WSRC-RP-89-1223

TRIMPWR - A POST PROCESSOR FOR TRIMHX (U)

NOVEMBER 1989

Patent Status

This internal management report is being transmitted without DOE patent clearance, and no further dissemination or publication shall be made of the report without prior approval of the DOE-SR patent counsel.

C. E. Apperson Jr.
DERIVATIVE CLASSIFIER

WESTINGHOUSE SAVANNAH RIVER COMPANY
SAVANNAH RIVER LABORATORY
AIKEN, SC 29808

PREPARED FOR THE U. S. DEPARTMENT OF ENERGY UNDER CONTRACT DE-AC09 88SR18035

DISCLAIMER

294558

NRTSC

NUCLEAR REACTOR TECHNOLOGY
AND SCIENTIFIC COMPUTATIONS

KEY WORDS: TRIMHX
TRANSIENT
SAFETY ANALYSIS
REACTOR
EDITOR

RETENTION PERIOD: LIFETIME

TRIMPWR - A POST PROCESSOR FOR TRIMHX (U)

By

E. F. Trumble

ISSUED: November 1989

SRL SAVANNAH RIVER LABORATORY, AIKEN, SC 29808
Westinghouse Savannah River Company
Prepared for the U. S. Department of Energy under Contract DE-AC09-88SR18035

PROJECT:

PROJECT:

DOCUMENT: WSRC-RP-89-1223

TITLE: TRIMPWR - A POST PROCESSOR FOR TRIMHX (U)

TASK: ASSEMBLY POWER CALCULATIONS FOR LIMITS ANALYSIS

APPROVALS

M. R. Buckner

DATE: 2-6-90

M. R. BUCKNER, MANAGER
SCIENTIFIC COMPUTATIONS SECTION

C. E. Apperson, Jr.

DATE: 2-5-90

C. E. APPERSON, MANAGER
REACTOR PHYSICS GROUP

Darin L. George

DATE: 11/9/89

D. L. GEORGE, TECHNICAL REVIEWER

INTRODUCTION:

The TRIMPWR code has been developed as a post processor for TRIMHX (transient 3D diffusion code) in support of the reactor limits program. TRIMPWR is designed to produce JOSHUA files containing: 1) core power as a function of time, 2) assembly power by hex as a function of time, 3) assembly power post peaking as a function of time, and 4) axial power shapes for each assembly as a function of time (formatted for use by the FLOWTRAN code) from the output of a TRIMHX run.

In an attempt to simplify the reactor limits process by reducing the number of assemblies which must be run through FLOWTRAN, TRIMPWR also sorts the assemblies by the product of the power post peaking and the maximum normalized axial power density for each assembly. This follows from the assumption that those assemblies having the maximum value of this product will have the most restrictive limits.

DISCUSSION:

The TRIMPWR code was developed to aid in the evaluation of TRIMHX results for the modeling of a safety rod scram. In order to use the TRIMPWR code, TRIMHX must be run with the "INPUT.REX.?case.EXECUTE. USERMOD" record. An example of this record and the module it points to are shown in Figures 1 and 2. The PWREDIT module allows TRIMHX to write assembly powers by axial level to unique files instead of overwriting the file at each timestep.

Figure 3 lists sample JCL used to execute the TRIMPWR code. Due to uncertainties in the final methodology which will be used for limits, the source code currently resides in the T9379.EDIT.LOAD module. If the TRIMHX methodology is accepted, TRIMPWR will be placed under configuration management.

TRIMPWR was designed to be run on TRIMHX output with a minimum amount of user input. The TRIMHX case name, number of axial layers and number of timesteps in the problem are the only input (Fig. A.1). TRIMPWR does an internal calculation of the number of axial levels and the number of timesteps and compares them to the user input for consistency. If the number of axial layers does not agree with the user's assumption, a message is written and the number of layers is set equal to the value used in TRIMHX. Since there may be times when the user does not wish output from all timesteps, if an inconsistency is found between

the entered and calculated number of timesteps, a message is written and the code uses the value supplied by the user.

Definitions of some of the calculational terms follows:

1) Power post peaking (PPP),
   $PPP^i(t) = [Passy^i(t)/Passy^i(0)] / [Pcore(t)/Pcore(0)]$ for assembly i, where the powers have been normalized

2) Normalized axial power density (NAPD),
   NAPD = power/cm at a particular axial layer normalized such that the sum of NAPD for an assembly is equal to the number of axial layers.

Appendix A contains the input and output templates used in executing TRIMPWR; these templates can be found in the SROD9379 dataset. Appendix B contains the source listing for the TRIMPWR code; Appendix C contains a brief programmers manual (Fig. C.1) and a simplified flowchart (Fig. C.2). Appendix D contains example output from the code.

The current dimensional limitations imposed in TRIMPWR are 889 assembly positions per axial layer, 12 axial layers, and 25 timesteps. These limits were set by array size limitations. The code currently prints only the first 433 values of the product of power post peaking and maximum axial power density. This number insures that all 432 fueled positions in a Mark 22 charge are listed, without listing several pages of unnecessary zeros. If this code is used for analysis of a Mark 16-31 charge and the user is interested in all fueled positions, this value will need to be increased.

QUALITY ASSURANCE

This project falls under Task number 89-029-1, Assembly Power Calculations for Limits Analysis. The sample power transient calculated in TRIMPWR and written to the file SVNR9379.TEMP.RIDS.?case.CPOWER was compared to that generated internally within TRIMHX. This comparison showed that the TRIMPWR core powers are within 0.05% of those generated in TRIMHX. The power/assembly calculated by TRIMPWR and written to SVNR9379.TEMP.RIDS.?case.APOWER was compared to the output of the XGRIM3D code which was run on the TRIMHX base case. The TRIMPWR power/assembly compared to within 0.03% of the numbers generated by XGRIM3D.

Figure 1 INPUT.REX.?case.EXECUTE.USERMOD Record

```
SROD9379.INPUT.REX.MK223DA.EXECUTE.USERMOD

  THIS RECORD IS USED TO TELL REX3T TO RUN THE SPECIFIED
  MODULE AT THE END OF EACH TRANSIENT TIMESTEP.

  REX3T PASSES TO THE MODULE THE FOLLOWING ARGUMENT LIST
     TDSN    = TEMPORARY DATASET NAME          REAL*8
     CASE    = CASE NAME FOR THIS PROBLEM      REAL*8
     TIME    = CURRENT PROBLEM TIME            REAL*4
     ITSTEP  = TIME STEP NUMBER                INTEGER

  THIS RECORD ALSO ALLOWS THE USER TO SPECIFY AN ERROR FLAG.
  THIS FLAG DETERMINES IF AN ERROR IN THE USERMOD SHOULD
  CAUSE TERMINATION OF THE GRASS PROBLEM.



SROD9379.INPUT.REX.MK223DA.EXECUTE.USERMOD

ENTER NAME OF THE MODULE TO BE EXECUTED :PWREDIT

ENTER ERROR FLAG :0     IF 0 DO NOT TERMINATE GRASS PROBLEM
                        IF 1 TERMINATE GRASS PROBLEM

-

NOT USED :0
NOT USED :0
```

Figure 2 Module PWREDIT

```
     PROJECT:  T9379                MEMBER:  POWREDIT                DATE: 89/08/14
     GROUP:    EDIT                 LEVEL:   01.11                   TIME: 13:40
     TYPE:     DATA                 USERID:  T9379                   PAGE: 01 OF 01

START                                                                                  MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  FLAGS
                                                                                00010009   *
   8          MODULE MAIN(TDSN,CASE,TIME,ITSTEP)                             00020000
   1  C       SUBROUTINE PWREDT                                              00030000
   1  C                                                                      00040001   *
   1  C       THIS SUBROUTINE WILL BE CALLED ONCE PER TIMESTEP DURING A TRIMHX 00040001  *
   1  C       RUN.  IT IS EXECUTED WITH THE INPUT.REX.CASE?.EXECUTE.USERMOD  00050001   *
   1  C       RECORD.  IT IS DESIGNED TO READ THE TEMP.RIDS.CASE?.RPWR RECORD 00060001  *
   1  C       GENERATED AT EACH TIMESTEP AND COPY IT TO TEMP.RIDS.CASE?.RPWRO. 00070001 *
   1  C       TIMESTEP? RECORD TO AVOID THE DATA BEING OVERWRITTEN.           00080001   *
   1  C                                                                      00090001   *
   1  C                                                                      00100000
   1  C                                                                      00110001   *
   8          REAL*8 TDSN,CASE                                               00120011   **
   8          DIMENSION RPWR(18000)                                          00130001   *
   1  C                                                                      00140001   *
   8          DO 10 J=1,18000                                                00150001   *
   8          RPWR(J) = 0.0                                                  00160001   *
   2    10    CONTINUE                                                       00170001   *
   1  C                                                                      00180011   **
   6          READ('TEMP','RIDS',CASE,'RPWR','0') NWORD,(RPWR(I),I=1,NWORD)  00200011   **
   8          WRITE('TEMP','RIDS',CASE,'RPWRO',ITSTEP) NWORD,(RPWR(I),I=1,   00210011   **
   6        1 NWORD)                                                         00220010   *
   1  4000    STOP                                                           00230010   *
   8          GO TO 4000                                                     00240000
   8          END
```

Figure 3: JCL for TRIMPWR Module

```
//SVNR9379 JOB (8632-EC,T9379,L034,02,60),'TR 3MIS/4LATE SR',
//   MSGLEVEL=(1,1),CLASS=O,MSGCLASS=T,NOTIFY=T9379
//STEP1 EXEC JOSHUA,DISP=N,RLSE=Y,USER=9379,MEMBER=TRIMPWR,
//   GTIME=02,BUFNO=20,GOSIZE=2800K,SSPACE=30,
//   JOB=SVNR9379,TIMER=OF,SPACE=30,PDS1='T9379.EDIT.LOAD',
//   GOUT='SYSOUT=*'
//GO.JOSIN DD *
DATASET=SVNR9379,AROD9379,SROD9379,STD
SCRATCH=SVNR9379
```

APPENDIX A
Input and Output Templates for TRIMPWR

FIGURE A.1

SROD9379.TEMPLATE.INPUT.TRIMPWR.JOB.?

```
CASE NAME FROM TRIMHX RUNS =  :AAAAAAAAAAA
NUMBER OF AXIAL LAYERS IN TRIMHX RUN =  :IIII
NUMBER OF TIMESTEPSS FROM TRIMHX RUN =  :IIII
```

FIGURE A.2

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.CPOWER

NUMBER OF TIMESTEPS = :III

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.CPOWER

TIMESTEP	TIME(SEC)	REL. CORE POWER
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF
:III	:FF.FF	:FF.FFFF

FIGURE A.3

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.APOWER.?

FOR THE TIME = :FF.FF SEC

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.APOWER.?

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF
:III	:FF.FFFF	:III	:FF.FFFF	:III	:FF.FFFF

FIGURE A.4

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.SORTPRD.?

FOR TIME = :FF.FF SEC

SROD9379.TEMPLATE.TEMP.RIDS.?.OUTPUT.SORTPRD.?

HEX	PROD	PPP	MAX AX POWER DENSITY
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF
:III	:FFF.FFFF	:FFF.FFFF	:FFF.FFFF

LASTPAGE 3

FIGURE A.5

SROD9379.TEMPLATE.TRIMPWR.?.OUTPUT.FLOWFILE.?

THE NUMBER OF AXIAL LEVELS = :III
THE NUMBER OF TIMESTEPS = :III

SROD9379.TEMPLATE.TRIMPWR.?.OUTPUT.FLOWFILE.?

THE TIMES ARE AS FOLLOWS

:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF
:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF	:FF.FFFF

SROD9379.TEMPLATE.TRIMPWR.?.OUTPUT.FLOWFILE.?

STEADY STATE CONDITIONS

LEVEL	REL. POWER
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF

SROD9379.TEMPLATE.TRIMPWR.?.OUTPUT.FLOWFILE.?

TRANSIENT POWERS

:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF	:F.FFFFF
:F.FFFFF	:F.FFFFF			

APPENDIX B
Source listing for TRIMPWR

PROJECT: T9379 MEMBER: TRIMPWR DATE: 89/11/08
GROUP: EDIT LEVEL: 01.88 TIME: 08:48
TYPE: DATA USERID: T9379 PAGE: 01 OF 06

```
TART                                                                                    MOD
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  FLAGS
                                                                                00010000
  8          MODULE MAIN                                                        00020001  *
  1  C       THIS ROUTINE WILL TAKE TRIMHX POWER RESULTS AT EACH TIME           00030001  *
  1  C       STEP AND COMPUTE ASSEMBLY POWERS, CORE POWERS AND RELATIVE         00040001  *
  1  C       AXIAL POWER DENSITIES.  THE POWER POST PEAKING WILL ALSO BE        00050001  *
  1  C       CALCULATED AND THE ASSEMBLIES WILL BE ORDERED BY THE PRODUCT OF    00060001  *
  1  C       THE POWER POST PEAKING AND THE MAXIMUM NORMILIZED POWER DENSITY    00070004  *
  1  C                                                                          00080020  *
  1  C       CURRENT DIMENSIONAL LIMITS ARE:  889 RADIAL POSITIONS, 12          00090020  *
  1  C       AXIAL LAYERS, AND 25 TIMESTEPS.                                    00100001  *
  1  C                                                                          00110052  *
  8          REAL POW(889,12,25),PCORE(25),A(11000),FTIME(20),PRCORE(25)        00120021  *
  8          REAL PPP(889,25),SEG(25),SEGAVG(25),W(25),TOTPOW(889,25),          00130020  *
  6        1 TIME(50),PASSY(889,25),AXMAX(889,25),MAX,KZ,PROD(889,25)           00140078  *
  8          REAL RELVL(12),RELVLS(12),REPOW(889,12,25),PPASSY(889,25)          00150039  *
  8          REAL*8 CASE,RXN1,RXN2,RXN3,WBCL,WBCR                               00160042  *
  8          INTEGER STEP(25),HEX(889),G1                                       00170039  *
  8          INTEGER*2 NDTM(20),EDFQ(20),NAX,NTIME,RXTYP,NPT(25)                00180002  *
  1  C       ---------------------------------------------------------          00190001  *
  1  C       READ THE JOB RECORD TO DETERMINE CASE,NUMBER OF AXIAL LAYERS       00200001  *
  1  C       AND NUMBER OF TIMESTEPS                                            00210002  *
  1  C       ---------------------------------------------------------          00220001  *
  8          READ('INPUT','TRIMPWR','JOB',$JOB) CASE,NAX,NTIME                  00230001  *
  1  C                                                                          00240030  *
  8          WRITE(6,1001)CASE,NAX,NTIME                                        00250030  *
  2   1001   FORMAT(1X,'CASE',A8,2X,'AX ',I3,' NTIME ',I3)                      00260084  *
  1  C       ---------------------------------------------------------          00270015  *
  1  C       DETERMINE TIME VALUES AT EACH TIMESTEP BY READING THE INPUT.       00280015  *
  1  C       REX.CASE.TRANS RECORD                                              00281084  *
  1  C       ---------------------------------------------------------          00300027  *
  8          READ('INPUT','REX',CASE,'TRANS') G1,NTIMDX,(NDTM(L),FTIME(L),      00310027  *
  6        1 EDFQ(L),L=1,NTIMDX)                                                00330084  *
  1  C       ---------------------------------------------------------          00648215  *
  1  C       DETERMINE TOTAL NUMBER OF TIMESTEPS AND CALCULATE THE TIMES        00648415  *
  1  C       CORRESPONDING TO EACH TIMESTEP                                     00648584  *
  1  C       ---------------------------------------------------------          00648628  *
  8          WRITE(6,1160)G1,NTIMDX,(NDTM(I),I=1,NTIMDX)                        00648724  *
  2   1160   FORMAT(1X,'READ CASE AND TRANS RECORD',6(I3,2X))                    00648815  *
  1  C                                                                          00648915  *
  8          NTIMST = 0                                                         00649015  *
  8          DO 160 JJ=1,NTIMDX                                                 00649115  *
  8          NTIMST = NDTM(JJ) + NTIMST                                         00649215  *
  2   160    CONTINUE                                                           00649315  *
  2   C                                                                        00649415  *
  8          NTIMST=NTIMST+1                                                    00649515  *
  8          TIME(1) = 0.0                                                      00649615  *
  8          STEP(1) = 1                                                        00649715  *
  8          KK = 1                                                             00649815  *
  8          DO 170 JK=1,NTIMDX                                                 00649915  *
  8          NSTOP = NDTM(JK)                                                   00650019  *
  8          IF(JK.EQ.1)GO TO 92                                                00650119  *
  8          TIMSTP=(FTIME(JK)-FTIME(JK-1))/NDTM(JK)                            00650219  *
  8          GO TO 93                                                           00650319  *
  2   92     TIMSTP=FTIME(JK)/NDTM(JK)                                          00650419  *
  2   93     DO 170 JJ=1,NSTOP                                                  00650515  *
 10            KK = KK+1                                                       00650619  *
 10            TIME(KK) = TIME(KK-1) + TIMSTP                                  00650715  *
 10            LJ=KK                                                           00650815  *
 10            STEP(KK) = LJ                                                   00650915  *
  2   170    CONTINUE                                                           00651023  *
  8          WRITE(6,1170)                                                      00651123  *
  2   1170   FORMAT(1X,'CALCULATED TIMESTEPS')                                 00651384  *
  1  C       ---------------------------------------------------------          00651419  *
  1  C       COMPARE CALCULATED NUMBER OF TIMESTEPS TO USER INPUT               00651684  *
  1  C       ---------------------------------------------------------          00651716  *
  8          IF(NTIMST.EQ.NTIME)GO TO 172                                       00652016  *
  8          WRITE(6,1172) NTIMST,NTIME                                         00660016  *
  2   1172   FORMAT(1X,'NUMBER OF TIMESTEPS DO NOT AGREE ',I3,3X,I3)           00670016  *
  8          NTIMST=NTIME                                                       00680016  *
  2   172    CONTINUE                                                           00690002  *
  1  C       ---------------------------------------------------------          00700001  *
  1  C       LOOP OVER THE NUMBER OF TIMESTEPS AND READ EACH POWER RECORD       00710002  *
  1  C       ---------------------------------------------------------          00720001  *
  8          DO 30 K=1,NTIME                                                    00730001  *
 10            READ('TEMP','RIDS',CASE,'RPWRO',K) NW,(A(L),L=1,NW)             00740002  *
  1  C       ---------------------------------------------------------          00750001  *
  1  C       CHECK THE NUMBER OF AXIAL LAYERS AGAINST INPUT DATA                00760002  *
  1  C       ---------------------------------------------------------          00770001  *
 10            IAX = NW/889                                                    00780001  *
 10            IF (IAX.EQ.NAX) GO TO 20                                        00790009  *
 10            WRITE(6,10) IAX,NAX                                             00800009  *
  2   10       FORMAT(1X,'AXIAL LEVELS DO NOT AGREE',I3,2X,I3)                 00810001  *
 10            NAX = IAX                                                       00820001  *
  2   20       L=0                                                             00830002  *
  1  C       ---------------------------------------------------------          00840001  *
  1  C       FILL THE POWER ARRAY WITH ASSEMBLY POWERS BY AXIAL LAYER FOR       00850001  *
  1  C       EACH TIME STEP                                                     00860002  *
  1  C       ---------------------------------------------------------          00870001  *
 10            DO 30 J=1,NAX                                                   00880001  *
 12              DO 30 I=1,889                                                 00890001  *
 14                L=L+1                                                       00900001  *
 14                POW(I,J,K) = A(L)                                           00910001  *
  2   30     CONTINUE                                                           00930023  *
  8          WRITE(6,1050)                                                      00940023  *
  2   1050   FORMAT(1X,'FILLED ARRAY AT AXIAL LEVEL')
```

```
1  C      ------------------------------------------------------------  00941084  *
1  C      SUM THE POWERS FOR EACH ASSEMBLY BY AXIAL LAYER TO PROVIDE    00950001  *
1  C      TOTAL ASSEMBLY POWER                                          00960001  *
1  C      ------------------------------------------------------------  00970002  *
1  C                                                                    00980001  *
8         DO 50 K=1,NTIME                                              00990001  *
10          DO 50 I=1,889                                              01000001  *
12            PASSY(I,K) = 0.0                                         01010001  *
12            DO 50 J=1,NAX                                            01020001  *
14              IF (POW(I,J,K).LE.0.0) GO TO 50                        01030001  *
14              PASSY(I,K) = POW(I,J,K) + PASSY(I,K)                   01040001  *
2    50     CONTINUE                                                   01050002  *
1  C      ------------------------------------------------------------  01060001  *
1  C      FIND TOTAL CORE POWER AT EACH TIME STEP BY SUMMING THE ASSEMBLY 01070001 *
1  C      POWERS                                                        01080002  *
1  C      ------------------------------------------------------------  01090001  *
8         DO 60 K=1,NTIME                                              01100001  *
10          PCORE(K) = 0.0                                             01110001  *
10          DO 60 I=1,889                                              01120001  *
12            PCORE(K) = PASSY(I,K) + PCORE(K)                         01130001  *
2    60     CONTINUE                                                   01140002  *
1  C      ------------------------------------------------------------  01150002  *
1  C      CONVERT TO RELATIVE ASSEMBLY POWER AND RELATIVE CORE POWER    01160002  *
1  C      ------------------------------------------------------------  01170002  *
8         DO 90 K=1,NTIME                                              01180002  *
10          DO 80 I=1,889                                              01190002  *
12            DO 70 J=1,NAX                                            01200002  *
14              IF (PASSY(I,K).LE.0.0) GO TO 70                        01210002  *
14              POW(I,J,K) = POW(I,J,K)/PASSY(I,K)                     01220002  *
2    70         CONTINUE                                               01230051  *
14              IF(PCORE(K).LE.0.0) GO TO 80                           01240002  *
14              PASSY(I,K) = PASSY(I,K)/PCORE(K)                       01240184  *
1  C      ------------------------------------------------------------  01240278  *
1  C        PPASSY IS USED TO PRINT ASSEMBLY POWERS.  THEY ARE          01240378  *
1  C        MULTIPLIED BY 1000 TO COMPARE THEM AGAINST XGRIM3D          01240478  *
1  C        OUTPUTS.                                                    01240584  *
1  C      ------------------------------------------------------------  01241079  *
14              PPASSY(I,K) = PASSY(I,K)*1000                          01250002  *
2    80       CONTINUE                                                 01260052  *
14              PRCORE(K) =  PCORE(K)/PCORE(1)                         01270002  *
2    90     CONTINUE                                                   01300002  *
1  C      ------------------------------------------------------------  01310001  *
1  C      FIND POWER POST PEAKING FOR EACH ASSEMBLY FOR EACH TIME STEP  01320002  *
1  C      ------------------------------------------------------------  01330002  *
8         DO 100 K=1,NTIME                                             01340002  *
10          DO 100 I=1,889                                             01350002  *
12            PPP(I,K) = 0.0                                           01360016  *
12            IF((PASSY(I,1).LE.0).OR.(PCORE(1).LE.0)) GO TO 100       01370001  *
12            PPP(I,K) = (PASSY(I,K)/PASSY(I,1))/(PCORE(K)/PCORE(1))   01380002  *
2    100    CONTINUE                                                   01390032  *
8         WRITE(6,1100)                                                01400032  *
2    1100   FORMAT(1X,'FOUND PPP')                                     01410002  *
1  C      ------------------------------------------------------------  01420002  *
1  C      DETERMINE THE HEIGHT OF THE CENTER OF EACH AXIAL SEGMENT      01421084  *
1  C      ------------------------------------------------------------  01430040  *
8         READ('INPUT','REX'.CASE.'GEOMETRY'.'REACTOR') NGIX,RXTYP,NSCX, 01440040 *
6       1 ND3X,NHPX,PITCH,RXN1,RXN2,RXN3,WBCL,WBCR,(SEG(I),NPT(I),      01450042  *
6       2 I=1,ND3X)                                                    01460041  *
8           DO 105 I=1,ND3X                                            01470042  *
8           WRITE(6,1110)ND3X,I,SEG(I),NPT(I)                          01480038  *
2    1110   FORMAT(1X,'N1 ',I3,2X,I3,2X,F6.1,2X,I3,2X,I3)              01490038  *
2    105    CONTINUE                                                   01500060  *
8           TOP = SEG(ND3X)                                            01510003  *
8           W(1) = SEG(1)                                              01520035  *
8           SEGAVG(1) = SEG(1)/2.0                                     01530003  *
8           DO 110 J=2,NAX                                             01540035  *
8           SEGAVG(J) = (SEG(J) - SEG(J-1))/2.0 + SEG(J-1)             01550003  *
8           W(J) = 0.0                                                 01560003  *
8           W(J) = SEG(J) - SEG(J-1)                                   01570003  *
2    110    CONTINUE                                                   01580003  *
8           DO 120 J=1,NAX                                             01590003  *
8           SEG(J) = SEGAVG(J)                                         01610003  *
2    120    CONTINUE                                                   01620035  *
8           WRITE(6,1120)(SEGAVG(J),J=1,NAX)                           01630088  **
2    1120   FORMAT(1X,'FOUND SEG MIDPNT',6(F8.2,2X))                   01650035  *
8           WRITE(6,1140) (W(J),J=1,NAX)                               01660088  **
2    1140   FORMAT(1X,'W(J) IS OK',6(F8.2,2X))                         01671084  *
1  C      ------------------------------------------------------------  01680088  **
1  C      CALCULATE THE AXIAL POWER DENSITY, THEN NORMALIZE TO NAX LAYERS 01691084 *
1  C      ------------------------------------------------------------  01700004  *
8           DO 140 K=1,NTIME                                           01710004  *
8           DO 140 I=1,889                                             01720004  *
10            TOTPOW(I,K) = 0.0                                        01730004  *
8           DO 140 J=1,NAX                                             01740004  *
10            POW(I,J,K) = POW(I,J,K)/W(J)                             01750004  *
10            TOTPOW(I,K) = POW(I,J,K) + TOTPOW(I,K)                   01760004  *
2    140    CONTINUE                                                   01770043  *
8           WRITE(6,1141)                                              01780043  *
2    1141   FORMAT(1X,'FOUND TOTPOW')                                  01790004  *
1  C                                                                    01800004  *
8           NORM = NAX                                                 01810044  *
8           WRITE(6,1142) NORM                                         01820044  *
2    1142   FORMAT(1X,'NORM = ',I3)                                    01830088  **
1  C                                                                    01840004  *
8           DO 150 K=1,NTIME                                           01850004  *
8           DO 150 I=1,889                                             01860044  *
10            IF(TOTPOW(I,K).LE.0.0) GO TO 150                         01870004  *
8           DO 150 J=1,NAX                                             01880004  *
10            POW(I,J,K) = POW(I,J,K) * (NORM/TOTPOW(I,K))             01890004  *
3     150   CONTINUE
```

```
8          DO 155 K=1,NTIME                                                    01890178  *
8          DO 155 I=1,889                                                      01890278  *
8          DO 155 LV=1,NAX                                                     01891078  *
8          LX=(NAX+1)-LV                                                       01892078  *
8          REPOW(I,LX,K) = POW(I,LV,K)                                         01893078  *
8          RELVL(LX) = (TOP - SEG(LV))/TOP                                     01893187  *
2   155    CONTINUE                                                            01894078  *
1   C                                                                          01895085  *
8          WRITE(6,1150)                                                       01900032  *
2   1150   FORMAT(1X,'FOUND APD')                                              01910032  *
1   C                                                                          01920017  *
1   C      ------------------------------------------------------------        01941085  *
1   C      THE FLOWTRAN FILE IS ONLY CREATED FOR THOSE ASSEMBLIES              01942085  *
1   C      WHICH PRODUCE POWER (PASSY(I,1)>0) AT TIME =0                       01943085  *
1   C      ------------------------------------------------------------        01944085  *
1   C                                                                          01950072  *
8          DO 999 I=1,889                                                      01960072  *
8          IF(PASSY(I,1).LE.0) GO TO 999                                       01980083  *
9           WRITE('TRIMPWR'.CASE.'OUTPUT'.'FLOWFILE'.I) NAX,NTIME,             01990082  *
6         1 (TIME(K),K=1,NTIME),(RELVL(J),REPOW(I,J,1),J=1,NAX),               02000078  *
6         2 ((REPOW(I,J,K),J=1,NAX),K=2,NTIME)                                 02020072  *
2   999    CONTINUE                                                            02041084  *
1   C      ------------------------------------------------------------        02050052  *
1   C      DETERMINE MAXIMUM NORMALIZED AXIAL POWER DENSITY FOR EACH           02060017  *
1   C      ASSEMBLY AT EACH TIME                                               02071084  *
1   C      ------------------------------------------------------------        02080017  *
8          DO 190 K=1,NTIME                                                    02090017  *
8          DO 190 I=1,889                                                      02100017  *
8          MAX = 0                                                             02110017  *
8          DO 190 J=1,NAX                                                      02120017  *
8          IF(POW(I,J,K).GT.MAX) MAX = POW(I,J,K)                              02130017  *
8          AXMAX(I,K) = MAX                                                    02140017  *
2   190    CONTINUE                                                            02151084  *
1   C      ------------------------------------------------------------        02160017  *
1   C      CALCULATE PRODUCT OF POWER POST PEAKING AND NAPD (MAX)              02171084  *
1   C      ------------------------------------------------------------        02180017  *
8          DO 200 K=1,NTIME                                                    02190019  *
8          DO 200 I=1,889                                                      02200017  *
8          PROD(I,K) = PPP(I,K) * AXMAX(I,K)                                   02210017  *
2   200    CONTINUE                                                            02271084  *
1   C      ------------------------------------------------------------        02280017  *
1   C      SORT ASSEMBLIES BY DECREASING VALUES OF THE PRODUCT                 02291084  *
1   C      ------------------------------------------------------------        02300058  *
8          DO 225 K=1,NTIME                                                    02310058  *
8          DO 210 J=1,889                                                      02320058  *
8          HEX(J) = J                                                          02330017  *
2   210    CONTINUE                                                            02340017  *
1   C                                                                          02350017  *
2   215    L = 0                                                               02360017  *
8          M = 0                                                               02370017  *
8          KZ = 0                                                              02380017  *
8          DO 220 I=1,888                                                      02390052  *
8          IF(PROD(I,K).GE.PROD(I+1,K)) GO TO 220                              02400017  *
8          KZ = PROD(I,K)                                                      02410017  *
8          M = HEX(I)                                                          02420052  *
8          PROD(I,K) = PROD(I+1,K)                                             02430052  *
8          HEX(I) = HEX(I+1)                                                   02440052  *
8          PROD(I+1,K) = KZ                                                    02450017  *
8          HEX(I+1) = M                                                        02460017  *
8          L=1                                                                 02470017  *
2   220    CONTINUE                                                            02480088  **
1   C      ------------------------------------------------------------        02481088  **
1   C      SORTPRD PRINTS OUT VALUES FOR 433 ASSEMBLIES, THIS IS               02482088  **
1   C      ENOUGH TO INSURE THE 432 FUELED ASSEMBLIES IN A MARK                02483088  **
1   C      22 CHARGE ARE ALL REPRESENTED.  IF ANOTHER CHARGE IS                02484088  **
1   C      TO BE STUDIED, THE INDEX FOR I IN THE FOLLOWING DO LOOP             02485088  **
1   C      SHOULD BE INCREASED TO INSURE ALL ASSEMBLIES OF INTEREST            02486088  **
1   C      ARE LISTED.                                                         02487088  **
1   C      ------------------------------------------------------------        02490017  *
8          IF(L.EQ.1) GO TO 215                                                02492086  *
8          WRITE('TEMP'.'RIDS'.CASE.'OUTPUT'.'SORTPRD'.K) TIME(K),             02493086  *
6         1 (HEX(I),PROD(I,K),PPP(HEX(I),K),AXMAX(HEX(I),K),                   02494086  *
6         2 I=1,433)                                                           02500058  *
2   225    CONTINUE                                                            02630006  *
1   C                                                                          02640049  *
8          WRITE(6,1181) NTIME                                                 02650049  *
2   1181   FORMAT(1X,'NTIME = ',I3)                                            02660047  *
8          WRITE('TEMP'.'RIDS'.CASE.'OUTPUT'.'CPOWER') NTIME,(STEP(KK),        02670052  *
6         1 TIME(KK),PRCORE(KK),KK=1,NTIME)                                    02680006  *
8          DO 180 K=1,NTIME                                                    02690017  *
8          WRITE('TEMP'.'RIDS'.CASE.'OUTPUT'.'APOWER'.K) TIME(K),              02700078  *
6         1 (I,PPASSY(I,K),I=1,889)                                            02710009  *
2   180    CONTINUE                                                            02810007  *
2   4000   STOP                                                                02820007  *
8          GO TO 4000                                                          02830008  *
8          END
```

APPENDIX C
Programmers Manual for TRIMPWR

Figure C.1
ABRIDGED PROGRAMMERS MANUAL FOR TRIMPWR

Variables used in TRIMPWR:

Variable	Type/Size	Description
POW	Real(889,12,25)	Power at each: axial level/assembly/time
REPOW	Real(889,12,25)	Relative power at: axial level/assy/time
PCORE	Real(25)	Total core power
A	Real(11000)	Dummy for RPWRO data
NTIMDX	Real	Number of time domains
FTIME	Real(20)	Final time in this time domain
PRCORE	Real(25)	Core relative power
PPP	Real(889,25)	Power post peaking
W	Real(25)	Layer thickness
SEGAVG	Real(25)	Layer midpoint (height from bottom)
TOTPOW	Real(889,25)	Sum for assembly power
TIME	Real(50)	Value of time at a timestep
NTIME	Integer*2	Number of timesteps
NAX	Integer*2	Number of axial layers
PASSY	Real(889,25)	Assembly power (normalized)
PPASSY	Real(889,25)	Assembly power (normalized) * 1000
AXMAX	Real(889,25)	Max normalized axial power density
PROD	Real(889,25)	Product of MNAPD and PPP
RELVL	Real(12)	Relative distance from top of assembly
STEP	Integer(25)	Timestep index
HEX	Integer(889)	Hex number
NPT	Integer*2(25)	Number of points per axial layer
CASE	Real*8	TRIMHX case name

Files read by TRIMPWR:

INPUT.TRIMPWR.JOB.$JOB
INPUT.REX.?case.TRANS
TEMP.RIDS.CASE.RPWR0.?timestep
INPUT.REX.?case.GEOMETRY.REACTOR

Files written by TRIMPWR:

TEMP.RIDS.?case.OUTPUT.CPOWER
TEMP.RIDS.?case.OUTPUT.APOWER.?timestep
TEMP.RIDS.?case.OUTPUT.SORTPRD.?timestep
TRIMPWR.?case.OUTPUT.FLOWFILE.?hexnumber

Figure C.1
ABRIDGED PROGRAMMERS MANUAL FOR TRIMPWR

The TRIMPWR output files and their contents are described below:

TEMP.RIDS.MK223DA.OUTPUT.CPOWER

This file contains the normalized core power [pcore(t)/pcore(0)] as a function of time for the transient. The format is as follows

time step time power

TEMP.RIDS.MK223DA.OUTPUT.APOWER.? ? = time step

This file contains, at each time step, the normalized assembly power [passy(t)/pcore(t)] value (multiplied by 1000) for each of the 889 assemblies. The format is as follows

hex # power hex # power hex # power

TEMP.RIDS.MK223DA.OUTPUT.SORTPRD.? ? = time step

This file contains, at each time step, the hex numbers sorted (in decreasing order) by the product (prod) of the power post peaking (ppp) and the maximum normalized axial power density (MNAPD). The format is as follows

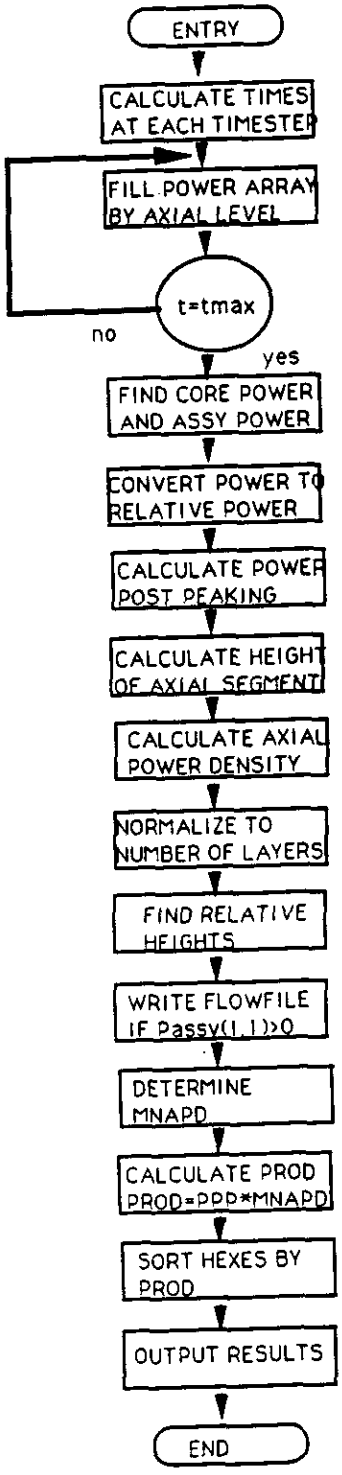
hex # prod ppp MNAPD

TRIMPWR.MK223DA.OUTPUT.FLOWFILE.? ? = hex #

This file contains, for each assembly, the complete FLOWTRAN input record for the assembly. The number of timesteps, axial layers, and time at each timestep are included in this file, as well as the static and transient normalized axial power shapes for each assembly. This file is created for all assemblies which have a non-zero initial (time=0) power.

FIGURE C.2

FLOWCHART FOR TRIMPWR



Appendix D
Example output from TRIMPWR

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.CPOWER

NUMBER OF TIMESTEPS = 21

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.CPOWER

TIMESTEP	TIME(SEC)	REL. CORE POWER
1	0.00	1.0000
2	.80	1.0000
3	.85	1.0000
4	.90	.9936
5	1.00	.9787
6	1.10	.9399
7	1.20	.8481
8	1.30	.7232
9	1.40	.5563
10	1.50	.4049
11	1.60	.3203
12	1.70	.2733
13	1.80	.2494
14	1.90	.2404
15	2.00	.2335
16	2.10	.2277
17	2.20	.2224
18	2.30	.2174

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.CPOWER

TIMESTEP	TIME(SEC)	REL. CORE POWER
19	2.40	.2128
20	2.50	.2084
21	2.60	.2042

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

FOR THE TIME = 0.00 SEC

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
1	0.0000	2	1.3031	3	1.3463
4	1.3746	5	0.0000	6	1.4405
7	1.4924	8	1.6264	9	1.6144
10	1.5871	11	1.6739	12	1.8404
13	0.0000	14	1.8098	15	0.0000
16	1.9478	17	2.1018	18	0.0000
19	2.0530	20	2.0207	21	2.1164
22	2.2691	23	2.4903	24	2.4612
25	2.3701	26	2.2961	27	2.3229
28	2.4324	29	0.0000	30	2.8536
31	3.1332	32	3.0866	33	2.6994
34	0.0000	35	2.6668	36	2.7971
37	2.9680	38	3.2849	39	3.3907
40	0.0000	41	3.3382	42	2.9583
43	2.9260	44	3.2291	45	3.2985
46	3.3317	47	0.0000	48	3.3338
49	3.4553	50	3.4794	51	3.4053
52	3.3297	53	3.3155	54	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
55	3.2208	56	3.0555	57	2.6869
58	2.9504	59	3.1108	60	3.2190
61	3.2824	62	0.0000	63	3.1895
64	3.1056	65	2.9676	66	2.5421
67	2.3502	68	2.0391	69	2.3118
70	2.6503	71	0.0000	72	2.8021
73	2.8572	74	2.8146	75	2.5114
76	2.4831	77	2.3489	78	2.0985
79	0.0000	80	0.0000	81	0.0000
82	0.0000	83	2.2655	84	2.3187
85	2.2251	86	2.2236	87	2.1459
88	2.1137	89	0.0000	90	0.0000
91	0.0000	92	0.0000	93	0.0000
94	0.0000	95	0.0000	96	0.0000
97	0.0000	98	0.0000	99	0.0000
100	0.0000	101	0.0000	102	0.0000
103	0.0000	104	0.0000	105	0.0000
106	0.0000	107	0.0000	108	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
109	0.0000	110	0.0000	111	0.0000
112	0.0000	113	0.0000	114	0.0000
115	0.0000	116	0.0000	117	0.0000
118	0.0000	119	0.0000	120	0.0000
121	0.0000	122	0.0000	123	0.0000
124	0.0000	125	0.0000	126	0.0000
127	0.0000	128	0.0000	129	0.0000
130	0.0000	131	0.0000	132	0.0000
133	0.0000	134	0.0000	135	0.0000
136	0.0000	137	0.0000	138	0.0000
139	0.0000	140	0.0000	141	0.0000
142	0.0000	143	0.0000	144	0.0000
145	0.0000	146	0.0000	147	0.0000
148	0.0000	149	0.0000	150	1.2868
151	1.3417	152	1.3429	153	0.0000
154	1.4190	155	1.4431	156	1.6463
157	1.6046	158	1.5486	159	1.6041
160	1.8736	161	0.0000	162	1.7829

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
163	0.0000	164	1.8531	165	2.1492
166	0.0000	167	2.0374	168	1.9739
169	2.0345	170	2.1478	171	2.5514
172	2.4915	173	2.3685	174	2.2629
175	2.2550	176	2.3279	177	0.0000
178	2.9298	179	3.1802	180	3.1016
181	2.6806	182	0.0000	183	2.5761
184	2.6680	185	2.7945	186	3.3805
187	3.4512	188	0.0000	189	3.3341
190	2.9195	191	2.8509	192	3.1079
193	3.1404	194	3.1334	195	0.0000
196	3.4060	197	3.4946	198	3.4879
199	3.3775	200	3.2689	201	3.2155
202	0.0000	203	3.0621	204	2.8735
205	2.7726	206	3.0203	207	3.1545
208	3.2371	209	3.2688	210	0.0000
211	3.1186	212	3.0029	213	2.8436
214	2.4131	215	2.2125	216	2.1065

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
217	2.3689	218	2.6934	219	0.0000
220	2.8027	221	2.8319	222	2.7666
223	2.4454	224	2.3959	225	2.2473
226	1.9905	227	0.0000	228	0.0000
229	0.0000	230	0.0000	231	2.2905
232	2.3271	233	2.2148	234	2.1952
235	2.1011	236	2.0518	237	0.0000
238	0.0000	239	0.0000	240	0.0000
241	0.0000	242	0.0000	243	0.0000
244	0.0000	245	0.0000	246	0.0000
247	0.0000	248	0.0000	249	0.0000
250	0.0000	251	0.0000	252	0.0000
253	0.0000	254	0.0000	255	0.0000
256	0.0000	257	0.0000	258	0.0000
259	0.0000	260	0.0000	261	0.0000
262	0.0000	263	0.0000	264	0.0000
265	0.0000	266	0.0000	267	0.0000
268	0.0000	269	0.0000	270	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
271	0.0000	272	0.0000	273	0.0000
274	0.0000	275	0.0000	276	0.0000
277	0.0000	278	0.0000	279	0.0000
280	0.0000	281	0.0000	282	0.0000
283	0.0000	284	0.0000	285	0.0000
286	0.0000	287	0.0000	288	0.0000
289	0.0000	290	0.0000	291	0.0000
292	0.0000	293	0.0000	294	0.0000
295	0.0000	296	0.0000	297	0.0000
298	1.2580	299	1.2977	300	1.2856
301	0.0000	302	1.3446	303	1.3534
304	1.5614	305	1.5070	306	1.4412
307	1.4780	308	1.7652	309	0.0000
310	1.6487	311	0.0000	312	1.6822
313	2.0128	314	0.0000	315	1.8741
316	1.8007	317	1.8411	318	1.9286
319	2.3778	320	2.2989	321	2.1686
322	2.0591	323	2.0374	324	2.0881

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
325	0.0000	326	2.7252	327	2.9207
328	2.8274	329	2.4340	330	0.0000
331	2.3101	332	2.3755	333	2.4682
334	3.1470	335	3.1717	336	0.0000
337	3.0212	338	2.6356	339	2.5590
340	2.7708	341	2.7809	342	2.7444
343	0.0000	344	3.1454	345	3.1914
346	3.1642	347	3.0511	348	2.9394
349	2.8735	350	0.0000	351	2.7021
352	2.5081	353	2.5879	354	2.7966
355	2.8947	356	2.9495	357	2.9593
358	0.0000	359	2.7964	360	2.6763
361	2.5202	362	2.1249	363	1.9363
364	1.9693	365	2.1968	366	2.4786
367	0.0000	368	2.5480	369	2.5579
370	2.4872	371	2.1875	372	2.1318
373	1.9889	374	1.7515	375	0.0000
376	0.0000	377	0.0000	378	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
379	2.0993	380	2.1205	381	2.0068
382	1.9790	383	1.8852	384	1.8319
385	0.0000	386	0.0000	387	0.0000
388	0.0000	389	0.0000	390	0.0000
391	0.0000	392	0.0000	393	0.0000
394	0.0000	395	0.0000	396	0.0000
397	0.0000	398	0.0000	399	0.0000
400	0.0000	401	0.0000	402	0.0000
403	0.0000	404	0.0000	405	0.0000
406	0.0000	407	0.0000	408	0.0000
409	0.0000	410	0.0000	411	0.0000
412	0.0000	413	0.0000	414	0.0000
415	0.0000	416	0.0000	417	0.0000
418	0.0000	419	0.0000	420	0.0000
421	0.0000	422	0.0000	423	0.0000
424	0.0000	425	0.0000	426	0.0000
427	0.0000	428	0.0000	429	0.0000
430	0.0000	431	0.0000	432	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
433	0.0000	434	0.0000	435	0.0000
436	0.0000	437	0.0000	438	0.0000
439	0.0000	440	0.0000	441	0.0000
442	0.0000	443	0.0000	444	0.0000
445	0.0000	446	1.2454	447	1.2568
448	1.2595	449	0.0000	450	1.2885
451	1.3118	452	1.4492	453	1.4122
454	1.3671	455	1.4194	456	1.6117
457	0.0000	458	1.5309	459	0.0000
460	1.6017	461	1.8116	462	0.0000
463	1.7101	464	1.6594	465	1.7178
466	1.8243	467	2.1180	468	2.0529
469	1.9476	470	1.8661	471	1.8671
472	1.9363	473	0.0000	474	2.4067
475	2.5787	476	2.5042	477	2.1742
478	0.0000	479	2.1048	480	2.1901
481	2.3100	482	2.7519	483	2.7811
484	0.0000	485	2.6651	486	2.3434

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
487	2.2956	488	2.5048	489	2.5453
490	2.5619	491	0.0000	492	2.7369
493	2.7868	494	2.7706	495	2.6791
496	2.5968	497	2.5650	498	0.0000
499	2.4700	500	2.3384	501	2.2456
502	2.4219	503	2.5091	504	2.5625
505	2.5797	506	0.0000	507	2.4679
508	2.3889	509	2.2747	510	1.9517
511	1.8031	512	1.7162	513	1.9061
514	2.1392	515	0.0000	516	2.2072
517	2.2269	518	2.1794	519	1.9420
520	1.9135	521	1.8062	522	1.6120
523	0.0000	524	0.0000	525	0.0000
526	0.0000	527	1.8062	528	1.8281
529	1.7459	530	1.7330	531	1.6643
532	1.6335	533	0.0000	534	0.0000
535	0.0000	536	0.0000	537	0.0000
538	0.0000	539	0.0000	540	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
541	0.0000	542	0.0000	543	0.0000
544	0.0000	545	0.0000	546	0.0000
547	0.0000	548	0.0000	549	0.0000
550	0.0000	551	0.0000	552	0.0000
553	0.0000	554	0.0000	555	0.0000
556	0.0000	557	0.0000	558	0.0000
559	0.0000	560	0.0000	561	0.0000
562	0.0000	563	0.0000	564	0.0000
565	0.0000	566	0.0000	567	0.0000
568	0.0000	569	0.0000	570	0.0000
571	0.0000	572	0.0000	573	0.0000
574	0.0000	575	0.0000	576	0.0000
577	0.0000	578	0.0000	579	0.0000
580	0.0000	581	0.0000	582	0.0000
583	0.0000	584	0.0000	585	0.0000
586	0.0000	587	0.0000	588	0.0000
589	0.0000	590	0.0000	591	0.0000
592	0.0000	593	0.0000	594	1.2621

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
595	1.2611	596	1.2928	597	0.0000
598	1.3107	599	1.3648	600	1.4237
601	1.4206	602	1.4083	603	1.4961
604	1.5681	605	0.0000	606	1.5584
607	0.0000	608	1.7078	609	1.7475
610	0.0000	611	1.7230	612	1.7101
613	1.8103	614	1.9627	615	2.0326
616	2.0066	617	1.9418	618	1.8998
619	1.9436	620	2.0575	621	0.0000
622	2.2973	623	2.5072	624	2.4743
625	2.1885	626	0.0000	627	2.2105
628	2.3448	629	2.5223	630	2.6129
631	2.6894	632	0.0000	633	2.6585
634	2.3827	635	2.3817	636	2.6497
637	2.7404	638	2.8213	639	0.0000
640	2.6291	641	2.7225	642	2.7469
643	2.7024	644	2.6634	645	2.6827
646	0.0000	647	2.6711	648	2.5837

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
649	2.1201	650	2.3178	651	2.4396
652	2.5267	653	2.5849	654	0.0000
655	2.5489	656	2.5123	657	2.4290
658	2.1182	659	1.9853	660	1.6170
661	1.8207	662	2.0718	663	0.0000
664	2.1946	665	2.2489	666	2.2318
667	2.0198	668	2.0206	669	1.9348
670	1.7522	671	0.0000	672	0.0000
673	0.0000	674	0.0000	675	1.7645
676	1.8084	677	1.7511	678	1.7620
679	1.7156	680	1.7081	681	0.0000
682	0.0000	683	0.0000	684	0.0000
685	0.0000	686	0.0000	687	0.0000
688	0.0000	689	0.0000	690	0.0000
691	0.0000	692	0.0000	693	0.0000
694	0.0000	695	0.0000	696	0.0000
697	0.0000	698	0.0000	699	0.0000
700	0.0000	701	0.0000	702	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
703	0.0000	704	0.0000	705	0.0000
706	0.0000	707	0.0000	708	0.0000
709	0.0000	710	0.0000	711	0.0000
712	0.0000	713	0.0000	714	0.0000
715	0.0000	716	0.0000	717	0.0000
718	0.0000	719	0.0000	720	0.0000
721	0.0000	722	0.0000	723	0.0000
724	0.0000	725	0.0000	726	0.0000
727	0.0000	728	0.0000	729	0.0000
730	0.0000	731	0.0000	732	0.0000
733	0.0000	734	0.0000	735	0.0000
736	0.0000	737	0.0000	738	0.0000
739	0.0000	740	0.0000	741	0.0000
742	1.2911	743	1.3066	744	1.3507
745	0.0000	746	1.3883	747	1.4559
748	1.5160	749	1.5252	750	1.5210
751	1.6249	752	1.6881	753	0.0000
754	1.7032	755	0.0000	756	1.8836

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
703	0.0000	704	0.0000	705	0.0000
706	0.0000	707	0.0000	708	0.0000
709	0.0000	710	0.0000	711	0.0000
712	0.0000	713	0.0000	714	0.0000
715	0.0000	716	0.0000	717	0.0000
718	0.0000	719	0.0000	720	0.0000
721	0.0000	722	0.0000	723	0.0000
724	0.0000	725	0.0000	726	0.0000
727	0.0000	728	0.0000	729	0.0000
730	0.0000	731	0.0000	732	0.0000
733	0.0000	734	0.0000	735	0.0000
736	0.0000	737	0.0000	738	0.0000
739	0.0000	740	0.0000	741	0.0000
742	1.2911	743	1.3066	744	1.3507
745	0.0000	746	1.3883	747	1.4559
748	1.5160	749	1.5252	750	1.5210
751	1.6249	752	1.6881	753	0.0000
754	1.7032	755	0.0000	756	1.8836

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
757	1.9006	758	0.0000	759	1.9028
760	1.8985	761	2.0163	762	2.1893
763	2.2302	764	2.2218	765	2.1644
766	2.1266	767	2.1830	768	2.3152
769	0.0000	770	2.5378	771	2.8011
772	2.7821	773	2.4675	774	0.0000
775	2.5083	776	2.6617	777	2.8567
778	2.9092	779	3.0178	780	0.0000
781	3.0186	782	2.7148	783	2.7223
784	3.0405	785	3.1380	786	3.2060
787	0.0000	788	2.9624	789	3.0863
790	3.1315	791	3.1032	792	3.0688
793	3.0948	794	0.0000	795	3.0659
796	2.9392	797	2.3732	798	2.6194
799	2.7795	800	2.8945	801	2.9787
802	0.0000	803	2.9508	804	2.9062
805	2.8025	806	2.4234	807	2.2590
808	1.8003	809	2.0520	810	2.3661

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
811	0.0000	812	2.5332	813	2.6061
814	2.5893	815	2.3334	816	2.3288
817	2.2218	818	2.0022	819	0.0000
820	0.0000	821	0.0000	822	0.0000
823	2.0312	824	2.0917	825	2.0224
826	2.0372	827	1.9827	828	1.9703
829	0.0000	830	0.0000	831	0.0000
832	0.0000	833	0.0000	834	0.0000
835	0.0000	836	0.0000	837	0.0000
838	0.0000	839	0.0000	840	0.0000
841	0.0000	842	0.0000	843	0.0000
844	0.0000	845	0.0000	846	0.0000
847	0.0000	848	0.0000	849	0.0000
850	0.0000	851	0.0000	852	0.0000
853	0.0000	854	0.0000	855	0.0000
856	0.0000	857	0.0000	858	0.0000
859	0.0000	860	0.0000	861	0.0000
862	0.0000	863	0.0000	864	0.0000

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.APOWER.1

HEX	ASSY POWER	HEX	ASSY POWER	HEX	ASSY POWER
865	0.0000	866	0.0000	867	0.0000
868	0.0000	869	0.0000	870	0.0000
871	0.0000	872	0.0000	873	0.0000
874	0.0000	875	0.0000	876	0.0000
877	0.0000	878	0.0000	879	0.0000
880	0.0000	881	0.0000	882	0.0000
883	0.0000	884	0.0000	885	0.0000
886	0.0000	887	0.0000	888	0.0000
889	0.0000				

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.SORTPRD.7

FOR TIME = 1.20 SEC

SVNR9379.TEMP.RIDS.MK223DA.OUTPUT.SORTPRD.7

HEX	PROD	PPP	MAX AX POWER DENSITY
231	2.9630	1.2135	2.4418
232	2.9630	1.2134	2.4420
197	2.9627	1.2076	2.4533
220	2.9626	1.2117	2.4449
209	2.9624	1.2089	2.4505
198	2.9621	1.2076	2.4528
208	2.9617	1.2106	2.4464
207	2.9616	1.2103	2.4471
206	2.9615	1.2080	2.4516
221	2.9610	1.2086	2.4499
196	2.9607	1.2059	2.4551
218	2.9606	1.2117	2.4434
187	2.9589	1.2026	2.4604
199	2.9581	1.2057	2.4535
189	2.9570	1.2041	2.4558
211	2.9566	1.1948	2.4746
222	2.9560	1.2030	2.4573
205	2.9558	1.2050	2.4529

HEX	PROD	PPP	MAX AX POWER DENSITY
186	2.9557	1.1995	2.4641
179	2.9555	1.1942	2.4749
46	2.9552	1.1917	2.4797
200	2.9547	1.2008	2.4606
180	2.9537	1.1992	2.4630
56	2.9529	1.1989	2.4631
201	2.9521	1.1939	2.4726
212	2.9513	1.1894	2.4812
55	2.9505	1.1914	2.4764
45	2.9502	1.1858	2.4880
354	2.9500	1.1785	2.5031
203	2.9491	1.1834	2.4920
61	2.9487	1.1780	2.5031
192	2.9484	1.1862	2.4855
344	2.9480	1.1814	2.4952
53	2.9475	1.1806	2.4967
50	2.9472	1.1771	2.5039
193	2.9471	1.1868	2.4832

HEX	PROD	PPP	MAX AX POWER DENSITY
345	2.9471	1.1804	2.4967
73	2.9469	1.1824	2.4924
204	2.9468	1.1810	2.4952
194	2.9467	1.1842	2.4884
65	2.9464	1.1847	2.4870
213	2.9462	1.1899	2.4759
334	2.9462	1.1784	2.5001
366	2.9461	1.1782	2.5006
346	2.9461	1.1763	2.5046
355	2.9460	1.1814	2.4937
335	2.9460	1.1768	2.5034
353	2.9458	1.1833	2.4894
63	2.9456	1.1828	2.4903
52	2.9455	1.1784	2.4997
356	2.9455	1.1774	2.5017
60	2.9455	1.1787	2.4990
72	2.9454	1.1838	2.4882
64	2.9453	1.1852	2.4850

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

THE NUMBER OF AXIAL LEVELS = 12
THE NUMBER OF TIMESTEPS = 21

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

THE TIMES ARE AS FOLLOWS

0.0000	.8000	.8500	.9000	1.0000
1.1000	1.2000	1.3000	1.4000	1.5000
1.6000	1.7000	1.8000	1.9000	2.0000
2.1000	2.2000	2.3000	2.4000	2.5000
2.6000				

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

STEADY STATE CONDITIONS

LEVEL	REL. POWER
.01107	0.00000
.03478	0.00000
.09893	.49194
.18958	1.02235
.28727	1.54476
.40439	2.05899
.52151	2.33977
.63862	2.12348
.73288	1.58106
.82009	.83765
.88423	0.00000
.94843	0.00000

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.49194	1.02235	1.54476
2.05899	2.33976	2.12347	1.58106	.83765
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.49194	1.02235	1.54476
2.05899	2.33976	2.12348	1.58107	.83765
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.47403	1.01047	1.54157
2.06308	2.34819	2.13265	1.58838	.84164
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.44091	.98760	1.53478
2.07039	2.36413	2.15030	1.60254	.84937
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.39933	.90922	1.50392
2.08234	2.40209	2.19482	1.63892	.86936
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.35486	.79489	1.38961
2.07272	2.46908	2.28740	1.71795	.91350
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.32270	.70946	1.21828
2.00839	2.53393	2.40654	1.82579	.97491
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.32013	.68946	1.13025
1.76525	2.52315	2.54005	1.97023	1.06148
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.35771	.75763	1.19600
1.73711	2.27922	2.51616	2.03912	1.11705
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.40379	.84741	1.30975
1.82360	2.23527	2.32744	1.95739	1.09534
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.44252	.92398	1.41132
1.91831	2.24955	2.17476	1.83208	1.04748
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.46672	.97249	1.47832
1.99029	2.29464	2.14035	1.67723	.97996
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.47186	.98267	1.49211
2.00468	2.30387	2.13662	1.65745	.95074
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.47591	.99081	1.50344
2.01727	2.31308	2.13478	1.64042	.92427
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.47943	.99789	1.51341
2.02860	2.32181	2.13374	1.62523	.89990
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.48266	1.00442	1.52263
2.03917	2.33009	2.13297	1.61107	.87700
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.48574	1.01064	1.53142
2.04925	2.33800	2.13221	1.59753	.85521
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.48867	1.01657	1.53981
2.05889	2.34556	2.13148	1.58456	.83446
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.49148	1.02224	1.54784
2.06812	2.35278	2.13075	1.57212	.81466
0.00000	0.00000			

SVNR9379.TRIMPWR.MK223DA.OUTPUT.FLOWFILE.2

TRANSIENT POWERS

0.00000	0.00000	.49418	1.02769	1.55554
2.07695	2.35970	2.13002	1.56017	.79575
0.00000	0.00000			

DISTRIBUTION

J. D. Spencer, 773-A
M. R. Buckner, 773-A
C. E. Apperson, 773-59A
W. E. Graves, 773-41A
J. R. Chandler, 707-C
N. H. Kuehn, 773-55A
P. B. Parks, 773-41A
M. V. Gregory, 773-41A
J. P. Church, 773-41A
A. O. Smetana, 773-42A
N. P. Baumann, 773-41A
R. L. Webb, 773-41A
L. R. Chandler, 773-59A
R. W. Ferarra, 773-41A
D. L. George, 773-59A
E. F. Trumble, 773-41A (4)
S. Skiles, 773-59A
G. Cefus, 773-59A
J. Wallace, 707-C
S. D. Painter, 773-55A
SRL File (4)
RPG File, 773-59A