-MAR-80 14:31 PAGE 27

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL STORE- STORE CHARACTER.  
;  
; FOR "READ DOCUMENT" REQUESTS,  
; STORE DETERMINES WHICH CHARACTERS ARE TO BE TRANSLATED BEFORE THEY ARE  
; STORED. IF THE CHARACTER IS A ZONE MARKER, SUB-ZONE MARKER, PARAGRAPH  
; OR SENTENCE MARKER, THE MARKER ITSELF IS TRANSLATED, BUT THE CHARACTER  
; FOLLOWING THE MARKER IS NOT. IN THE CASE OF A SUB-DOCUMENT MARKER, THE  
; FOLLOWING 11 CHARACTERS ARE NOT TRANSLATED.  
;  
; FOR "DELETE SUB-DOCUMENT" REQUESTS, THE "STORE" SUBROUTINE'S  
; PURPOSE IS TO LOOK FOR SUB-DOCUMENT MARKERS. WHEN A SUB-DOCUMENT  
; START SEQUENCE IS FOUND, ITS START ADDRESS IS SAVED (BYTE OFFSET  
; IN SECTOR, SECTOR ADDRESS).  
;  
; INPUT: R2 CONTAINS THE CHARACTER.  
; R5 -> SUB-DOC ID STRING (FOR DELETE SUB-DOC REQUEST)  
;  
; OUTPUT: (R4) WILL HAVE THE TRANSLATED OR UNTRANSLATED CHARACTER.  
; R4 HOLDS THE ADDRESS OF THE NEXT BYTE IN THE OUTPUT BUFFER.  
;  
STORE: SAVE R0  
; TST NOCNVT ; NO CONVERT FLAG SET?  
; BEQ 1$ ; NO  
; CMP #AR,DSD,RQSTWD ; DELETE SUB-DOC?  
; BNE 6$ ; BRANCH IF NO  
; TST SDID ; SUB-DOC ID?  
; BLE 6$ ; BRANCH IF NO  
;  
; POSSIBLE SUB-DOC ID MATCH FOR "DELETE SUB-DOC" REQUEST.  
;  
; MOV #11,R0  
; SUB NOCNVT,R0 ; R0=BYTE COUNT INTO SUB-DOC ID STRING  
; TST R0 ; FLAG BYTE OF SUB-DOC?  
; BNE 11$ ; BRANCH IF NO  
; MOV UNPTMP,DSDTMP ; SAVE WORD OFFSET AND DISK ADD'S  
; MOV UNPTMP+2,DSDTMP+2 ; OF 3-WORD GROUP CONTAINING  
; MOV UNPTMP+4,DSDTMP+4 ; FLAG WORD OF SUB-DOC ID  
; ADD R5,R0 ; R0->NEXT BYTE OF SUB-DOC ID TO  
; ; BE MATCHED  
; CMPB (R0),R2 ; SUB-DOC ID BYTE MATCH?  
; BEQ 10$ ; BRANCH IF YES  
; CLR SDID ; THIS IS NOT THE RIGHT SUB-DOC ID  
; BR 6$  
;  
; SUB-DOC ID MATCH STILL POSSIBLE.  
;  
; CMP #2,NOCNVT ; LAST BYTE IN STRING ?  
; BLT 6$ ; BRANCH IF NO  
;  
; SUB-DOC ID MATCH !!!!  
;  
; CLR NOCNVT  
; SEC  
; BR 7$  
;  
; 6$: MOVB R2,(R4)+ ; YES, STORE UNTRANSLATED CHARACTER IN OUTPUT  
; DEC NOCNVT ; DECREMENT NO CONVERT FLAG
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - D SUBROUTINES.
STORE - STORE CHARACTER.

MACRO M1110 27-MAR-88 14:31 P 27-1

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
1309 006402 000433 BR 4$ ; CHECK OUTPUT BUFFER FULL?
1310 006404 ;
1311 006404 122702 000000G CMPB #ZMK,R2 ; ZONE MARKER?
1312 006410 001423 BEQ 2$ ; YES
1313 006412 122702 000000G CMPB #SZMK,R2 ; SUB-ZONE MARKER?
1314 006416 001420 BEQ 2$ ; YES
1315 006420 122702 000000G CMPB #PRGH,R2 ; PARAGRAPH MARKER?
1316 006424 001420 BEQ 3$ ; YES
1317 006426 122702 000000G CMPB #SHTC,R2 ; SENTENCE MARKER?
1318 006432 001415 BEQ 3$ ; YES
1319 006434 122702 000000G CMPB #SDMK,R2 ; SUB-DOCUMENT MARKER?
1320 006440 001012 BNE 3$ ; NO
1321 006442 012737 000013 000000G MOV #11,,NOCNVT ; DO NOT CONVERT NEXT 11 CHARACTERS
1322 006450 012737 000001 006564' MOV #1,SDID ; SUB-DOC ID START SEQUENCE
1323 006456 000403 BR 3$ ; TRANSLATE SUB-DOC MARKER
1324 006460 ;
1325 006460 012737 000001 000000G MOV #1,NOCNVT ; DO NOT TRANSLATE NEXT CHARACTER ONLY
1326 006466 ;
1327 006466 116224 000000G MOVB TRANSA(R2),(R4)+ ; TRANSLATE THIS CHARACTER
1328 006472 ;
1329 006472 062737 000001 000002G ADD #1,TEMP+WORD1 ; INCREMENT DOCUMENT CHARACTER COUNT
1330 006500 005537 000000G ADC TEMP ; ADJUST FOR CARRY
1331 006504 022737 000000G 000000G CMP #AR,DSD,RQSTWD ; DELETE SUB-DOC REQUEST?
1332 006512 001411 BEQ 8$ ; BRANCH IF YES
1333 006514 020427 000000C CMP R4,#RTDBUF+SLB ; OUTPUT BUFFER FULL?
1334 006520 103410 BLO 5$ ; NO
1335 006522 012700 000000G MOV #RTDFDB,R0 ; R0 -> FDB
1336 006526 004737 007740' JSR PC,WRTRTD ; WRITE ONE OUTPUT BUFFER
1337 006532 005237 000000G INC R9 ; INC # OF BLOCKS WRITTEN
1338 006536 012704 000000G MOV #RTDBUF,R4 ; R4 -> OUTPUT BUFFER
1339 006542 000241 8$: CLC
1340 006544 5$: RESTOR R0
1341 006546 000207 7$: RTS PC
1342 ;
1343 ;
1344 006550 UNPTMP::BLKW 3
1345 006556 DSDTMP::BLKW 3
1346 006564 000000 SDID::WORD 0
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

DBUSBR - DBU SUBROUTINES
 UNPACK- UNPACK DATA BASE DOCUMENT

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

1348          .SBTTL UNPACK- UNPACK DATA BASE DOCUMENT.
1349          ;
1350          ; UNPACK ROUTINE UNPACKS THE 6-BIT CHARACTERS, TRANSLATES THEM TO THE 7-OR-8-
1351          ; BIT ASCII CHARACTER, PLACES THEM IN AN OUTPUT BUFFER AND CALLS THE WRITE
1352          ; MODULE TO WRITE EACH OUTPUT BUFFER.
1353          ;
1354          ; INPUT:
1355          ; R3 CONTAINS THE ADDR OF THE FIRST PACKED WORD.
1356          ; R4 CONTAINS THE POINTER TO THE OUTPUT BUFFER.
1357          ; R5 -> SUB-DOC ID STRING (FOR DELETE SUB-DOC)
1358          ; PCCT HAS NUMBER OF PACKED WORDS IN INPUT BUFFER.
1359          ; UPCT HAS THE EXPECTED NUMBER OF UNPACKED CHARACTERS.
1360          ;
1361          ; OUTPUT:
1362          ; TEMP,TEMP+2 CONTAINS TEXT CHARACTER COUNT.
1363          ;
1364          ; REGISTERS DESTROYED: R1,R2.
1365          ;
1366          UNPACK::
1367          CLR R9 ; CLEAR RETRIEVED DOCUMENT BLOCK COUNT.
1368          CLR TEMP ; CLEAR TEMPORARY STORAGE AREA FOR CHARACTER COUNT.
1369          CLR TEMP+WORD1 ;
1370          JMP U.7 ; START UNPACKING 5TH WORD 6TH CHARACTER.
1371
1372          U.1:
1373          MOV R3,UNPTMP ; SAVE WORD OFFSET AND SECTOR ADD'S.
1374          MOV DOCADR,UNPTMP+2 ; OF 3-WORD GROUP.
1375          MOV DOCADR+2,UNPTMP+4
1376          MOV (R3),R2 ; 1ST WORD OF 3 WORD BLOCK REPEAT PATTERN.
1377          UNPAK ; 1ST CHAR.
1378          MOV (R3),R2 ; STILL 1ST WORD OF 3 WORD BLOCK.
1379          ASL R2 ; MOVE 2ND SIX-BIT CHARACTER--
1380          ASL R2 ; -- TO LOWER 6 BITS OF R2
1381          SWAB R2 ;
1382          UNPAK ; 2ND CHAR.
1383          MOV (R3)+,R1 ; NEED UPPER 4 BITS OF FIRST WORD.
1384          CMP R3,#RDBUF+SLB ; END OF INPUT BUFFER?
1385          BLO 4# ; NOT YET.
1386          JSR PC,READN ; READ NEXT SECTOR OF PACKED DOC.
1387          CMP #AS,SUP,STATWD ;
1388          BEQ 4# ;
1389          JMP U.3 ;
1390
1391          4$:
1392          SUB #1,PCT+WORD1 ; DECREMENT PACKED WORD COUNT.
1393          SBC PUCT ;
1394          MOV (R3),R2 ; 2ND WORD OF 3 WORD REPEAT PATTERN.
1395          ROL R1 ; 4TH BIT TO C-BIT
1396          ROL R2 ; C-BIT TO BIT0 OF R2.
1397          ROL R1 ; 3RD BIT TO C-BIT
1398          ROL R2 ; C-BIT TO BIT0 OF R2.
1399          ROL R1 ; 2ND BIT TO C-BIT
1400          ROL R2 ; C-BIT TO BIT0 OF R2.
1401          ROL R1 ; 1ST BIT TO C-BIT.
1402          ROL R2 ; C-BIT TO BIT0 OF R2.
1403          UNPAK ; 3RD CHAR.
1404          MOV (R3),R2 ; STILL 2ND WORD OF 3 WORD BLOCK.
1405          ASL R2 ; MOVE 4TH 6-BIT CHAR. TO
1406          ASL R2 ; -- LOWER 6 BITS

```

```

1405 007122. UNPAK. ;4TH CHAR.
1406 007172. 011302. ; STILL 2ND WORD OF 3 WORD BLOCK.
1407 007174 000302. UNPAK. R2. ; GET 5TH CHAR. IN LOWER 6 BITS OF R2.
1408 007176 UNPAK. ;5TH CHAR.
1409 007246 012301 U.7: MOV. (R3)+,R1 ; NEED UPPER 2 BITS OF 2ND WORD.
1410 007250 020327 000000C. CMP. R3,#RDBUF+SLB. ; END OF INPUT BUFFER?.
1411 007254 103406 BLO. 10$ ; NO.
1412 007256 004737 005476' JSR. PC,READN. ; YES, READ NEXT SECTOR OF PACKED DOC.
1413 007262 022737 000000G. 000000G. CMP. #AS,SUP,STATWD
1414 007270 001167 BNE. U.3
1415 007272. 10$:
1416 007272. 162737 000001 000002G. SUB. #1,PWCT+WORD1 ; DECREMENT PACKED WORD COUNT.
1417 007300 005637 000000G. SBC. PWCT
1418 007304 011302. MOV. (R3),R2. ; 3RD WORD OF 3 WORD BLOCK.
1419 007306 006101 ROL. R1 ; 2ND BIT TO C-BIT
1420 007310 006102. ROL. R2. ; C-BIT TO BIT0 OF R2.
1421 007312. 006101 ROL. R1 ; 1ST BIT TO C-BIT
1422 007314 006102. ROL. R2. ; C-BIT TO BIT0 OF R2.
1423 007316 UNPAK. ;6TH CHAR.
1424 007366 011302. MOV. (R3),R2. ; STILL 3RD WORD OF 3 WORD BLOCK.
1425 007370 006202. ASR. R2. ; GET 7TH CHAR. IN LOWER --
1426 007372. 006202. ASR. R2. ; -- 6 BITS OF R2.
1427 007374 006202. ASR. R2. ;
1428 007376 006202. ASR. R2. ;
1429 007400 UNPAK. ;7TH CHAR.
1430 007450 012302. MOV. (R3)+,R2. ; STILL 3RD WORD OF 3 WORD BLOCK.
1431 007452. 006202. ASR. R2. ;
1432 007454 006202. ASR. R2. ;
1433 007456 000302. SWAB. R2. ; GET 8TH CHAR. IN LOWER 6 BITS OF R2.
1434 007460 162737 000001 000002G. SUB. #1,PWCT+WORD1 ; DECREMENT PACKED WORD COUNT.
1435 007466 005637 000000G. SBC. PWCT
1436 007472. UNPAK. ;8TH CHAR.
1437 007542. 020327 000000C. CMP. R3,#RDBUF+SLB. ; END OF READ BUFFER?.
1438 007546 103406 BLO. 10$ ; NO.
1439 007550 004737 005476' JSR. PC,READN. ; YES, READ NEXT PACKED DOCUMENT SECTOR.
1440 007554 022737 000000G. 000000G. CMP. #AS,SUP,STATWD
1441 007562. 001032. BNE. U.3
1442 007564. 14$:
1443 007564 000137 006606' JMP. U.1 ; CONTINUE.
1444 007570 U.15:
1445 007570 022737 000000G. 000000G. CMP. #AR,DSD,ROSTWD ; DELETE SUB-DOC.?
1446 007576 001424 BEQ. U.3 ; BRANCH IF YES.
1447 007600 012700 000000G. MOV. #RTDFDB,R0 ; R0 -> FDB.
1448 007604 004737 007740' JSR. PC,WRTTRD. ; WRITE FINAL BLOCK OUT.
1449 007610 005237 000000G. INC. R9 ; INC. # OF BLOCKS WRITTEN.
1450
1451 ;
1452 ; MAKE SURE FILE LENGTH IS A MULTIPLE OF 4 SECTORS.
1453 007614 016002. 000002G. 1$: MOV. F,EFBK+2(R0),R2. ; R2 = END-OF-FILE BLOCK NUMBER
1454 007620 005302. DEC. R2. ; R2 = BLOCK LENGTH OF FILE.
1455 007622. 032702. 0000003 BIT. #3,R2. ; MULTIPLE OF 4 BLOCKS?.
1456 007626 BOFF. U.3 ; BRANCH IF YES.
1457 007630 062760 000001 000002G. ADD. #1,F,EFBK+2(R0) ; INCREMENT BLOCK LENGTH.
1458 007636 005560 000000G. ADC. F,EFBK(R0)
1459 007642. 005237 000000G. INC. R9 ; INC. # OF BLOCKS WRITTEN.
1460 007646 000762. BR. 1$ ; TEST AGAIN.
1461 007650 000241 U.3: CLC.
    
```

DBUSBR - DBU-SUBROUTINES
UNPACK- UNPACK-DATA-BASE-DOCUMENT

MACRO-N1110-27-000-00 PAGE 01
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1462 007652 000207
1463 007654 000261
1464 007655 000207

U:2: RTS PC
SEC PC
RTS PC

DBUSBR - IN SUBROUTINES
WRTDB - WRITE DATA BASE

MACRO M1110 27-MAR-80 14:31 P. 29
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
.SBTTL WRTDB- WRITE DATA BASE
;
; WRTDB WRITES ONE SECTOR OF PACKED DOCUMENT TO THE DATA BASE DISK. IT THEN
; ZEROS OUT THE WORKING BUFFER AND RE-ESTABLISHES THE POINTERS FOR THE NEXT
; DATA BASE BUFFER.
;
WRTDB:
CLR WRTST ;
MOV #10,WLB,DQ10+Q.10FN ; WRITE FUNCTION CODE
JSR PC,ISSDIR ; ISSUE WRITE DIRECTIVE
TST BRSTAT ; GOOD WRITE?
BEQ 1$ ; YES
JSR PC,EXIT ; FATAL ERROR
1$:
ADD #1,DQ10+Q.10PL+10 ; GET ADDR OF NEXT SECTOR
ADC DQ10+Q.10PL+6 ; ACCOUNT FOR CARRY
MOV #WBUF,R0 ; CLEAR WORKING BUFFER
JSR PC,CLEARB ;
MOV #WBUF,R1 ; RE-ESTABLISH POINTER
RTS PC ;
;
;
.SBTTL WRTTRD- WRITE OUTPUT BUFFER
;
; THIS ROUTINE WRITES ONE SECTOR OF A RETRIEVED DOCUMENT AS IT IS
; UNPACKED. IT THEN CLEARS THE OUTPUT BUFFER FOR MORE OUTPUT.
;
; ON ENTRY, R0->FDB
; ON RETURN, R0->FDB
;
WRTTRD::SAVE R0
WRITE$ R0 ; WRITE OUTPUT BUFFER
WAIT$ R0 ; FOR I/O COMPLETION
BCC 1$ ; SUCCESSFUL WRITE
MOV# @#WIOST,PAR1 ; WRITE ERROR
MOV #EMSG2,R1
CALL ERROR
MOV #AS,UWD,STATWD ; UNABLE TO READ DISK
1$:
MOV #RTDBUF,R0 ; CLEAR OUTPUT BUFFER
JSR PC,CLEARB ; CLEAR BUFFER
RESTOR R0
RTS PC ;
.END
```

ACKBUF = ***** GX.	CF.UPD = 000043	FN.NMB = 000022.	004 NDCADR = ***** GX.	READN = 005476RG.
ACKECT = ***** GX.	CINDXE 001064RG	FN.QLS = 000006	004 NDOC = ***** GX.	RPLFLG = 004614R.
ACKFDB = ***** GX.	CIN005 001134R.	FN.RDC = 000014	004 NOCNVT = ***** GX.	ROSTWD = ***** GX.
ACKPTR = ***** GX.	CIN010 001144R.	FN.UPD = 000012.	004 NSBUF = ***** GX.	ROJWS = ***** GX.
ADDRID = ***** GX.	CIN020 001266R.	F.BKVB = ***** GX.	NSREAD = ***** GX.	RSTEOC = 006032RG.
AR.ADD = ***** GX.	CIN025 001306R.	F.EFBK = ***** GX.	NS2RD = ***** GX.	RTDBUF = ***** GX.
AR.DSD = ***** GX.	CIN030 001316R.	F.ERR = ***** GX.	N.BFAC = 000004	RTDFDB = ***** GX.
AR.RPD = ***** GX.	CIN040 001354R.	IBFADR = ***** GX.	N.BHGH = 000006	R\$IND = ***** GX.
AS.DNE = ***** GX.	CIN050 001414R.	IDXADR = ***** GX.	N.BHGH = 000006	R\$IPR = ***** GX.
AS.FAT = ***** GX.	CIPRE = 001452RG	IE.ITS = ***** GX.	N.BIPE = ***** GX.	R.SUTN = 000002
AS.SUP = ***** GX.	CIP005 001522R.	INCOUT = ***** GX.	N.BTCH = 000004	R8 = ***** GX.
AS.URD = ***** GX.	CIP010 001536R.	IN.CEC = ***** GX.	N.BUFB = 004000	R9 = ***** GX.
AS.UUD = ***** GX.	CIP015 001632R.	IN.FE = ***** GX.	N.BUFW = 002000	SAVEPT = ***** GX.
BITST = ***** GX.	CIP020 001636R.	IN.SAL = ***** GX.	N.BXE = ***** GX.	SADATA = 005622RG.
BITVAL = 000000	CIP030 001726R.	IN.SAN = ***** GX.	N.FOS = 000764	SDID = 006564RG.
BIT0 = 000001	CIP040 001756R.	IN.TYP = ***** GX.	N.QURY = 000031	SDMK = ***** GX.
BIT1 = 000002	CIP050 001772R.	IDST = ***** GX.	N.SUNT = 000002	SLB = ***** GX.
BIT10 = 002000	CIP060 001776R.	ID.LBL = ***** GX.	OPEACK = ***** GX.	SLW = ***** GX.
BIT11 = 004000	CIP070 002036R.	ID.UILB = ***** GX.	PACK6 = 004676R.	SNTC = ***** GX.
BIT12 = 010000	CIP080 002056R.	IPRBUF = ***** GX.	PAR\$\$\$ = 000000	SREC = ***** GX.
BIT13 = 020000	CIP090 002126R.	IP.CEC = ***** GX.	PAR1 = ***** GX.	SR.ARE = 000114
BIT14 = 040000	CLEARB 002130RG	IP.FE = ***** GX.	PCCT = ***** GX.	SR.ARS = 000106
BIT15 = 100000	CLOSEF 002150RG	IP.HID = ***** GX.	PCK1 = 004700R.	SR.DAY = 000010
BIT2 = 000004	CMPST = ***** GX	IP.LIA = ***** GX.	PCK10 = 005220R.	SR.DLT = 000014
BIT3 = 000010	CMP2ID 002204RG	IP.SRL = ***** GX.	PCK2 = 004726R.	SR.ECB = 000047
BIT4 = 000020	CMP2WD 002322RG	IP.TYP = ***** GX.	PCK3 = 004754R.	SR.ECH = 000046
BIT5 = 000040	DBACKB = ***** GX	ISSDR = 002476RG	PCK4 = 005022R.	SR.ECL = 000050
BIT6 = 000100	DBSLEN = 000116	IS.SUC = ***** GX.	PCK5 = 005050R.	SR.FIB = 000012
BIT7 = 000200	DELTA = ***** GX.	I.BIDH = ***** GX.	PCK6 = 005076R.	SR.GRE = 000100
BIT8 = 000400	DG.ERR = 001000	I.BIDL = ***** GX.	PCK7 = 005144R.	SR.GRS = 000072
BIT9 = 001000	DG.SDF = 002000	I.BIDM = ***** GX.	PCK8 = 005172R.	SR.LEN = 000122
BUFADR = ***** GX.	DG.TDF = 004000	I.PGI = ***** GX.	PC.IDH = ***** GX.	SR.LIN = 000066
BYTE0 = 000000	DOCADR = ***** GX.	I.RWS = ***** GX.	PC.IDL = ***** GX.	SR.LIP = 000062
BYTE1 = 000001	DOCID = ***** GX.	I.SANH = ***** GX.	PC.IDM = ***** GX.	SR.MON = 000006
BYTE2 = 000002	DQIO = ***** GX.	I.SANL = ***** GX.	PC.PAD = ***** GX.	SR.NDC = 000042
BYTE3 = 000003	DRSTAT = ***** GX.	I.UUS = ***** GX.	PC.PWC = ***** GX.	SR.NDS = 000036
BYTE4 = 000004	DSDTMP 006556RG	LINDXE = 002622RG	PC.STW = ***** GX.	SR.NIN = 000030
BYTE5 = 000005	EMSG0 = ***** GX.	LIPRE = 003254RG	PC.TCC = ***** GX.	SR.NIP = 000022
BYTE6 = 000006	EMSG1 = ***** GX.	LIP020 003374R.	PC.TYP = ***** GX.	SR.SDB = 000032
BYTE7 = 000007	EMSG15 = ***** GX.	LIP040 003464R.	PC.WSC = ***** GX.	SR.SUN = 000000
BYTE8 = 000010	EMSG2 = ***** GX.	LIP060 003630R.	PPDATA = 004004RG.	SR.TWS = 000056
BYTE9 = 000011	ENDBFA = ***** GX.	LIP080 003640R.	PPD010 004130R.	SR.UJL = 000052
BYTVAL = 000012	END = ***** GX.	LIP085 003676R.	PPD030 004462R.	SR.YR = 000004
CACKE = 000000RG.	EOT = ***** GX.	LIP090 003736R.	PPD035 004512R.	SR.11N = 000024
CAC010 000142R.	ERROR = ***** GX.	LIX010 002664R.	PPD040 004526R.	SR.11P = 000016
CDSTAT = 000262RG.	ETRYCT = ***** GX.	LIX020 002730R.	PPD041 004570R.	STATWD = ***** GX.
CDS010 000316R.	EXIT = 002440RG	LIX035 003050R.	PRB = ***** GX.	STORE = 006254R.
CDS020 000334R.	FD.FID 000000	LIX045 003060R.	PRGH = ***** GX.	SZMK = ***** GX.
CDS025 000426R.	FD.FNB 000006	LIX050 003072R.	PWCT = ***** GX.	TD\$CTR = 176370
CDS030 000464R.	FD.FYR 000004	LIX060 003126R.	Q.IOFN = ***** GX.	TD\$CTW = 176360
CDS035 000470R.	FD.LEN 000010	LNSBUF = ***** GX.	Q.IOPL = ***** GX.	TD\$INL = 004000
CF.COT = 000041	FN.ACK 000016	MECIND = ***** GX.	RDBUF = ***** GX.	TD\$MEM = 000270
CF.DGN = 000046	FN.FSA 000000	MSGOUT = ***** GX.	RDEOC = 001126RG.	TD\$ORD = 176344
CF.DHR = 000042.	FN.FSB 000002.	MSQ = ***** GX.	RDSTAT = ***** GX.	TD\$OTR = 176346
CF.DMC = 000047	FN.FSC 000004	MS.DGN = 010000	READ = 005224RG.	TD\$ORD = 000274
CF.HBR = 000045	FN.FSD 000020	N = 000012.	READF = 005424RG.	TD\$RST = 176366
CF.HRL = 000044	FN.MHR 000010			

DBUSBR - D SUBROUTINES
SYMBOL TABLE

MACRO: M1110 27-MAR-88 14:31 P 38-3
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

TD\$SW = 176376	T\$EMEM = 010000	T\$RNB = 000040	U:3 007650R	WRTDB = 007660R
TD\$TAR = 176372	T\$FSAA = 000000	T\$RSET = 040000	U:7 007246R	WRTTRD = 007740R6
TD\$TAU = 176362	T\$FSAB = 000004	T\$SC = 000022	WBUF = ***** GX	WRTST = ***** GX
TD\$TDR = 176374	T\$FSAC = 000014	T\$SCL = 020000	WDOFF = ***** GX	XMTACK = ***** GX
TD\$TDW = 176364	T\$FSB2 = 000010	T\$SEG1 = 000000	WDOFND = ***** GX	XTEMP = ***** GX
TEMP = ***** GX	T\$IB = 000026	T\$SEG2 = 000001	WHTSPC = 004616R	ZMK = ***** GX
TRANSA = ***** GX	T\$IBAR = 000024	T\$SEG3 = 000002	WIOST = ***** GX	\$DDIV = ***** GX
T\$AD = 000020	T\$IBF = 020000	T\$SO = 000001	WORD0 = 000000	\$DMUL = ***** GX
T\$BA = 000002	T\$IBF = 040000	T\$UBUS = 100000	WORD1 = 000002	\$DSW = ***** GX
T\$BD = 000010	T\$ICD = 000040	T\$ICLK = 000400	WORD2 = 000004	\$MUL = ***** GX
T\$BSO = 100000	T\$MODE = 004000	T\$IBEN = 000020	WORD3 = 000006	\$#\$ = 000006R 005
T\$BT = 000020	T\$OB = 000036	UNPACK = 006566RG	WORD4 = 000010	\$#\$OST = 000006
T\$BTAR = 000030	T\$OBE = 004000	UNPTMP = 006550RG	WORD5 = 000012	\$#\$T1 = 000004
T\$BTD = 002000	T\$OBF = 010000	UNWS = ***** GX	WORD6 = 000014	.CLOSE = ***** G
T\$CD = 000100	T\$OBRA = 000034	UPCCT = ***** GX	WORD7 = 000016	.READ = ***** G
T\$CLK = 002000	T\$OBWA = 000032	U:1 006606R	WORD8 = 000020	.WAIT = ***** G
T\$DISK = 000200	T\$OUTA = 100000	U:15 007570R	WORD9 = 000022	.WRITE = ***** G
T\$DRD = 000004	T\$RBDO = 000200	U:2 007654R	WRDVAL = 000024	

. ABS . 000000 000
010014 001
SRCOFF 000122 002
FDSCOF 000010 003
FNOFFS 000022 004
\$DPB\$# 000014 005
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 6646 WORDS (26 PAGES)
DYNAMIC MEMORY: 8084 WORDS (31 PAGES)
ELAPSED TIME: 00:01:21
DBUSBR, DBUSBR / - SP=C20, 1JP, C, DBUSBR

.MAIN: MACRO-M1110 27-MAR-80 14:34
TABLE OF CONTENTS:

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

8-	1	TASK NAME: INITDB - INITIALIZE DATA BASE.
8-	2	
8-	3	
8-	4	
8-	5	
8-	6	
8-	7	CREATION DATE: 10 AUG 78
8-	8	MODIFICATIONS: DATE:
8-	9	TIME:
9-	11	TASK OVERVIEW:
11-	43	TASK IMMEDIATES:
12-	120	TASK VARIABLES:
13-	136	TASK BUFFERS:
15-	152	MESSAGES TO CONSOLE:
16-	192	ERROR MESSAGES:
17-	249	INITDB - MAIN CONTROL LOOP:
18-	307	CLRBUF - CLEAR BUFFER:
19-	322	DIVIDE - GENERAL DIVIDE ROUTINE:
20-	344	GCLIN - GET COMMAND LINE INPUTS:
21-	475	IINDEX - INITIALIZE INDEX RECORDS:
22-	529	IIPR - INITIALIZE INDEX POINTER RECORDS:
23-	584	INIBLK - INITIALIZE INITIALIZATION BLOCK:
24-	707	VFYDSK - VERIFY DATA BASE DISK PACK:
25-	818	.OD3CT - CONVERT FROM ASCII TO THREE WORD BINARY:

INITDB-- INITIALIZE DATA BASE D MACRO M1110 27-MAR-80 14:34 E 8
TASK NAME INITDB-- INITIALIZE DATA BASE
Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

1
2
3
4
5
6
7
8
9

.SBTTL TASK NAME: INITDB-- INITIALIZE DATA BASE
.SBTTL
.SBTTL
.SBTTL
.SBTTL PHONE:
.SBTTL
.SBTTL CREATION DATE: 10 AUG 79
.SBTTL MODIFICATIONS: DATE:
.SBTTL TIME:

```
11          .SBTTL - TASK OVERVIEW.  
12          ;  
13          ;  
14          ;  
15          ; INITDB INITIALIZES AND FORMATS THE DISK ON THE HSTS SEARCH SUBSYSTEM.  
16          ; WHEN INITDB IS INITIATED BY A COMMAND FROM THE OPERATOR CONSOLE, INITDB  
17          ; PROMPTS FOR THREE INPUTS FROM THE OPERATOR:  
18          ;  
19          ;         DOCUMENT ID RANGE START (DECIMAL)  
20          ;         DOCUMENT ID RANGE END (DECIMAL)  
21          ;         NUMBER OF DOCUMENTS EXPECTED FOR THIS DISK (DECIMAL)  
22          ;  
23          ;  
24          ; INITDB DETERMINES THE NUMBER OF IPR S AND INDEX RECORDS NEEDED FOR THE  
25          ; NUMBER OF DOCUMENTS TO BE STORED ON THE DISK. THE TASK CREATES THE  
26          ; INITIALIZATION BLOCK WHICH PRECEDES THE IPR SECTION ON THE DISK. INITDB  
27          ; ALSO STORES THE SECTOR ADDRESS OF EACH IPR AND INDEX RECORD IN THE  
28          ; CORRESPONDING SECTOR AND WRITES IT TO THE DISK.  
29          ;  
30          ; ALL INFORMATION PERTINENT TO THE DATA BASE IS KEPT IN THE INITIALIZATION  
31          ; BLOCK WHICH IS INITIALIZED IN INITDB. THE INITIALIZATION BLOCK WILL  
32          ; CONTAIN THE CURRENT NUMBER OF DOCUMENTS ON THE DISK, THE ACTUAL DOCUMENT  
33          ; ID RANGE, THE FIRST SECTOR CONTAINING THE SEARCHABLE DATA, THE AMOUNT OF  
34          ; WHITE SPACE, THE SECTOR CONTAINING THE END-OF-COLLECTION SEQUENCE, ETC.  
35          ;
```

INITDB -- INITIALIZE DATA BASE DISK MACRO M1110 27-MAR-80 14:34 E. 10
TASK OVER

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

37
38 000000
39
40
41

.TITLE .INITDB -- INITIALIZE DATA BASE DISK
.PSECT .INITDB
.ENABL .AMA : ASSEMBLY MODE ABSOLUTE
.NLIST .MEB
.NLIST .BEX

```
43 .SBTTL - TASK IMMEDIATES
44 .MCALL - Q10W$, Q10U$, DIR$, ABRT$, EXIT$, GTIM$.
45 ;
46 ;
47 000000 DSW = 0 ; DIRECTIVE STATUS WORD
48 000001 DPLUN = 1 ; DATA-BASE DISK LUN
49 000005 TTLUN = 5 ; SYSTEM CONSOLE LUN
50 000001 EFN1 = 1 ; EVENT FLAG FOR DISK QIO'S
51 000005 EFN5 = 5 ; EVENT FLAG FOR CONSOLE QIO'S
52 000062 R1PRI = 50 ; READ PRIORITY
53 000062 W1PRI = 50 ; WRITE PRIORITY
54 000062 WSPRI = 50 ; WRITE PRIORITY
55 000040 VFC = 40 ; VERTICAL FORMAT CONTROL
56 000015 CR = 15 ; CARRIAGE RETURN
57 000012 LF = 12 ; LINE FEED
58 000015 NCHIN = 15 ; ALLOWABLE NUMBER OF INPUT CHARACTERS
59 001000 EOC = 001000 ; END-OF-COLLECTION MARKER
60 ;
61 ;
62 000000 000000 000000 RSW: .WORD 0,0 ; READ STATUS WORD
63 000004 000000 000000 WSW: .WORD 0,0 ; WRITE STATUS WORD
64 ;
65 ;
66 INDEX POINTER RECORDS AND INDEX RECORDS IMMEDIATES
67 ;
68 000012 DELTA = 10 ; IPR ENTRIES POINT TO EVERY 10TH INDEX RECORD
69 001000 LSCTR = 512 ; SECTOR LENGTH IN BYTES
70 000006 N.BIPE = WORD3 ; NUMBER OF BYTES PER IPR ENTRY
71 000016 N.BXE = WORD7 ; NUMBER OF BYTES PER INDEX ENTRY
72 000044 MECIND = 36 ; MAXIMUM ENTRY COUNT FOR INDEX RECORDS
73 000121 MECIPR = 81 ; MAXIMUM ENTRY COUNT FOR IPR RECORDS
74 000200 R$IPR = BIT7 ; IPR RECORD TYPE
75 000100 R$IND = BIT6 ; INDEX RECORD TYPE
76 ;
77 INDEX POINTER RECORD OFFSET DEFINITIONS
78 ;
79 000000 .PSECT: IPROFF,ABS
80 000000 IP.SAN: .BLKB 1 ; HIGH ORDER SECTOR ADDRESS
81 000001 IP.TYP: .BLKB 1 ; RECORD TYPE
82 000002 IP.SAL: .BLKW 1 ; LOW ORDER SECTOR ADDRESS
83 000004 IP.DLT: .BLKW 1 ; DELTA FOR # OF X RCDS REPRESENTED
84 000006 IP.CEC: .BLKW 1 ; CURRENT ENTRY COUNT
85 000010 IP.SRL: .BLKW 1 ; # OF SECTORS REPRESENTED IN LAST ENTRY
86 000012 IP.HID: .BLKW 3 ; HIGHEST DOCUMENT ID REPRESENTED
87 000020 IP.BPE: .BLKW 1 ; BYTES PER IPR ENTRY
88 000022 IP.LIA: .BLKW 2 ; LAST INDEX RECORD ADDRESS REPRESENTED
89 000026 IP.PAD: .BLKW 1 ; PAD FILL
90 000030 IP.FE: .BLKW 1 ; FIRST ENTRY OFFSET
91 ;
92 INDEX POINTER RECORD ENTRY OFFSETS
93 .PSECT: IPEOFF,ABS
94 000000 PE.IDH: .BLKW 1 ; HIGH ORDER WORD OF DOCUMENT ID
95 000002 PE.IDM: .BLKW 1 ; MIDDLE ORDER WORD OF DOCUMENT ID
96 000004 PE.IDL: .BLKW 1 ; LOW ORDER WORD OF DOCUMENT ID
97 ;
98 INDEX RECORD OFFSET DEFINITIONS
99 000000 .PSECT: INDOFF,ABS
```

INITDB: - INITIALIZE DATA BASE D MACRO M1110 27-MAR-80 14:34 E: 11-1
TASK: IMMEDIATE

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
100 000000      IN.SAN: .BLKB 1      ; HIGH ORDER SECTOR ADDRESS  
101 000001      IN.TYP: .BLKB 1      ; RECORD TYPE  
102 000002      IN.SAL: .BLKW 1      ; LOW ORDER SECTOR ADDRESS  
103 000004      IN.CEC: .BLKW 1      ; CURRENT ENTRY COUNT  
104 000006      IN.MEC: .BLKW 1      ; MAXIMUM ENTRY COUNT  
105 000010      IN.FE: .BLKW 1      ; FIRST ENTRY OFFSET  
106  
107      ; INDEX RECORD ENTRY OFFSETS DEFINITIONS FOR EACH DOCUMENT IN RECORD  
108 000000      .PSECT INEOFF,ABS  
109 000000      I.BIDH: .BLKB 1      ; BINARY DOCUMENT ID HIGH ORDER  
110 000001      I.PGI: .BLKB 1      ; BYTE CONTAINING PURGE INDICATOR  
111 000002      I.BIDM: .BLKW 1      ; BINARY DOCUMENT ID MIDDLE WORD  
112 000004      I.BIDL: .BLKW 1      ; BINARY DOCUMENT ID LOW ORDER  
113 000006      I.SANH: .BLKB 1      ; HIGH ORDER SECTOR ADDRESS NUMBER  
114 000007      I.BC: .BLKB 1      ; BYTE COUNT WITHIN SECTOR  
115 000010      I.SANL: .BLKW 1      ; LOW ORDER SECTOR ADDRESS NUMBER  
116 000012      I.RWS: .BLKW 1      ; REQUIRED WHITE SPACE FOR THIS DOCUMENT  
117 000014      I.UWS: .BLKW 1      ; UNUSED WHITE SPACE FOR THIS DOCUMENT  
118 000010      .PSECT INITDB
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
120                                     .SBTTL TASK VARIABLES
121                                     :
122                                     :
123 000010                               ASCSTR: .BLKB 10.           ; ASCII BUFFER FOR OCTAL INPUT
124 000022 000000 000000 000000        BIN3WD: .WORD 0,0,0       ; THREE WORD BINARY FIELD FOR CONVERSION
125                                     :
126 000030 000000                       ABORT: .WORD 0           ; ABORT TASK FLAG
127 000032 000000 000000                TEMP: .WORD 0,0         ; TEMPORARY WORKING AREA
128                                     :
129                                     ; THE FOLLOWING TWO VARIABLES ARE SUBJECT TO CHANGE WITHOUT NOTICE
130                                     :
131 000036 000000 165140                 IBLK: .WORD 0,165140     ; INITIALIZATION BLOCK NUMBER
132 000042 000007 010500                 NSECTR: .WORD 7,010500    ; TOTAL # OF SECTORS ON DISK (463,168)
133                                     :
134 000046 111 104 102 PRMPT: .ASCII </IDB></> ; PROMPT ASCII STRING FOR TERMINAL INPUT
```

INITDB--INITIALIZE-DATA-BASE-D-MACRO-M1110 27-MAR-80 14:34 E:13
TASK-BUFF

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

136
137
138
139 000052.
140 000174
141 000214
142 001214

.SBTTL--TASK BUFFERS:
:
:
CLBUF: .BLKB 82. : COMMAND LINE BUFFER
DTBUF: .BLKW 8. : DATE TIME BUFFER
INITBF: .BLKB 512: : INITIALIZATION BUFFER
SECTOR: .BLKB 512: : SECTOR BUFFER FOR IPR AND INDEX INITIALIZATION

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
144      :  
145      :  
146      :  
147 002214      : TTQIOW: QIOW$ IO,WVB,TTLUN,EFN5,W5PRI,WSW,,<.,VFC> ; QIO-PARAMETER-BLOCK  
148      :  
149 002244      : DPQIOW: QIOW$ ,DPLUN,EFN1,R1PRI,RSW,,<.,512,.,.,>  
150      :
```



```
152.                                     .SBTTL- MESSAGES TO CONSOLE.
153.                                     ;
154.                                     ;
155 002274      120      114      105  STR1: .ASCIZ /PLEASE ENTER DOCUMENT ID RANGE START FOR THIS DISK/
156 002357                                     STR1E:
157.                                     .EVEN.
158.                                     ;
159 002360      120      114      105  STR2: .ASCIZ /PLEASE ENTER DOCUMENT ID RANGE END FOR THIS DISK/
160 002441                                     STR2E:
161.                                     .EVEN.
162.                                     ;
163 002442      120      114      105  STR3: .ASCIZ /PLEASE ENTER NUMBER OF DOCUMENTS EXPECTED FOR THIS DISK/
164 002532                                     STR3E:
165.                                     .EVEN.
166.                                     ;
167 002532 002274'  MADDR: .WORD  STR1
168 002534 002360'  .WORD  STR2
169 002536 002442'  .WORD  STR3
170.                                     ;
171 002540 000063  LTBL: .WORD  STR1-STR1
172 002542 000061  .WORD  STR2-STR2
173 002544 000070  .WORD  STR3-STR3
174.                                     ;
175 002546      104      111      123  STR4: .ASCII /DISK INITIALIZED ON /
176 002572      071      071      055  S4: .ASCII /99-AAA-99/CR><LF>
177 002605      104      117      040  STR4E: .ASCIZ /DO YOU WANT TO CONTINUE WITH THIS DISK (Y OR N)?/
178 002666                                     .EVEN.
179.                                     ;
180 002666 000120  LEN4: .WORD  STR4E-STR4
181.                                     ;
182 002670      120      114      105  STR5: .ASCII /PLEASE MOUNT A NEW DISK PACK/CR><LF>
183 002726      122      105      120  STR5E: .ASCIZ /REPLY WHEN NEW DISK IS READY (Y OR N)/
184 002774                                     .EVEN.
185.                                     ;
186 002774 000104  LEN5: .WORD  STR5E-STR5
187 002776      040      111      116  STR6: .ASCIZ /INITIALIZATION OF THE DISK IS COMPLETE /
188 003047                                     STR6E:
189.                                     .EVEN.
190 003050 000051  LEN6: .WORD  STR6E-STR6
```

```
192. ; .SBTTL ERROR MESSAGES
193. ;
194. ;
195 003052 000057 MSG1: .WORD LN1E-LN1
196 003054 003122' .WORD LN1
197 003056 000032' MSG2: .WORD LN2E-LN2
198 003060 003202' .WORD LN2
199 003062 000042' MSG3: .WORD LN3E-LN3
200 003064 003234' .WORD LN3
201 003066 000040' MSG4: .WORD LN4E-LN4
202 003070 003276' .WORD LN4
203 003072 000041' MSG5: .WORD LN5E-LN5
204 003074 003336' .WORD LN5
205 003076 000060' MSG6: .WORD LN6E-LN6
206 003100 003400' .WORD LN6
207 003102 000050' MSG7: .WORD LN7E-LN7
208 003104 003460' .WORD LN7
209 003106 000052' MSG8: .WORD LN8E-LN8
210 003110 003530' .WORD LN8
211 003112 000055' MSG9: .WORD LN9E-LN9
212 003114 003602' .WORD LN9
213 003116 000022' MSG10: .WORD LN10E-LN10
214 003120 003660' .WORD LN10
215. ;
216 003122 104 111 122 LN1: .ASCIZ /DIRECTIVE ERROR ON QIO TO CONSOLE USW = %1D/
217 003201 LN1E: .ASCIZ /
218. ;
219 003202 116 125 114 LN2: .ASCIZ /NULL INPUT - TASK ABORTED/
220 003234 LN2E: .ASCIZ /
221. ;
222 003234 104 117 103 LN3: .ASCIZ /DOCUMENT ID RANGE START TOO LARGE/
223 003276 LN3E: .ASCIZ /
224. ;
225 003276 104 117 103 LN4: .ASCIZ /DOCUMENT ID RANGE END TOO LARGE/
226 003336 LN4E: .ASCIZ /
227. ;
228 003336 105 122 122 LN5: .ASCIZ /ERROR CONVERTING INPUT TO BINARY/
229 003377 LN5E: .ASCIZ /
230. ;
231 003400 104 117 103 LN6: .ASCIZ /DOCUMENT ID RANGE START EQUAL TO OR > RANGE END/
232 003460 LN6E: .ASCIZ /
233. ;
234 003460 104 111 123 LN7: .ASCIZ /DISK NOT READY - TASK ABORTED RSW = %1D/
235 003530 LN7E: .ASCIZ /
236. ;
237 003530 104 111 122 LN8: .ASCIZ /DIRECTIVE ERROR ON READ TO DISK RSW = %1D/
238 003602 LN8E: .ASCIZ /
239. ;
240 003602 104 111 122 LN9: .ASCIZ /DIRECTIVE ERROR ON WRITE TO DISK USW = %1D/
241 003657 LN9E: .ASCIZ /
242. ;
243 003660 124 101 123 LN10: .ASCIZ /TASK IS ABORTING/
244 003702 LN10E: .ASCIZ /
245. ;
246. ;
247 003702 000000 PAR1: .WORD 0
```

INITDB-- INITIALIZE DATA BASE D MACRO M1110 27-MAR-80 14:34 E:17
INITDB-- MAIN CONTROL LOOP

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
249 .SBTTL- INITDB-- MAIN CONTROL LOOP.
250 ;
251 ;
252 003704          ;
253 003704 005001  ;
254 003706 004737 005462' CLR R1 ; INITIALIZE INDEX FOR INPUT SUBROUTINE.
255 003712 005737 000030' JSR PC,VFYDSK ; VERIFY DATA DISK PACK.
256 003716 001026 TST ABORT ; ERROR ON INPUT?
257 003720 004737 004326' BNE IDB100 ; YES.
258 003724 005737 000030' JSR PC,GCLIN ; GET INPUT FROM OPERATOR.
259 003730 001021 TST ABORT ; ANY PROBLEMS?
260 003732 004737 005764' BNE IDB100 ; YES.
261 003736 005737 000030' JSR PC,INIPLK ; INITIALIZE INITIALIZATION BLOCK.
262 003742 001014 TST ABORT ; PROBLEM?
263 003744 004737 005514' BNE IDB100 ; YES.
264 003750 005737 000030' JSR PC,IIPR ; INITIALIZE INDEX POINTER RECORDS.
265 003754 001007 TST ABORT ; PROBLEM WITH DISK?
266 003756 004737 005252' BNE IDB100 ; YES.
267 003762 005737 000030' JSR PC,IINDEX ; INITIALIZE INDEX RECORDS.
268 003766 001002 TST ABORT ; PROBLEM WITH DISK?
269 003770 000137 004016' BNE IDB100 ; YES.
270 003774 JMP IDB200 ; NO.
271 ;
272 004012 000137 004246' IDB100: MOUT$# #EMSG10 ; ISSUE ABORT MESSAGE.
273 ; JMP IDB300 ; EXIT.
274 ;
275 ; WRITE EOC INTO FIRST BLOCK
276 ;
277 004016 012704 001214' IDB200: MOV #SECTOR,R4 ;R4-> OUTPUT BUFFER.
278 004022 012714 001000 MOV #EOC,(R4) ; INSERT EOC MARKER.
279 004026 012737 000000G 002246' MOV #IO.WLB,DPQ10W+0,IOFN ; WRITE FUNCTION CODE
280 004034 012737 000004' 002254' MOV #WSW,DPQ10W+0,IOSB.
281 004042 012737 001214' 002260' MOV #SECTOR,DPQ10W+0,IOPL.
282 004050 012737 001000 002262' MOV #S12.,DPQ10W+0,IOPL+2.
283 004056 013737 000000C 002266' MOV SR,ECH+INITBF,DPQ10W+0,IOPL+6
284 004064 013737 000000C 002270' MOV SR,ECL+INITBF,DPQ10W+0,IOPL+10
285 004100 103014 DIR$ #DPQ10W.
286 004102 113737 000000 003702' BCC 1$
287 004110 MOV# @#DSW,PAR1
288 004130 000446 MOUT$# #EMSG9,#PAR1
289 004132 123727 000004' 000000G 1$: BR IDB300
290 004140 001414 CMP# WSW,#IS:SUC.
291 004142 113737 000004' 003702' BEQ 2$
292 004150 MOV# @#WSW,PAR1
293 004170 000426 MOUT$# #EMSG9,#PAR1
294 ; BR IDB300
295 ;
296 ; PRINT COMPLETION MESSAGE.
297 ;
298 004172 012737 002776' 002230' 2$: MOV #STR6,TTQ10W+0,IOPL ; ADDR OF MSG.
299 004200 013737 003050' 002232' MOV LENS,TTQ10W+0,IOPL+2 ; LENGTH OF MSG.
300 004214 103014 DIR$ #TTQ10W ; ISSUE MSG.
301 004216 113700 000004' BCC IDB300 ; SUCCESSFUL.
302 004222 010037 003702' MOV# WSW,R0 ; STATUS WORD TO R0 TO EXTEND SIGN.
303 004226 MOUT$# #EMSG1,#PAR1 ;
304 004246 MOUT$# #EMSG1,#PAR1
305 004246 IDB300: EXIT$#
```

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```
307                                     .SBTTL CLRBUF-- CLEAR BUFFER.
308                                     :
309                                     : INPUTS:
310                                     : R4 - ADDRESS OF BUFFER TO BE CLEARED.
311                                     : R5 - LENGTH OF BUFFER.
312                                     :
313 004254 CLRBUF:
314 004254 10$: SAVE R4,R5 ; SAVE REGISTERS.
315 004260 CLRBUF:
316 004260 105024 CLRB (R4)+ ; CLEAR BUFFER.
317 004262 005305 DEC R5 ;
318 004264 003375 BGT 10$ ; NOT DONE.
319 004266 RESTOR R4,R5 ; YES.
320 004272 000207 RTS PC ;
```

```
322 .                                .SBTTL- DIVIDE- GENERAL-DIVIDE-ROUTINE-
323 .                                ;
324 .                                ; INPUTS:
325 .                                ; STACK-CONTAINS-LOW-ORDER-DIVIDEND-
326 .                                ; HIGH-ORDER-DIVIDEND-
327 .                                ; DIVISOR
328 .                                ; OUTPUTS:
329 .                                ; R0 - REMAINDER-
330 .                                ; R1 - HIGH-ORDER-QUOTIENT-
331 .                                ; R2 - LOW-ORDER-QUOTIENT-
332 .                                ; STACK-IS-CLEARED-
333 .                                ;
334 .                                ;
335 004274                                ; DIVIDE:
336 004274 016602 000002                MOV. WORD1(SP),R2. ; LOW-ORDER-DIVIDEND-
337 004300 016601 000004                MOV. WORD2(SP),R1. ; HIGH-ORDER-DIVIDEND-
338 004304 016600 000006                MOV. WORD3(SP),R0. ; DIVISOR-
339 004310 004737 000000G                JSR. PC,$DDIV. ; OBTAIN-QUOTIENT-
340 004314 011666 000006                MOV. (SP),WORD3(SP) ; MOV-RETURN-ADDRESS-
341 004320 062706 000006                ADD. #WORD3,SP. ; CLEAR-STACK-
342 004324 000207                RTS. PC. ;
```

```

344      ,SBTTL -- GCLIN -- GET COMMAND LINE INPUTS
345
346      ; THIS TASK GETS THE COMMAND LINE INPUTS FROM THE OPERATOR CONSOLE
347      ; VALIDATES THE INPUTS, AND CONVERTS THE INPUTS TO BINARY
348
349      ; THE FOLLOWING INPUTS, UNIQUE TO EACH DISK BEING INITIALIZED, ARE RECEIVED:
350
351      ; DOCUMENT ID RANGE START
352      ; DOCUMENT ID RANGE END
353      ; TOTAL NUMBER OF DOCUMENTS SLATED FOR THIS DISK
354
355      ; IF ANY ERRORS ARE DETECTED ON INPUT, A MESSAGE IS SENT TO THE OPERATOR
356      ; CONSOLE AND THE TASK IS ABORTED.
357
358 004326 GCLIN:
359 004326
360 004336 005037 000030*   SAVE R2,R3,R4,R5 ; SAVE REGISTERS
361 004342 010100         CLR ABORT          ; CLEAR ABORT FLAG
362 004344 010102         MOV R1,R0          ; SET UP INDEX FOR LENGTH TABLE
363 004346         MOV R1,R2          ; SAVE INDEX VALUE
364 004346 062701 002532*   GCL05:
365 004352 011137 002230*   ADD #MADDR,R1      ; GET ADDR OF MSG ADDR TABLE
366 004356 062700 002540*   MOV (R1),TTQIOW+Q,IOPL ; GET ADDR OF MSG
367         ADD #LTBL,R0 ; GET ADDR OF MSG LENGTH TABLE
368 004362 011037 002232*   MOV (R0),TTQIOW+Q,IOPL+2 ; GET LENGTH OF MSG
369 004366         DIR# #TTQIOW ; PROMPT OPERATOR FOR INPUT
370 004374 103020         BCC GCL10 ; QIO SUCCESSFUL
371 004376 113700 000004*   MOV# WSJ,R0 ;
372 004402 010037 003702*   MOV R0,PAR1 ;
373 004406         MOUT#S #EMSG1,#PAR1 ; ISSUE ERROR MESSAGE
374 004426 005237 000030*   INC ABORT ; SET ABORT FLAG
375 004432 000137 005240*   JMP RTN ; RETURN
376 004436
377 004436
378 004516 103022         GCL10:
379 004520 105737 000004*   QIOW#S #IO,RPR,#TTLUN,#EFNS, #WSJ, <#CLBUF,#02, #PRMPT,#4,#44>
380 004524 003017         BCC 1$ ; SUCCESSFUL QIO
381 004526 113737 000004* 003702*   TSTB WSJ ; SUCCESSFUL RECEIPT?
382 004534         BGT 1$ ; YES
383 004554 005237 000030*   MOV# @#WSJ,PAR1 ; WRITE STATUS ERROR
384 004560 000137 005240*   MOUT#S #EMSG1,#PAR1 ;
385 004564         INC ABORT ; SET ABORT FLAG
386 004564         JMP RTN ; RETURN
387 004570 013704 000006*   1$:
388 004574 012701 000052*   SAVE R0,R1 ; SAVE REGISTERS
389 004600 012700 000010*   MOV WSJ+WORD1,R4 ; LENGTH OF INPUT
390 004604         MOV #CLBUF,R1 ; ADDR OF COMMAND LINE BUFFER
391 004604         MOV #ASCSTR,R0 ; ADDR OF ASCII BUFFER
392 004606 112120         10$:
393 004610 005304         MOV# (R1)+,(R0)+ ; GET NEXT ASCII BYTE
394 004612         DEC R4 ; DECREMENT CHARACTER COUNT
395 004612         BGT 10$ ; MORE TO MOVE
396 004616 013704 000006*   20$:
397 004622 005704         RESTOR R0,R1 ; RESTORE REGISTERS
398 004624 003013         MOV WSJ+WORD1,R4 ; LENGTH OF INPUT CHARACTERS
399 004626         TST R4 ; ANYTHING INPUT?
400         BGT 30$ ; YES
         MOUT#S #EMSG2 ; ISSUE ERROR MESSAGE

```

```

401 004644 005237 000030'      INC  ABORT      ; SET ABORT FLAG
402 004650 000137 005240'      JMP  RTN      ; RETURN
403 004654                                30$:
404 004654 020227 000002      CMP  R2,#2    ; WHICH INPUT?
405 004660 003401      BLE  40$    ; 1ST OR 2ND INPUT
406 004662 000431      BR   50$    ; 3RD INPUT
407 004664                                40$:
408 004664 020427 000015      CMP  R4,#NCHN ; CORRECT # OF CHARACTERS INPUT?
409 004670 003426      BLE  50$    ; YES
410 004672 020227 000002      CMP  R2,#2    ; 1ST OR 2ND INPUT?
411 004676 001410      BEQ  45$    ; 2ND INPUT
412 004700      MOUT$S #EMSG3 ; ISSUE ERROR MSG FOR 1ST INPUT
413 004716 000407      BR   46$    ; ABORT TASK
414 004720                                45$:
415 004720      MOUT$S #EMSG4 ; ISSUE ERROR MSG FOR 2ND INPUT
416 004736                                46$:
417 004736 005237 000030'      INC  ABORT      ; SET ABORT FLAG
418 004742 000137 005240'      JMP  RTN      ; RETURN
419 004746                                50$:
420 004746 012703 000022'      MOV  #BIN3WD,R3 ; OUTPUT FIELD
421 004752 012705 000010'      MOV  #ASCSTR,R5 ; ASCII INPUT STRING
422 004756 020227 000004      CMP  R2,#4    ; THIRD INPUT?
423 004762 002405      BLT  55$    ; NO, 1ST OR 2ND
424 004764 004737 007520'      JSR  PC,.DD3CT ; YES, DO DECIMAL CONVERT
425 004770 103020      BCC  60$    ; SUCCESSFUL
426 004772 000137 005004'      JMP  56$    ; ERROR
427 004776                                55$:
428 004776 004737 007520'      JSR  PC,.DD3CT ; CONVERT ASCII DECIMAL TO 3 WORD BINARY
429 005002 103013      BCC  56$    ; SUCCESSFUL CONVERSION
430 005004                                56$:
431 005004      MOUT$S #EMSG5 ; ERROR CONVERTING OCTAL #
432 005022 005237 000030'      INC  ABORT      ; SET ABORT FLAG
433 005026 000137 005240'      JMP  RTN      ; RETURN
434 005032                                60$:
435 005032 020227 000002      CMP  R2,#2    ; WHICH INPUT?
436 005036 002411      BLT  62$    ; 1ST INPUT
437 005040 001430      BEQ  64$    ; 2ND INPUT
438 005042 013737 000024' 000000C'      MOV  BIN3WD+2,INITBF+SR,ND5 ; 3RD INPUT - # OF DOCUMENTS
439 005050 013737 000026' 000000C'      MOV  BIN3WD+4,INITBF+SR,ND5+2 ;
440 005056 000137 005160'      JMP  62$    ;
441 005062                                62$:
442 005062 012704 000214'      MOV  #INITBF,R4 ;
443 005066 012705 001000      MOV  #SCTR,R5 ;
444 005072 004737 004254'      JSR  PC,CLRBUF ; CLEAR INITIALIZATION BUFFER
445 005076 013737 000022' 000000C'      MOV  BIN3WD,INITBF+SR,GR5 ; 1ST INPUT - DOC ID RANGE START
446 005104 013737 000024' 000000C'      MOV  BIN3WD+2,INITBF+SR,GR5+2 ;
447 005112 013737 000026' 000000C'      MOV  BIN3WD+4,INITBF+SR,GR5+4 ;
448 005120 000411      BR   70$    ;
449 005122                                64$:
450 005122 013737 000022' 000000C'      MOV  BIN3WD,INITBF+SR,GRE ; 2ND INPUT - DOC ID RANGE END
451 005130 013737 000024' 000000C'      MOV  BIN3WD+2,INITBF+SR,GRE+2 ;
452 005136 013737 000026' 000000C'      MOV  BIN3WD+4,INITBF+SR,GRE+4 ;
453 005144                                70$:
454 005144 062702 000002      ADD  #2,R2    ; INCREMENT INDEX FOR NEXT INPUT
455 005150 010201      MOV  R2,R1    ; RE-INITIALIZE R1 --
456 005152 010100      MOV  R1,R0    ; -- AND R0
457 005154 000137 004346'      JMP  GCL05    ; GET NEXT INPUT

```

INITDB -- INITIALIZE DATA BASE D MACRO M1110 27-MAR-80 14:34 PAGE 20-2
GCLIN -- GET COMMAND LINE INPUTS

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

458 005160			80\$:		
459 005160	023737	000000C-000000C	CMP	INITBF+SR.GRS,INITBF+SR.GRE	; RANGE START < RANGE END?
460 005166	001023		BNE	90\$; MAYBE
461 005170	023737	000000C-000000C	CMP	INITBF+SR.GRS+2,INITBF+SR.GRE+2	; HIGH WORD ZERO
462 005176	001017		BNE	90\$; MAYBE <
463 005200	023737	000000C-000000C	CMP	INITBF+SR.GRS+4,INITBF+SR.GRE+4	; 2ND WORD ZERO
464 005206	001013		BNE	90\$; MAYBE LESS THAN
465 005210			85\$:		
466 005210			MOUT\$S	#EMSG6	; RANGE START = OR > RANGE END
467 005226	005237	000030'	INC	ABORT	; SET ABORT FLAG
468 005232	000137	005240'	JMP	RTN	; RETURN
469 005236			90\$:		
470 005236	101364		BHI	85\$; RANGE START > RANGE END
471 005240			RTN:		
472 005240			RESTOR	R2,R3,R4,R5	; RESTORE REGISTERS
473 005250	000207		RTS	PC	; RETURN

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

```

475 ;
476 ;
477 ;
478 ;
479 ;
480 ;
481 ;
482 ;
483 ;
484 ;
485 ;
486 005252. ; SBTTL- IINDEX- INITIALIZE INDEX RECORDS.
487 005252. 012704 001214' ;
488 005256 012705 001000 ; IINDEX INITIALIZES EACH INDEX RECORD (SECTOR) ALLOTTED FOR THE DISK BEING
489 005262. 004737 004254' ; INITIALIZED. THE SECTOR IS ZEROED, THEN THE RECORD TYPE, THE SECTOR ADDRESS
490 005266 013764 000000C 000000 ; OF THE INDEX RECORD, AND THE MAXIMUM ENTRY COUNT IS STORED IN EACH INDEX
491 005274 013764 000000C 000002 ; RECORD AND WRITTEN TO THE DISK.
492 005302. 013737 000000C 000032' ;
493 005310 013737 000000C 000034' ; IF ANY ERROR OCCURS WRITING TO THE DISK, THE DISK IS CONSIDERED INOPERABLE
494 005316 112764 000100 000001 ; AND THE TASK IS ABORTED.
495 005324 012764 000044 000006 ;
496 005332. 013700 000000C ;
497 ;
498 ;
499 ;
500 005336 012737 000000C 002246' IINDEX:
501 005344 112737 000062 002253' MOV. #SECTOR,R4 ; CLEAR THE INDEX SECTOR.
502 005352. 012737 000004' 002254' MOV. #LSECT,R5 ; LENGTH OF SECTOR BUFFER.
503 005360 012737 001214' 002260' JSR. PC,CLRBUF. ;
504 005366 005037 002266' MOV. INITBF+SR.1IN,IN.SAN(R4) ; ADDRESS OF FIRST INDEX SECTOR.
505 ;
506 ;
507 ;
508 005372. MOV. INITBF+SR.1IN+WORD1,IN.SAL(R4) ; LOW ORDER ADDR OF 1ST X-RCO.
509 005372. 013737 000032' 002266' MOV. INITBF+SR.1IN,TEMP. ;
510 005400 013737 000034' 002270' MOV. INITBF+SR.1IN+WORD1,TEMP+WORD1 ;
511 005406 DIR# #R#IND,IN,TYP(R4) ;
512 005414 103020 MOV. #MECIND,IN,MEC(R4) ; SET MAXIMUM ENTRY COUNT.
513 005416 113700 000004' MOV. INITBF+SR.NIN,R0 ; # OF INDEX RECORDS TO INITIALIZE.
514 005422. 010037 003702' ;
515 005426 ;
516 005446 005237 000030' MOV. #IO.WLB,DPQIOW+Q.IOFN. ; SET WRITE FUNCTION CODE.
517 005452. 000137 005512' MOV. #WIPRI,DPQIOW+Q.IOPR. ; SET WRITE PRIORITY.
518 005456 ; MOV. #WSW,DPQIOW+Q.IOSB. ; SET WRITE STATUS WORD.
519 005456 062737 000001 000034' MOV. #SECTOR,DPQIOW+Q.IOPL. ; ADDRESS OF SECTOR TO WRITE.
520 005464 005537 000032' CLR. DPQIOW+Q.IOPL+6 ; CLEAR HIGH ORDER ADDRESS OF SECTOR.
521 005470 062764 000001 000002. ;
522 005476 005564 000000 ;
523 005502. 005300 ;
524 005504 001402. ;
525 005506 000137 005372' ;
526 005512. ;
527 005512. 000207 ;

```

```

529.          .SBTTL- IIPR-   INITIALIZE INDEX POINTER RECORDS-
530.          ;
531.          ; IIPR- INITIALIZES EACH INDEX POINTER RECORD ALLOTTED FOR THE DISK BEING
532.          ; INITIALIZED, EACH IPR IS ZEROED, THEN THE RECORD TYPE, THE SECTOR ADDRESS
533.          ; OF THE IPR, THE DELTA OF INDEX RECORDS BEING REPRESENTED, AND THE ENTRY
534.          ; LENGTH OF EACH IPR ENTRY ARE PLACED IN THE IPR AND WRITTEN TO DISK.
535.          ;
536.          ; IF ANY ERROR OCCURS WRITING TO THE DISK, THE DISK IS CONSIDERED INOPERABLE
537.          ; AND THE TASK IS ABORTED.
538.          ;
539.          ;
540. 005514          IIPR:
541. 005514 012704 001214'  MOV. #SECTOR,R4      ; ADDR OF BUFFER TO WRITE TO THE DISK
542. 005520 012705 001000  MOV. #LSCTR,R5      ; LENGTH OF THE BUFFER
543. 005524 004737 004254'  JSR. PC,CLRBUF      ; CLEAR THE BUFFER
544. 005530 013764 000000C-000000  MOV. INITBF+SR,IIP,IP,SAN(R4) ; FIRST IPR SECTOR ADDRESS
545. 005536 013764 000000C-000002  MOV. INITBF+SR,IIP+WORD1,IP,SAL(R4) ;
546. 005544 013737 000000C-000032'  MOV. INITBF+SR,IIP,TEMP ;
547. 005552 013737 000000C-000034'  MOV. INITBF+SR,IIP+WORD1,TEMP+WORD1 ;
548. 005560 112764 000200 000001  MOV. #R$IIPR,IP,TYP(R4) ; SET IPR RECORD TYPE
549. 005566 012764 000012 000004  MOV. #DELTA,IP,DLT(R4) ; SET DELTA OF INDEX RECORDS
550. 005574 012764 000006 000020  MOV. #N,BIPE,IP,BPE(R4) ; PUT BYTES PER ENTRY IN IPR
551. 005602 013700 000000C-  MOV. INITBF+SR,NIP,R0 ; # OF IPRS TO INITIALIZE
552.          ;
553.          ; SET UP THE QIO TO WRITE TO THE DISK
554.          ;
555. 005606 012737 000006C-002246'  MOV. #IO,MLB,DPOIOW+Q,IOFN ; WRITE FUNCTION CODE
556. 005614 112737 000062 002253'  MOV. #IIPR1,DPOIOW+Q,IOPR ; WRITE PRIORITY
557. 005622 012737 000004' 002254'  MOV. #WSJ,DPOIOW+Q,IOSB ; WRITE STATUS WORD
558. 005630 012737 001214' 002260'  MOV. #SECTOR,DPOIOW+Q,IOPL ; ADDR OF BUFFER TO WRITE
559. 005636 005037 002266'  CLR. DPOIOW+Q,IOPL+6 ; CLEAR HIGH ORDER DISK ADDR
560.          ;
561.          ; WRITE EACH INITIALIZED IPR TO THE DISK
562.          ;
563. 005642          IIPR10:
564. 005642 013737 000032' 002266'  MOV. TEMP,DPOIOW+Q,IOPL+6 ; LOW ORDER DISK ADDR TO QIO
565. 005650 013737 000034' 002270'  MOV. TEMP+WORD1,DPOIOW+Q,IOPL+10 ; HIGH ORDER ADDR TO QIO
566. 005656          DIR# #DPOIOW ; ISSUE WRITE
567. 005664 103020          BCC. IIPR20 ; SUCCESSFUL
568. 005666 113700          MOV. WSJ,R0 ;
569. 005672 010037 003702'  MOV. R0,PAR1 ;
570. 005676          MOUT#S: #EMSG9,#PAR1 ; ISSUE ERROR MSG
571. 005716 005237 000030'  INC. ABORT ; SET UP FOR ABORT
572. 005722 000137 005762'  JMP. IIPR30 ; RETURN
573. 005726          IIPR20:
574. 005726 062737 000001 000034'  ADD. #1,TEMP+WORD1 ; INCREMENT ADDR FOR WRITE
575. 005734 005537 000032'  ADD. TEMP ;
576. 005740 062764 000001 000002  ADD. #1,IP,SAL(R4) ; INCREMENT ADDR FOR IPR
577. 005746 005564 000000  ADD. IP,SAN(R4) ;
578. 005752 005300  DEC. R0 ; MORE IPRS TO INITIALIZE?
579. 005754 001402  BEQ. IIPR30 ; NO
580. 005756 000137 005642'  JMP. IIPR10 ; YES
581. 005762          IIPR30:
582. 005762 000207  RTS. PC ; RETURN
  
```

```
584 ;
585 ;
586 ;
587 ;
588 ;
589 ;
590 ;
591 ;
592 005764 ;
593 005764 012704 000214* ;
594 005770 013764 000174* 000004 ;
595 005776 013764 000176* 000006 ;
596 006004 013764 000200* 000010 ;
597 ;
598 ;
599 ;
600 ;
601 ;
602 ;
603 006012 ;
604 006026 004737 004274* ;
605 ;
606 ;
607 ;
608 ;
609 ;
610 ;
611 ;
612 006032 005700 ;
613 006034 001401 ;
614 ;
615 006036 005202 ;
616 006040 ;
617 ;
618 006040 010264 000030 ;
619 ;
620 ;
621 ;
622 ;
623 006044 005001 ;
624 006046 ;
625 006056 004737 004274* ;
626 006062 005700 ;
627 006064 001401 ;
628 006066 005202 ;
629 006070 ;
630 006070 005001 ;
631 006072 ;
632 006102 004737 004274* ;
633 006106 005700 ;
634 006110 001401 ;
635 006112 005202 ;
636 006114 ;
637 006114 010264 000022 ;
638 ;
639 ;
640 ;
```

.SBTTL INIBLK INITIALIZE INITIALIZATION BLOCK.

INIBLK INITIALIZES THE BLOCK PRECEDING THE INDEX SECTIONS AND THE SEARCHABLE DATA BASE. THIS BLOCK WILL CONTAIN THE LATEST INFORMATION ABOUT THE SEARCHABLE DATA BASE.

INIBLK:

MOV #INITBF,R4 ; ADDR OF INITIALIZATION BUFFER
MOV DTBUF,SR,YR(R4) ; MOVE DATE (YR,MON,DAY) TO THE--
MOV DTBUF+2,SR,MON(R4) ; -- INITIALIZATION BUFFER
MOV DTBUF+4,SR,DAY(R4) ;

DIVIDE NUMBER OF DOCUMENTS BY NUMBER OF ENTRIES PER INDEX RECORD TO DETERMINE NUMBER OF INDEX RECORDS NECESSARY FOR THIS DATA BASE DISK.

*** FOR THE 11/04. USE THE SOFTWARE DIVIDE.

SAVE #MECIND,SR,NDS(R4),SR,NDS+2(R4) ;
JSR PC,DIVIDE ; RESULT = # OF INDEX RECORDS.

*** FOR THE 11/45. USE HARDWARE DIVIDE.

MOV SR,NDS(R4),R0 ; * HIGH ORDER OF DIVIDEND
MOV SR,NDS+2(R4),R1 ; * LOW ORDER OF DIVIDEND
DIV #MECIND,R0 ; * RESULT = # OF INDEX RECORDS NEEDED
TST R1 ; * REMAINDER?
R0 ; REMAINDER? (11/04)
BEQ 10\$; NO
INC R0 ; * YES, NEED ANOTHER INDEX RECORD
INC R2 ; * YES, NEED ANOTHER INDEX RECORD

10\$:
MOV R0,SR,NIN(R4) ; NUMBER OF INDEX RECORDS ALLOTTED
MOV R2,SR,NIN(R4) ; NUMBER OF INDEX RECORDS ALLOTTED

DIVIDE TOTAL NUMBER OF INDEX RECORDS BY THE DELTA TO DETERMINE TOTAL NUMBER OF ENTRY POINTERS TO THE INDEX RECORDS.

CLR R1 ; CLEAR HIGH ORDER DIVIDEND
SAVE #DELTA,R1,R2 ; PUSH DIVISOR AND DIVIDENDS ON STACK
JSR PC,DIVIDE ; RESULT = # OF IPR ENTRY POINTERS
TST R0 ; REMAINDER?
BEQ 20\$; NO
INC R2 ; * YES, NEED ANOTHER IPR ENTRY POINTER

20\$:
CLR R1 ;
SAVE #MECIPR,R1,R2 ; PUSH DIVISOR AND DIVIDENDS ON STACK
JSR PC,DIVIDE ; RESULT = # OF IPR SECTORS NEEDED
TST R0 ; REMAINDER?
BEQ 30\$; NO
INC R2 ; * YES, NEED ANOTHER IPR SECTOR

30\$:
MOV R2,SR,NIP(R4) ; TOTAL # OF IPR SECTORS
MOV SR,NIN(R4),R1 ; * # OF INDEX RECORDS
R0 ; * CLEAR HIGH ORDER
DIV #DELTA,R0 ; * # OF IPR ENTRY POINTERS NEEDED

```

641          :TST R1          ; * REMAINDER?
642          :BEQ 20$         ; * NO
643          :INC R0          ; * YES, NEED ANOTHER ENTRY POINTER
644          :                ; *
645          : 20$:
646          :MOV R0,R1        ; * MOVE TO LOW ORDER WORD OF DIVIDEND
647          :CLR R0          ; * CLEAR HIGH ORDER WORD OF DIVIDEND
648          :DIV #MECIPR,R0  ; * = NUMBER OF IPR SECTORS
649          :TST R1          ; * REMAINDER?
650          :BEQ 30$         ; * NO
651          :INC R0          ; * YES, NEED ANOTHER IPR SECTOR
652          :                ; *
653          : 30$:
654          :MOV R0,SR,NIP(R4) ; * STORE # OF IPR SECTORS
655          :MOV IBLK,R0     ; * HIGH ORDER SECTOR ADDR OF INIT BLOCK
656          :MOV IBLK+2,R1   ; * LOW ORDER SECTOR ADDR OF INIT BLOCK
657          :ADD #1,R1       ; * = ADDR OF FIRST IPR SECTOR
658          :ADC R0          ;
659          :MOV R0,SR,1IP(R4) ;
660          :MOV R1,SR,1IP+WORD1(R4) ;
661          :ADD SR,NIP(R4),R1 ; * = ADDR OF FIRST INDEX SECTOR
662          :ADC R0          ;
663          :MOV R0,SR,1IN(R4) ;
664          :MOV R1,SR,1IN+WORD1(R4) ;
665          :ADD SR,NIN(R4),R1 ; * = ADDR OF 1ST SEARCHABLE DATA SECTOR
666          :ADC R0          ;
667          :MOV R0,SR,SDB(R4) ;
668          :MOV R1,SR,SDB+2(R4) ;
669          :MOV SR,SDB(R4),SR,ECH(R4) ; SET END-OF-COLLECTION SEQUENCE
670          :MOV SR,SDB+2(R4),SR,ECL(R4) ; TO 1ST SEARCHABLE DATA SECTOR
671          :MOV NSECTR,R2   ; R2,R3 = MAX. NO. OF SECTORS ON DISK
672          :MOV NSECTR+2,R3 ;
673          :SUB R1,R3      ; R2,R3 = NO. OF SECTORS LEFT FOR DB
674          :SBC R2        ;
675          :SUB R0,R2     ;
676          :MOV #LSCTR,R0 ; R0 = LENGTH OF SECTOR
677          :JSR PC,$DMJL  ; * = NUMBER OF AVAILABLE BYTES
678          :MOV R0,SR,WSL(R4) ; BYTES OF W.S. AT END OF DB
679          :MOV R1,SR,WSL+2(R4) ;
680          :MOV R0,SR,TWS(R4) ; TOTAL WHITE SPACE COUNT - INITIALLY
681          :MOV R1,SR,TWS+2(R4) ; -- EQUAL TO WHITE SPACE COUNT
682          :MOV SR,1IP(R4),SR,LIP(R4) ; LATEST IPR SECTOR = 1ST IPR SECTOR
683          :MOV SR,1IP+WORD1(R4),SR,LIP+WORD1(R4) ; INITIALIZE LATEST IPR SECTOR
684          :MOV SR,1IN(R4),SR,LIN(R4) ; LATEST INDEX RCD. = 1ST INDEX RCD
685          :MOV SR,1IN+WORD1(R4),SR,LIN+WORD1(R4) ;
686          :
687          : WRITE THE INITIALIZATION BLOCK TO DISK
688          :
689          :MOV #IO,WLB,DPQIOW+0,IOPN ; WRITE FUNCTION CODE TO DPB
690          :MOV #NIPRI,DPQIOW+0,IOPR ; WRITE PRIORITY
691          :MOV #UJW,DPQIOW+0,IOSB ; WRITE STATUS WORD
692          :MOV #INITBF,DPQIOW+0,IOPL ; ADDR OF INITIALIZATION BLOCK
693          :MOV IBLK,DPQIOW+0,IOP+6 ; HIGH ORDER ADDR OF BLOCK
694          :MOV IBLK+2,DPQIOW+0,IOP+10 ; LOW ORDER ADDR OF BLOCK
695          :DIR $ #DPQIOW ;
696          :BCC 40$        ; ISSUE WRITE
697          :MOV #UJW,PAR1 ; SUCCESSFUL
698          :MOUT $ #MSG9,#PAR1 ; ISSUE ERROR MESSAGE

```

INITDB- INITIALIZE DATA BASE D MACRO M1110 27-MAR-80 14:34 E-23-2
INIBLK- INITIALIZE INITIALIZATION

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

698	006416	005237	000030*	INC-	ABORT-	;	SET UP FOR ABORT-
699	006422			40\$:			
700	006422	123727	000004*	000000G-	CHPB-	WSW,#IS:SUC	; SUCCESSFUL WRITE?
701	006430	001413			BEQ-	INEND-	; YES-
702	006432	113737	000004*	003702*	MOVB-	@*WSW,PAR1	;
703	006440				MOVB\$S-	#EMSG9,#PAR1	;
704	006460			INEND:			
705	006460	000207		RTS-	PC-		

```

707 .SBTTL - VFYDSK- VERIFY DATA BASE DISK PACK
708 ;
709 ;
710 ; THIS SUBROUTINE VERIFIES THAT A DISK IS MOUNTED AND HAS NOT BEEN INITIALIZED
711 ; BEFORE. IF THE DISK HAS BEEN INITIALIZED, A MESSAGE IS PLACED ON THE
712 ; CONSOLE WITH THE DATE OF THE PREVIOUS INITIALIZATION. THE OPERATOR IS THEN
713 ; ASKED IF HE WISHES TO CONTINUE THE INITIALIZATION WITH THIS DISK. IF THE
714 ; OPERATOR DOES NOT WISH TO CONTINUE ON THIS DISK, HE IS ASKED TO MOUNT
715 ; ANOTHER DISK AND NOTIFY THE PROGRAM THAT THE NEW DISK IS READY. IF NO REPLY
716 ; IS GIVEN, THE TASK IS ABORTED. OTHERWISE, THE PROCESS IS BEGUN AGAIN TO
717 ; VERIFY THE NEW DISK.
718 ;
719 ;
720 VFYDSK:
721     SAVE R0,R1 ; SAVE REGISTERS
722     CLR ABORT ; CLEAR ABORT FLAG
723     GTIM#S #DTBUF ; GET CURRENT DATE AND TIME
724     VDSK05:
725     MOV #IO,RLB!IO,X,DPQIOW+0,IOFN ; READ FUNCTION CODE WITH NO RETRIES
726     MOV #INITBF,DPQIOW+0,IOPL ; ADDRESS OF INITIALIZATION BUFFER
727     MOV IBLK,DPQIOW+0,IOPL+6 ; BLOCK HIGH ADDRESS
728     MOV IBLK+2,DPQIOW+0,IOPL+10 ; BLOCK LOW ADDRESS
729     DIR# #DPQIOW ; ISSUE READ
730     BCC VDSK06 ; SUCCESSFUL
731     MOV# R0,R0 ;
732     MOV R0,PAR1 ;
733     MOUT#S #EMSG9,#PAR1 ;
734     JMP VDSK40 ;
735     VDSK06:
736     CMPB #IS,SUC,RSW ; SUCCESSFUL READ?
737     BEQ VDSK20 ; YES
738     CMPB #IE,DNR,RSW ; DEVICE NOT READY?
739     BEQ VDSK10 ; YES, DEVICE IS NOT READY
740     MOV# R0,R0 ; STATUS WORD TO ERROR MSG
741     MOV R0,PAR1 ;
742     MOUT#S #EMSG7,#PAR1 ; ISSUE ERROR MSG
743     JMP VDSK40 ; SET UP FOR ABORT
744     VDSK10:
745     MOV# R0,R0 ; STATUS WORD TO ERROR MSG
746     MOV R0,PAR1 ;
747     MOUT#S #EMSG8,#PAR1 ; ISSUE ERROR MSG
748     JMP VDSK40 ; SET UP FOR ABORT
749     VDSK20:
750     TST #INITBF+SR,YR ; DISK ALREADY INITIALIZED?
751     BNE 20$ ; YES
752     JMP VDSK50 ; NO
753     20$:
754     MOV #S4,R0 ; ADDRESS FOR DATE CONVERSION
755     MOV #INITBF+SR,YR,R1 ; INTERNAL DATE TO BE CONVERTED
756     JSR PC,$DAT ; CONVERT DATE TO ASCII DISPLAY
757     MOV #STR4,TTQIOW+0,IOPL ; ADDR OF MESSAGE
758     MOV LEN4,TTQIOW+0,IOPL+2 ; LENGTH OF MSG
759     DIR# #TTQIOW ; ISSUE MSG WITH PREV. INIT. DATE
760     BCC VDSK25 ; SUCCESSFUL
761     MOV# R0,R0 ;
762     MOV R0,PAR1 ;
763     MOUT#S #EMSG1,#PAR1 ; ISSUE ERROR MSG
    
```

```

764 007012 000137 007506*      JMP      VDSK40      ; SET UP FOR ABORT
765 007016                                VDSK25:
766 007016      Q10W$#  #10,RPR,#TTLUN,#EFNS, #WSW, <#CLBUF,#82, #PRMPT,#4,#44>
767 007076 103017      BCC     25$        ; SUCCESSFUL DIRECTIVE
768 007100 113737 000004* 003702*  MOV#    @#WSW,PAR1  ;
769 007106      MOUT$#  #EMSG1,#PAR1 ;
770 007126 005237 000030*      INC     ABORT      ;
771 007132 000137 007512*      JMP     VDSK50      ; RETURN
772 007136                                25$:
773 007136 105737 000004*      TSTB   WSW         ; GOOD Q10?
774 007142 003017      BGT     26$        ; YES
775 007144 113737 000004* 003702*  MOV#    @#WSW,PAR1  ; NO
776 007152      MOUT$#  #EMSG1,#PAR1 ;
777 007172 005237 000030*      INC     ABORT      ;
778 007176 000137 007512*      JMP     VDSK50      ;
779 007202                                26$:
780 007202 123727 000052* 000015  CMP#    CLBUF,#CR   ; CARRIAGE RETURN (IMPLIES YES)?
781 007210 001540      BEQ     VDSK50      ; YES, CONTINUE WITH INITIALIZATION
782 007212 123727 000052* 000131  CMP#    CLBUF,#Y    ; CONTINUE WITH INITIALIZATION?
783 007220 001534      BEQ     VDSK50      ; YES, CONTINUE
784 007222 012737 002670* 002230*  MOV#    #STR5,TTQ10W+Q,IOPL  ; ADDR OF MSG TO MOUNT NEW DISK
785 007230 013737 002774* 002232*  MOV#    LENS,TTQ10W+Q,IOPL+2  ; LENGTH OF MESSAGE
786 007236      DIR#    #TTQ10W    ; ISSUE MSG
787 007244 103016      BCC     VDSK35      ; SUCCESSFUL
788 007246 113700 000004*      MOV#    WSW,R0     ;
789 007252 010037 003702*      MOV#    R0,PAR1   ;
790 007256      MOUT$#  #EMSG1,#PAR1 ; ISSUE ERROR MSG
791 007276 000137 007506*      JMP     VDSK40      ; SET UP FOR ABORT
792 007302                                VDSK35:
793 007302 103015      Q10W$#  #10,RPR,#TTLUN,#EFNS, #WSW, <#CLBUF,#82, #PRMPT,#4,#44>
794 007362 103015      BCC     33$        ; SUCCESSFUL Q10
795 007364 113737 000004* 003702*  MOV#    @#WSW,PAR1  ;
796 007372 000137 007506*      MOUT$#  #EMSG1,#PAR1 ;
797 007412 000137 007506*      JMP     VDSK40      ; ABORT
798 007416                                33$:
799 007416 105737 000004*      TSTB   WSW         ; GOOD RECEIPT?
800 007422 003015      BGT     34$        ; YES
801 007424 113737 000004* 003702*  MOV#    @#WSW,PAR1  ;
802 007432      MOUT$#  #EMSG1,#PAR1 ;
803 007452 000137 007506*      JMP     VDSK40      ;
804 007456                                34$:
805 007456 123727 000052* 000015  CMP#    CLBUF,#CR   ; CARRIAGE RETURN (IMPLIED YES)?
806 007464 001002      BNE    VDSK40      ; NO
807 007466 000137 006504*      JMP     VDSK05      ; YES, VERIFY NEW DISK
808 007472                                35$:
809 007472 123727 000052* 000131  CMP#    CLBUF,#Y    ; NEW DISK MOUNTED?
810 007500 001002      BNE    VDSK40      ; NO
811 007502 000137 006504*      JMP     VDSK05      ; YES, VERIFY NEW DISK
812 007506                                VDSK40:
813 007506 005237 000030*      INC     ABORT      ; SET UP FOR ABORT
814 007512                                VDSK50:
815 007512      RESTOR  R0,R1    ; RESTORE REGISTERS
816 007516 000207      RTS     PC         ; RETURN
  
```

```

818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855 007520
856 007520
857 007526 012701 000012
858 007532 000405
859 007534
860 007534
861 007542 012701 000010
862
863 007546
864 007546 010502
865 007550 060402
866 007552 124227 000056
867 007556 001003
868 007560 012701 000012
869 007564 005304
870
871 007566
872 007566 005046
873 007570 121527 000053
874 007574 001404
    
```

```

.SBTTL .OD3CT - CONVERT FROM ASCII TO THREE WORD BINARY
*** - .DD3CT - CONVERT A DECIMAL NUMBER TO THREE BINARY WORDS.
THIS SUBROUTINE CONVERTS AN ASCII DECIMAL NUMBER
STRING TO A TRIPLE LENGTH BINARY NUMBER. A
TRAILING DECIMAL POINT IS PERMITTED.
*** - .OD3CT - CONVERT AN OCTAL OR DECIMAL NUMBER TO THREE BINARY
WORDS. THIS SUBROUTINE CONVERTS AN ASCII OCTAL OR
DECIMAL NUMBER STRING TO A TRIPLE LENGTH BINARY
NUMBER. RADIX IS CONTROLLED BY THE PRESENCE OR
ABSENCE OF A TRAILING DECIMAL POINT.
BOTH ENTRIES ACCEPT A LEADING "+" OR "-" AND PRODUCE A TWO'S
COMPLEMENT NEGATIVE FOR THE LATTER.
INPUT:
R3 = ADDRESS OF THREE WORD FIELD TO HOLD CONVERTED NUMBER.
WORD1 = HIGH ORDER 16 BITS.
WORD2 = MIDDLE ORDER 16 BITS.
WORD3 = LOW ORDER 16 BITS.
R4 = CHARACTER COUNT OF STRING TO BE CONVERTED.
R5 = ADDRESS OF STRING.
OUTPUT:
CC = C.
CLEAR = SUCCESSFUL CONVERSION.
SET = ILLEGAL CHARACTERS IN STRING.
ALL REGISTERS ARE PRESERVED
.DD3CT:
SAVE R0,R1,R2 ; SAVE REGISTERS.
MOV R4,R2 ; SET UP FOR DECIMAL RADIX ONLY.
BR CNVT
.OD3CT:
SAVE R0,R1,R2 ; SAVE REGISTERS.
MOV R4,R1 ; SET UP FOR OCTAL RADIX.
CNVT:
MOV R5,R2 ; COPY THE STRING POINTER.
ADD R4,R2 ; POINT TO THE END OF STRING.
CMPB -(R2),#' ; LOOK FOR TRAILING DECIMAL POINT.
BNE 20# ; NO.
MOV #10,R1 ; YES - SET DECIMAL.
DEC R4 ; REMOVE "." FROM STRING.
20#:
CLR -(SP) ; CLEAR NEGATIVE FLAG.
CMPB (R5),#' + ; LOOK FOR LEADING "+".
BEQ 30# ; YES.
    
```



```

875 007576 121527 000055      CMPB (R5),#1+ ; NO. LOOK FOR LEADING "-"
876 007602 001003      BNE 40$ ; NO
877 007604 005216      INC (SP) ; YES, SET NEGATIVE FLAG
878 ;
879 007606      30$:
880 007606 005205      INC R5 ; EITHER CASE - DISPOSE OF CHARACTER
881 007610 005304      DEC R4 ;
882 ;
883 007612      40$:
884 007612 005023      CLR (R3)+ ; ZERO HIGH ORDER 16 BITS
885 007614 005023      CLR (R3)+ ; ZERO MIDDLE 16 BITS
886 007616 005013      CLR (R3) ; ZERO LOW ORDER 16 BITS
887 ;
888 007620      50$:
889 007620 112502      MOVB (R5)+,R2 ; GET NEXT CHARACTER
890 007622 162702      SUB #0,R2 ; CONVERT DIGIT TO VALUE
891 007626 103465      BLO 90$ ; NOT A LEGAL DIGIT CHARACTER
892 007630 020201      CMP R2,R1 ; IN RANGE?
893 007632 103063      BHIS 90$ ; DEFINITELY NOT LEGAL
894 ;
895 007634 006313      ASL (R3) ; MULTIPLY CURRENT BINARY NUMBER BY 2
896 007636 006143      ROL -(R3) ; USE 32 BIT ACCURACY
897 007640 006143      ROL -(R3) ; USE FULL 48 BIT ACCURACY
898 007642 012346      MOV (R3)+,(SP) ; SAVE HIGH ORDER BITS
899 007644 012346      MOV (R3)+,(SP) ; SAVE MIDDLE BITS
900 007646 011346      MOV (R3),(SP) ; SAVE LOW ORDER BITS
901 ;
902 007650 006313      ASL (R3) ; MULTIPLY AGAIN BY 2
903 007652 006163 177776      ROL -2(R3) ; USE 32 BIT ACCURACY
904 007656 006163 177774      ROL -4(R3) ; USE FULL 48 BIT ACCURACY
905 ;
906 007662 006313      ASL (R3) ; MAKE TOTAL MULTIPLICATION BY 8
907 007664 006163 177776      ROL -2(R3) ; TAKE CARE OF CARRY BIT
908 007670 006163 177774      ROL -4(R3) ; NOW: NEW NUMBER = (OLD NUMBER) * 8
909 ;
910 007674 020127 000012      CMP R1,#10 ; DECIMAL CONVERSION?
911 007700 001403      BEQ 60$ ; YES
912 007702 022626      CMP (SP)+,(SP)+ ; THROW AWAY OLD NUMBER --
913 007704 005726      TST (SP)+ ; -- FOR OCTAL CONVERSION
914 007706 000411      BR 70$ ;
915 ;
916 007710      60$:
917 007710 062613      ADD (SP)+,(R3) ; ADD BACK (OLD NUMBER)*2
918 007712 005543      ADC -(R3) ; ADD ANY OVERFLOW TO MIDDLE 16 BITS
919 007714 005563 177776      ADC -2(R3) ; ADD ANY OVERFLOW TO HIGH ORDER
920 ;
921 007720 062613      ADD (SP)+,(R3) ; ADD BACK MIDDLE 16 BITS
922 007722 005543      ADC -(R3) ; ADD ANY OVERFLOW TO HIGH ORDER
923 ;
924 007724 062613      ADD (SP)+,(R3) ; NOW: NEW NUMBER = (OLD NUMBER)*10
925 007726 062703 000004      ADD #4,R3 ; POINT TO LOW ORDER
926 ;
927 007732      70$:
928 007732 060213      ADD R2,(R3) ; ADD THIS DIGIT TO NEW TOTAL
929 007734 005563 177776      ADC -2(R3) ; ADD ANY OVERFLOW
930 007740 005563 177774      ADC -4(R3) ; ADD ANY OVERFLOW TO HIGH ORDER
931 007744 005304      DEC R4 ; COUNT CHARACTERS
    
```

```
932 007746 001324          BNE 50$          ; GO GET NEXT DIGIT
933
934 007750 005726          TST (SP)+        ; CHECK NEGATIVE FLAG
935 007752 001412          BEQ 80$          ; POSITIVE
936 007754 005463 177774   NEG -4(R3)       ; COMPLEMENT HIGH ORDER
937 007760 005443          NEG -(R3)        ; COMPLEMENT MIDDLE BITS
938 007762 005643          SBC -(R3)        ; PROPAGATE CARRY TO HIGH ORDER
939 007764 005463 000004   NEG 4(R3)        ; COMPLEMENT LOW ORDER
940 007770 005663 000002   SBC 2(R3)        ; PROPAGATE CARRY TO MIDDLE BITS
941 007774 005613          SBC (R3)         ; PROPAGATE CARRY TO HIGH ORDER
942 007776 000257          CCC
943
944 010000
945 010000 000402          80$: BR 100$     ; RETURN
946
947 010002          90$:
948 010002 005726          TST (SP)+        ; CLEAN THE STACK
949 010004 000261          SEC              ; SET THE CARRY
950
951 010006          100$:
952 010006          RESTOR R0,R1,R2 ; RESTORE REGISTERS
953 010014 000207          RTS PC           ; RETURN
954 003704          .END INITDB
```

ABORT = 000030R	005 EMSG3 003062R	005 IO.RPR = ***** GX	NCHIN = 000015	STR1E = 002357R	005
ASCSTR = 000010R	005 EMSG4 003066R	005 IO.WLB = ***** GX	NSECTR = 000042R	STR2 = 002360R	005
BIN3WD = 000022R	005 EMSG5 003072R	005 IO.UWB = ***** GX	N.BFAC = 000004	STR2E = 002441R	005
BITVAL = 000000	EMSG6 003076R	005 IP.BPE = 000020	005 N.BHGH = 000006	STR3 = 002442R	005
BIT8 = 000001	EMSG7 003102R	005 IP.CEC = 000006	006 N.BIPE = 000006	STR3E = 002532R	005
BIT1 = 000002	EMSG8 003106R	005 IP.DLT = 000004	006 N.BTCH = 000004	STR4 = 002546R	005
BIT10 = 002000	EMSG9 003112R	005 IP.FE = 000030	006 N.BUFW = 004000	STR4E = 002666R	005
BIT11 = 004000	EOC = 001000	IP.HID = 000012	006 N.BUFW = 002000	STR5 = 002670R	005
BIT12 = 010000	FD.FID 000000	003 IP.LIA = 000022	006 N.BXE = 000016	STR5E = 002774R	005
BIT13 = 020000	FD.FNB 000006	003 IP.PAD = 000026	006 N.FOS = 000764	STR6 = 002776R	005
BIT14 = 040000	FD.FYR 000004	003 IP.SAL = 000002	006 N.OURY = 000031	STR6E = 003047R	005
BIT15 = 100000	FD.LEN 000010	003 IP.SAN = 000000	006 N.SUNT = 000002	S4 = 002572R	005
BIT2 = 000004	FN.ACK 000016	004 IP.SRL = 000010	006 PAR1 = 003702R	005 TD\$CTR = 176370	
BIT3 = 000010	FN.FSA 000000	004 IP.TYP = 000001	006 PE.IDH = 000000	007 TD\$CTLW = 176360	
BIT4 = 000020	FN.FSB 000002	004 IO.X = ***** GX	006 PE.IDL = 000004	007 TD\$INL = 004000	
BIT5 = 000040	FN.FSC 000004	004 IS.SUC = ***** GX	006 PE.IDM = 000002	007 TD\$MEM = 000270	
BIT6 = 000100	FN.FSD 000020	004 I.BC = 000007	011 PRMPT = 000046R	005 TD\$OR = 176344	
BIT7 = 000200	FN.MHR 000010	004 I.BIDH = 000000	011 Q.IOAE = 000012	TD\$ORD = 176346	
BIT8 = 000400	FN.NMB 000022	004 I.BIDL = 000004	011 Q.IOEF = 000006	TD\$ORD = 000274	
BIT9 = 001000	FN.QLS 000006	004 I.BIDM = 000002	011 Q.IOFN = 000002	TD\$RST = 176366	
BYTE0 = 000000	FN.RDC 000014	004 I.PGI = 000001	011 Q.IOLU = 000004	TD\$SW = 176376	
BYTE1 = 000001	FN.UPD 000012	004 I.RUS = 000012	011 Q.IOPL = 000014	TD\$STAR = 176372	
BYTE2 = 000002	GCLIN = 004326R	005 I.SANH = 000006	011 Q.IOPR = 000007	TD\$STAR = 176362	
BYTE3 = 000003	GCL05 = 004346R	005 I.SANL = 000010	011 Q.IOSB = 000010	TD\$TDR = 176374	
BYTE4 = 000004	GCL10 = 004436R	005 I.UUS = 000014	011 RSJ = 000000R	005 TD\$TDW = 176364	
BYTE5 = 000005	G.TICP = 000016	LEN4 = 002666R	005 RTN = 005240R	005 TEMP = 000032R	005
BYTE6 = 000006	G.TICT = 000014	LEN5 = 002774R	005 R\$IND = 000100	TTLUN = 000005	
BYTE7 = 000007	G.TIDA = 000004	LEN6 = 003050R	005 R\$IPR = 000200	TTIOUJ = 002214R	005
BYTE8 = 000010	G.TIHR = 000006	LF = 000012	005 R\$PRI = 000062	T\$AD = 000020	
BYTE9 = 000011	G.TIMI = 000010	LN1 = 003122R	005 SECTOR = 001214R	005 T\$BA = 000002	
BYTVAL = 000012	G.TIMO = 000002	LN1E = 003201R	005 SR.ARE = 000114	002 T\$BD = 000010	
CF.COT = 000041	G.TISC = 000012	LN10 = 003660R	005 SR.ARS = 000106	002 T\$BSO = 100000	
CF.DGN = 000046	G.TIYR = 000000	LN10E = 003702R	005 SR.DAY = 000010	002 T\$BT = 000020	
CF.DHR = 000042	IBLK = 000036R	005 LN2 = 003202R	005 SR.DLT = 000014	002 T\$BTAR = 000030	
CF.DMC = 000047	IDB100 = 003774R	005 LN2E = 003234R	005 SR.ECB = 000047	002 T\$BT = 002000	
CF.HBR = 000045	IDB200 = 004016R	005 LN3 = 003234R	005 SR.ECH = 000046	002 T\$CD = 000100	
CF.HRL = 000044	IDB300 = 004246R	005 LN3E = 003276R	005 SR.ECL = 000050	002 T\$CLK = 002000	
CF.UPD = 000043	IE.DHR = ***** GX	LN4 = 003276R	005 SR.FIB = 000012	002 T\$DISK = 000200	
CLBUF = 000052R	005 INDEX = 005252R	005 LN4E = 003336R	005 SR.GRE = 000100	002 T\$DID = 000004	
CLRBUF = 004254R	005 IIPR = 005514R	005 LN5 = 003336R	005 SR.GRS = 000072	002 T\$EMEM = 010000	
CHVT = 007546R	005 IIPR10 = 005642R	005 LN5E = 003377R	005 SR.LEN = 000122	002 T\$FSAB = 000004	
CR = 000015	IIPR20 = 005726R	005 LN6 = 003400R	005 SR.LIN = 000066	002 T\$FSAC = 000014	
DBSLN = 000116	IIPR30 = 005762R	005 LN6E = 003460R	005 SR.LIP = 000052	002 T\$FSB = 000010	
DELTA = 000012	INDX10 = 005372R	005 LN7 = 003460R	005 SR.LON = 000066	002 T\$FSB2 = 000010	
DG.ERR = 001000	INDX20 = 005456R	005 LN7E = 003530R	005 SR.MDN = 000062	002 T\$IB = 000026	
DG.SDF = 002000	INDX30 = 005512R	005 LN8 = 003530R	005 SR.NDC = 000042	002 T\$IBAR = 000024	
DG.TDF = 004000	INEND = 006460R	005 LN8E = 003602R	005 SR.NIN = 000030	002 T\$IBE = 020000	
DIVIDE = 004274R	005 INIBLK = 005764R	005 LN9 = 003602R	005 SR.NIP = 000022	002 T\$IBF = 040000	
DPLUN = 000001	INITBF = 000214R	005 LN9E = 003657R	005 SR.SDB = 000032	002 T\$ICD = 000040	
DPQ10W = 002244R	005 INITDB = 003704R	005 L\$CTR = 001000	005 SR.SRC = 000002	002 T\$INDP = 004000	
DSW = 000000	IN.CEC = 000004	010 LTBL = 002540R	005 SR.SUN = 000000	002 T\$IOB = 000036	
DTBUF = 000174R	005 IN.FE = 000010	010 MADDR = 002532R	005 SR.TUS = 000056	002 T\$OB = 004000	
EFN1 = 000001	IN.MEC = 000006	010 MECIND = 000044	005 SR.WSL = 000052	002 T\$OBF = 010000	
EFN5 = 000005	IN.SAL = 000002	010 MEIPR = 000121	005 SR.YR = 000004	002 T\$OBRA = 000034	
EMSG1 = 003052R	005 IN.SAN = 000000	010 MSGOUT = ***** GX	005 SR.IIN = 000024	002 T\$OBUA = 000032	
EMSG10 = 003116R	005 IN.TYP = 000001	010 MS.DGN = 010000	005 SR.IIP = 000016	002 T\$OUTA = 100000	
EMSG2 = 003056R	005 IO.RLB = ***** GX	N = 000012	STR1 = 002274R	005 T\$RBD0 = 000200	

T\$RNB = 000040	T\$1CLK = 000400	VDSK50 007512R	005 WORD5 = 000012	WSPRI = 000062
T\$RSET = 040000	T\$OBEN = 000020	VFC = 000040	WORD6 = 000014	\$DAT = ***** GX
T\$SC = 000022	VDSK05 006504R	005 VFYDSK 006462R	005 WORD7 = 000016	\$DDIV = ***** GX
T\$SCLK = 020000	VDSK06 006600R	005 WORD0 = 000000	WORD8 = 000020	\$DMUL = ***** GX
T\$SEGI = 000000	VDSK10 006654R	005 WORD1 = 000002	WORD9 = 000022	\$\$\$ARG = 000002
T\$SEG2 = 000001	VDSK20 006710R	005 WORD2 = 000004	WRDVAL = 000024	\$\$\$OST = 000020
T\$SEG3 = 000002	VDSK25 007016R	005 WORD3 = 000006	WSW = 000004R	005 DD3CT: 007520R 005
T\$SO = 000001	VDSK35 007302R	005 WORD4 = 000010	WIPRI = 000062	.DD3CT: 007534R 005
T\$UBUS = 100000	VDSK40 007506R	005		

. ABS. 000000 000
 000000 001
 SRCOFF 000122 002
 FDSCOF 000010 003
 FNOFFS 000022 004
 INITDB 010016 005
 IPROFF 000032 006
 IPEOFF 000006 007
 INDOFF 000012 010
 INEOFF 000016 011
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4572 WORDS (18 PAGES)
 DYNAMIC MEMORY: 5972 WORDS (22 PAGES)
 ELAPSED TIME: 00:00:57
 INITDB, INITDB / -SP=C20, 1JP.C, INITDB

READIB· MACRO· M1110 27·MAR·80 14:29
TABLE OF CONTENTS·

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

11- 33 ISRECC· INITIALIZE· SREC· COMMON·

1
2-000000
3
4
5

.TITLE- READIB
.PSECT-
.ENABL- AMA
.LIST- MEB
.MCALL- QIDW\$S,EXIT\$S

Approved For Release 2005/07/14 : CIA-RDP85-00514R000100040001-2

7	000000	000000	000000	I0ST: .WORD.	0.0
8	000004	000000	000000	DSW: .WORD.	0.0
9	000010	000000		PAR: .WORD.	0
10		000001		EFN: =.	1
11		001000		SLB: =.	512.
12		000001		LUN: =.	1
13	000012.			BUF: .BLKW.	256.
14	001012.	000000	165140	IBLK: .WORD.	0.165140

```

16 001016
17 001016          READIB:
001016 005046          QIOW$$ #IO,RLB,#LUN,#EFN,,#IOST,,<#BUF,#SLB,#IBLK,IBLK+WORD1>
001020 013746 001014' CLR - (SP)
001024 013746 001012' MOV IBLK+WORD1,-(SP)
001030 005046          MOV IBLK,-(SP)
001032 012746 001000 CLR - (SP)
001036 012746 000012' MOV #SLB,-(SP)
001042 005046          MOV #BUF,-(SP)
001044 012746 000000' CLR - (SP)
001050 005046          MOV #IOST,-(SP)
001052 112716 000001 CLR - (SP)
001056 012746 000001 MOV #EFN,(SP)
001062 012746 000000G MOV #LUN,-(SP)
001066 012746          MOV #IO,RLB,-(SP)
001070 003 014        MOV (PC)+,-(SP)
001072 104377        .BYTE 3,12
18 001074 103015        EMT +D<377>
19 001076 113737 000004' 000010' BCC RD01 ; SUCCESSFUL DIRECTIVE
20 001104          MOV @#DSJ,PAR1 ;
001104 012746 001232' MOUT$$ #MSG1,#PAR1 ;
001110 012746 000010' MOV #MSG1,-(SP) ; PUSH ADDRESS OF ASCIZ STRING
001114 004737 000000G MOV #PAR1,-(SP) ; PUSH ADDRESS OF ARGUMENT BLOCK
001120 062706 000004 JSR PC,MSGOUT ; EDIT OUTPUT #MSG1 STRING AND PRINT IT
21 001124 000137 001176' ADD #4,SP ; RESTORE STACK POINTER
22 001130          JMP DONE ;
23 001130 123727 000000' 000000G RD01: CMPB IOST,#IS,SUC ; SUCCESSFUL READ?
24 001136 001415 BEQ RD02 ; YES
25 001140 113737 000000' 000010' MOV @#IOST,PAR1 ; ERROR
26 001146          MOUT$$ #MSG1,#PAR1 ;
001146 012746 001232' MOV #MSG1,-(SP) ; PUSH ADDRESS OF ASCIZ STRING
001152 012746 000010' MOV #PAR1,-(SP) ; PUSH ADDRESS OF ARGUMENT BLOCK
001156 004737 000000G JSR PC,MSGOUT ; EDIT OUTPUT #MSG1 STRING AND PRINT IT
001162 062706 000004 ADD #4,SP ; RESTORE STACK POINTER
27 001166 000137 001176' JMP DONE ;
28 001172          RD02: JSR PC,ISRECC ; INITIALIZE SREC COMMON
29 001172 004737 001204' DONE:
30 001176          EXIT$$
31 001176          MOV (PC)+,-(SP)
001200 063 001        .BYTE 51,1
001202 104377        EMT +D<377>

```



```
33                                     .SBTTL--ISRECC- INITIALIZE-SREC-COMMON.
34                                     ;
35                                     ;
36 001204                               ISRECC:
37 001204 012701 000016'                MOV.   #BUF+WORD2,R1 ;
38 001210 012702 000004G                MOV.   #SREC+WORD2,R2 ;
39 001214 012703 000116'                MOV.   #DBSLN,R3 ;
40 001220 006303                          ASL.   R3 ;
41 001222.                               ISREC1:
42 001222. 012122.                       MOV.   (R1)+,(R2)+ ;
43 001224 005303                          DEC.   R3 ;
44 001226 003375                          BGT.  ISREC1 ;
45 001230 000207                          RTS.  PC ;
46 001232 000067                          MSG1: .WORD LN1E-LN1 ;
47 001234 001236'                          .WORD LN1 ;
48
49 001236      105      122      122.   LN1: .ASCIZ /ERROR-READING INITIALIZATION-BLOCK-ON-DB- ERROR-=%ID /
50 001241      117      122      040
51 001244      122      105      101
52 001247      104      111      116
   001252      107      040      111
   001255      116      111      124
   001260      111      101      114
   001263      111      132      101
   001266      124      111      117
   001271      116      040      102.
   001274      114      117      103
   001277      113      040      117
   001302      116      040      104
   001305      102.     040      055
   001310      040      105      122.
   001313      122.     117      122.
   001316      040      075      045
   001321      061      104      040
   001324      000
50 001325
51
52.      001016'                               ;
                                     .END.  READIB. ;
```

BITVAL = 000000	CF.UPD = 000043	N.BFAC = 000004	SR.TWS = 000056	002.T#IBAR = 000024
BIT0 = 000001	DBSLEN = 000116	N.BHGH = 000006	SR.WSL = 000052	002.T#IBE = 020000
BIT1 = 000002	DG.ERR = 001000	N.BTCH = 000004	SR.YR = 000004	002.T#IBF = 040000
BIT10 = 002000	DG.SDF = 002000	N.BUFB = 004000	SR.IIN = 000024	002.T#ICD = 000040
BIT11 = 004000	DG.TDF = 004000	N.BUFW = 002000	SR.IIP = 000016	002.T#MODE = 004000
BIT12 = 010000	DONE = 001176R	N.FOS = 000764	TD#CTR = 176370	T#0B = 000036
BIT13 = 020000	DSW = 000004R	N.QURY = 000031	TD#CTW = 176360	T#0BE = 004000
BIT14 = 040000	EFN = 000001	N.SUNT = 000002	TD#INL = 004000	T#0BF = 010000
BIT15 = 100000	EMSG1 = 001232R	PAR1 = 000010R	TD#MEM = 000270	T#0BRA = 000034
BIT2 = 000004	FD.FID 000000	003 RD01 001130R	TD#0AR = 176344	T#0BWA = 000032
BIT3 = 000010	FD.FNB 000006	003 RD02 001172R	TD#0TR = 176346	T#0OUTA = 100000
BIT4 = 000020	FD.FVR 000004	003 READIB 001016R	TD#0RD = 000274	T#RBD0 = 000200
BIT5 = 000040	FD.LEN 000010	003 SLB = 001000	TD#RST = 176366	T#RNB = 000040
BIT6 = 000100	FN.ACK 000016	004 SREC = ***** GX	TD#SW = 176376	T#RSET = 040000
BIT7 = 000200	FN.FSA 000000	004 SR.ARE 000114	002.TD#TAR = 176372	T#SC = 000022
BIT8 = 000400	FN.FSB 000002	004 SR.ARS 000106	002.TD#TAW = 176362	T#SCLK = 020000
BIT9 = 001000	FN.FSC 000004	004 SR.DAY 000010	002.TD#TDR = 176374	T#SEG1 = 000000
BUF = 000012R	FN.FSD 000020	004 SR.DLT 000014	002.TD#TDW = 176364	T#SEG2 = 000001
BYTE0 = 000000	FN.MHR 000010	004 SR.ECB 000047	002.T#AD = 000020	T#SEG3 = 000002
BYTE1 = 000001	FN.NMB 000022	004 SR.ECH 000046	002.T#BA = 000002	T#S0 = 000001
BYTE2 = 000002	FN.NLS 000006	004 SR.ECL 000050	002.T#BD = 000010	T#UBUS = 100000
BYTE3 = 000003	FN.RDC 000014	004 SR.FIB 000012	002.T#BS0 = 100000	T#ICLK = 000400
BYTE4 = 000004	FN.UPD 000012	004 SR.GRE 000100	002.T#BT = 000020	T#BBEN = 000020
BYTE5 = 000005	IBLK = 001012R	SR.GRS 000072	002.T#BTAR = 000030	WORD0 = 000000
BYTE6 = 000006	IOST = 000000R	SR.LEN 000122	002.T#BTD = 002000	WORD1 = 000002
BYTE7 = 000007	IO.RLB = ***** GX	SR.LIN 000066	002.T#CD = 000100	WORD2 = 000004
BYTE8 = 000010	ISRECC 001204R	SR.LIP 000062	002.T#CLK = 002000	WORD3 = 000006
BYTE9 = 000011	ISRECI 001222R	SR.MON 000006	002.T#DISK = 000200	WORD4 = 000010
BYTVAL = 000012	IS.SUC = ***** GX	SR.NDC 000042	002.T#DRD = 000004	WORDS = 000012
CF.COT = 000041	LN1 = 001236R	SR.NDS 000036	002.T#MEM = 010000	WORD6 = 000014
CF.DGN = 000046	LN1E = 001325R	SR.NIN 000030	002.T#FSA = 000000	WORD7 = 000016
CF.DHR = 000042	LUN = 000001	SR.NIP 000022	002.T#FSAB = 000004	WORD8 = 000020
CF.DMC = 000047	MSGOUT = ***** GX	SR.SDB 000032	002.T#FSAC = 000014	WORD9 = 000022
CF.HBR = 000045	MS.DGN = 010000	SR.SRC 000002	002.T#FSB2 = 000010	WRDVAL = 000024
CF.HRL = 000044	N = 000012	SR.SUN 000000	002.T#IB = 000026	\$\$\$ARG = 000002

. ABS. 000000 000
001325 001
SRCOFF 000122 002
FDSCOF 000010 003
FNOFFS 000022 004
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 2936 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:18