Random Access Memory
GENERAL PURPOSE DISK CONTROL WITH FORTRAN

prepared for
presentation at

SWAP-14 Conference
Chicago, Illinois
April 22-24, 1968

cc-68-16

by

R. E. Schoenborn

Computer Center
Oregon State University
Corvallis, Oreg. 97331

R. E. Schoenborn, Research Associate
Computer Center
Oregon State Universtiy
Corvallis, Oregon 97331

Random Access Memory
GENERAL PURPOSE DISK CONTROL WITH FORTRAN*

The original purpose of this program was to exchange computer time for more memory storage space. The requirement of the specific program for which this subroutine was developed was the need for quick access to varible length records which were to be modified on-line and returned to storage. To speed up this process no attempt was made in this application to keep track of space no longer used or needed and while the data was of varible length, the indices were of fixed length and location. The reasoning behind this procedure was that any system that can afford the costs of on-line modification (such as with data display devices) must also afford the costs of back-up dumps to protect itself from machine failure.

The characteristics of this subroutine are that it allows the user to a] modify the program easily to match any random access memory, b] provide the user with the facility to write in either fixed length or varible length mode, with or without the user being aware of the present state of the files in the random access memory and finally c] to provide the user with the facility for detecting errors.

In any general purpose program there is the problem of how much control such a program should have and how much latitude is allowed the user. This latitude may prove a burden to some users and a limit to others. Also, one must weigh the overhead added by a general purpose program against the benefits provided to the user.

Perhaps an example of how this subroutine is used will be most useful in explaining how it works. A listing of the HISTDATA program is attached. The purpose of this program is to build a file of data on the RAM device in such a way that it is readily callable on the data displays. This calls for opening the previously initiated RAM device and testing it to determine that it was properly closed when last used. Next, data is collected in fixed blocks, stored, and the locations noted in an index. Finally, the last odd sized block is stored and the index and count blocks returned to the RAM. The RAM is then closed.

*The work described in this article was completed under a grant from the National Science Foundation.

In the opening, GENRAMOI gets a small block of data (presently set at 50 words as developed and stored during initialization) which contains, a] a flag word, b] number of words available on the RAM device (amount allocated), c] number of words used, d] next available block and word locations and e] additional space that the systems designer may use. The flag word is checked to determine if the RAM was properly initalized or closed the last time it was used.

The flag is returned in the operations parameter as to the status of the RAM. The user then may proceed to do the required work or take corrective action if so indicated.

In another example, the user may elect to write in an area of his own choosing and if he does, GENRAMOI will test to see that such a write will not exceed the available space and that the "Next available location address" which it maintains is properly updated, if necessary.

The six operations available to the user are (in the order of their use):

4 - Set up new RAM
2 - Open RAM previously set up or used
1 - Write on RAM
5 - Read from RAM
3 - Close RAM (also does Emergency Close)
6 - Write on RAM at location indicated by user

The Call for each operations parameter and the possible flag responses and operations of the subroutine are shown in the table which is attached to the documentation.

Again, this program was the result of a situation which required a system to receive, store, and retrieve data from four data display units simultaneously. Before it was written the disk used required a space allocation of 850 of the 1000 available tracks. After this program was installed the block and word address, which are returned by the subroutine and stored on the disk in fixed format index with the #6 function, indicated that only 60 fully packed tracks of data were required.

The listing of GENRAMOI which is presented here is for users without BDP units. Oregon State University's CDC 3300 happens to have one and we make use of it with a call to a small COMPASS routine for moving, blanking and zeroing blocks of memory in core. Listings for the BDP user, with or without the COMPASS routines, are also available.

Briefly then, GENRAMOI attempts to be readily adaptable to any RAM device, or system using such devices. It attempts to provide the user with as much or as little control as they wish to exercise with as little overhead as is possible in any abstracted language, such as FORTRAN.

1.0 Identification:
   1.1 GENRAMOI
   1.2 R. E. Schoenborn
   1.3 Computer Center, Oregon State University
   1.4 20 June 1967

2.0 Purpose:
   2.1 The purpose of this program is to provide a general subroutine in FORTRAN, to be called by FORTRAN, to allow I/O with any RAM unit of variable length records without gaps in the RAM. To provide a useful subroutine to use, as efficiently as possible, all available space provided by a RAM device, whether for temporary or permanent storage. Program length: approx. 1,000 words plus 2 blocks.

3.0 Usage:
   3.1 Calling Sequence: Call GENRAMOI (Request and response codes, NR of words, list, track #, word #)
   3.2 Inputs and formats are: (See example) Function Code--1 to 6, NR of words to be handled, BUFFER to read/write from, track and word related to flag.
   3.3 Outputs and formats are: (See attached examples) Flag returned in first parameter location.
   3.4 Process used on Inputs to get Outputs:
      3.4.1 Output: Data moved from table to physical record size area, packed consecutively with previous data and written to RAM.
      3.4.2 Input: Physical size records read from RAM and requested data unpacked from consecutive locations and moved to table of requesting program.
   3.5 List of error conditions, messages and operator actions: Response codes are returned to the calling program as noted on explanation example sheet.
   3.6 List of time constraints and order of operation with **respect to** other programs: User need not be concerned with any RAM I/O operation since this program lists for conclusion of operations before RAM is used and does not return to user until all operations are concluded.
   3.7 List of Equipment (Computer, Peripherals, off-line) to be used: Random Access Memory device equipped in EQUIP card and parameters described to program via COMMON/ DATA/ statements.
   3.8 List of systems, programs & subroutines available for use: GENMOVE (See attached listing).

	NFUNT	NBLOKSIZ	NAMBUFR	NRTRK	NWDPTR	CONDITION
	Call GENRAMOI(4,,,,)					
Req	4	--	--	--	--	Set up new RAM
Resp	1	--	--	--	--	O.K.
Resp	4	--	--	--	--	No go--System not able to find RAM track
Req	*	--	--	--	--	Illegal Request
Resp	3	--	--	--	--	No action
	Call GENRAMOI(2,,,,)					
Req	2	--	--	--	--	Initialize RAM previously set up or used
Resp	1	--	--	--	--	O.K.
Resp	5	--	--	--	--	RAM not originally set up or closed after last usage. Next available location provided for Reqd. write out may write on previous records. Write at your own risk. Read Req. will not go beyond Next Available Location. (See Call Option 3)
	Call GENRAMOI(1,1000,NLIST,NTRACK,NRWRD)					
Req	1	1000	NLIST	--	--	Write 1000 Word Buffer from NLIST
Resp	1	1000	NLIST	75	342	O.K. 1000 Words are on RAM Starting at Track 75 Word 342
Resp	2	1000	NLIST	--	--	Blocksize would Exceed available RAM size or limit NO Action.

Figure 1

	NFUNT	NBLOKSIZ	NAMBUFR	NRTRK	NWDPTR	CONDITION
	Call GENRAMOI(5,660,MYBUFFER,819,737)					
Req	5	660	MYBUFFR	819	737	Read and pack 660 word into MYBUFFR starting from Track 819 Word 737
Resp	1	660	MYBUFFR	819	737	O.K.
Resp	2	660	MYBUFFR	819	737	Read Req. goes beyond next available location No Action
Resp	4	660	MYBUFFR	819	737	No Action, System not able to locate RAM Track
	Call GENRAMOI(3,,,,)					
Req	3	--	--	--	--	Close Shop--Return next available locations to RAM.
	Call GENRAMOI(3,*,,**,***)					
Req	3	*Computed or estimated no. of words used	--	**Next available track, to be inserted	***Next available word, to be inserted	When RAM was not closed after a previous usage (program or machine failure) this emergency closing option may be requested Might be called after getting a Resp 5 code to an initialize Call (#2).
Resp	1	--	--	--	--	O.K.--Goodbye
	Call GENRAMOI(6,4745,INDEX,2,1)					
Req	6	4745	INDEX	2	1	Write 4745 word buffer from INDEX to RAM starting at Track 2 Word 1.
Resp	1	4745	INDEX	2	1	O.K. (See Notes on NFUNT = 1)
Resp	2	4745	INDEX	2	1	No go. (See Notes on NFUNT = 2)

Figure 2

```
      SUBROUTINE GENRAMOI (NFUNT,NBLOKSIZ,NAMBUFR,NRTRK,NWDPTR)          KIT00010
C * * *                          22 - G E N     WITHOUT B D P  UNIT      KIT00020
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KIT00030
C * * * EQUIP RANDOM ACCESS MEMORY(RAM) TO  7 FOR THIS ROUTINE            KIT00040
C * * *  THIS ROUTINE IS A FILE ASSIGNMENT PROGRAM FOR USE WITH           KIT00050
C * * *  A RANDOM ACCESS MEMORY DEVICE.  CHANGES TO THE PROGRAM ARE       KIT00060
C * * *  ARE NECESSARY ON CARDS MARKED ****    TO DEFINE MAX NR WORDS     KIT00070
C * * *  ON RAM DEVICE AND MAX NR OF WORDS ON A TRACK/SECTOR              KIT00080
C * * *  FILES ARE PACKED IN CONSECUTIVE LOCATIONS W/O LOSS OF SPACE      KIT00090
C * * *  MAXNR = MAX NR OF MACH WDS ON RAM OR SIZE OF ASSIGNED AREA       KIT00100
C * * *  MTRKSIZ = MAX NR WORDS ON A TRACK OR SECTOR                      KIT00110
C * * *  INPTR = 0 POINTERS NOT READ IN YET                               KIT00120
C * * *  NSTART = TRACK TO START WORKING FROM                             KIT00130
C * * *  1ST 50 WORDS ARE RESERVED FOR THIS PROGRAM. 1ST AVAIL WD = 51.   KIT00140
C * * *     REQUEST IS - - -                                              KIT00150
C * * *  NFUNT = 1 TO ADD NEW BLOCK OF DATA TO R A M                      KIT00160
C * * *  NFUNT = 2 OPEN       R A M AND GET PREVIOUS  FILE DATA           KIT00170
C * * *  NFUNT = 3 CLOSE SHOP AND SAVE INFO ON RAM, IF NRTRK =0 OR BLK    KIT00180
C * * *  NFUNT = 4 START UP A NEW DISC PACK OR RAM                        KIT00190
C * * *  NFUNT = 5 READ 1 NBLOKSIZE RECORD FROM NRTRK AT NWDPTR INTO      KIT00200
C * * *  NAMBUFR.                                                        KIT00210
C * * *  NFUNT = 6  USER CONTROLLED WRITE.. NEXTAVAIL TRACK AND WORD      KIT00220
C * * *     MODIFIED ONLY IF NECESSARY                                    KIT00230
C * * *  NBLOKSIZ = NR OF WORDS IN/OUT  TO/FROM BUFFER = NAMBUFR          KIT00240
C * * *  NRTRK AND NWDPTR = TRACK AND WORD STARTING LOCATION OF RECORD    KIT00250
C * * *  USER CAN PUT IT IN AN INDEX IF NECESSARY AFTER WRITE AND         KIT00260
C * * *  SUPPLY THEM FOR NFUNT =5  CALL                                   KIT00270
C * * *     RESPONSE IS - - -                                             KIT00280
C * * *  NFUNT = 1  IF O K   =2 IF RAM AREA EXCEEDED =3 IF REQ NOT        KIT00290
C * * *  COMPLETE OR CORRECT                                              KIT00300
C * * *   NFUNT =4 TRACK NOT FOUND   =5 NXAVAIL POINTER NOT RETURNED      KIT00310
C * * *  TO DISC LAST TIME.                                               KIT00320
C *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *KIT00330
      DIMENSION NAMBUFR(2),NTRBUF(1024,2),INBUF(2)                       KIT00340
      GO TO (900,40,10,600,942,300,400,900)  NFUNT+1                     KIT00350
   10 MAXNR=1000000 $ MTRKSIZ=1024 $ NSTART=1 $ INPTR=0                  KIT0****
      GOTO (11,960) LOCATEF (7,NSTART)                                   KIT00370
C * * *  READ IN NEXT AVAILA LOCTIONS AND INITALIZE                       KIT00390
   11 BUFFER IN (7,1) (NTRBUF(1,1),NTRBUF (MTRKSIZ,1))                   KIT0040*
 1100 GOTO (1100,1110) UNITSTF (7)                                       KIT0041*
 1110 IF (NTRBUF(5,1)   .EQ. 4HOKOK) 1120,1130                           KIT00420
 1120 NXAVTRK= NTRBUF(1,1)  $NXAVWD=NTRBUF(2,1)  $INBUF(1)=0             KIT00430
      JY = 1$  NRLEFT=MTRKSIZ $ MAXNR = NTRBUF (3,1)                     KIT00440
      NRUSED = NTRBUF (4,1)                                              KIT00450
      GOTO (1140,960) LOCATEF (7,NSTART)                                 KIT0046*
C * * *   RAM IS NOT SET UP RIGHT                                         KIT00470
 1130 NFUNT=5 $ RETURN                                                   KIT00480
 1140 NTRBUF (5,1) =0                                                    KIT00490
      BUFFER OUT (7,1)(NTRBUF(1,1),NTRBUF(MTRKSIZ , 1))                  KIT0050*
C * * *  SET FLAG O. K.                                                   KIT00510
   13 NFUNT=1     $ RETURN                                               KIT00520
C * * *  SET UP TO MOVE BUFFER AND WRITE TRACKS                           KIT00530
   40 NRNEED = NBLOKSIZ                                                  KIT00540
      KPTR =1                                                            KIT00550
      ITEMTRK= NRTRK=NXAVTRK                                             KIT00560
      ITEMWD = NWDPTR = NXAVWD                                           KIT00570
      IF ((NRUSED+NRNEED) .GT. MAXNR) 950, 50                            KIT00580
C * * *   TEST IF NEXT AVAILABLE TRACK IN BUFFERS                         KIT00590
```

```
   50 IF (INBUF(1) .EQ. NXAVTRK) 90,52                                  KIT00600
   52 IF (INBUF(2) .EQ. NXAVTRK) 92,70                                  KIT00610
C * * *  READ IN PARTLY FILLED TRACK                                    KIT00620
   70 IF (NXAVWD .EQ. 1) 120,72                                         KIT00630
   72 GOTO (74,76) JY                                                   KIT00640
   74 JY=2 $ GOTO 78                                                    KIT00650
   76 JY =1                                                             KIT00660
   78 GOTO (80,960)LOCATEF (7,NXAVTRK)                                  KIT0067*
   80 BUFFER IN (7,1)(NTRBUF(1,JY),NTRBUF(MTRKSIZ, JY))                 KIT0068*
      INBUF(JY) = ITEMTRK                                               KIT00690
   88 GOTO (88,120) UNITSTF(7)                                          KIT0070*
   90 JY = 1  $ GOTO 120                                                KIT00710
   92 JY = 2                                                            KIT00720
  120 NRLEFT = MTRKSIZ - ITEMWD  +1                                     KIT00730
  125 IF (NRNEED .GT. NRLEFT) 130,160                                   KIT00740
C * * *  MOVE OUT PART OF BUFFER                                        KIT00750
C*130 CALL GENMOVE (NAMBUFR(KPTR),NTRBUF(ITEMWD,JY),NRLEFT)             KIT0076*
  130 IP=KPTR $ JP=ITEMWD $ IT=KPTR+NRLEFT-1 $ GOTO 132                 KIT00761
  131 IP=IP+1 $ JP=JP+1                                                 KIT00762
  132 NTRBUF(JP,JY)=NAMBUFR(IP) $ IF (IT-IP) 131,132                    KIT00763
  133 GOTO (133,134) UNITSTF(7)                                         KIT0077*
  134 GOTO (136,900) LOCATEF (7,ITEMTRK)                                KIT0078*
  136 BUFFER OUT (7,1)(NTRBUF(1,JY), NTRBUF(MTRKSIZ,JY))                KIT0079*
      INBUF(JY) = ITEMTRK                                               KIT00800
      GOTO (140,144) JY                                                 KIT00810
  140 JY=2 $ GOTO 150                                                   KIT00820
  144 JY=1                                                              KIT00830
  150 ITEMTRK   = ITEMTRK + 1                                           KIT00840
      NRNEED = NRNEED - NRLEFT                                          KIT00850
      KPTR = KPTR + NRLEFT                                              KIT00860
      ITEMWD = 1                                                        KIT00870
      NRLEFT = MTRKSIZ                                                  KIT00880
      GOTO 125                                                          KIT00890
C*160 CALL GENMOVE (NAMBUFR(KPTR),NTRBUF(ITEMWD,JY),NRNEED)             KIT0090*
  160 IP=KPTR $ JP=ITEMWD $ IT=KPTR+NRNEED-1 $ GOTO 162                 KIT00901
  161 IP=IP+1 $ JP=JP+1                                                 KIT00902
  162 NTRBUF(JP,JY)=NAMBUFR(IP) $ IF (IT-IP) 161,163                    KIT00903
  163 GOTO (163,164) UNITSTF(7)                                         KIT0091*
  164 GOTO (166,900) LOCATEF(7,ITEMTRK)                                 KIT0092*
  166 BUFFEROUT (7,1) (NTRBUF(1,JY),NTRBUF(MTRKSIZ,JY))                 KIT0093*
      INBUF(JY) = ITEMTRK                                               KIT00940
C * * *  TEST IF TRACK COUNT SHOULD BE CHANGED                          KIT00950
      NXAVWD =  NXAVWD + NBLOKSIZ                                       KIT00960
  170 IF (NXAVWD .LE. MTRKSIZ) 175,172                                  KIT00970
  172 NXAVTRK=NXAVTRK+1                                                 KIT00980
      NXAVWD=NXAVWD-MTRKSIZ $ GOTO 170                                  KIT00990
  175 NRUSED=NRUSED+NBLOKSIZ                                            KIT01000
  177 GOTO (177,13) UNITSTF(7)                                          KIT0101*
C * * *  READ IN REQUEST  =5                                            KIT01020
  300 NRNEED=NBLOKSIZ                                                   KIT01030
      ITEMWD=NWDPTR+NBLOKSIZ-1                                          KIT01040
      ITEMTRK=NRTRK $JY=1                                               KIT01050
      KPTR=1                                                            KIT01060
  302 IF (ITEMWD .LE. MTRKSIZ) 308,304                                  KIT01070
  304 ITEMWD=ITEMWD-MTRKSIZ                                             KIT01080
      ITEMTRK=ITEMTRK+1 $GO TO 302                                      KIT01090
  308 IF (ITEMTRK-NXAVTRK) 316,312,950                                  KIT01100
  312 IF (ITEMWD.LT.NXAVWD) 316,950                                     KIT01110
  316 ITEMTRK=NRTRK $ ITEMWD=NWDPTR                                     KIT01120
      IF (NRTRK .EQ. INBUF(1)) 321,320                                  KIT01130
  320 IF (NRTRK.EQ.INBUF(2)) 322,380                                    KIT01140
  321 NRBUF=1 $JY = 2  $ GOTO 324                                       KIT01150
  322 NRBUF=2 $  JY = 1                                                 KIT01160
  324 NRLEFT=MTRKSIZ+1-ITEMWD                                           KIT01170
  328 IF (NRNEED.GT.NRLEFT) 338,333                                     KIT01180
C*333 CALL GENMOVE (NTRBUF(ITEMWD,NRBUF),NAMBUFR(KPTR),NRNEED)          KIT0119*
  333 IP=KPTR $ JP=ITEMWD $ IT=KPTR+NRNEED-1 $ GOTO 335                 KIT01191
  334 IP=IP+1 $ JP=JP+1                                                 KIT01192
  335 NAMBUFR(IP)=NTRBUF(JP,NRBUF) $ IF (IT-IP) 334,13                  KIT01193
C*    GO TO 13                                                          KIT01200
  338 ITEMTRK=ITEMTRK+1                                                 KIT01210
      GOTO (340,960) LOCATEF(7,ITEMTRK)                                 KIT0122*
  340 BUFFER IN (7,1)(NTRBUF(1,JY),NTRBUF(MTRKSIZ,JY))                  KIT0123*
C*    CALL GENMOVE (NTRBUF(ITEMWD,NRBUF),NAMBUFR(KPTR),NRLEFT)          KIT0124*
      IP=KPTR $ JP=ITEMWD $ IT=KPTR+NRLEFT-1 $ GOTO 344                 KIT01241
  342 IP=IP+1 $ JP=JP+1                                                 KIT01242
  344 NAMBUFR(IP)=NTRBUF(JP,NRBUF) $ IF (IT-IP) 342,346                 KIT01243
  346 KPTR=KPTR+NRLEFT                                                  KIT01250
      NRNEED=NRNEED-NRLEFT                                              KIT01260
      INBUF(JY)=ITEMTRK                                                 KIT01270
      NRLEFT=MTRKSIZ                                                    KIT01280
      GO TO (350,352)JY                                                 KIT01290
  350 JY=2 $NRBUF=1 $GO TO 356                                          KIT01300
  352 JY=1 $NRBUF=2                                                     KIT01310
  356 ITEMWD=1                                                          KIT01320
  360 GO TO (360,328) UNITSTF(7)                                        KIT0133*
  380 GO TO (380,384) UNITSTF(7)                                        KIT0134*
  384 GOTO (388,960) LOCATEF(7,ITEMTRK)                                 KIT0135*
  388 BUFFER IN (7,1)(NTRBUF(1,1),NTRBUF(MTRKSIZ,1))                    KIT0136*
      INBUF(1)=ITEMTRK                                                  KIT01370
  390 GO TO (390,321) UNITSTF(7)                                        KIT0138*
C * * *    USER CONTROLLED WRITE (REQ = 6). IF WRITE GOES BEYOND        KIT01390
C * * *    NEXTAVAIL TRACK AND WORD THESE WILL BE RESET. OTHERWISE      KIT01400
C * * *    NOTHING IS AFFECTED.  RETURN FLAGS SAME AS REGULAR WRITE.    KIT01410
  400 LPTR=1 $ NNRTRK=NRTRK                                             KIT01420
      NRA=NRTRK*MTRKSIZ+NWDPTR+NBLOKSIZ                                 KIT01430
      NRB=NBLOKSIZ $ NRC=NWDPTR                                         KIT01440
      IF (NRA .GT. MAXNR) 950,402                                       KIT01450
C * * *    TEST FOR FULL TRACK OUTPUT                                   KIT01460
  402 IF (NRC .EQ. 1) 404,420                                           KIT01470
  404 IF (NRB .LT. MTRKSIZ) 420,406                                     KIT01480
C * * *    SET UP AND MOVE FULL TRACK FROM USERS TABLE                  KIT01490
  406 GOTO (410,960) LOCATEF(7,NNRTRK)                                  KIT0150*
  410 NRMOV=LPTR+MTRKSIZ-1                                              KIT01510
      BUFFEROUT (7,1) (NAMBUFR(LPTR),NAMBUFR(NRMOV))                    KIT0152*
      LPTR=LPTR+MTRKSIZ $ NNRTRK=NNRTRK+1                               KIT01530
      NRB=NRB-MTRKSIZ                                                   KIT01540
  412 GOTO (412,453) UNITSTF(7)                                         KIT0155*
C * * *    SET UP TO MOVE PARTIAL TRACK...TEST IF TRACK IN CORE         KIT01560
  420 IF (INBUF(1) .EQ. NNRTRK) 440,422                                 KIT01570
  422 IF (INBUF(2) .EQ. NNRTRK) 442,425                                 KIT01580
  425 GOTO (427,428) JY                                                 KIT01590
  427 JY=2 $ GOTO 430                                                   KIT01600
  428 JY=1                                                              KIT01610
  430 GOTO (433,960) LOCATEF(7,NNRTRK)                                  KIT0162*
  433 BUFFERIN (7,1) (NTRBUF(1,JY),NTRBUF(MTRKSIZ,JY))                  KIT0163*
      INBUF(JY)=NNRTRK                                                  KIT01640
  435 GOTO (435,445) UNITSTF(7)                                         KIT0165*
  440 JY=1 $ GOTO 445                                                   KIT01660
  442 JY=2                                                              KIT01670
  445 NRMOV=MTRKSIZ-NRC+1                                               KIT01680
      IF (NRMOV .GT. NRB) 447,450                                       KIT01690
  447 NRMOV=NRB                                                         KIT01700
C*450 CALL GENMOVE (NAMBUFR(LPTR),NTRBUF(NRC,JY),NRMOV)                 KIT0171*
  450 IP=LPTR $ JP=NRC $ IT=LPTR+NRMOV-1 $ GOTO 452                     KIT01711
  451 IP=IP+1 $ JP=JP+1                                                 KIT01712
  452 NTRBUF(JP,JY)=NAMBUFR(IP) $ IF (IT-IP) 451,453                    KIT01713
  453 GOTO (454,960) LOCATEF(7,NNRTRK)                                  KIT0172*
  454 BUFFEROUT (7,1) (NTRBUF(1,JY),NTRBUF(MTRKSIZ,JY))                 KIT0173*
      NRC=1 $ NRB=NRB-NRMOV                                             KIT01740
      NNRTRK=NNRTRK+1 $  LPTR=LPTR+NRMOV                                KIT01750
  455 GOTO (455,456) UNITSTF(7)                                         KIT0176*
  456 IF (NRB) 402,460                                                  KIT01770
C * * *    TEST IF NEXTAVAIL INFO NEEDS UPDATING                        KIT01780
  460 NTK=NRTRK $ NWD=NWDPTR+NBLOKSIZ-1                                 KIT01790
  462 IF (NWD .LT. MTRKSIZ) 470,465                                     KIT01800
  465 NWD=NWD-MTRKSIZ $ NTK=NTK+1 $ GOTO 462                            KIT01810
  470 IF (NXAVTRK-NTK) 472,476,13                                       KIT01820
  472 NXAVTRK=NTK $ GOTO 480                                            KIT01830
  476 IF (NWD .LT. NXAVWD) 13,480                                       KIT01840
  480 NXAVWD=NWD+1 $ GOTO 13                                            KIT01850
C * * *  ALL DONE - CLEAN UP ...  RETURN POINTERS TO DISC               KIT01860
  600 GOTO (600,602) UNITSTF (7)                                        KIT0189*
  602 GOTO (604, 960) LOCATEF (7,NSTART)                                KIT0190*
  604 BUFFER IN (7,1)(NTRBUF(1,1), NTRBUF (MTRKSIZ,1))                  KIT0191*
  606 GOTO (606,608) UNITSTF(7)                                         KIT0192*
  608 IF (NRTRK .EQ. 4H    ) 617, 610                                   KIT01921
  610 IF (NRTRK .EQ. 0)  617, 612                                       KIT01922
C              REBUILD DISK OPEN AFTER BLOW UPR OR SUMTHIN              KIT01923
  612 NTRBUF (1,1) = NRTRK $NTRBUF(2,1)=NWDPTR$NTRBUF(4,1)=NBLOKSIZ      KIT01924
      GOTO 620                                                          KIT01925
  617 NTRBUF(1,1) = NXAVTRK $ NTRBUF(2,1)=NXAVWD                        KIT01930
      NTRBUF(4,1) = NRUSED                                              KIT01940
  620 NTRBUF (3,1) = MAXNR                                              KIT01950
  629 NTRBUF(5,1)= 4HOKOK                                               KIT01960
      GOTO (930,960) LOCATEF(7,NSTART)                                  KIT0197*
  930 BUFFEROUT (7,1) (NTRBUF(1,1),NTRBUF(MTRKSIZ,1))                   KIT0198*
  940 GOTO (940,13) UNITSTF(7)                                          KIT0199*
  942 NTRBUF(1,1) = 1  $ NTRBUF (2,1)=51                                KIT02000
      NTRBUF(3,1)=1000000 $ NTRBUF(4,1)=0 $ MTRKSIZ=1024                KIT0****
      NSTART=1                                                          KIT0$$$$
      GOTO 629                                                          KIT02020
C * * *  INPUT REQUEST ERRONEOUS. RETURN BAD FLAG.                      KIT02022
  900 NFUNT= 3      $ RETURN                                            KIT02025
C * * *  ALLOCATED RAM AREA TO SMALL FOR NEXT RECORD,SET FLAG           KIT02030
  950 NFUNT= 2  $NRTRK=NWDPTR=0  $ RETURN                               KIT02040
C * * *     CANNOT FIND TRACK, SET FLAG                                 KIT02050
  960 NFUNT=4 $ RETURN                                                  KIT02060
      END                                                               KIT02070
```

```
         IDENT     MOVE
         ENTRY     GENFILL,GENMOVE
*       1/26/68
************************************************************************
***    HI SPEED XERO FILL,BLANK FILL OR BUFFER MOVE                  ***
***    BY USE OF B.D.P. UNIT.    ANY PLACE A DO LOOP IS USED FOR THESE ***
***    PURPOSES GREATER EFFICIENCY CAN BE EFFECTED BY USE OF THIS ROUTINE.   ***
***              USE IN FORTRAN PROG AS FOLLOWS...                   ***
***    CALLGENFILL(8 OR 16, BUFF, NRWORDS)
***            8= BLANK FILL      16= ZERO FILL                       ***
***    CALL GENMOVE(FROMBUFF, TOBUFF, NRWORDS)                        ***
***       BUFFER ADDRESS MAYBE SUBSCRIPTED.        NRWORDS .LE.  1023 ***
***       EXAMPLE              BLANK A   4000 WORD BUFFER             ***
***    DIMENSION MATRIX (4000)                                        ***
***       DO 6 I=1,4000,1000                                          ***
***       CALL GENFILL (1,MATRIX(I), 1000)                            ***
***    6   CONTINUE                                                   ***
************************************************************************
GENMOVE  UJP       **
         STI       TEMP,3              SAVE INDEX
         LDI       GENMOVE,3
         LDA       0,3                 GET FROM ADDRESS
         SHA       2                   CONVERT TO CHAR. ADD.
         ANA,S     77774B              MASK IT AND
         SCHA      MOVE                STORE
         ENA,S     0
         ENQ,S     70000B
SAME     SACH      MOVE+4
         LDA,I     2,3                 GET NR OF WORDS TO MOVE
         SHA       2
         AQA
         STA       MOVE+2
         INI       3,3                 SET INDEX TO RETURN LOCATION
         STI       GENFILL,3
         LDA       -2,3                GET BUFFER ADD/TO ADD.
         SHA       2                   CONVERT TO CHAR. ADD.
         ANA,S     77774B              MASK IT
         SCHA      MOVE+1
MOVE     MVE       MOVE+4,0,0,0,0,0     MOVE OR BLANK/ZERO FILL
TEMP     ENI       **,3
GENFILL  UJP       **
         STI       TEMP,3              SAVE INDEX
         LDI       GENFILL,3
         LDA,I     0,3                 GET OPTION. 8=BLANK  16= ZERO
         ENQ,S     0
         UJP       SAME
         END
         IDENT     MOVE
         ENTRY     GENMOVE,GENFILL
********************************************************************************
***   HI SPEED XERO FILL,BLANK FILL OR BUFFER MOVE                            ***
***              USE IN FORTRAN PROG AS FOLLOWS...                            ***
***   CALLGENFILL(8 OR 16, BUFF, NRWORDS)
***            8= BLANK FILL      16= ZERO FILL                               ***
***   CALL GENMOVE(FROMBUFF, TOBUFF, NRWORDS)                                 ***
***      BUFFER ADDRESS MAYBE SUBSCRIPTED.           NRWORDS .LE.  1023       ***
***      EXAMPLE                BLANK A   4000 WORD BUFFER                    ***
***   DIMENSION MATRIX (4000)                                                 ***
***      DO 6 I=1,4000,1000                                                   ***
***      CALL GENFILL (1,MATRIX(I),` 1000)                                    ***
***   6   CONTINUE                                                            ***
********************************************************************************
GENMOVE  UJP       **
         STI       TEM,1                     SAVE INDEX
         LDI       GENMOVE,1                LOAD ADDRESS OF PARAMETER LIST
         STI       GENFILL,1                STORE RETURN ADDRESS
         LDAQ      0,1                      GET ADDRESS OF FROM AND TO BUFS
         SWA       LOAD                     STORE LOAD ADDRESS
         SHAQ      24
         SWA       STORE                    STORE STORE ADDRESS
         LDA,I     2,1                      LOAD NO OF WORDS TO MOVE
         TAI       1                        TRANSFER WORD COUNT TO INDEX
         INI       -1,1
LOAD     LDA       **,1                     LOAD WORD
STORE    STA       **,1                     STORE WORD
         IJD       *-2,1
OUT      ENA       3
         RAD       GENFILL                  INCREASE RETURN ADDRESS BY THREE
TEM      ENI       **,1
GENFILL  UJP       **
         STI       TEM,1                    SAVE INDEX
         LDI       GENFILL,1                 LOAD ADDRESS OF PARAMETER LIST
         LDA       1,1                       LOAD ADDRESS OF BUFFER AND
         SWA       STOR                       STORE
         LDQ,I     0,1                       LOAD FLAG DATA
         LDA,I     2,1                       LOAD COUNT
         TAI       1                         TRANSFER COUNT TO INDEX
         INI       -1,1                      DECREASE BY ONE
         ENA       0                         LOAD A WITH ZERO
         QSE       16                        IF FLAG IS 16 STORE ZERO
         LDA       =H                        OTHERWISE STORE BLANKS
STOR     STA       **,1                      STORE BLANKS OR ZEROS
         IJD       *-1,1
         UJP       OUT
         END
```

```
C * * * * * * *     DEMONSTRATION OF PROGRAM USING G E N R A M O I   * * * *
      PROGRAM HISTDATA
C * * * * * * * * * * * * * * * * * *  27 RESV   * * * * * * * * * * * *
C *   *   *   *  THIS PROGRAM IS TO ENTER HISTORY DATA TO THE FILE TO BE
C *   *   *  CALLED FROM DATA DISPLAY UNIT
C  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
      INTEGER GENRAMOI,DISKFIX
      COMMON/DATA/ITOTAL(10),IBILPTCT,IBSORTCT,ITR(26),IWD(26),HIGH(17)
     1IDAUTH(12),IDS(3,24),IDELAY(18),IDOT,IDOTT,ITRPT,IWDPT
      DATA (IDOT=4H....),(IDOTT=4H....)
      DIMENSION IPAGE(500),INDEX(500),BUF(200)
      EQUIVALENCE (DOT,IDOT),(TRWDPT,ITRPT),(BUF,INDEX)
    1 FORMAT (A8)
    2 FORMAT (13A4)
    3 FORMAT (R2,12A4)
C * * * * *   OPEN PREVIOUSLY INITALIZED RAM AND TEST FOR POSSIBLE ERROR
 1000 IF (GENRAMOI (2,0,0,0,0)-5) 1020,1010
C * * * * *  RAM WAS NOT CLOSED LAST TIME. (POSSIBLY DUE TO COMPUTER
C * * * * *         FAILURE )    RESTART.
 1010 M=DISKFIX(INDEX) $ GOTO 1000
C * * * * *   READ IN BLOCK
 1020 M=GENRAMOI (5,200,ITOTAL,1,51)
      IP=1 $ IPW=101 $ IPC=401
   10 READ (20,1) BUF(IP) $ IF (BUF(IP)) 15,70
   15 IPTOP=IFLAG=1 $ INDEX(IPC)=0
   20 ITOP=IPTOP+11 $ READ (20,2) (IPAGE(I),I=IPTOP,ITOP),ITEM
      IF (IPAGE(IPTOP)-4H****) 25,50
   25 IPTOP=ITOP+1 $ ITOP=IPTOP+12 $ READ (20,3) (IPAGE(J),J=IPTOP,ITOP)
      IF (IPAGE(IPTOP)-4HOO**) 30,60
   30 IPAGE(IPTOP)=AND(ITEM,77770000B)+IPAGE(IPTOP)
      IPTOP=ITOP+1 $ IF (ITOP-500) 20,80
   50 IF (IPTOP-1) 52,58
   52 IPTOP=IPTOP-1
   54 INDEX(IPC)=INDEX(IPC)+IPTOP
C * * * * *  WRITE LAST NEW PAGE TO RAM...GET BLOCK(TRACK) AND WORD
C * * * * *        ADDRESS IN ITRPT  AND  IWDPT
      M=GENRAMOI (1,IPTOP,IPAGE,ITRPT,IWDPT)
      IF (IFLAG) 57,58
   57 BUF(IPW)=TRWDPT
   58 IP=IP+1 $ IPW=IPW+1 $ IPC=IPC+1 $ GOTO 10
   60 IPAGE(IPTOP)=ITEM $ GOTO 54
   70 ITOTAL(7)=IP-1
      DO 100 I=IP,100
      BUFF (I)=DOT $ BUF (I+100)=0. $ INDEX(I+400)=0
  100 CONTINUE
C * * * * *        FORCED WRITE OF FIXED SIZE INDEX
      M=GENRAMOI (6,500,INDEX,ITR(23),IWD(23))
C * * * *  WRITE TOTALS FOR COMMON TABLES USED FOR ALL TYPES OF ACCTS.
C * * * *  CLOSE RAM (RETURN NEXT AVAILABLE ADDRESS ETC  TO STORAGE ON RAM)
      M=GENRAMOI (6,200,ITOTAL,1,51) $ M=GENRAMOI (3,0,0,0,0) $CALL EXIT
C * * * * *   WRITE 1 FULL BUFFER (FIXED SIZE) AS NEW PAGE ...
   80 M=GENRAMOI (1,500,IPAGE,ITRPT,IWDPT)
      INDEX(IPC)=INDEX(IPC)+500 $ IPTOP=1 $ IF (IFLAG) 85,20
   85 BUF(IPW)=TRWDPT $ IFLAG=0 $ GOTO 20
      END
```