COMPUTER RETRIEVAL
OF
CSE MULTIPLEXER DATA

W. R. Weissenberger
E. L. Wells

SEPTEMBER 1967

AEC RESEARCH &
DEVELOPMENT REPORT

AUTHOR		TITLE	

BATTELLE

NORTHWEST

BATTELLE MEMORIAL INSTITUTE

PACIFIC NORTHWEST LABORATORY

3000 STEVENS DRIVE, P. O. BOX 999, RICHLAND, WASHINGTON 99352

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

operated by

BATTELLE MEMORIAL INSTITUTE

for the

UNITED STATES ATOMIC ENERGY COMMISSION UNDER CONTRACT AT(45-1)-1830

PRINTED BY/ FOR THE U. S. ATOMIC ENERGY COMMISSION

COMPUTER RETRIEVAL
OF
CSE MULTIPLEXER DATA

By

W. R. Weissenberger

and

E. L. Wells

Systems Programming Unit
Mathematical Analysis Section

September 1967

88

COMPUTER RETRIEVAL
OF
CSE MULTIPLEXER DATA

By

W. R. Weissenberger

and

E. L. Wells

Systems Programming Unit
Mathematical Analysis Section

September 1967

COMPUTER RETRIEVAL
OF
CSE MULTIPLEXER DATA

ABSTRACT

Descriptions of PDP-7 computer programs to allow the retrieval of multiplexer data from the Containment System Experiment are given. Program listings and flow-diagrams of the programs are included.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	ii
I. INTRODUCTION	1
II. DESCRIPTION OF THE MULTIPLEXER DATA RETRIEVAL PROGRAM (MPLX)	1
III. DESCRIPTION OF SUBROUTINE FETCH	4
IV. DESCRIPTION OF SUBROUTINE WTAPE	5
V. DESCRIPTION OF SUBROUTINE LFEED	5
REFERENCES	30

LIST OF TABLES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1.	MPLX Program Listing	7
2.	FETCH Program Listing	18
3.	WTAPE Program Listing	22
4.	LFEED Program Listing	25

LIST OF FIGURES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1.	MPLX Flow-Diagram	6
	APPENDIX A	
	Operating Procedure	26
	APPENDIX B	
	Nomenclature	28

I. Introduction

A PDP-7 computer interfaced with a low and a high speed multiplexer is used to log data⁽¹⁾ for the "Containment System Experiment," a program initialized by Battelle Memorial Institute, Pacific Northwest Laboratory late in 1963.⁽²⁾ The data logged by the PDP-7 low and high speed multiplexer combination is stored on magnetic tapes. The purpose of the Multiplexer Data Retrieval Program is to retrieve the data which has been stored on magnetic tapes by the low speed multiplexer program, convert the data to decimal voltages, and print the output on the IBM Selectric typewriter. Due to the limited amount of storage available in the PDP-7, the program will only retrieve data for twenty major points and the associated subpoints. The purpose of this document is to give a general description of the Multiplexer Data Retrieval Program and its associated subroutines.

II. Description of the Multiplexer Data Retrieval Program (MPLX)

The first step in this program is to read in the MPGEN2 paper tape.⁽¹⁾ This tape contains the CLPER table, ISCTM table, and the PERT table. The CLPER table is used to determine which major points are to be checked and the scan class and period in which they are to be found. The ISCTM table and the PERT table are used in computing the time at which each major point was checked. The MPGEN2 tape is read in at address 15700₈.

After the MPGEN2 tape has been read into core, the program requests the number of magnetic tapes to be processed, the number of bad blocks on each tape, (as noted by the CSEMPLEX code),⁽¹⁾ and the octal block numbers of each bad block. At this point the program checks to see if the current block on the input tape is a bad block. If the block is bad, the next sequential block is checked. This process is continued until a

-2-

a good block has been found.

After a good block has been found, subroutine FETCH is called to retrieve the good block (256_{10} words). The program then checks the first word of this block. This word contains a major point and the first subpoint. The CLPER table is then checked to see if this major point was requested. If the major point was not requested then the next three words are bypassed and the next major point is checked.

When a requested major point has been found, the next word is scanned. This word contains the period, scan class, and the second subpoint. The period is then checked to see if it is the same as the previous period. (This is done only after the first requested major point has been found).

If the period is not the same, the program checks to see if all of the scan classes for the previous period were checked. If all six of the scan classes have not been checked, the currently stored data is typed out and the tape is rewound to the beginning of the previous period. This is done until all six of the scan classes have been checked. After the period changes and all six scan classes have been checked, the currently stored data is typed out and the program continues with the new period. This procedure is followed for all of the periods except period 0 which has only one scan class.

If the current period has not changed, the scan class is checked against the previous scan class. If the scan classes are not the same, the program bypasses the next two words and looks for a major point in the same scan class as the previous one. If the scan class has not changed, then the next word is checked. This word contains ISCAN (the

-3-

number of times the current major point has been scanned), and the third subpoint data. The fourth word is now checked. This word contains the fourth subpoint data, and $IT\emptyset$ (the last bit of word four, used to determine the time of rupture).⁽¹⁾

After five major points have been checked, they are written on one line with the IBM typewriter. The next fifteen major points checked are stored in three separate memory blocks. After these blocks are filled, or after the period changes, the program calls subroutine WTape, and the three memory blocks are written on three magnetic tapes. When the period has changed, these three tapes are read and the data is written using the IBM typewriter. This is done by using FETCH and writing one block of data at a time from the first tape (until all the data on this tape has been written). This process is repeated for each of the two remaining tapes.

The time of the scan for the twenty major points is computed each time the first major point is checked. The time of scan is approximately the same for all the major points which follow the first major point.

All of the previously mentioned procedures are repeated until the end of the current tape is reached, or until the last word of the input is reached indicated by either the word being equal to 777777_8 or 777000_8 .⁽¹⁾ When a word is found that equals 777000_8 , all of the current y stored data is typed out, and the operator is told to change to the next data tape. If the word matches a 777777_8 , the same procedure follows except that there are no remaining data tapes, and $T\emptyset$ is printed

out and the program ends. T_0 is set equal to the time of scan when IT_0 changes from a 0 to a 1. (i.e., when the disc is ruptured).⁽¹⁾

III. Description of Subroutine FETCH

This subroutine was written to retrieve data from a magnetic tape one block at a time. The main program supplies FETCH with the number of the unit which contains the data tape, and the number of the block of data to be retrieved.

When the main program calls FETCH, it first checks to see if the correct unit is selected. When the unit is selected, FETCH searches for the current block requested. FETCH searches for this block by first searching forward on the tape until a block has been found. If the number of the block found is less than the block requested, the tape continues searching forward until the correct block has been found. If the block number is greater than the one requested, the tape searches in reverse until a block number two less than the one requested has been found. At this point the tape is again searched forward. If the block requested is not found on the tape, an error message is written and the program halts.

When the right block has been found, it is stored in one of two memory locations. One location is used for input tapes and the other for the three storage output tapes. FETCH determines which location is to be used by checking the unit the tape is on. Unit five is always used for input tapes and units one, two, and three are always used for output tapes.

After the block has been found, retrieved, and stored in memory, control is returned to the main program.

IV. Description of Subroutine WTAPE

WTAPE was written to allow the main program to store data and scan times on magnetic tapes. The main program supplies WTAPE with the unit number and the block number.

When WTAPE is called, it first checks to see if the correct unit is selected. WTAPE then finds the correct block, positions the magnetic tape and prepares for writing. The main program stores the data from fifteen major points and the associated subpoints in three memory areas. Each area contains five major points, their scan times, and their subpoints. WTAPE looks for and writes the contents of the area corresponding to the unit number (either one, two, or three). After the memory area has been written on tape, WTAPE fills the locations with 777777's. This is done so the last word to be written on a tape will be a 777777 and the main program can end.

After WTAPE has filled the memory location with 777777's, it returns control to the main program.

V. Description of Subroutine LFEED

LFEED was written to line feed the IBM typewriter at the end of 45 lines of print. This was done to assure uniform length for each page of output. The main program supplies LFEED with the current number of lines (45) subtracted from the total number of lines available on a page (52). LFEED then skips the typewriter one line at a time until the number of skipped lines equals the total obtained above (7).

LFEED was not written exclusively for this program. This is the reason the number of lines printed is subtracted from 52 to obtain the total. Another program may require more than 45 lines printed before skipping to the next page.

After LFEED has finished skipping lines on the IBM typewriter it returns control to the main program.

FIGURE 1
Retrieval ("Fetch") Flow Diagram

/MULTIPLEXER DATA RETRIEVAL PROGRAM

```

2WORD
S      OCTAL
S      XND=17777
S      DECIMAL
      DIMENSION IDUM(48),JUMP2(2136),JUMP3(9),MASK(7),DATA
$      (4,20),SCTM(7,7),DATA1(105),DATA2(105),DATA3(105),
$      KBFA(256),IPT(48),KBFB(256),JUMP1(448),ICLPER(64),
$      IPERT(7,2),ISCTM(7,7),NBAD(10),NBLOCK(10,10),II0(7),
$      II(7),PERT(7),IPO(4),IPOT(20,8),JPT(48)
$      ,JUMP4(107)
      EXTEND MODE NBLOCK,NBAD,JUMP3,PERT,II0,IPT,MAJOR,IT0,
$      ISCAN,
$      KBFB,JUMP1,ICLPER,K,IPERT,ISCTM,SCTM,JPT,JUMP2,
$      IPOT,IPO,II,JUMP4,KBFA,IDUM,DATA1,DATA2,DATA3,DATA
C      KBFB=17100
C      ICLPER=16100
C      IPERT=16061
C      ISCTM=16000
C      NBAD=15756
C      NBLOCK=15622
C      II0=15615
C      II=15606
C      PERT=15577
C      LOAD READ IN TAPE AT 15700
      N3=1
S      LAC (MSK
S      DAC MASK.+2
S      HLT
C      CLPER,SCTM,AND PERT TABLES ARE READ IN AT VHS TIME
      DO 20 I=1,7
      II0(I)=0
      II(I)=0
      K=IPERT(I,2)
S      RCR
S      DAC K.
      PERT(I)=(262144.*IPERT(I,1)+K*2.)+0.5
      DO 10 J=1,7
      SCTM(J,I)=ISCTM(J,I)
10     CONTINUE
20     CONTINUE
      WRITE2,2010
      READ1,1020,NTAPES
C      NTAPES=NUMBER OF INPUT TAPES
      DO 40 I=1,NTAPES
      DO 30 J=1,10
      NBLOCK(I,J)=0
30     CONTINUE
      NBAD(I)=0
C      NBAD=NUMBER OF BAD BLOCKS ON TAPE I

```

-8-

```

40      CONTINUE
      DO 50 I=1,4
      IPO(I)=-I+1
50      CONTINUE
      DO 60 I=1,48
      JPT(I)=-1
      IPT(I)=-1
60      CONTINUE
      DO 70 I=1,NTAPES
      WRITE2,2020
      READ1,1020,NBAD(I)
      WRITE2,2000
70      CONTINUE
      DO 140 I=1,NTAPES
      N=NBAD(I)
      IF(N)140,140,80
80      WRITE2,2030
      DO 130 K=1,N
      READ1,1020,NBLOCK(I,K)
      NADD1=0
      NADD2=0
      NB=NBLOCK(I,K)
      IF(NB-1000)100,90,90
90      NADD2=512
      NB=NB-1000
100     IF(NB-100)120,110,110
110     NADD1=(NB/100)*64
      NB=NB-(NB/100)*100
120     KB=NB/10
      NBLOCK(I,K)=8*KB+(NB-10*KB)+NADD1+NADD2
C      NBLOCK IS THE NUMBER OF THE BAD BLOCK
      WRITE 2,2000
130     CONTINUE
140     CONTINUE
      C=10./4096.
      C0005=0.0005
      N3=1
      JK=100
      KK=1
      K5=1
      KX=0
      KX1=0
      KX2=0
      KOUNT=0
      ICOUNT=0
      JJ=100
      J0=1
      IPERP=0
      IS=0
      IP=0

```

```

      ITT=0
      NTM=4
      N=1
      INS=0
      DO 150 JX=1,105
      DATA1(JX)=4096
      DATA2(JX)=4096
      DATA3(JX)=4096
150   CONTINUE
      DO 710 I=1,NTAPES
      JSTART=5
160   CONTINUE
      DO 695 J=JSTART,576
      IUNIT=5
      N=NBAD(I)
      IF(N)190,190,170
170   DO 180 K=1,N
      IF(NBLOCK(I,K)-J)180,695,180
C      CHECK TO SEE IF THIS BLOCK IS BAD
180   CONTINUE
190   CALL FETCH(IUNIT,J,M,IE,IH)
      IF(M)200,200,695
200   CONTINUE
C      LOW SPEED MULTIPLEXER COMPUTATIONS
      K1=0
      DO 690 K=1,64
      K1=K1+1
      IDAT=KBFA(K1)
S      OCTAL
S      LAC (777777
S      SAD IDAT.
S      JMP .780
S      LAC (777000
S      SAD IDAT.
S      JMP .700
S      LAC IDAT.
S      AND (77
S      DAC MAJOR.
S      LAC IDAT.
S      RCL
S      CML
S      RAR
S      RTR
S      RTR
S      RTR
S      AND (7777
S      DAC MDATA1.
S      DECIMAL
      K1=K1+1
      IDAT=KBFA(K1)

```

```
S      OCTAL
S      AND (7
S      DAC IPER. /IPER=PERIOD
S      LAC JK.
S      SAD IPER. /SKIP WHEN THE PERIOD CHANGES
S      JMP .205
S      JMP .585
S      DECIMAL
205    CONTINUE
S      OCTAL
S      LAC IDAT.
S      RCL
S      CML
S      RAR
S      RTR
S      RTR
S      RTR
S      AND (7777
S      DAC MDATA2.
S      LAC IDAT.
S      RTR
S      RAR
S      AND (7
S      DAC ICL. /ICL=SCAN CLASS
S      DECIMAL
C      IF(ICL-KK)210,220,210
210    CHECK TO SEE IF SCAN CLASS HAS CHANGED
      K1=K1+2
      GO TO 690
220    CONTINUE
      K1=K1+1
      IDAT=KBFA(K1)
S      OCTAL
S      AND (77
S      DAC ISCAN. /NUMBER OF TIMES POINT HAS BEEN SCANNED
S      LAC IDAT.
S      RCL
S      CML
S      RAR
S      RTR
S      RTR
S      RTR
S      AND (7777
S      DAC MDATA3.
S      DECIMAL
      K1=K1+1
      IDAT=KBFA(K1)
S      OCTAL
S      AND (1
S      DAC IT0.
```



```

S      LAC IDAT.
S      RCL
S      CML
S      RAR
S      RTR
S      RTR
S      RTR
S      AND (7777
S      DAC MDATA4.
S      DECIMAL
225    IF(MAJOR-JPT(J0))230,240,230
230    IF(MAJOR-JPT(1))690,530,690
240    IF(JJ-NK)250,250,530
250    DATA(1,JJ)=MDATA1*C+C0005
      DATA(2,JJ)=MDATA2*C+C0005
      DATA(3,JJ)=MDATA3*C+C0005
      DATA(4,JJ)=MDATA4*C+C0005
      J0=J0+1
      IF(IT0)260,280,260
260    IF(ITT)280,270,280
270    ITT=5
      T0=T
280    IF(NK-5)290,290,300
290    IF(JJ-NK)350,305,360
300    IF(JJ-5)350,305,360
C      WRITE ON PAGE 1
305    IF(KOUNT-45)310,340,340
310    IF(ICOUNT-5)330,320,320
320    WRITE9,2000
      ICOUNT=0
      KOUNT=KOUNT+2
330    WRITE9,3002,T,((DATA(I8,I9),I8=1,4),I9=1,MK)
      KOUNT=KOUNT+1
      ICOUNT=ICOUNT+1
      GO TO 350
340    KOUNT=52-KOUNT
S      LAS
S      AND (1
S      SAD (1
S      JMP .785
      CALL LFEED (KOUNT)
      WRITE9,3005,K5
      WRITE9,3000,IP,ISS,DT
      WRITE9,3001,((IPOT(IB,IC),IC=1,8),IB=1,MK)
      K5=K5+1
      KOUNT=5
      ICOUNT=0
      GO TO 330
350    JJ=JJ+1
      GO TO 690
360    IF(NK-10)370,370,380
370    IF(JJ-NK)350,390,420

```

```
380      IF(JJ-10)350,390,420
390      CONTINUE
C      WRITE ON TAPE 1
        IUNIT=1
        KX=KX+1
        DATA1(KX)=T
        DO 410 IX=6,10
        DO 400 JX=1,4
        KX=KX+1
        DATA1(KX)=DATA(JX,IX)
400      CONTINUE
410      CONTINUE
        IF(KX-105)350,510,510
420      IF(NK-15)430,430,440
430      IF(JJ-NK)350,450,470
440      IF(JJ-15)350,450,470
450      CONTINUE
C      WRITE ON TAPE 2
        IUNIT=2
        KX1=KX1+1
        DATA2(KX1)=T
        DO 460 IX=11,15
        DO 460 JX=1,4
        KX1=KX1+1
        DATA2(KX1)=DATA(JX,IX)
460      CONTINUE
        IF(KX1-105)350,520,520
470      IF(NK-20)480,480,490
480      IF(JJ-NK)350,500,350
490      IF(JJ-20)350,500,350
500      CONTINUE
C      WRITE ON TAPE 3
        IUNIT=3
        KX2=KX2+1
        DATA3(KX2)=T
        DO 505 IX=16,20
        DO 505 JX=1,4
        KX2=KX2+1
        DATA3(KX2)=DATA(JX,IX)
505      CONTINUE
        IF(KX2-105)350,525,525
510      NTM=NTM+1
        KX=0
515      CALL WTAPE(IUNIT,NTM)
        GO TO 350
520      CALL WTAPE(IUNIT,NTM)
        KX1=0
        GO TO 350
525      CALL WTAPE(IUNIT,NTM)
        KX2=0
```

```
GO TO 350
530 JJ=1
    J0=1
    IF(ISCAN-II(ICL))540,535,535
535 II(ICL)=ISCAN
    XI1=II(ICL)+II0(ICL)
    GO TO 545
540 II0(ICL)=II0(ICL)+64
    GO TO 535
545 IPER=IPER+1
    T=XI1*SCTM(IPER,ICL)
    IF(IPER-1)555,555,550
550 T=T+PERT(IPER-1)
555 T=T/1000.
    IPER=IPER-1
    IF(IS)250,560,250
560 IS=1
    IP=IPER
    ISS=KK
    JJ=1
C WRITE HEADING FOR PAGE 1
    ICOUNT=0
    KOUNT=52-KOUNT
S LAS
S AND (1
S SAD (1
S JMP .785
    CALL LFEED(KOUNT)
    WRITE 9,3005,K5
    DT=SCTM(IPER+1,ICL)
    WRITE 9,3000,IP,ISS,DT
    DO 565 IB=1,NK
    IPOT(IB,1)=IPT(IB)
    IPOT(IB,2)=IPO(1)
    IPOT(IB,3)=IPT(IB)
    IPOT(IB,4)=IPO(2)
    IPOT(IB,5)=IPT(IB)
    IPOT(IB,6)=IPO(3)
    IPOT(IB,7)=IPT(IB)
    IPOT(IB,8)=IPO(2)
    IPOT(IB,5)=IPT(IB)
    IPOT(IB,6)=IPO(3)
    IPOT(IB,7)=IPT(IB)
    IPOT(IB,8)=IPO(4)
565 CONTINUE
    IF(NK-5)575,575,570
570 MK=5
    GO TO 580
575 MK=NK
580 CONTINUE
```

```

WRITE9,3001,((IPOT(IB,IC),IC=1,8),IB=1,MK)
K5=K5+1
KOUNT=5
GO TO 250
585 IF(IPERP-IPER)590,590,210
590 DO 595 IQ=1,7
      II0(IQ)=0
      II(IQ)=0
595 CONTINUE
      K3=1
      IF(JK-100)600,620,600
C      WRITE PAGES 2,3,AND4,AND TO
600 K4=0
      GO TO 715
605 IF(JK)610,615,610
610 IF(KK-6)670,620,670
615 IPERP=IPER
620 N=MASK(IPERP+1)
      NK=0
      DO 650 NN=1,48
      N1=ICLPER(NN)
S      AND N.
S      DAC N1.
      IF(N1)625,650,625
625 IF(IPERP-1)645,635,630
630 KNT=IPERP*3-4
S      CMA
S      DAC KNT.
S      LAC N1.
S      RAR
S      ISZ KNT.
S      JMP .-2
S      DAC N1.
635 CONTINUE
      IF(N1-KK)650,645,650
645 NK=NK+1
      IPT(NK)=NN-1
      JPT(NK)=NN-1
      NKJ=IPT(NK)/8
      IPT(NK)=2*NKJ+IPT(NK)
650 CONTINUE
      IF(NK)670,670,655
655 IF(K3)660,665,665
660 K3=1
      GO TO 160
C      REINITIALIZE FOR THE NEXT PERIOD
665 JK=IPER
      IPERP=IPER
      KK=1
      IS=0

```

```
JJ=100
GO TO 205
670 KK=KK+1
IF(KK-7)675,680,680
C REWIND THE DATA TAPE
C PERIOD IS STILL THE SAME
675 IS=0
JJ=100
K3=-1
GO TO 620
680 KK=1
IF(ICL-8)685,785,705
685 JK=0
JSTART=J
GO TO 585
690 CONTINUE
695 CONTINUE
700 ICL=9
GO TO 585
C TELL THE OPERATOR TO CHANGE TAPES
705 WRITE 2,3004
JSTART=1
S HLT
710 CONTINUE
715 NTM=NTM+1
N=1
DO 770 NM=1,3
ICOUNT=0
N=N+5
NK1=N+4
IF(NK1-NK)725,720,720
720 NK1=NK
725 IF(NK-N)775,730,730
730 KOUNT=52-KOUNT
S LAS
S AND (1
S SAD (1
S JMP .785
CALL LFEED(KOUNT)
WRITE9,3005,K5
K5=K5+1
WRITE9,3000,IP,ISS,DT
WRITE 9,3001,((IPOT(IB,IC),IC=1,8),IB=N,NK1)
KOUNT=5
IUNIT=NM
CALL WTAPE(IUNIT,NTM)
DO 765 MN=5,NTM
IM1=1
CALL FETCH(IUNIT,MN,M,IE,IH)
DO 760 NI=1,5
```

-16-

```

735      IF(DATA1(IM1)) 770, 735, 735
        CONTINUE
740      IF(KOUNT-45) 750, 745, 745
        ITN=(NK1-5*NM)*4+IM1
        WRITE9, 3002, (DATA1(K3), K3=IM1, ITN)
        KOUNT=KOUNT+1
        ICOUNT=ICOUNT+1
        IM1=IM1+21
        GO TO 760
745      KOUNT=52-KOUNT
S        LAS
S        AND (1
S        SAD (1
S        JMP .785
        CALL LFEED(KOUNT)
        WRITE9, 3005, K5
        K5=K5+1
        WRITE9, 3000, IP, ISS, DT
        WRITE 9, 3001, ((IPOT(IB, IC), IC=1, 8), IB=N, NK1)
        KOUNT=5
        ICOUNT=0
        GO TO 740
750      IF(ICOUNT-5) 740, 755, 755
755      WRITE 9, 2000
        ICOUNT=0
        KOUNT=KOUNT+2
        GO TO 740
760      CONTINUE
765      CONTINUE
770      CONTINUE
        NTM=4
775      IUNIT=5
        GO TO 605
780      ICL=8
        GO TO 585
C        WRITE THE LAST THREE PAGES AND TO
S        OCTAL
SMSK,    777777
S        000007
S        000070
S        000700
S        007000
S        070000
S        700000
SMSK1,   000001
S        000002
S        000003
S        000004
S        000005
S        000006

```

```
S      000007
S      DECIMAL
785    CONTINUE
      WRITE(9,3003,T0)
2010    FORMAT(14HNO. OF TAPES =)
2020    FORMAT(17HNO. OF BAD BLOCKS)
2000    FORMAT(1H //)
2030    FORMAT(13HBAD BLOCK NOS)
1020    FORMAT(I4)
3000    FORMAT(1H@,8HPERIOD= ,I2/1H@,12HSCAN CLASS= ,I2/1H@
$      ,8HDELTA T=,F13.0,6H MSEC )
3001    FORMAT(4X,1H†,8HT (SEC),3X,I3,I2,I5,I2,I5,I2,I5,
$      I2,I6,I2,I5,I2,I5,I2,I5,I2,I6,I2,I5,I2,I5,I2,I5,
$      I2,I6,I2,I5,I2,I5,I2,I5,I2,I6,I2,I5,I2,I5,I2,I5,I2)
3002    FORMAT(F13.3,4F7.3,F8.3,3F7.3,F8.3,3F7.3,F8.3,3F7.3,
$      F8.3,3F7.3)
3003    FORMAT(4HT0= ,F13.3)
3004    FORMAT(15HMOUNT NEXT TAPE)
3005    FORMAT(1H@,1H@,5HPAGE ,I3)
      END
```

```

/TAPE RETRIEVAL PROGRAM FOR THE MULTIPLEXER
/CALL ARGUMENTS.
/IUNIT-THE TAPE UNIT ON WHICH HAS THE MULTIPLEXER
/DAIS MOUNTED, ALWAYS UNIT 5
/J-THE BLOCK NO. TO BE READ OFF TAPE I
/RETURN ARGUMENTS
/M-INDICATES THAT THE DATA JUST READ IN IS FROM
/LOW SPEED MULTIPLEXER OR HIGH SPEED MULTIPLEXER.
/M=0 INDICATES LOW SPEED MULTIPLEXER
/M=1 INDICATES HIGH SPEED MULTIPLEXER
/IE=1 INDICATES THE END OF THE RUN. IE=0 MEANS
/MORE DATA WILL BE READ IN, IE=1 MEANS NO MORE
/DAIS LEFT.
/IH=1 INDICATES THERE WAS A HALT DURING THE DATA
/COLLECTION AND THERE WON'T BE ANYMORE DATA
/ON TAPE I. CHANGE TO NEXT SEQUENTIAL TAPE
/AND CONTINUE.
      FIDEC
      EXTERNAL .I01,.I02,.I03,.I04,.I05,.I06,.I07
      OCTAL
      INTERNAL FETCH.
FETCH., JMS GTARG
      JMP .AAA
IUNIT,  0
J,      0
M,      0
IE,     0
IH,     0
.AAA,   JMP M10
IBUF,   JMS CALSB
      LAW 1
      .BFF
M10,    NOP
      LAC BFF
      DAC IBUF+2
      LFM
      CLA          /DESELECT ALL TAPES
      MMLC
      MMSE
      JMS OLC
      CLL
      CLA
      LAC I IUNIT      /LOAD AC WITH TAPE UNIT
      DAC SU
      RTL          RTL          RTL          RTL
      RTL          RTL          /ROTATE LEFT 12
      ADD I J        /ADD BLOCK NO. TO BE FOUND
      DAC SALL
      LAC SU
      SAD C5

```



```

        JMP .+3
        LAC COR1
        DAC IBUF+2
        LAC SALL
        JMS MTAPE
        LAC IBUF+2
        LAC 10403
        AND (40
        SAD (00040
        JMP A8
A6,     DZM I M
        NOP
        NOP
        SAD (777000
        JMP A12
        LAC I J /NO-MORE BLOCKS OF DATA WILL FOLLOW
        SAD (1100 /IS THIS THE LAST BLOCK ON TAPE
        NOP
A7,     RETUR /NO-CONTINUE
        HLT
A8,     LAC (1
        DAC I M /M=1-DATA IS FROM HIGH SPEED MULT.
        JMP A6+2
A10,    LAC (1
        DAC I IE /IE=1-END OF RUN-NO MORE DATA
        JMP A7
A12,    LAC (1
        DAC I IH /IH=1 NO MORE DATA ON THIS TAPE
        JMP A7
OLC,    0
        LAC TTC
        DAC COUNS
        ISZ COUNS
        JMP .-1
        JMP I OLC
COUNS,  0
TTC,    754300
SU,     0
BFF,    10400
SALL,   0
COR1,   7776
MTAPE,  0
        MMSE
        AND (1777
        DAC BLOCK
S1,     LAW 41
        MMLC
SEARCH, JMS RED
        AND (1777
        SAD BLOCK

```

```

S2,      JMP FOUND
          CMA
          TAD BLOCK
          SMA
          JMP SEARCH
          LAW 61
          MMLC
          JMS RED
          AND (1777
          CMA
          TAD BLOCK
          TAD MN2
          SPA
          JMP S2
          JMP S1
MN2,     777776
BLOCK,   0
MN1,     LAM -0
CHECK,   0
FOUND,   XCT I MTAPE      /LAC CORE ADDRESS
          TAD MN1
          DAC 10
          LAM -377
          DAC COUNS
          LAW 42
          MMLC
          JMS RED
          DAC CHECK
LOAD,    JMS RED
          DAC I 10
          ADD CHECK
          DAC CHECK
          ISZ COUNS
          JMP LOAD
          MMBF
          JMP --1
          MMRD
          ADD CHECK
          DAC CHECK
          CLA
          MMLC
          MMSE
          JMS OLC
          ISZ MTAPE
          JMP I MTAPE
RED,     0
          MMEF
          SKP
          JMP ERR
          MMDF
```

```

        JMP RED+1
        MMRD
        JMP I RED
ERR,    MMRS
        SAD (150000
        JMP .+2
        JMP ER1
        MMSE
        JMS OLC
        JMP S1
ER1,    WRITE
        JMS .I02
        2
        FOR .ER
        ARX I IUNIT
        ARX BLOCK
        ENDIO
        JMP M10
        OCTAL
.ER,    320016      / (14H
        TEXT
BAD TAPE UNIT #
        020002      / , I2
        120016      / , 14H
        TEXT
BLOCK #
        020004      / I4
        600000      / )
        JMP A7
START

```

/ TAPE WRITE SUBROUTINE---WRITES ON UNIT 1 ,2,OR 3

```

      FIODEC
      OCTAL
      INTERNAL WTAPE.
WTAPE., JMS GTARG
      JMP .AAA
IUNIT,  0
IBLK,   0
.AAA,   NOP
      LAC I IBLK
      DAC BLOCK
      CLA
      MMLC
      MMSE
      JMS DELA
      LAC I IUNIT
      DAC SU
      SAD (1
      LAC COR1
      SAD (2
      LAC COR2
      SAD (3
      LAC COR3
      DAC CORE
      LAC SU
      CLL
      RTR          RTR          RTR          RAR
      TAD (60
      NOP
      MMSE
S2,     MMLC
      MMEF
      JMP .-1
      MMRS
      AND (40000
      SZA
      JMP S1
      LAW 60
      JMP S2
S1,     LAW 41
      MMLC
SEARCH, JMS RED
      AND (1777
      SAD BLOCK
      SKP
      JMP SEARCH
FOUND,  LAC CORE
      DAC 12
      LAM -377
      DAC COUNS

```

```
B3,      CLC
          DAC CHECK
          LAW 43
          MMLC
          LAC I 12
          MMEF
          SKP
          JMP RONG
          MMBF
          JMP DAT
          DAC TX
          LAC CHECK
          CMA
          MMWR
          CLC
          DAC CHECK
          LAC TX

DAT,      MMDF
          JMP B3+1
          MMWR
          ADD CHECK
          DAC CHECK
          ISZ COUNS
          JMP B3
          MMBF
          JMP .-1
          CMA
          MMWR
          MMDF
          JMP .-1
          CLA
          MMLC
          CLC
          LAC CORE
          DAC 12
          LAM -312
          DAC COUNS
          LAC (777777
          DAC I 12
          ISZ COUNS
          JMP .+2
          NOP
          RETUR
          HLT

RED,      0
          MMEF
          SKP
          JMP RONG
          MMDF
```

-24-

	JMP --4
	MMRD
	JMP I RED
RONG,	LAC SEVENO
	HLT
DELA,	0
	LAC TC
	DAC TX
	ISZ TX
	JMP --1
	JMP I DELA
TC,	754000
BLOCK,	0
COUNS,	0
SEVENO,	707070
TX,	0
CORE,	0
MN2,	777776
CHECK,	0
COR1,	7775
COR2,	7453
COR3,	7131
SU,	0
START	

-25-

/LINE FEED SUBROUTINE FOR REPORT

```
      FIODEC
      OCTAL
      INTERNAL LFEEED.
LFEEED., JMS GTARG
      JMP .AAA
KOUNT,  0
.AAA,   NOP
      LAC (1
      DAC IX
.AAM,   NOP
      LAC (001000
      703201
      JMP .-1
      703206
      LAC IX
      ADD (1
      DAC IX
      CMA
      ADD I KOUNT
      ADD (1
      SMA
      JMP .AAM
      RETUR
      HLT
IX,
START 0
```

OPERATING PROCEDURE

The following steps are to be followed when running the Multiplexer Data Retrieval program.

1. Place the DSD magnetic tape on any of the tape units.
2. Place the DSD paper tape in the paper tape reader.
3. Set the Address switches to 17600. Set the Accumulator switches 1-5 to the unit number of the DSD tape. Set the Accumulator switches 6-17 to 140₈. Press the Read-In switch.
4. The computer will halt at location 000231. Set the Address switches to 15700₈. Place the MPGEN2 paper tape in the paper tape reader and press the Read-In switch.
5. Set the Address switches to 100₈ and press the start switch.
6. The computer will again halt at location 000231. Replace the DSD magnetic tape with a good Scratch tape. Place the Input Data tape on unit 5, and place good scratch tapes on the remaining units. There should now be a Data tape on unit 5 and scratch tapes on units 1, 2, and 3.
7. Press the Continue switch.
8. A message will now be printed asking for the number of input tapes. Type the appropriate number and line-feed, carriage return.
9. A message will now ask for the number of bad blocks on tape (I). Type the appropriate number and line-feed, carriage return. If there are no bad blocks type a 0 and line-feed, carriage return.

-27-

10. A message will now ask for the bad block numbers on tape (I). Type the number and line-feed, carriage return. This message will not print out if a Ø was typed in step 9.
11. When the current tape has finished, a message will print out asking that the next tape be mounted on unit 5. After changing tapes, press the Continue switch.
12. When all of the tapes have been finished, the program will write the time of rupture (TØ) and end.
13. The program can be halted at the end of a page of output by placing AC switch 17 in the UP position.

NOMENCLATURE

The following list describes frequently used program symbols and their functions.

1. ICLPER = The storage area for the CLPER table, used in determining which major points have been checked and the period and scan class in which they were checked.
2. PERT = The storage area for the PERT table, used in computing the time at which a major point was scanned.
3. SCTM = The storage area for the SCTM table, also used in computing the time of scan.
4. NBAD = The variable set equal to the number of bad blocks on a given tape.
5. NBLOCK = The variable set equal to the bad block numbers on a given tape.
6. IIØ and II = Variables used in computing the time of scan.
7. KOUNT = Variable used to determine when the typewriter needs to skip to a new page for output.
8. ICOUNT = Variable used to determine when the typewriter needs to skip the output page two lines.
9. MAJOR = Variable set equal to the current major point.
10. MDATA1 = Variable set equal to the first subpoint of the current major point.
11. MDATA2 = Variable set equal to the second subpoint of the current major point.

-29-

- 12. MDATA3 = Variable set equal to the third subpoint of the major point.
- 13. MDATA4 = Variable set equal to the fourth subpoint of the current major point.
- 14. IPER = Variable set equal to the current period.
- 15. ICL = Variable set equal to the current scan class.
- 16. ISCAN = Variable set equal to the number of times the current major point was scanned.
- 17. ITØ = Constant set to 1 at time of rupture.
- 18. T = Variable set equal to the computed time of scan.

REFERENCES

1. N. P. Wilburn. Multiplexer Codes for the PDP-7, Document BNWL-CC-796, August 24, 1966.
2. G. J. Rogers. Program for Containment Systems Experiment, USAEC Report HW-83607, September, 1964.

DISTRIBUTION

Number
of Copies
40

Battelle-Northwest

R. T. Alleman
J. M. Atwood
J. M. Batch
F. P. Brauer
J. C. Campbell
L. D. Coffin (3)
B. L. Combs
L. F. Coleman
R. E. Connally
R. Y. Dean
J. L. Deichman
B. H. Duane
G. E. Driver
R. K. Hilliard
F. E. Holt
J. G. Kaye
C. E. Linderoth
W. L. Nicholson
J. D. McCormack
H. J. Rieck
G. J. Rogers
J. T. Russell
S. J. Simmons
R. W. Walker
W. R. Weissenberger (8)
E. L. Wells
N. P. Wilburn
M. E. Witherspoon
Central Files (3)

43

AEC - Washington, D.C.

Advisory Committee on Reactor Safeguards
R. F. Fraley (17)
Division of Reactor Development and Technology
R. L. Ednie
J. W. Crawford
M. J. Whitman
E. E. Sinclair
J. A. Lieberman (5)
Division of Compliance
L. Kornblith, Jr.
Division of Operational Safety
H. Gilbert

Number
of Copies

43	<u>AEC - Washington, D.C., (cont'd)</u> Division of Reactor Standards E. G. Case (10) M. Bolotsky A. B. Holt Naval Reactors Branch, RDT R. S. Brodsky (2) Division of Production G. B. Pleat
1	<u>AEC - Denver, Colorado</u> Division of Compliance, Region IV J. W. Flora
1	<u>AEC - Whiteshell Branch, Manitoba, Canada</u> USAEC Technical Representative G. D. Boyer
5	<u>Argonne National Laboratory</u> R. C. Vogel P. Lottes C. E. Dickerman R. O. Ivins S. Fistedis
1	<u>Babcock and Wilcox Company</u> R. Wascher
6	<u>Battelle Memorial Institute</u> D. N. Sunderman (2) D. L. Morrison S. Paprocki (2) A. R. Duffy
1	<u>Brookhaven National Laboratory</u> A. W. Castleman
1	<u>Chicago Operations Office</u> D. M. Gardiner

Number
of Copies

1	<u>Combustion Engineering, Inc. - Windsor, Conn.</u> <u>Nuclear Division</u> M. F. Valerino
2	<u>General Electric Company - San Jose, Calif.</u> G. E. Wade S. Vandenberg
3	<u>General Electric Corp. - San Jose, Calif.</u> <u>Atomic Products Division</u> <u>Atomic Power Equipment Department</u> P. Brau M. Siegler W. A. Sutherland
1	<u>Harvard Air Cleaning Laboratory</u> <u>Harvard University - Boston, Mass.</u>
2	<u>Idaho Operations Office - Idaho Falls, Idaho</u> D. Williams (2)
2	<u>IIT Research Institute - Chicago, Illinois</u> T. A. Zaker E. Gallagher
1	<u>Los Alamos Scientific Laboratory</u> J. H. Russel, K Div.
1	<u>MPR Associates, Inc.</u> <u>815 Connecticut Avenue, N.W.</u> <u>Washington, D.C.</u> T. Rockwell, III
1	<u>National Bureau of Standards - Washington, D.C.</u> C. Muehlhause
1	<u>Naval Ordnance Laboratory, White Oak,</u> <u>Silver Spring, Maryland</u> J. Proctor

-34-

Number
of Copies

1	<u>North Carolina State University</u> <u>Raleigh, North Carolina</u> <u>Department of Mechanical Engineering</u> M. N. Ozisik
1	<u>Nuclear Fuels Services - West Valley, N.Y.</u> R. P. Wischow
7	<u>Oak Ridge National Laboratory</u> <u>Nuclear Safety Information Center</u> J. Buchanan Chemical Technology Division D. Ferguson R. Blanco HTGR Safety Program Office W. B. Cottrell (4)
1	<u>Oak Ridge Operations Office</u> W. L. Smalley
4	<u>Pacific Northwest Laboratories</u> J. C. Spanner R. Nightingale (2) L. Schwendiman
9	<u>Phillips Petroleum Co. - Idaho Falls, Idaho</u> C. Haire S. Forbes (2) T. R. Wilson (2) N. K. Sowards Water Reactor Safety Program F. Schroeder (3)
2	<u>Richland Operations Office</u> C. Robinson A. S. Waterhouse
1	<u>San Francisco Operations Office - Berkeley, Calif.</u> J. B. Radcliffe

Number
of Copies

1

Savannah River Laboratories
E. I. duPont deNemours and Co.
Aiken, South Carolina 29802

A. H. Peters

2

TRW, Inc., One Space Park, Redondo Beach, Calif.
TRW Systems Group

D. B. Langmuir
S. M. Zivi

2

University of California - Berkeley, Calif.
Institute of Engineering Research

H. A. Johnson
V. E. Schrock

3

Westinghouse Electric Corp. - Pittsburgh, Pa.
Atomic Power Division

R. A. Wieseman
D. Fletcher
E. Beckjord