MONTE-CARLO FREQUENCY ANALYSIS USING WATAND

BY

DJUNYEN YON

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Electrical Engineering

Program

Philip C. Munro 2/22/89

Advisor Date

Sally M. Hotchkiss February 27, 1989

Dean of the Graduate School Date

YOUNGSTOWN STATE UNIVERSITY

MARCH, 1989

ABSTRACT

MONTE-CARLO FREQUENCY ANALYSIS USING WATAND

Djunyen Yon
Master of Science, Electrical Engineering
Youngstown State University, 1989

A Monte-Carlo Frequency Response (MCFR) analysis and Monte-Carlo DIsplay (MCDI) post-processor based on the Watand macro facility are designed to meet the requirements of flexibility, speed, and minimum computer storage. The MCFR analysis provides statistical results in the frequency domain for circuits with linear and/or nonlinear elements. Initial, minimum, maximum, mean and standard deviation values of the varied elements/parameters and of the specified outputs are generated. The number of tolerance elements/parameters to be simultaneously simulated is limited only by computer memory. All Watand output actions (NONE, PRINT, PPRINTER, PTERMINA, PLOTTEKT, POVERLAY, PVOLKER, and CALL), five output types (MAGNITUD, PHASE, REAL, IMAGINAR and DB) and six statistical output modes (initial, minimum, maximum, average, standard deviation, and all values) are available.

The MCDI post-processor is designed to display the results of an MCFR analysis without re-calculating the circuit responses of the sample population. This saves computer time. MCDI also allows the display of statistical results of selected outputs with different output actions, output types, and/or statistical modes. Moreover, a zoom capability is available to view a smaller/larger portion of a Tektronix graphic display. Repeated operation of the zoom is possible. The Watand DI post-processor may be used to display the individual frequency responses of an entire sample population numerically or graphically.

A simple RC-filter example introduces the analysis and is used to show how statistical results are generated. A Boctor high-pass notch filter example is used to illustrate the use of the analysis and the post-processor. Statistical results for sample populations of 10 and 100 are compared. Two figures are used to show the capability of displaying the individual frequency responses of the entire sample population generated by the MCFR analysis.

ACKNOWLEDGEMENTS

First I wish to gratefully thank my thesis adviser, Dr. Philip Munro, for his ongoing guidance and noteworthy amount of time and patience to this thesis. Without his guidance, the existence of this paper would probably not have been feasible.

I would like to express my sincere gratitude to Dr. Hojjat Mehri in the Department of Industrial Engineering, and Drs. Robert Foulkes and Jalal Jalali in the Department of Electrical Engineering for their valuable comments on the statistics that were so helpful for the completion of the thesis.

I must acknowledge the assistance of Mr. David K. Rehard who carefully corrected my writing. Also, my special note of appreciation goes to my parents, who gave support to my study in the United States.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

WATAND PACKAGE

1.1 OVERVIEW

Watand (WATerloo ANalysis and Design) is an interactive computer program for linear and/or nonlinear electrical circuits simulation [1]. Founded by the Natural Science and Engineering Research Council of Canada and developed by the Electrical Engineering Department of the University of Waterloo since 1972 [2], Watand has been proven to be widely applicable and extremely convenient to the user. Version V1.11-03 with YSU changes (V1.11-3g) and later were used for this work.

1.2 WATAND FACILITIES USED IN MONTE-CARLO FR ANALYSIS

The FR and DI analyses, wexec macros, #RUSER, and the option commands are used in the implementation of the Monte-Carlo FR analysis and its post-processor. Their functions and definitions are discussed in the following sections.

1.2.1 FR ANALYSIS

This Watand analysis command performs small-signal frequency domain analysis at an operating point. The FR analysis help file [3] is listed in Table 1-1. For more detailed information about the FR analysis, see Ref. [1].

Table 1-1

FR Analysis Help Menu [3]

FR ANALYSIS 11/15/88

This WATAND analysis command performs small-signal frequency-domain analysis at an operating point.

```
+------+----------------------------------------------------------------+
| FR   | NOne             OUtputs V p n         BEgin bgn  ENd end      |
|      | PRint n                  V p                                   |
|      | PPrinter                 V type.n      LInear   incr           |
|      | PTermina n m             I type.n      LOgarith npts           |
|      | PLottekt n m             G gname                               |
|      | POverlay n m                                                   |
|      | PVolker n                               <options ...>          |
|      | CAll rname ...                                                 |
+------+----------------------------------------------------------------+
```

```
options:    (defaults listed first)

            VSFreque | VSOmega
            TRue | GAin
            MAgnitud | DB | PHase | REal | IMaginar | COmplex | CPolar
            XAxis  DEfault | out-spec
            VBounds  vl  vh
            IBounds  il  ih
            XBounds  xl  xh
            IPoint   ipname
            KEep  ALl          IN  FR      ON  A
                  OUtputs          fname       fmode
                  out-specs
                  NOne
```

where: (defaults)

NONE ...	tells WATAND what to do with output. See 'OUTPUT' helps.
OUTPUTS	specifies which outputs to display. See 'OUTPUT' helps.
BEGIN END	specifies the beginning and ending frequencies for the analysis. Must have bgn > 0 and end > bgn.
LINEAR LOGARITH	Kind of frequency increment followed by increment or number of points per decade. Must have incr > 0 or npts >= 1.
VSFREQUE VSOMEGA	BEGIN, END, and LINEAR incr values in Hz. (Default) BEGIN, END, and LINEAR incr values in radians.

TRUE	True amplitude respose to all SIN/COS sources. (Default)
GAIN	Transfer response to one sinusoidal source.
MAGNITUD	Absolute values of complex output displayed. (Default)
DB	Decibel values of output displayed, i.e., 20*log(|output|).
PHASE	Phase in radians (-PI to +PI) displayed.
REAL	Real part of output values displayed.
IMAGINAR	Imaginary part of output values displayed.
COMPLEX CPOLAR	For PR output, real and imaginary (or magnitude and phase) parts of output values are displayed. For plots, the real part is the x-axis value (XA DE must be in effect) and the imaginary part is the y-axis value.
XAXIS	The x-axis value for plots. May be any output value. (DEFAULT)
VBOUNDS	The y-axis voltage minimum and maximum values for plots.
IBOUNDS	The y-axis current minimum and maximum values for plots.
XBOUNDS	The x-axis minimum and maximum values for plots if XA DE is not in effect.
IPOINT	The initial-point name which defines the point around which nonlinear circuits are linearized. (ipname=ZERO for linear, ipname=DC for nonlinear)
KEEP	specifies keeping of the results of the analysis for post-processing. General form is 'KE what IN fname ON fmode'. (what=AL) (fname=FR) (fmode=A)

Notes:

1. For PHase, IMaginar, COmplex, or CPolar output, sine sources will be shifted in phase since cosine is the reference.

2. For linear circuits the IP specification is immaterial (default is IP ZERO). For nonlinear circuits the initial point (default IP DC) specifies the operating point around which the circuit is linearized.

1.2.2 DI POST-PROCESSOR

This Watand post-processor command reads kept results from the disk and then displays them according to parameters entered. The Watand help file for the DI post-processor [3] is shown in Table 1-2. For details, see Ref. [1].

1.2.3 WEXEC MACROS

A wexec macro is most commonly used as a command in the WEXEC interactive environment, but it may also be called from the #Execute section of a Watand source file [4]. It can be used to input a sequence of lines, but it can also be used to check values, create and use variables, branch conditionally, etc.

Detailed information about the macro language for creating an wexec macro may be found in Ref. [4]. One especially important macro command for the MCFR and MCDI commands is &CALL. See Table 1-3 for a summary of this command's action.

1.2.4 REPEATING AN ANALYSIS WITH #RUSER

The #RUSER command is used to repeat an analysis a number of times. Command #RUSER does not alter an element's value in each individual run as #REPEAT does, but instead calls a user subroutine before each analysis run.

Table 1-2

DI Analysis Help Menu [3]

DI POST-PROCESSOR 2/08/89

This WATAND post-processor command reads kept results from disk and then displays them according to the original keeping analysis and parameters entered.

```
+------+-------------------------------------------------------------+
| DI   |  NOne                  OUtputs V p n        < options ... > |
|      |  PRint n                       V p                          |
|      |  PPrinter                      V type.n                     |
|      |  PTermina n m                  I type.n                     |
|      |  PLottekt n m                  G gname                      |
|      |  POverlay n m                                               |
|      |  PVolker n             USe fname < ON fmode >               |
|      |  CAll rname ...                                             |
+------+-------------------------------------------------------------+
```

```
options:   (defaults listed first)

           MAgnitud | DB | PHase | REal | IMaginar | COmplex | CPolar
           XAxis DEfault | out-spec
           VBounds vl vh
           IBounds il ih
           XBounds xl xh
           ZOom
           KEep  OUtputs  IN DI    ON A
                 NOne        fname    fmode
```

where: (defaults)

NONE ...	tells WATAND what to do with output. See 'OUTPUT' helps.
OUTPUTS	specifies which outputs to display. See 'OUTPUT' helps.
USE	specifies the kept file to use. (fmode=A)
MAGNITUD	Absolute values of complex output displayed. (Default)
DB	Decibel values of output displayed, i.e., 20*log(|output|).
PHASE	Phase in radians (-PI to +PI) displayed.
REAL	Real part of output values displayed.
IMAGINAR	Imaginary part of output values displayed.
COMPLEX	Real and imaginary values of output are displayed.
CPOLAR	Magnitude and phase of output are displayed.

XAXIS	The x-axis value for plots. May be any output value. (DEFAULT)
VBOUNDS	The y-axis voltage minimum and maximum values for plots.
IBOUNDS	The y-axis current minimum and maximum values for plots.
XBOUNDS	The x-axis minimum and maximum values for plots if XA DE is not in effect.
ZOOM	is used in the WATAND environment to redefine a viewing window. See note below.
KEEP	specifies keeping of the results for further post-processing or subsequent display. General form is 'KE what IN fname ON fmode'. (what=OUTPUTS) (fname=DI) (fmode=A)

Notes:

1. For PHase, IMaginar, COmplex, or CPolar output, sine sources will be shifted in phase since cosine is the reference.

2. Plot bounds need not be specified. DI selects bounds automatically. To restore automatic bounds selection after other settings of VB, IB, and/or XB were made, specify equal high and low bounds (eg. VB 0 0).

3. The ZOOM option is used to display a portion of a plot. This option is designed for Tektronix graphics terminals using PL and PO output action. Enter 'DI ZOom' after the initial display appears. Position the crosshair cursor at each corner of the desired window and press any key except the ENTER (or carriage return) key.

4. When using previous DF or FS results, 'DI KEep' is not allowed since no new outputs are possible.

Both #REPEAT and #RUSER open a keep file during the repetitions if the analysis specifies the KEEP option. Therefore, anything written to the keep file during the repetitions is together in one file. A help menu for the usage of the #RUSER command is shown on Table 1-4 [4].

Table 1-3

&CALL Macro Directive [4]

```
+-----------------------------------------------------------------+
|  &CALL  rname  outvals  < in1 in2 ... >                          |
+-----------------------------------------------------------------+
```

The &CALL directive shifts execution to a user's compiled subroutine. The prefix '&' is not used when specifying variable names outvals, in1, in2, etc. The arguments of this directive are as follows:

rname — is the name of the subroutine which Watand calls.

outvals — is the output variable name. If more than one value is returned by the subroutine, &outvals is indexed accordingly.

in1 ... — are values may be passed to the subroutine as input. All the values of each &in1, &in2, etc., are passed and indices may not be specified.

The values of &outvals and &in must be numeric. The subroutine must contain coding similar to:

```
SUBROUTINE rname(IN,NIN,OUT,NOUT)
INTEGER NIN,NOUT
REAL*8 IN(1),OUT(1)
  test NIN and NOUT, use IN, and
  evaluate values of OUT.
NOUT=number of values in OUT
RETURN
END
```

where

IN — is an array of the input variable values.
NIN — is set by Watand to the number of values in the array IN.
OUT — is evaluated and set by the subroutine.
NOUT — is initialized by Watand to the maximum possible number of OUT values. On exit from the routine, it must be set by the subroutine to the actual number of values returned in OUT.

Watand sets NIN to 0 when no &in values are entered. Likewise, the subroutine may set NOUT to 0 to indicate that &outvals is empty.

For an example of non-zero NIN, suppose &in1(1) = 1D2, &in1(2) = 2D4, and &in2 = 27.5. The values of IN passed to the subroutine will be IN(1) = 1D2, IN(2) = 2D4, and IN(3) = 27.5. The value of NIN = 3.

Table 1-4

#RUser Control Word [4]

```
+---------------------------------------------------------------------+
|   #RUser nrep rname val1 val2 ... analysis-command ...              |
+---------------------------------------------------------------------+
```

where:

nrep	is the of times that the Watand analysis is repeated.
rname	is the name of the subroutine that is called before each repetition of the Watand analysis.
val1 ...	are numeric values which can be passed to the subroutine.
anal...	is the normal format of any analysis except DI, FO, PZ, SY, and user-defined analyses, which may not be repeated with #RUser.

The subroutine must contain coding similar to:

```
      SUBROUTINE rname(IREP,PAR)
      INTEGER IREP
      REAL*8 PAR(1)
      IF (IREP.LT.0) GOTO 700
        do the normal execution
        ...
      GOTO 999
 700    prepare for exit due to attention
        code, ATTCD, or error interrupt.
 999  RETURN
      END
```

where

IREP	is the repetition number telling which repetition is about to be performed - 1st, 2nd, etc. In the case of a user interrupt, Watand sets IREP = -1 for a final call to rname.
PAR	is an array of the values passed to the subroutine.

1.2.5 #OPTION COMMANDS

Options are available in Watand to control the operation, which affects the Watand environment [3]. Two options commands are used in the wexec macros which perform the Monte-Carlo FR analysis and post-processing. They are ANID and TIMEST.

The option command, ANID ON|OFF, controls the printing of the analysis ID (including version number and date) at the beginning of an analysis output. When ANID is set ON, which is the default, the analysis name, current Watand version number, current date, and the TABLE file name of the analysis are printed at the user's terminal.

The other option command is TIMESTAM ON|OFF. TIMESTAM controls the printing of the execution time of an analysis at the end of an analysis. If TIMESTAM is set ON, the default condition, the execution time stamp of the analysis is printed. Otherwise, TIMESTAM is set OFF and the time stamp will not be displayed.

1.2.6 FORTRAN AND ASSEMBLER SUBROUTINES

There are several subroutines written in Fortran and in Assembler in the Watand package that are used directly in the Monte-Carlo FR analysis and its post-processor (e.g., DCMAIN, DISPLA). Using these subroutines saves computer memory because all of these subroutines are loaded in computer memory already and Watand allows such use.

1.3 MEMORY CLASSIFICATION IN WATAND PACKAGE

Generally speaking, there are four types of memory used for Watand package. The first is what will be called the Watand table memory. It is a block consisting of 1200 REAL*8, equivalenced to 2400 INTEGER*4, memory locations named RDATA/IDATA. By calling the Watand subroutine GSTORE, CSTORE, and/or FSTORE, the user can define any amount (less or equal to the current largest Watand table storage block) of Watand table storage in the user's subroutine. This storage can be re-used in another user's subroutine after a proper storage release operation on that storage block.

For the CMS system, command 'WE fname(storage nn' allows Watand access to nn percent of the CMS storage which is free after the WEXEC module is loaded. The default is nn = 0, and nn may be an integer from 0 to 99. Therefore, the Watand table storage can be extended into free CMS storage with this option. This option finds use for very large circuits and for special user subroutines where a large amount of IDATA/RDATA storage may be needed. However, the amount of nn% of the CMS storage is not always greater than that of Watand default table memory (1200 REAL*8). It depends on the size of the CMS storage and the nn value. Storage shortage problems might occur if this option is not properly used.

The second type of storage which may be used is called macro storage. There are up to 750 REAL*8 memory locations in the macro storage. The macro storage is available during the execution of the macro facilities. Macro variables are stored in literal format.

The third type of storage which may be used is called disk storage. The amount of this kind of storage depends on user's definition. Compared to the other kinds of memory, disk storage has the lowest operation speed.

The last type of storage which may be used is called subroutine memory. This kind of memory is created by the declaration instructions of any Fortran subroutines, for example, REAL*8, INTEGER, etc. Therefore, these memory locations do not necessarily occupy consecutive memory blocks. They may be discrete in their locations distribution. Moreover, once they are reserved for a Fortran subroutine, they can't be used for any other Fortran subroutines. This is different from that of the Watand table storage.

CHAPTER II

IMPLEMENTATION OF MONTE-CARLO FR ANALYSIS

2.1 OVERVIEW

One wexec macro and eight Fortran subroutines are used in the MCFR analysis. The wexec macro is the main control program and is named MCFR WEXEC. The eight Fortran subroutines which work with MCFR WEXEC are MDCHCK, MDINTL, MDREAD, MFRUSR, MFRNEW, MFDADS, MFSTAS and MFTIME. The functions of this wexec macro are described in this chapter. Fortran subroutines are described in Chapter IV.

Three rules were always kept in mind during the development of the MCFR analysis. The first is to achieve the maximum execution speed of the analysis. The second one is to make memory used in the analysis a minimum. The last one is to provide the user the greatest flexibility when using the MCFR analysis.

Due to the large amount of data to be processed, a large memory block is required. Disk and Watand table storage are the two possible ways to fulfill this need. The memory selection has been carefully considered to achieve the maximum execution speed. Watand table storage was considered the first choice except when the data that needs to be stored will be re-processed later on.

By the use of dynamic Watand table storage access and release, the storage block used in one part of the wexec program can be reused at the other part of the wexec program. At the end of the MCFR analysis these temporary storage blocks are released completely. Thus, memory needed for the MCFR analysis is kept to a minimum.

All eight Watand output actions (NONE, PRINT, PPRINTER, PTERMINA, PLOTTEKT, POVERLAY, PVOLKER and CALL) are available for MCFR analysis [1]. Five output types (MAGNITUD, dB, PHASE, REAL and IMAGINAR) are enabled [1]. Furthermore, six analysis modes (ALL, INI, MIN, MAX, AVG, and STD) are provided. The MCFR input format retains Watand's spirit and is easy to use. All of these features mentioned above give the user the greatest flexibility and the most convenience in using the MCFR analysis.

Watand version V1.11-03 with YSU changes (V.1.11-3g) and later are used for the MCFR analysis. Since the new Watand version is compiled with VS-Fortran, the programming of the MCFR analysis uses the Fortran 77 language with the VS-Fortran compiler.

2.2 PROGRAMS COMPRISING MCFR ANALYSIS

The MCFR analysis is based on the work of J. Suen [6] who developed the MCDC analysis. Some of the Fortran subroutines created for the MCDC analysis have been changed so that they can be used either in the MCFR analysis or in the MCDC analysis. These Fortran subroutines are MDCHCK, MDINTL and MDREAD. Subroutines MFRNEW and MFDADS are basically derived from MCDC's subroutine MDRUSR. For more detailed description, see Chapter IV.

2.3 MCFR WEXEC MACRO

MCFR WEXEC macro program is the main routine that controls the MCFR analysis. The macro program is shown in Table 2-1.

Lines 1 and 2 in the MCFR WEXEC macro define the input-line parameter names and assign default values for the parameters DSEED, NS, IPDC and AS.

Subroutine MDCHCK is called for the first time in the macro at line 6 with &ICODE = 0 as it's input. Two jobs are done by this subroutine:

1. Check to be sure the macro is executing in the Watand WE environment.
2. Check the macro memory.

Variable &OUT is set to empty if current macro memory is not enough (less than one location) for MDCHCK execution. If variable &OUT(1) returned from MDCHCK is not equal to 0, the MCFR analysis is not executing in the WE environment. Therefore, the MCFR analysis is forced to stop by branching to line 133 (-ERR7) where an error message will be displayed.

Line 11 checks for positional parameters in the MCFR argument list. If any of positional parameter is found, the MCFR execution branches to line 121 (-ERR1) where an error message is displayed and the execution is stopped there. The parameter DSEED cannot be a number less than zero. This is checked on line 12. An error message is displayed and the execution stop if a negative DSEED is

Table 2-1

MCFR Wexc Macro

Program

```
&PARAM DSEED NS WRTD WRTM FR IPDC AS
&DEFAULT DSEED 0 NS 10 IPDC OFF AS ALL
#*
#* Check environment.
&ICODE = 0
&CALL MDCHCK OUT ICODE
&IF &OUT EM &GOTO -ERR8
&IF &OUT NE 0 &GOTO -ERR7
#*
#* Check position parameter, DSEED, NS, WRTD and WRTM.
&IF &1 NM &GOTO -ERR1
&IF &DSEED LT 0 &GOTO -ERR2
&IF &NS LT 1 &GOTO -ERR3
&IF &WRTD EM &IF &WRTM EM &GOTO -ERR4
#*
#* Encode AS into numerical data
&IF &AS(2) NM &GOTO -ERR9
&MODE = 0
&IF &AS EQ INI &MODE = 1
&IF &AS EQ MIN &MODE = 2
&IF &AS EQ MAX &MODE = 3
&IF &AS EQ AVG &MODE = 4
&IF &AS EQ STD &MODE = 5
&IF &AS EQ ALL &MODE = 6
&IF &MODE EQ 0 &GOTO -ERR9
#*
#* Disable MCDI FR ZO option.
&CKZOFL = DSAZO
&GLOBAL CKZOFL
#*
#* Update the FR analysis and indicate error if any.
#E FR KE ALL IN MCFR VB 1 2 IB 1 2 *
 &FR |
#E DI US MCFR NO OU KE OU IN MFST
#*
#* Get FR ready flag, output type, linear code and action code.
&ICODE = 3
&IF &MODE EQ 6 &ICODE = -3
&CALL MDCHCK OUT ICODE
&IF &OUT(1) EQ -1 &EXIT
&IF &OUT(2) EQ 5 &GOTO -ERR5
&IF &OUT(2) EQ 7 &GOTO -ERR5
&ACTION = &OUT(4)
```

```
#*
#* Check KEEP.
&ICODE = 3
&CALL MDINTL RSLT ICODE
&IF &RSLT(1) EQ -2 &GOTO -ERR6
#*
#* Set DI outputs the same as FR outputs, US file(#RU) KE in MFST
&DCODE = 12
&CALL MDCHCK DMMY ICODE DCODE
#*
#* Encode parameters following WRTD/WRTM.
&CALL MDREAD PAR RSLT
&IF &PAR(1) EQ -1 &EXIT
#*
#* Turn off analysis ID and time stamp before go. Encode IPDC.
#O ANID OFF TIMEST OFF
&RSLT(2) = 0
&RSLT(3) = &ACTION
&IF &IPDC EQ OFF &GOTO -GO
&IF &OUT(3) LT 1 &GOTO -GO
&RSLT(2) = 1
#E DC NO KE NO IP DC
-GO
#*
#* One more time to put initial values back to elements altered.
&NSS = &NS + 1
#RU &NSS MFRUSR &RSLT &NS &DSEED &PAR FR NO
#*
&IER = 0
&IF &ACTION EQ 1 &GOTO -NO
#*
#* Calculate and display the statistical results of the output.
&RSLT(1) = &MODE
&RSLT(2) = &NS
&IF &MODE EQ 6 &IF &ACTION NE 2 &GOTO -TWO
&CALL MFSTAS IER RSLT
&IF &IER EM &GOTO -ERR8
&IF &IER EQ -1 &EXIT
&IF &ACTION EQ 2 &GOTO -NO
&GOTO -SET
-TWO
#*
#* Use generalized outputs' name to create the title.
#GV INITIAL 0
#GV MINIMUM 0
#GV MAXIMUM 0
#GV MEAN 0
#GI ST.DEV 0
#*
#* MFST initialization(MFST WKEEPID).
```

```
&CALL MFSTAS DIPAR RSLT
&IF &DIPAR EM &GOTO -ERR8
&IF &DIPAR EQ -1 &EXIT
#E DI OU G INITIAL G MINIMUM G MAXIMUM G MEAN G ST.DEV
&RSLT(1) = -6
#*
#* Fill in MFST WKEEP.
&CALL MFSTAS IER RSLT DIPAR
&IF &IER EM &GOTO -ERR8
&IF &IER EQ -1 &EXIT
#E DI OU G INITIAL G MINIMUM G MAXIMUM G MEAN G ST.DEV
-SET
#E DI US MFST
-NO
&MESSAGE
#E DI KE NO IN DI
#E FR KE NO IN FR
#*
#* Perform PP/PO/PT/PL/PV/CALL action. Display time stamp.
&RSLT(1) = 1
&CALL MFTIME DMMY RSLT
#*
#* Job done. Turn on analysis ID and time stamp before exit.
#O ANID ON TIMEST ON
&IF &IER ZE &EXIT
&WARNING ONLY &IER OUTPUT SAMPLES USED - INITIAL VALUES LOST
&EXIT
-ERR1 &ERROR PARAMETER &1 NOT RECOGNIZED
&EXIT
-ERR2 &ERROR DSEED CANNOT BE LESS THAN ZERO
&EXIT
-ERR3 &ERROR NS MUST BE GREATER THAN ZERO
&EXIT
-ERR4 &ERROR MISSING WRTD/WRTM SPECIFICATION
&EXIT
-ERR5 &ERROR CO/CP OUTPUT TYPE NOT AVAILABLE FOR MCFR
&EXIT
-ERR6 &ERROR MCFR ANALYSIS MUST USE 'KE OU/ALL' ONLY
&EXIT
-ERR7 &ERROR MCFR ANALYSIS VALID IN WEXEC ENVIRONMENT ONLY
&EXIT
-ERR8 &ERROR INSUFFICIENT STORAGE
&EXIT
-ERR9 &ERROR AS PARAMETER '&AS' INVALID
&EXIT
#* 02-FEB-1989  BY  D. YON    EE DEPT.  YSU.
```

detected. The parameter NS is examined on line 13 and is required to be a number greater than or equal to 1. If this requirement is not satisfied, an error message is displayed and the execution stop by branching the execution to line 125 (-ERR3). Line 14 makes sure that at least one of WRTD/WRTM parameters is entered. If both WRTD and WRTM are empty, the execution branches to line 127 (-ERR4), displays an error message, and stops the MCFR analysis.

Lines 17 to 25 decode parameter AS into numerical data which defines the MCFR analysis mode. Parameter AS is required to be equal to one of the six key words -- INI/MIN/MAX/AVG/STD/ALL. If none of these six key words, or more than one of these six key words are entered, the MCFR execution branches to line 137 (-ERR9). On line 137, an error message indicates that parameter AS is incorrectly entered.

Line 28 disables the Monte-Carlo zoom option by setting the check flag to DSAZO. The value is then made available to the other wexec macros on line 29.

On lines 32 and 33, the FR analysis specification is updated to the user's new specification if the parameter FR is not empty. On line 34, the DI analysis is partly defined to cooperate with the FR analysis.

Subroutine MDCHCK is called for the second time on line 39. At this time MDCHCK checks whether the FR analysis is ready or not. Variable &ICODE is set equal to 3 or -3 to tell subroutine MDCHCK to check the FR analysis specifications. If there is any problem that indicates the FR analysis is not ready to run, variable &OUT(1) is set to -1 by MDCHCK. A -1 value in &OUT(1) causes the MCFR execution to stop immediately. In this case, Watand will have already displayed any related error message to indicate why the FR analysis is not ready to run. With the help of this error message, the user will know how to correct the error and redefine the FR analysis specification. MCFR analysis's output type and action code are read out here, too. Line 41 and line 42 examine these parameters and make sure that COMPLEX and CPOLAR output types are not entered by the user. Also, a warning message '## WARNING ## MORE THAN 1 OUTPUT;REST IGNORED' will be displayed by MDCHCK if the number of outputs is greater than one when AS ALL is specified.

The KEEP option is checked by calling MDINTL on line 47. To indicate that the FR analysis KEEP option is to be checked, &ICODE is set to equal to 3 on line 46. MCFR analysis requires either KE OU or KE ALL is in effect, but not both. Otherwise, the macro branches to line 131 (-ERR6) where an error message is displayed and the execution is stopped. Parameters follow WRTD and/or WRTM are also checked in this subroutine.

On line 52, subroutine MDCHCK is called for the third time to define DI post-processor. DI post-processor's outputs are set by this subroutine to be the same as FR analysis's outputs. The USE file name for DI is set to the file name where the FR analysis will keep its results. The disk mode for DI USE/KEEP is set to be the same as the disk mode for FR KEEP.

On line 55, subroutine MDREAD encodes the element/parameter values following parameter WRTD/WRTM into element/parameter pointers. Each pointer consists of three integer and two real numbers for each element/parameter. These pointers are stored into a memory block obtained from Watand table memory storage. The memory block's pointer and its size are returned in the output parameter &PAR if there is no error associated with the elements information. If there is any error during the pointers encoding such as incorrect parameter, an error message is displayed by the MDREAD subroutine and &PAR(1) is assigned a -1 to stop the execution. If the virtual storage defined is not enough to store the pointers, &PAR is set to empty to branch the execution to line 135 (-ERR8). On line 135, an error message is displayed and the execution is stopped.

Line 59 turns off ANID and TIMEST to prevent the analysis ID and the execution time stamp from displaying for each individual run of the repeated FR analysis.

Line 61 puts the &ACTION value into variable &RSLT(3) for transmission into subroutine MFRUSR during Watand #RUSER command on line 70.

Lines 62 through 65 check the parameter IPDC. If IPDC is not off and the circuit under analysis is a non-linear one, variable &RSLT(2) will be set to equal to 1 to force MCFR analysis to re-calculate the DC solution (the bias point for FR analysis) immediately before each individual FR run.

Line 70 executes #RUSER Watand control command which repeats FR analysis &NS + 1 times. MFRUSR is called immediately before each FR analysis. MFRUSR replaces elements\parameters with random values according to the statistical distribution function. The normal distribution function is the default one. However, the user is allowed to define his/her own statistical distribution function by creating a new subroutine MDRAND [6]. The repeated FR analyses outputs are stored on the disk under the file name and file mode defined by the user. MCFR analysis executes one time more than &NS. The extra FR execution allows MCFR to restore the original elements/parameters values and to calculate the initial response of the FR analysis with normal element/parameter values.

Line 72 initials variable &IER, the error flag, to 0. Line 73 branches the execution to line 107 (-NO) if output action NONE was specified. In that case, no statistical results calculated or generated. Only all of the sampling outputs were kept on disk.

If any output action except NONE is specified by the user, the MCFR execution continues on line 76. Lines 76 and 77 transfer the values of variables &MODE and &NS into variables &RSLT so that their values can be available in the subroutine MFSTAS.

Subroutine MFSTAS needs to be called twice if AS ALL and actions other than NONE and PRINT are specified. Otherwise, subroutine MFSTAS is called once only. Line 78 examines this condition and does the necessary branching.

If the conditions on line 78 are not true, line 79 will be executed. Subroutine MFSTAS is called on line 79 to calculate the statistical results for the output and print the results on the user's terminal when action PRINT is specified. If action PRINT is not specified, two files named MFST WKEEPID and MFST WKEEP will be created on disk by subroutine MFSTAS for further processing. Line 80 tests whether the current Watand table storage is adequate for subroutine MFSTAS to calculate statistical results or not. If the Watand table storage is not enough for the current analysis set up, variable &IER is empty and a branch to line 105 (-SET) occurs. On the other hand, if there is any error during the I/O read/write in MFSTAS, -1 is assigned to &IER to stop the execution immediately. If output action PRINT (&ACTION=2) is specified, then output action has been completed by subroutine MFSTAS on line 79. So, the execution branches to line 107 (-NO). Otherwise, the execution branches to line 105 (-SET) where the file name for DI USE is re-defined.

If conditions on line 78 are true, which means AS ALL and output action except NONE and PRINT are specified, the execution branches to line 84 (-TWO). Line 87 through line 91 create five special output names (INITIAL, MINIMUM, MAXIMUM, MEAN, ST.DEV) by using Watand's generalized outputs option. The only purpose of these lines is to create these nameplates. The values assigned to these generalized outputs are not important. On line 94, subroutine MFSTAS is called for the first time in this case to store the parameters in DI OU. Lines 95 and 96 examine storage problems and I/O read/write errors, respectively. Line 97 defines DI OU and makes Watand think that these five generalized outputs are going to be kept. On line 98, variable &RSLT(1) is assigned a value -6 to indicate subroutine MFSTAS is being called the second time. On line 101, subroutine MFSTAS is called. At this time, a file named MFST WKEEPID (with the five generalized output names) is created first. Then, user's OU parameters are restored into DI analysis data structure for proper output reading. After statistical calculation, these outputs are transformed into statistical results and stored into the file MFST WKEEP. Line 103 examines I/O read/write errors, if any. DI OU is set to the five generalized outputs again on line 104 to prepare for actions performed by the DI post-processor inside the subroutine MFTIME on line 114. The DI analysis will then use and display the results of MFST files.

A blank line will be typed on the user terminal by line 108 in the wexec macro. Line 109 and line 110 reset FR and DI analyses back to the default KEEP specification option. On line 113, variable &RSLT(1) is assigned a value 1 to indicate to subroutine MFTIME the current wexec macro is MCFR. Subroutine MFTIME is called on line 114 to perform Watand output actions (except NONE/PRINT) and to print the time stamp.

On line 117, MCFR analysis has completed. So, the analysis ID and the time stamp are set on before exiting from the macro.

Lines 118 and 119 check whether the sample number read and processed by subroutine MFSTAS agrees with the user's sample number (&NS) specification or not. A normal exit occurs if they agree, or a warning message will be displayed to report the sample number returned from subroutine MFSTAS.

any of the six Monte-Carlo analysis modes. In addition, a option, ZOOM, is also provided for the post-processor. With the ZOOM option, the user is able to display a portion of a Tektronix plot.

A minor change in the input format of the MCDC post-processor has been made so that it can be called by the new MCDI wexec macro. The new input format is to enter a key word DC in addition to the old input format. The key word DC can be entered with free format (i.e., key word DC is a non-positional variable).

3.2 MCDI WEXEC MACRO

MCDI WEXEC is the main control macro for the Monte-Carlo post-processor. It does the necessary branching and data transferring for Monte-Carlo DC, FR, and ZO post-processing. The macro program is shown in Table 3-1.

Lines 3 and 4 in MCDI WEXEC assign default values and define the parameter names for the parameters DC, FR, and ZO. The default values for these three parameters are not important. The point is to assign an unusual value to each of these three parameters so that any change in the parameters can be detected. Such a change would occur when the parameter name is entered with no input or with specific input.

Line 7 checks for the key word ZO. If the key word ZO does not show on the input line, &ZO remains at the default value, YON, and the execution branches to line 18

Table 3-1

MCDI Wexc Macro

Program

```
#*
#* Monte Carlo analysis post-processor.
&PARAM DC FR ZO
&DEFAULT DC $SUEN$ FR $YON$ ZO $YON$
#*
#* MCDI ZO?
&IF &ZO EQ $YON$ &GOTO -NOZO
#*
#* Any error on user's input command?
&IF &1 NM &GOTO -ERR1
&IF &DC EQ $SUEN$ &IF &FR EQ $YON$ &IF &ZO EM &GOTO -ZOOM
&IF &ZO EM &GOTO -ERR1
&POSITION = &ZO
&GOTO -ERR2
-ZOOM
&STACK MFRZO
&EXIT
-NOZO
#*
#* Branching
&IF &DC NE $SUEN$ &GOTO -DIDC
&IF &FR NE $YON$ &GOTO -DIFR
&GOTO -ERR1
-DIDC
&IF &FR NE $YON$ &GOTO -ERR1
&STACK MDCDI &POSITION &DC
&EXIT
-DIFR
&STACK MFRDI &POSITION FR &FR
&EXIT
-ERR1 &ERROR Invalid format - Must be: MCDI ZO or MCDI DC|FR ...
&EXIT
-ERR2 &ERROR PARAMETER '&POSITION' NOT RECOGNIZED
&EXIT
#* 02-FEB-1989  BY  D. YON    EE DEPT.  YSU.
```

(-NOZO). On the other hand, if the key word ZO does show on the input line, the execution continues on line 10.

If the execution branches to line 18 (-NOZO), the macro is preparing to branch to either MCDI FR or MCDI DC.

Line 21 checks for the key word DC. If the key word DC is entered by the user, &DC will not remain at its default value, $SUEN$, so the execution branches to line 24 (-DIDC). Line 25 then checks if the key word FR is also entered. If key word FR is entered, &FR not equal to YON, the execution branches to line 31 (-ERR1) to display an error message and then to stop the execution. Otherwise, the execution continues on line 26. On line 26, the post-processor for MCDC analysis, MDCDI WEXEC, is called to perform the MCDC post processing.

On the other hand, if the user does not enter key word DC, the execution continues on line 22. Line 22 checks whether or not the key word FR is entered. The execution continues on line 28 (-DIFR) if key word FR is entered by the user. Otherwise, neither FR nor DC was entered. Line 23 detects this error and branches the execution to line 31 (-ERR1) to display an error message and to stop the execution. However, if the execution continues on line 28 (-DIFR), MFRDI WEXEC is called to perform the MCFR post-processing.

In the case of zoom branching from line 7, the execution continues on line 10. On line 10, the positional parameter is checked. No positional variables are allowed in the zoom option. If there are any positional variables entered by the user (&1 not empty), the execution branches to line 31 (-ERR1). On the other hand (&1 is empty), the execution will continue on line 11.

Line 11 makes sure that none of the key words (DC and FR) is entered on the input line. If neither DC nor FR is entered and there is nothing following the key word ZO (i.e., &ZO is empty), the execution branches to line 15 (-ZOOM) where the command MFRZO is stacked to perform the zoom option immediately after exiting from the MCDI wexec macro. Otherwise, the execution continues on line 12.

Two types of errors can cause the execution to continue on line 12. One is that at least one of the key words (DC and FR) is entered beside key word ZO by the user. The other is that something unexpected is entered following the key word ZO on the user's input line.

Line 12 branches the execution to line 31 (-ERR1) to indicate to the user that at least one of the key words (DC and FR) is shown on the user's input line (&ZO is empty). Otherwise (&ZO is not empty), the execution continues on line 13. This is the case when something unexpected following key word ZO is found on the user's input line.

On line 13, the error input content (&ZO) is first transferred into the variable &POSITION. Then, line 14 branches the execution to line 34 (-ERR2) to display the error input.

3.3 MFRDI WEXEC MACRO

The MFRDI WEXEC macro is used as the control macro to perform the MCDI FR post-processor which displays the input data and the statistical results of the MCFR analysis. This macro is designed to be called by MCDI WEXEC macro. The program of the MFRDI WEXEC macro is shown in Table 3-2.

Lines 1 and 2 in the MFRDI WEXEC macro define the parameter names and assign default values for the parameters FR, NOIN, and AS.

Line 9 makes sure that there are no positional variables entered by the user. If there are any positional variable entered by the user, the execution branches to line 116 (-ERR1) to display an error message and to stop the execution.

On line 6, subroutine MDCHCK is called for the first time in the macro with &ICODE=0 as its input. MDCHCK checks if the macro is executing inside the Watand WE environment. It will set &OUT=0 if it is in the Watand WE environment, or &OUT=-1 if it is not in the Watand WE environment. Line 8 branches the execution to line 120 (-ERR3) if the current environment is not in the Watand WE (&OUT = -1). Otherwise (&OUT = 0), the execution continues on line 12.

If the current macro storage capacity is less than 1 during the execution of MDCHCK, &OUT is set to empty by subroutine MDCHCK. The instruction on line 7 checks for this condition and does the necessary branching. The

Table 3-2

MFRDI Wexc Macro

Program

```
&PARAM AS FR NOIN
&DEFAULT NOIN EMTY AS ALL
#*
#* Check environment and storage
&ICODE = 0
&CALL MDCHCK OUT ICODE
&IF &OUT EM &GOTO -ERR2
&IF &OUT NE 0 &GOTO -ERR3
&IF &1 NM &GOTO -ERR1
#*
#* Set up INPUT on/off code. This controls subroutine MDINPT's action
&IF &NOIN NE EMTY &IF &NOIN NM &GOTO -ERR6
&UPCODE = 0
&IF &NOIN NM &UPCODE = 1
#*
#* Encode AS into numerical data
&IF &AS(2) NM &GOTO -ERR5
&MODE = 0
&IF &AS EQ INI &MODE = 1
&IF &AS EQ MIN &MODE = 2
&IF &AS EQ MAX &MODE = 3
&IF &AS EQ AVG &MODE = 4
&IF &AS EQ STD &MODE = 5
&IF &AS EQ ALL &MODE = 6
&IF &MODE EQ 0 &GOTO -ERR5
#*
#* Disable MCDI ZO.
&CKZOFL = DSAZO
&GLOBAL CKZOFL
#*
#* Update the FR analysis and indicate error if any.
#E DI &FR
#*
#* Get DI ready flag. Check output type code and store action code.
&PCODE = 12
&IF &MODE EQ 6 &PCODE = -12
&CALL MDCHCK OUT PCODE
&IF &OUT EM &GOTO -ERR2
&IF &OUT(1) LT 0 &EXIT
&IF &OUT(2) EQ 5 &GOTO -ERR4
&IF &OUT(2) EQ 7 &GOTO -ERR4
&ACTION = &OUT(4)
#*
#* Display element statistical data if &UPCODE=1
&PCODE = 12
```

```
&CALL MDINPT INPT UPCODE PCODE PCODE
&IF &INPT EQ -1 &EXIT
#*
&IER = 0
&PTR = 0
&IF &ACTION EQ 1 &GOTO -NO
#*
#* Initialize global parameters for DI.
#O ANID OFF TIMEST OFF
DI NO KE NO
#O ANID ON TIMEST ON
#E DI &FR KE OU IN MFST
&CALL MDCHCK DMMY PCODE ACTION
#*
#* Calculate and display the statistical results for the output.
&INPT(1) = &MODE
&INPT(2) = &INPT(3)
&IF &MODE EQ 6 &IF &ACTION GT 2 &GOTO -TWO
&DIPAR = 0
&CALL MFSTAS IER INPT
&IF &IER EM &GOTO -ERR2
&IF &IER EQ -1 &EXIT
&IF &ACTION EQ 2 &GOTO -NO
&GOTO -SET
-TWO
#*
#* Use generalized outputs' name to create the title.
#GV INITIAL 0
#GV MINIMUM 0
#GV MAXIMUM 0
#GV MEAN 0
#GI ST.DEV 0
#*
#* MFST initialization (MFST WKEEPID).
&CALL MFSTAS DIPAR INPT
&IF &DIPAR EM &GOTO -ERR2
&IF &DIPAR EQ -1 &EXIT
#E DI OU G INITIAL G MINIMUM G MAXIMUM G MEAN G ST.DEV
&INPT(1) = -6
#*
#* Fill in MFST WKEEP file.
&CALL MFSTAS IER INPT DIPAR
&IF &IER EM &GOTO -ERR2
&IF &IER EQ -1 &EXIT
#E DI OU G INITIAL G MINIMUM G MAXIMUM G MEAN G ST.DEV
-SET
#*
#* Store file name for DI US.
&CALL MDCHCK PTR PCODE ICODE
&IF &PTR EM &GOTO -ERR2
#E DI US MFST
-NO
&MESSAGE
```

```
099 #E DI KE NO IN DI
100 #*
101 #* Perform PP/PT/PL/PO/PV/CALL action if required. Display time stamp.
102 &INPT(1) = 2
103 #O ANID OFF TIMEST OFF
104 &CALL MFTIME DMMY INPT PTR DIPAR
105 #*
106 #* Turn on analysis title and time stamp before exit.
107 #O ANID ON TIMEST ON
108 &IF &IER EM &EXIT
109 #*
110 #* Job done without error -- Enable MCDI ZO.
111 &CKZOFL = ENAZO
112 &GLOBAL CKZOFL ACTION MODE
113 &IF &IER NE 0 &WARNING
114 * ONLY &IER OUTPUT SAMPLES USED - INITIAL VALUES LOST
115 &EXIT
116 -ERR1 &ERROR PARAMETER &1 NOT RECOGNIZED
117 &EXIT
118 -ERR2 &ERROR INSUFFICIENT STORAGE
119 &EXIT
120 -ERR3 &ERROR MCDI ANALYSIS VALID IN WEXEC ENVIRONMENT ONLY
121 &EXIT
122 -ERR4 &ERROR CO/CP OUTPUT TYPE NOT AVAILABLE FOR MCFR
123 &EXIT
124 -ERR5 &ERROR AS PARAMETER '&AS' INVALID
125 &EXIT
126 -ERR6 &ERROR PARAMETER '&NOIN' NOT RECOGNIZED
127 &EXIT
128 #* 02-FEB-1989  BY  D. YON    EE DEPT.  YSU.
```

---

execution is branched to line 118 (-ERR2) if the macro storage is not enough for the run of MDCHCK (&OUT is empty). On line 118, an error message is displayed. Then, the execution is stopped following that.

Lines 12 through 14 encode parameter NOIN into numerical data and store the numerical data into &UPCODE. The value of &UPCODE controls the display of the input elements' statistical results. If &UPCODE is not equal to 0, the input elements' statistical results will be displayed on the user's terminal. On the other hand, the input elements' statistical results will be suppressed if &UPCODE is equal to 0. The value of &UPCODE depends on the user's input entering. If the user does not enter the key word NOIN on the input line, &UPCODE is set to 1. &UPCODE is set to 0 if the user does enter the key word NOIN with nothing following it. However, if something unexpected is following the key word NOIN on the user's input line, the execution will be branched to line 126 (-ERR6) by the instruction on line 12. An error message is displayed and the execution is stopped.

Line 17 checks if there is more than one value following the parameter AS. Only one value is allowed for the parameter AS. If the value of AS is incorrectly entered, the execution branches to line 124 (-ERR5) and is stopped after the display of an error message.

Lines 18 through 25 encode the parameter AS into numerical data. The numerical data which is the MCFR analysis mode code is kept by variable &MODE. No error occurs if parameter AS equals one of these six key words -- INI, MIN, MAX, AVG, STD, or ALL. Otherwise, the execution branches to line 124 (-ERR5) and is stopped after the display of an error message.

Lines 28 and 29 disable the check flag of the Monte-Carlo zoom option and make the value global (i.e., this value is available for all of the wexec macros).

The DI analysis specifications are updated on line 32. New analysis specifications are transmitted in the variable &FR.

Subroutine MDCHCK is called for the second time on line 37 with variable &PCODE as its input. Variable &PCODE is set equal to 12 or -12 to tell subroutine MDCHCK to check the DI analysis's ready flag. If the DI analysis is ready to run, variable &OUT(1) is set equal to 0 by MDCHCK. In addition, a warning message, ## WARNING ## MORE THAN 1 OUTPUT; REST IGNORED, is displayed on the user's terminal if the number of outputs is greater than 1 and AS ALL is in effect. The output type code and the output action code are also returned by MDCHCK to the macro.

However, if the DI analysis is not ready to run, only one value, &OUT(1) = -1, is returned by subroutine MDCHCK. A negative value in &OUT(1) forces the execution to stop immediately on line 39. In this case, Watand will have already displayed any related error message due to macro line 32 to indicate why the DI analysis was not ready to run.

Lines 40 and 41 make sure that neither COMPLEX nor CPOLAR output type is specified by the user. The execution branches to line 122 (-ERR4) to display an error message and to stop the execution if either COMPLEX or CPOLAR is in effect. Otherwise, the execution continues on line 42. On line 42, the value of &OUT(4) is moved into the variable &ACTION.

Subroutine MDINPT is called on line 46 with three input values. MDINPT intends to display the input elements' statistical results. As mentioned before, the value of &UPCODE tells subroutine MDINPT how to display the input elements statistical results. If there is any error during the disk data reading, &INPT with the value -1 is returned by MDINPT to stop the execution immediately on line 47. The related error messages will be displayed by MDINPT.

Lines 49 and 50 initialize the variables &IER and &PTR to zero, so that they won't be empty variables. This is to prevent a severe error from occurring if the subroutine called by &CALL has any empty variable as its input. For example, supposing action NONE is in effect, line 51 branches the execution to line 97 (-NO). After a few instruction lines, line 104 which is a &CALL command will be executed. It calls subroutine MFTIME with &PTR as one of its input. If line 50 didn't initialize &PTR to a variable not empty, a severe error occurs in this situation. To prevent this &CALL error from occurring, the instruction of line 50 must be presented.

Line 51 branches the execution to line 97 (-NO) if action NONE is specified by the user (&ACTION = 1). Otherwise, the execution continues on line 54. On line 54, the analysis ID and the time stamp are both turned off.

Line 55 initializes the global parameters for the DI analysis. Some DI analysis parameters (e.g., the output names) that are necessary for the run of some &CALL Fortran subroutines in a wexec macro won't be available to these

Fortran subroutines until the DI analysis is run. Therefore, before a wexec macro calls these Fortran subroutines, the DI analysis must be initialized.

Line 56 turns ANID and TIMEST back on in case of an error exit. Line 57 resets the KEEP option and the KEEP file name for the DI analysis. Line 58 restores the output action code and defines the disk mode onto which the output data is kept for the DI analysis by calling the subroutine MDCHCK the third time with &PCODE = 12 and &ACTION as its input.

Lines 61 through 107 have the same program structure as lines 76 through 116 in the MCFR WEXEC macro. However, lines 94 and 95 in the MFRDI WEXEC macro are different from that of the MCFR WEXEC macro. Line 94 calls subroutine MDCHCK for the fourth time to store the current DI USE file name. It is stored into Watand table memory. The reason for doing this is to be able to reset after the post-processing. The memory location pointer is returned with the variable &PTR by subroutine MDCHCK if no error occurs. If the macro storage is insufficient to return the location pointer or there is no Watand table memory available to store the file name, variable &PTR is set empty by subroutine MDCHCK and the execution branches to line 118 (-ERR2) to display an error message and to terminate the execution.

Line 108 terminates the execution immediately if the variable &IER is empty, i.e., macro storage was not enough when Fortran subroutine MFSTAS was called.

On line 111, since the Monte-Carlo post-processing is successfully done without severe error, the check flag of the Monte-Carlo zoom option is enabled. Line 112 makes variables CKZOFL, ACTION, and MODE global so that these variables can be available to the other wexec macros.

The execution is terminated on line 115 when no errors occur during the Monte-Carlo post-processing. Otherwise, the execution is terminated after a warning message. This warning message indicates that the sample number obtained by reading and calculation of the data that was kept on disk is not equal to the user's sample number. If this is the case, the output statistical results displayed by the Monte-Carlo post-processor will be based on the data available on disk.

3.4 MFRZO WEXEC MACRO

Macro MFRZO WEXEC is used to mimic the Watand DI ZOOM option with the MCDI ZO command. This is possible only after the execution of MCDI FR with PLOTTEKT action without any errors. Like DI zoom, this option won't be disabled until the user runs the MCFR analysis or the MCDI FR post-processor without the PLOTTEKT action. The macro program is shown in Table 3-3.

Line 1 on the MFRZO WEXEC macro checks for the check-zoom-flag, &CKZOFL, of the MCDI ZO to be sure that this

Table 3-3

MFRZO Wexc Macro

Program

```
&IF &CKZOFL NE ENAZO &GOTO -ERR1
&IF &ACTION NE 6 &GOTO -ERR1
#*
#* Store parameters in DI OU.
&ZCODE = -1
&CALL MFSTAS DIPAR MODE MODE ZCODE
&IF &DIPAR EM &GOTO -ERR2
&IF &MODE EQ 6 #E DI OU G INITIAL G MINIMUM G MAXIMUM G MEAN G ST.DEV
#*
#* Store initial file name for DI US.
&DCODE = 12
&ZCODE = 0
&CALL MDCHCK PTR DCODE ZCODE
&IF &PTR NM &GOTO -GOON
&PTR = 0
&IER =
&GOTO -TIME
-GOON
#E DI US MFST
#*
#* Check output compatibility and file reading error.
&CALL MDCHCK IER DCODE
&IF &IER NE 0 &GOTO -TIME
#O TIMEST OFF
DI ZO
#O TIMEST ON
-TIME
#*
#* Restore parameters for DI OU and filename for DI US.
&DCODE = 3
&ZCODE = &DIPAR(25)
&CALL MFTIME DMMY DCODE DCODE DCODE ZCODE PTR DIPAR
&IF &IER EM &GOTO -ERR2
&IF &IER GE 0 &EXIT
&CKZOFL = DSAZO
&GLOBAL CKZOFL
&EXIT
-ERR1 &ERROR CANNOT ZOOM
&EXIT
-ERR2 &ERROR INSUFFICIENT STORAGE
&EXIT
#* 02-FEB-1989  BY  D. YON    EE DEPT.  YSU.
```

zoom option has been enabled. If it is not enabled, the execution branches to line 38 (-ERR1) to display an error message. Line 2 checks if the action code of the last MCDI FR execution equals 6 (i.e., PLOTTEKT). If not, the execution branches to line 38 (-ERR1) where the execution is stopped after the display of an error message.

If the check-zoom-flag is enabled and the last MCDI FR command action was PLOTTEKT, the zoom execution will continue on line 5. Line 5 initializes the variable &ZCODE.

Subroutine MFSTAS is called on line 6 to store the current output specification parameters of the DI analysis. These parameters are returned in the variable &DIPAR. The MCDI ZO analysis start time is also obtained by subroutine MFSTAS. However, if the macro storage is not adequate for the subroutine MFSTAS to run (i.e., number of free memory locations is less than 25), variable &DIPAR is set to empty by subroutine MFSTAS. If &DIPAR is empty, the execution branches to line 40 (-ERR2) to display an error message and then stop the execution.

Line 8 checks the Monte-Carlo analysis mode in the last execution of the MCDI FR command. The output specifications of the DI analysis are properly set to the statistical outputs if the mode was 6 (i.e., AS ALL was in effect).

Subroutine MDCHCK is called on line 13 with &DCODE= 12 and &ZCODE=0 as its input values. With this input combination, subroutine MDCHCK stores the current file name for the DI analysis USE parameter into a Watand table memory location and returns the memory location pointer. If the macro storage is not available to return the pointer, variable &PTR is set to empty by subroutine MDCHCK. The execution continues on line 15 if the variable &PTR is empty (error). If the variable &PTR is not empty (no error), the execution branches to line 18 (-GOON).

When the execution continues on line 15, the following steps must be done before exiting from the macro. First, variable &PTR is reset to 0 to prevent incorrect Watand table memory release during the execution of the subroutine MFTIME (cf. line 32). Second, variable &IER is reset to empty to produce an error message to be displayed after the execution of the subroutine MFTIME. Third, subroutine MFTIME is called to restore the original DI analysis output specifications and its USE file name. Finally, the execution branches from line 33 to line 40 (-ERR2) to display the error message and stop the execution.

The other case is when the execution continues on line 18 (-GOON) which is the non-error path for the MCDI ZO. On line 19, the USE parameter of the DI analysis is reset to MFST.

Subroutine MDCHCK is called for the second time on line 22 to check for the file reading error and the output compatibility between the output specification in the DI analysis and the available outputs in file on the disk. If either of these two errors (file reading and output compatibility) is detected by MDCHCK, &IER is returned with a value not equal to 0. The error message describing the error is displayed by Watand. If &IER is not equal to 0, the execution branches to line 27 (-TIME). When the execution continues on line 27, the original DI output specifications and USE file name are restored. The check-zoom-flag, &CKZOFL, is set to disable before exiting from the macro to prevent re-execution without the necessary modification.

If &IER returned by subroutine MDCHCK on line 22 is error free (equal to zero), the execution continues on line 24. The Watand analysis time stamp is turned off on line 24 before executing the Watand DI ZO command on line 25. Line 25 performs the zoom action. The Watand analysis time stamp is turned back on at line 26 after the completion of the zoom action. At this point, the MCDI ZO execution is successfully completed.

Before exiting from the macro, the original output specifications and the file name for parameter USE of the DI analysis is restored by MFTIME. In addition, MFTIME prints the time stamp for MCDI ZO at the end of the execution.

CHAPTER IV

FORTRAN SUBROUTINES

4.1 OVERVIEW

Seven Fortran subroutines are used to perform the MCFR analysis and the MCDI post-processing. They are MDCHCK, MDINTL, MDREAD, MFRUSR, MFSTAS, MFTIME, and MDINPT. Their functions are described in this chapter. Functions of subroutines MFRNEW and MFDADS are also discussed. They are the two main subroutines called by MFRUSR. These Fortran programs are listed on Appendix D.

4.2 MDCHCK FORTRAN SUBROUTINE

Subroutine MDCHCK was derived from J. Suen's subroutine of the same name in his Monte-Carlo DC analysis [6]. Many changes have been made in this subroutine. The purpose of these changes is to integrate some MCFR analysis's jobs into this subroutine so that subroutine MDCHCK can work for MCFR analysis as it does for MCDC analysis without creating another new subroutine. Subroutine MDCHCK is a multi-purpose routine. Five functions are done in this subroutine.

The first function is to check the environment in which the wexec macro is executing. Subroutine MDCHCK will make the environment check if and only if 0 is the only input transmitted into the subroutine MDCHCK. If the variable WIWEFL in a Watand common is equal to 2, the current environment in which the wexec macro is running is the WE environment. The output with the name OUT is returned to the wexec macro that calls subroutine MDCHCK. OUT is equal to 0 if it is executing in the WE environment. Otherwise, OUT is equal to -1. However, if the current available macro storage for MDCHCK is less than one, no value is returned to the wexec macro by the subroutine MDCHCK and OUT is set to empty.

If a non-zero number is the input for subroutine MDCHCK, MDCHCK will perform the second function. It takes the input number as the Watand analysis type code and checks for the ready flag of that particular Watand analysis. The ready flag is not equal to 0 if and only if the Watand analysis is ready to run. MDCHCK will set OUT(1) equal to 0 if the analysis is ready. Otherwise, OUT(1) is set to -1. Furthermore, the output type code and the action code of that particular Watand analysis are read and stored into variables OUT(2) and OUT(4), respectively. There are seven output types for Watand analysis FR. They are MAGNITUD, PHASE, REAL, IMAGINAR, COMPLEX, DB, and CPOLAR. The output type code will indicate which output type is currently preferred by the user. Eight output actions are available for Watand FR analysis. They are NONE, PRINT, PPRINTER, PTERMINA, PLOTTEKT, POVERLAY, PVOLKER and CALL. The action code tells which one of these eight output actions is in effect. Subroutine MDCHCK also puts the value of NPWL into output variable OUT(3). NPWL is a Watand common variable which indicates the number of nonlinear elements in the circuit under analysis. If the circuit contains linear elements only, NPWL will equal 0. If NPWL is greater than or equal to 1, the circuit must be a nonlinear one. This means that the FR analysis's operating point will be important.

If the input value for subroutine MDCHCK is less than zero, MDCHCK takes its absolute value and then gets the four values for OUT. A negative input value indicates to subroutine MDCHCK to check the output number and the output action to determine whether to print the warning message, ## WARNING ## MORE THAN 1 OUTPUT; REST IGNORED., or not. If MDCHCK is not called by MCDC or by MCDI, the availability of the IPOINT (initial point) is checked for that analysis. Furthermore, if 12 is the input number for subroutine MDCHCK (meaning that DI analysis is currently under investigation), the file reading error and output compatibility are checked. OUT(1) is set to -2 if a file reading error or output incompatible error is detected. However, if the macro storage is less than four, no values are returned to the wexec macro by subroutine MDCHCK. MOUT is set to 0 in this case.

The third function of subroutine MDCHCK happens when the input of MDCHCK has two values with the first value not equal to 12. In this case, the two input values represent two Watand analysis type codes. The first input value indicates the particular Watand analysis whose global parameters will be copied into the other Watand analysis which is determined by the second input value. For MCFR analysis, the second input value is equal to 12 which means the Watand DI analysis will be used. The file name and file mode in which the first Watand analysis keeps its output values are copied into the DI analysis parameter and become the file name and the file mode which the DI analysis is going to use or process. The file mode on which the DI analysis keeps its output values is set the same as the file mode on which the first Watand analysis keeps its output results. In short, analysis DI is set to process the output results of the first Watand analysis and then keep its output results on the same disk mode. The DI analysis output specification and output type are also defined the same as that of the first Watand analysis. No OUT value is returned by subroutine MDCHCK in this function.

The fourth function of subroutine MDCHCK is to store the current DI analysis USE parameter, which is a file name, into a free Watand table memory location. This function is activated whenever subroutine MDCHCK has two values, with the first value equal to 12 and the second value equal to 0, as its input. One value, the pointer of the memory location where the file name is stored, is returned at the end of this routine. However, if the macro storage has storage capacity less than one, there is no value returned by subroutine MDCHCK. OUT is set to empty.

If subroutine MDCHCK has two values as its input, the first one equal to 12 and the second one not equal to 0, the fifth function of the subroutine MDCHCK starts. In this case, the second input value is the Watand analysis action code and becomes the new action code for the DI analysis. Furthermore, the disk mode that the DI analysis keeps its output results in is defined the same as the disk mode where the DI analysis is going to read/process the file. No value is returned by subroutine MDCHCK at the end of the execution to the wexec macro in this function. OUT is set to empty.

4.3 MDINTL FORTRAN SUBROUTINE

This subroutine was originally created for the MCDC analysis [6]. A few changes have been made to accommodate the MCFR analysis. The MDINTL Fortran subroutine is called to check if a correct KEEP specification is present in the DC/FR analysis for the MCDC/MCFR analysis, respectively. The correct specification is either KE OU or KE ALL be specified in DC/FR analysis. In addition, subroutine MDINTL counts the number of elements and parameters in WRTD and WRTM and sends them to the MDREAD subroutine as input data.

If an improper specification of condition is involved, MDINTL assigns an output code, -1, to display an error message and to stop the MCDC or the MCFR from processing. If no error occurs in the analysis KEEP options, MDINTL delivers to the subroutine MDREAD the number of parameters following WRTD and WRTM.

4.4 MDREAD FORTRAN SUBROUTINE

This is another subroutine basically designed for the MCDC analysis [6]. It was originally programmed with Fortran IV language for the MCDC analysis. Now it is updated with the Fortran 77 language and can be used either for the MCDC analysis or for the MCFR analysis. There was a limitation of 20 on the maximum number of WRT elements/parameters that could be simulated simultaneously with the old MDREAD subroutine. This updated MDREAD subroutine allows the user to enter as many WRT elements/parameters at one time as desired as long as there is enough Watand table memory available to support this attempt.

One-third of the maximum free table memory block is reserved by MDREAD to store the WRT elements/parameters pointers. To satisfy the memory request, MDREAD looks for system table memory first, then for Watand table memory if system table memory is not available. The rest of this memory block is reserved for the statistical results of input WRT elements/parameters. These two memory blocks are released after the statistical results of input WRT elements/parameters are obtained so that this memory can be re-used for output statistical results calculation in subroutine MFSTAS.

What the other job subroutine MDREAD will do is verify whether or not the WRT element/parameter type, the WRT element/parameter name, the two values following the WRT element/parameter name, and the qualifier value of a linear or nonlinear WRT element/parameter are correctly entered. These WRT element/parameter pointers are stored into Watand table memory storage if all of the WRT elements/parameters are correctly entered. The storage address pointer and the storage size are then returned by subroutine MDREAD. However, if there is any error with any of the WRT element/parameter pointers, an error message is displayed and the output code is assigned a -1 to stop the MCDC/MCFR analysis.

An END indicates the termination of the current model and the beginning of the new model when the current model is not the last model during the verification of a modeled element. It is used to prevent ambiguity between a parameter name and an element type. The reason for this was described by Suen [6].

4.5 MFRUSR FORTRAN SUBROUTINE

Subroutine MFRUSR controls the performance of Watand control word #RUSER. The number of executions of MFRUSR is one more than the number of samples (NS) because all of the WRT elements/parameters whose values have been changed are reset to their original values during the extra execution (NS+1). The extra execution also provides the MCDC/MCFR analysis the 'initial' output. MFRUSR calls subroutine MFRNEW if the execution number (IREP) is less than or equal to NS. For IREP greater than NS, it calls subroutine MFDADS and then releases two Watand table storage blocks which were reserved by subroutines MDREAD and MFRNEW to store WRT elements/parameters pointers and WRT elements/parameters statistical results. MFRUSR also checks if the user has enabled the IPDC option. If the IPDC option is enabled by the user and the circuit under analysis is a nonlinear one, MFRUSR will calculate the new DC bias point by calling the Watand subroutine DCMAIN before returning. This allows the user to have the correct DC bias point after the WRT elements/parameters are changed. On the other hand, if the IPDC option is not enabled by the user or the circuit under analysis is a linear one, MFRUSR will return with an unchanged DC bias point. This feature allows the user to have the same DC bias point for every individual FR analysis sample; it also saves a little computer time.

If the MCFR analysis is interrupted during the execution of #RUSER for some reason by hitting the ENTER key, Watand sets IREP=-1 [7]. Subroutine MFRUSR will acknowledge this situation and call subroutine MFDADS immediately to reset all of the WRT elements/parameters to their original values and then release those Watand table memories as mentioned before. There may be some input statistical results displayed on the terminal. However, they may not have much meaning in this case since the sampling is not complete yet.

4.6 MFRNEW FORTRAN SUBROUTINE

Subroutine MFRNEW is derived from the subroutine MDRUSR created for the MCDC analysis [6]. To remove the limitation on the maximum WRT element/parameter number from 20, subroutine MFRNEW reserves the current maximum Watand table memory block. This memory block is used to store the WRT element/parameter statistical results. If the storage reserved by MFRNEW is not enough to store all of the WRT element/parameter statistical results, the MCFR/MCDC analysis will be stopped after an error message is displayed.

The basic job of MFRNEW is to reset Watand values of specified WRT elements/parameters according to the statistical distribution function. The normal statistical distribution function is the default, but the user may rewrite the subroutine MDRAND for any or even several statistical distribution functions [6].

An error message is displayed by MFRNEW if the random variable routine MDRAND returns an IER not equal to 0.

4.7 MFDADS FORTRAN SUBROUTINE

Subroutine MFDADS is also derived from the subroutine MDRUSR created for the MCDC analysis [6].

The WRT elements/parameters statistical results are calculated and displayed by subroutine MFDADS. The WRT elements/parameters original values are restored by this subroutine, too. Furthermore, MFDADS stores the MCDC/MCFR analysis output results onto the disk with user assigned file name and file mode or with the default file name MCDC/MCFR and the default file mode A. However, the file types are fixed to the special types, MDKPINFO and MDKPVALS. Two files are created at the end of the execution of MFDADS. The file with MDKPINFO as its file type contains the Watand file identification information and the other file with MDKPVALS as its file type contains the WRT elements/parameters statistical results.

If there is any error during the creation of these two files, the analysis will be stopped with an error message displayed on the user's terminal.

4.8 MFSTAS FORTRAN SUBROUTINE

Subroutine MFSTAS calculates output statistical results and performs the PRINT output action if such is specified. This is a multipurpose subroutine like MDCHCK.

The sign of the third input value of MFSTAS tells whether it is called by the MCDI ZO (zoom) command or not. A positive sign for this input value indicates to MFSTAS that the current command is MCFR or MCDI without ZO option. A negative sign indicates that the current command is MCDI ZO.

The MCFR command or the MCDI command without zoom option examines the MCFR analysis mode code and the Watand output action code. If the mode code is not equal to 6 (i.e., AS ALL is not true) or the action code is less than or equal to 2 (action NONE or PRINT is specified), subroutine MFSTAS is executed only once. Otherwise, MFSTAS is executed twice.

In case that MFSTAS is executed once, the Watand action code directs MFSTAS where to store the output statistical results. The output statistical results are stored into Watand table storage if the action code is less than or equal to 2. Otherwise, disk memory will be used to store the output statistical results. In fact, two files (MFST WKEEPID and MFST WKEEP) are created by MFSTAS if disk memory is used. They are created with the same format as the general Watand KEEP files so that they can be processed by the DI analysis later.

In the other case that subroutine MFSTAS is executed twice, the DI analysis output specification is stored into macro memory by MFSTAS at the first execution. The mode code is then set to -6 by the wexec macro to indicate to MFSTAS the second execution. At the second execution, the original DI output specification will not be restored until the MFST WKEEPID file has been created on the disk. Consequently, MFSTAS won't begin to read the files created by Watand control command #RUSER and calculate the output statistical results until the DI analysis original output specification has been restored. MFSTAS stores the output statistical results onto the disk under file name MFST WKEEP.

Subroutine MFSTAS converts the statistical results into complex format before storing them onto the disk. The purpose of this data conversion is to satisfy the reading format of the DI analysis. This data conversion makes sure that the value read from the disk by the DI analysis is the same as the value calculated by subroutine MFSTAS no matter what kind of output type is in effect.

In other words, two files (MFST WKEEPID and MFST WKEEP) will be created on the disk if and only if the Watand output action code is greater than 2. Otherwise, no file with file name MFST will be created on the disk. In the case that the output action code is less or equal to 2, the current maximum block of Watand table storage will be reserved to store the output statistical results. The storage block will be released after the output action has been completed.

For the MCDI ZO command, subroutine MFSTAS will get the start time and store the DI analysis output specification into macro memory.

When the amount of the Watand table storage reserved by subroutine MFSTAS is not enough to store all of the output statistical results, an error message is displayed on the user's terminal and the output variable IIER with a value -1 will be returned to the wexec macro to stop the MCFR analysis. The same situation occurs with a proper error message displayed when the current Watand table storage amount cannot meet the storage requirement for reading data from files on disk or the user's output specification does not agree with the contents in the files on disk.

Likewise, this might happen when the disk memory is not enough to store all of the output statistical results or part of the output data was lost for some reason. To check this situation, subroutine MFSTAS will count the number of samples actually read out from files on disk and then compare the number of counting with user's sample number. Nothing unusual will happen if these two numbers agree with each other. However, if these two numbers are not equal, the MCFR analysis will calculate the output statistical results based on the data obtained from the disk and display a warning message on the user's terminal to report how many samples are actually used and processed.

In case the amount of the macro storage cannot support the execution of subroutine MFSTAS, no value is returned from MFSTAS and the MCFR analysis is stopped immediately.

4.9 MFTIME FORTRAN SUBROUTINE

Subroutine MFTIME is used to perform various tasks for MCFR, MCDI FR or MCDI ZO. The first value of the input for subroutine MFTIME indicates which command is in effect. It is equal to 1, 2, and 3 for MCFR, MCDI FR, and MCDI ZO commands, respectively.

For the MCFR command, subroutine MFTIME performs one of the PPRINTER, PTERMINA, PLOTTEKT, POVERLAY, PVOLKER or CALL output actions at the user's option by calling the Watand subroutine DISPLA. If output action PLOTTEKT is in effect, MFTIME will turn on the analysis ID code, ANID, and pause in the execution until the user hits the ENTER key. The time stamp with proper description is displayed at the end of the output by MFTIME. Before returning to the MCFR wexec macro, MFTIME resets some parameters for the FR and the DI analyses. The action code of the FR analysis is reset to the user's current selection and the VB/IB parameters of the FR analysis are reset to undefined. The ready flag of the DI analysis is reset to not ready and the file name for DI USE is reset to undefined. The output specification of DI analysis is also cleared (i.e., no output specified). These settings described above are similar to the Watand default settings. The user is therefore required to make some modifications to the analysis parameters in order to run FR or DI analysis whenever the Monte-Carlo analysis has been run.

For the MCDI FR command, the way MFTIME deals with output actions is the same as that for the MCFR command. The difference between them is the parameter settings for the FR and the DI analyses. In case of the MCDI FR command, MFTIME will restore the action code for the DI analysis. The file name for DI USE and the output specification for the DI analysis are also reset to the user's specification by MFTIME. MFTIME also prints the time stamp at the end of the execution.

In case of the MCDI ZO command, subroutine MFTIME will not perform any output action. However, it restores the file name for DI USE and the output specifications for DI OUT only. Of course, time stamp for MCDI ZO is printed by MFTIME, too.

No values are returned to the macro by MFTIME in all of these three cases.

4.10 MDINPT FORTRAN SUBROUTINE

Subroutine MDINPT is the same as when it was created for the MCDC analysis [6] except some minor changes have been made so that this subroutine can also be used to run the MCFR analysis.

The first input value for this subroutine, IDCODE, determines how to display the WRT elements/parameters statistical results. The WRT elements/parameters statistical results will be displayed if IDCODE=1. Otherwise, the WRT elements/parameters statistical results won't be displayed. The data to be displayed is obtained by reading two files on the disk with file types MDKPINFO and MDKPVALS. They were created by subroutine MFDADS.

CHAPTER V

EXAMPLES

5.1 ASSUMPTIONS

The simulation examples in this chapter are based on the assumption that every element in the example circuits is randomly picked and its statistical distribution function is Gaussian (normal distribution). Another assumption is that there are no correlations existing between any circuit elements.

According to Becker and Jensen [8], the component statistical distribution functions for discrete components are generally Gaussian with few exceptions. This is especially true when the component-manufacturing process has exited a long time and the process has become established. Thus, the statistical distribution function for each element in these examples is taken to be the normal distribution function. To perform simulations with elements having different kinds of statistical distribution functions, the Monte-Carlo subroutine MDRAND can be easily changed. See Ref. [6].

It is also possible to perform the Monte-Carlo simulation of circuits with correlations between element values by remodeling the element models [9]. This can be done in Watand by using common models and with user-defined elements.

5.2 THE FIRST EXAMPLE - A SIMPLE RC-FILTER

A simple RC-filter shown in Fig. 5.1 is used as an example to describe how the MCFR analysis calculates statistical results. The Watand file for this simple RC-filter is listed at Table 5-1.

5.2.1 DESIGN PURPOSE

This circuit is designed to provide 45°, or 0.7854 rad of phase lead at a frequency of 1000 rad/s. The transfer function of this circuit can be expressed as

$$T(s) = \frac{V_2}{V_1} = \frac{s+z_1}{s+p_1}$$

where

$$z_1 = \frac{1}{R_1C_1} \quad \text{and} \quad p_1 = z_1 + \frac{1}{R_2C_1}$$

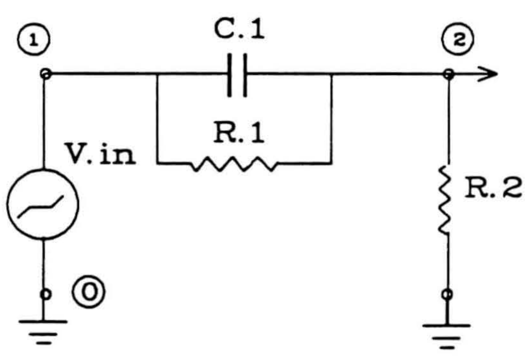

Fig. 5.1 A simple RC–filter

TABLE 5-1

Simple RC-Filter Watand File

```
#T SIMPLE RC-FILTER
#D
R.1  1  2  1.5K
R.2  2  0  230
C.1  1  2  1U
V.IN 1  0  SIN 1 1K 0
#E
FR BE 800 END 1200 LIN 100 VSO OUT V 2 PHASE KEEP OUT ON D PR
#S
```

The frequency response of this filter is obtained by doing the Watand FR analysis and is listed below. The response agrees with the design. Output is in radian.

```
fr

FR   V1.11-3g  SR137020 31-JAN-89 15:21:07  YSUCMS    FILE: RC

    OMEGA            V 2-0

    8.00000E+02     -8.52943E-01
    9.00000E+02     -8.15138E-01
    1.00000E+03     -7.84842E-01
    1.10000E+03     -7.60808E-01
    1.20000E+03     -7.41988E-01

FR   EXECUTION TIME=   0.007 SEC.
```

5.2.2 A SIMPLE MCFR ANALYSIS

The following are the responses to a simple MCFR analysis of this RC-filter. For simplicity, the number of samples, NS, is set equal to 4, and the number of sampled frequencies is the same as the previous FR analysis since no new FR input is given on the MCFR line.

```
mcfr wrtd r.1 5 r.2 5 c.1 v.in 5 ns 4 dseed 1963924

MCFR V1.11-3g  SR137020 31-JAN-89 15:24:29  YSUCMS    FILE: RC

   NUMBER OF SAMPLES =     4              GAUSSIAN  DISTRIBUTION

INPUT                  INITIAL    MINIMUM    MAXIMUM     MEAN      ST.DEV

R.1                   1.500E+03  1.491E+03  1.520E+03  1.501E+03  1.157E+01
R.2                   2.300E+02  2.268E+02  2.408E+02  2.320E+02  5.731E+00
C.1                   1.000E-06  9.454E-07  1.050E-06  9.992E-07  3.738E-08
V.IN                  1.000E+00  9.511E-01  1.024E+00  9.936E-01  2.722E-02

PHASE: V 2-0

    OMEGA        INITIAL       MINIMUM       MAXIMUM        MEAN         ST.DEV

  8.00000E+02 -8.52943E-01 -8.73144E-01 -8.38553E-01 -8.54560E-01  1.24368E-02
  9.00000E+02 -8.15138E-01 -8.33374E-01 -8.02638E-01 -8.16915E-01  1.09941E-02
  1.00000E+03 -7.84842E-01 -8.00913E-01 -7.74351E-01 -7.86763E-01  9.45252E-03
  1.10000E+03 -7.60808E-01 -7.74624E-01 -7.52361E-01 -7.62857E-01  4.89671E-03
  1.20000E+03 -7.41988E-01 -7.53534E-01 -7.35566E-01 -7.44153E-01  6.39464E-03

MCFR EXECUTION TIME=   0.217 SEC.
```

The statistical results shown above are calculated by the MCFR analysis based on the results of the four individual FR analyses. The entire numerical data of the four individual runs are listed as follows. The data is obtained using the DI analysis. See section 5.3.5. Note that the initial response is automatically run by MCFR and always appearing as the last data set.

```
di us mcfr on d ou v 2 ph pr

FR  V1.11-3g   SR137020 31-JAN-89 15:25:52  YSUCMS    FILE: RC

   OMEGA          V 2-0           OMEGA          V 2-0

   8.00000E+02   -8.55897E-01     8.00000E+02   -8.73144E-01
   9.00000E+02   -8.17608E-01     9.00000E+02   -8.33374E-01
   1.00000E+03   -7.86795E-01     1.00000E+03   -8.00913E-01
   1.10000E+03   -7.62232E-01     1.10000E+03   -7.74624E-01
   1.20000E+03   -7.42886E-01     1.20000E+03   -7.53534E-01

   OMEGA          V 2-0           OMEGA          V 2-0

   8.00000E+02   -8.38553E-01     8.00000E+02   -8.50645E-01
   9.00000E+02   -8.02638E-01     9.00000E+02   -8.14042E-01
   1.00000E+03   -7.74351E-01     1.00000E+03   -7.84991E-01
   1.10000E+03   -7.52361E-01     1.10000E+03   -7.62209E-01
   1.20000E+03   -7.35566E-01     1.20000E+03   -7.44626E-01

   OMEGA          V 2-0

   8.00000E+02   -8.52943E-01
   9.00000E+02   -8.15138E-01
   1.00000E+03   -7.84842E-01
   1.10000E+03   -7.60808E-01
   1.20000E+03   -7.41988E-01

DISPLAY TIME=   0.017 SEC.
```

The statistical mean value in the MCFR analysis is defined as

$$\overline{X} = \frac{1}{n} * \sum_{i=1}^{n} X_i \qquad (5.1)$$

where the X_i's represent the individual observed samples, and n is the total number of samples.

The variance, σ^2, square of the standard deviation, is defined as

$$\sigma^2 = \frac{1}{n} * \sum_{i=1}^{n} [X_i - \overline{X}]^2 \qquad (5.2)$$

or equivalently as

$$\sigma^2 = [\frac{1}{n} * \sum_{i=1}^{n} X_i^2] - (\overline{X})^2 \qquad (5.3)$$

MCFR analysis finds the mean and the standard deviation at every sampling frequencies one by one by using Eq.(5.1) and Eq.(5.3). For example, at the sampling frequency 1000 rad/s, the 4 sampled phase outputs of V 2-0 are -7.86795E-01, -8.00913E-01, -7.74351E-01, and -7.84991E-01. These are the X_is for Eq.(5.1) and Eq.(5.2). Therefore, the mean at 1000 rad/s is

$$\overline{X} = -7.867625E-01$$

and the variance, σ^2, is equal to

$$\sigma^2 = 8.935531E-05$$

or the standard deviation is equal to

$$\sigma = 9.452794E-03$$

The results are slightly different from that of the MCFR analysis results. However, they are agree with each other since the digit precision is different between these two cases. The MCFR uses greater precision in its calculation than the example described above.

5.3 THE SECOND EXAMPLE - BOCTOR NOTCH CIRCUIT

A notch circuit design, which finds use in practical realizations, is due to Boctor [5], and is shown in Fig. 5.2. The circuit realizes the high-pass notch (HPN) and provides a high-frequency (above notch frequency) gain, G, greater than 1. The transfer function can be expressed as

$$T(s) = \frac{V_2}{V_1} = G\frac{s^2 + w_z^2}{s^2 + (w_o/Q)s + w_o^2}$$

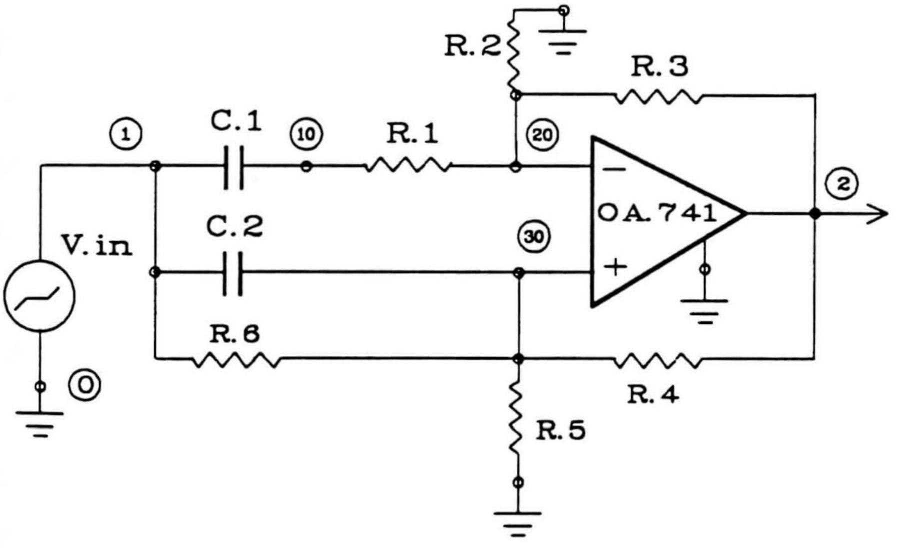

Fig. 5.2 Boctor high-pass notch filter

5.3.1 DESIGN PURPOSE

Suppose that a design requires a high-pass notch filter for which Q=1 and $w_o=2^{\frac{1}{2}}*w_z$. The filter is to provide 12 dB (G=4) of gain at high frequencies and 6 dB (G=2) of gain at low frequencies. The notch frequency, w_z, corresponds to 60 Hz or 377 rad/s.

According to the above specifications, the circuit element values are determined [10] and shown in Table 5-2. The Watand circuit file is listed in Table 5-3. It uses a large-signal op-amp model developed by B. Hsu [11].

To see how well the designed circuit agrees with the specifications, the Watand FR (frequency response) analysis is used. However, before the FR analysis is run, the DC analysis should be run first. Since the circuit is a non-linear one, the FR analysis will call for the DC solution as it's initial point. The frequency response of the completed design filter is shown in Fig. 5.3 which agrees with the specifications.

5.3.2 MCFR SIMULATION WITH 10 SAMPLES

If there is tolerance associated with component values, how will it affect the filter's frequency response? One of the ways to answer the question is to simulate the effects on the computer.

Suppose the HPN Boctor filter is going to be manufactured in large number with resistors and capacitors

Table 5-2

Nominal Element Values for the Boctor Filter

Element	Nominal value
R_1	12.50k
R_2	16.67k
R_3	50.00k
R_4	50.00k
R_5	10.00k
R_6	25.00k
C_1	0.15uF
C_2	0.15uF

Table 5-3

Boctor Filter Watand File

```
#T BOCTOR HIGH-PASS NOTCH CIRCUIT
#M
OA.741
#D
R.1 10 20  12.5K
R.2 20  0  16.67K
R.3 20  2  50K
R.4  2 30  50K
R.5 30  0  10K
R.6  1 30  25K
C.1  1 10  0.15U
C.2  1 30  0.15U
OA.741.G   30 20  2  0
V.IN  1 0  SIN 1 1K 0
#E
DC OUT ALL PR
FR BE 5 END 500 LOG 25 OUT V 2 MAG PL VB 0 4 KEEP ALL ON D
#S
```

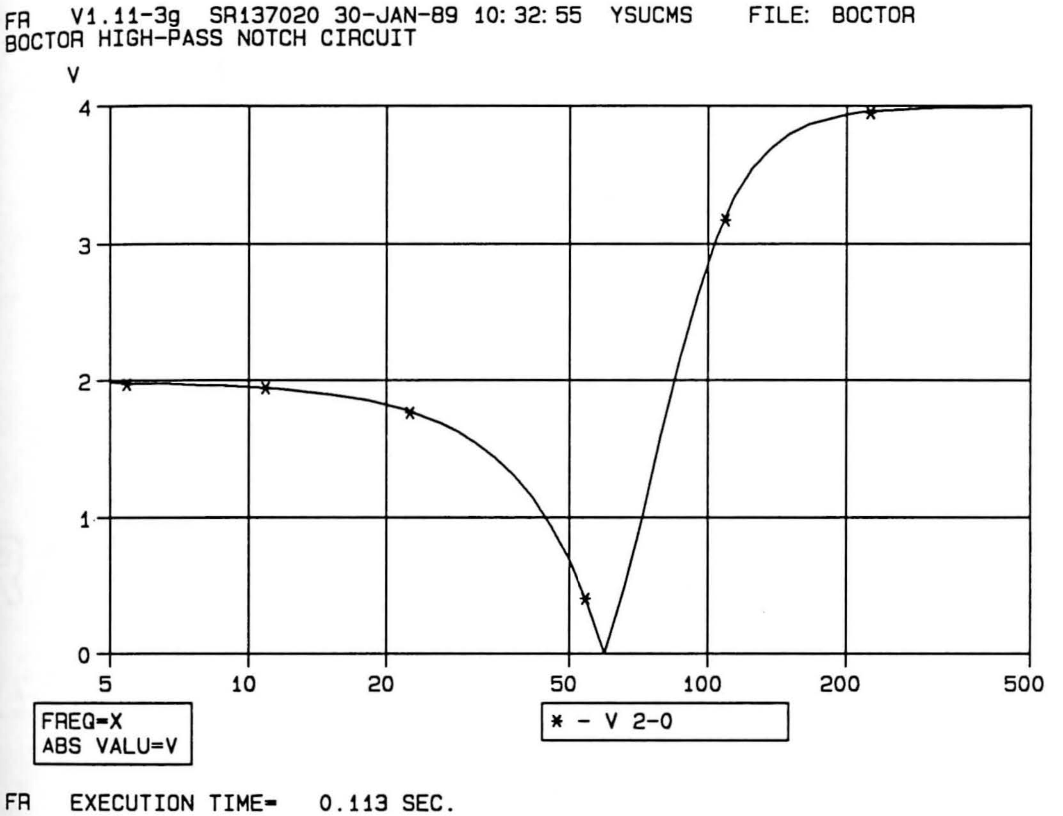

Fig. 5.3 Magnitude versus frequency for the Boctor HPN filter with exact design component values.

having ±10% tolerance in their values and ±50% tolerance in the OP amp's gain. Using the MCFR Monte-Carlo frequency analysis to produce such a simulation, the following interactive input should be entered.

```
MCFR WRTD R.1 R.2 R.3 R.4 R.5 R.6 C.1 C.2 *
WRTM OA.741 A0 50 FR KEEP ALL IN BOCTOR *
IPDC ON DSEED 1963924
```

MCFR is the command word that tells Watand to run the Monte-Carlo Frequency Response analysis. Changed #DATA level elements, R.1, R.2, R.3, R.4, R.5, R.6, C.1, and C.2, are entered following the key word WRTD. Since no numbers follow each linear element, the default tolerance, ±10%, is assigned to each linear elements. The '*' at the end of the first and the second lines indicate continuation on the next line.

Following the key word WRTM is the model OA.741 and model parameter A0. The number following A0 means there is ±50% tolerance for the parameter A0. 'FR KEEP ALL IN BOCTOR' gives Watand new FR analysis settings. Note that previously defined FR parameters will be used except those changed by the MCFR input line. The Watand analysis option KEEP ALL saves the complete circuit solution at all frequencies. Since a KEEP filename, BOCTOR, is specified, the Monte-Carlo simulation results will be kept in files named BOCTOR. Otherwise, the simulation results would be kept in files named MCFR. Key word IPDC with value ON turns on the option that causes the bias point (DC solution) to be regenerated before each FR analysis. Finally, the seed number for the random number generator, DSEED, is assigned as 1963924. A fixed seed number is used for every example in this chapter to allow some comparison of the pseudo-random numbers generated in all examples.

Key words NS and AS are not entered on the input command line. Therefore, their defaults will be in effect. The default for NS is 10 which means the sample population will be 10 samples. Likewise, ALL is the default for AS which means that all of the five statistical outputs, INITIAL, MINIMUM, MAXIMUM, MEAN, and ST.DEV, for each electrical output will be displayed. The input lines and output response is shown below and in Fig. 5.4.

```
mcfr wrtd r.1 r.2 r.3 r.4 r.5 r.6 c.1 c.2 *
wrtm oa.741 a0 50 fr ke all in boctor ipdc on dseed 1963924

MCFR V1.11-3g  SR137020 25-JAN-89 13:31:24  YSUCMS    FILE: BOCTOR

  NUMBER OF SAMPLES =   10              GAUSSIAN  DISTRIBUTION
```

INPUT		INITIAL	MINIMUM	MAXIMUM	MEAN	ST.DEV
R.1		1.250E+04	1.187E+04	1.287E+04	1.228E+04	3.200E+02
R.2		1.667E+04	1.527E+04	1.764E+04	1.650E+04	7.121E+02
R.3		5.000E+04	4.702E+04	5.203E+04	4.966E+04	1.472E+03
R.4		5.000E+04	4.511E+04	5.242E+04	4.986E+04	1.849E+03
R.5		1.000E+04	9.334E+03	1.094E+04	1.001E+04	4.045E+02
R.6		2.500E+04	2.387E+04	2.705E+04	2.521E+04	9.481E+02
C.1		1.500E-07	1.409E-07	1.571E-07	1.494E-07	5.987E-09
C.2		1.500E-07	1.427E-07	1.569E-07	1.505E-07	4.199E-09
OA.741	A0	2.000E+05	1.471E+05	2.714E+05	1.940E+05	4.085E+04

```
ABS VALU: V 2-0

Hit ENTER key when ready to display output...
```

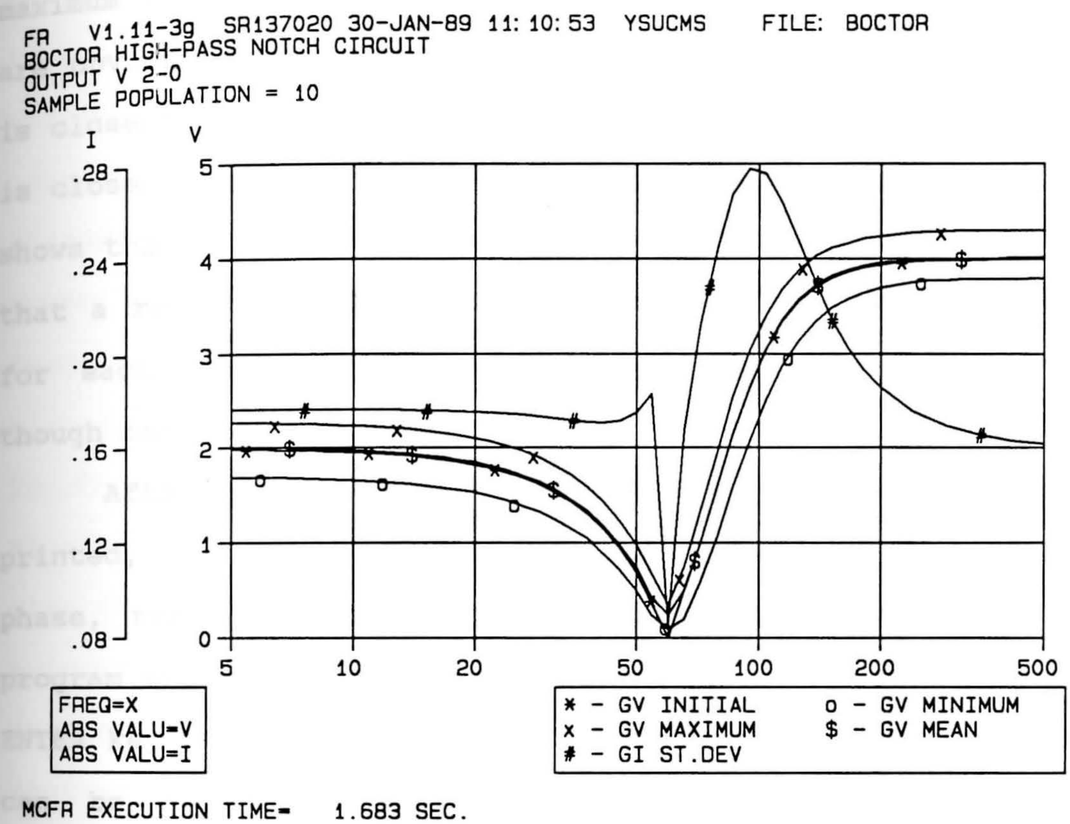

Fig. 5.4 MCFR statistical results of output V 2-0 in magnitude versus frequency for the Boctor HPN filter based on ten samples.

From the results, the changed elements and model parameter can be seen to vary within the tolerance error ranges, ±10% and ±50%, respectively. For example, the resistor, R.3 with the minimum value 47.02 kilohms and maximum value 52.03 kilohms, shows that the random values are obviously within the interval 50±5 kilohms. The mean is close to the initial value while the standard deviation is close to (INITIAL*10/100)/3 or 1.667E+03 for R.3, which shows that 3σ is about equal to the 10% range. This shows that a reasonable Gaussian distribution has been obtained for each component to perform a good simulation, even though only ten samples are used.

After all components' statistical results have been printed, the output's name and its type (absolute value, phase, real, imaginary, or dB value) is displayed. The program execution is then postponed until the user hits the ENTER key on the keyboard. With this feature, the screen can be held for the user to study the components' statistical results before going to the Tektronix (PL action) graphical output which clears the screen.

Fig. 5.4 shows the five statistical outputs for the output voltage V 2-0. In Fig. 5.4, the MEAN and the INITIAL output lines are very close except at the notch frequency 60Hz. The MEAN is interpreted as the average of the responses at each frequency for the produced filters. At 60 Hz, the MEAN is much higher than that of INITIAL which means the produced filters' attenuation of output V

2-0 at 60Hz will most likely be less (higher voltage) than that of the designed filter. The frequency response of the designed filter (Fig. 5.3) which is the same as the INITIAL in Fig. 5.4 can be viewed as 'perfect' response since the nominal component values are the best match for the circuit to satisfy the design purposes. Lines representing MINIMUM and MAXIMUM show how far up-and-down the output responses ranged at each frequency. Line ST.DEV represents the standard deviation at each frequency. It gives exact numerical information to tell how concentrated the sampling frequency responses will be around the line MEAN. With higher values, more uncertain responses should be expected. In Fig. 5.4, the lowest ST.DEV occurs at 60Hz and the highest at 95 Hz where the narrowest and the widest spreads also occur. It should be noted that these five statistical outputs except the INITIAL do not represent frequency response of a particular circuit. Rather, they give the statistical results at each frequency obtained from the entire sample population.

In most cases, the numerical values for ST.DEV are quite different from those of the other statistical results. Therefore, a different axis has been established for ST.DEV, the I axis.

It should be noted that the computer time for the simulation is 1.683 seconds which is reasonable. The simulation computer time depends primarily on the size of the circuit and the number of the sampling frequencies.

There are 52 sampling frequencies between 50 Hz and 500 Hz in total for this second example.

The entire ten samples' numerical data is shown on Appendix A in printed output (PR) format and the statistical results can be confirmed by traditional calculation. The way to get this complete numerical data for the sample population in a Monte-Carlo simulation is shown in section 5.3.5 where the use of DI analysis is explained.

5.3.3 DC, FR AND DI PARAMETER SETTINGS AFTER MCFR

Three Watand analyses (DC, FR and DI) global parameters are affected by doing the MCFR analysis. To illustrate this, the queries of these analyses are performed right after the MCFR command shown in section 5.3.2. The following is a list of the response to the #Q DC command which query the DC parameters setting:

```
#q dc

--> DC   ANALYSIS
ITERATIO   200
SMULTIPL   1.000E+01
NITERATI     0
NNOLU        1
NERROR     1.000E-12
IPOINT DC
NO OUTPT
OUTPUTS: ALL
KEEP NONE
    IN DC        ON A
```

since the IPDC option is set to ON in the MCFR analysis, the following resets of the DC analysis are performed after exiting the MCFR analysis. The bias point (IPOINT) is set to DC and the output action is set to NO for the DC analysis. Furthermore, DC analysis' KEEP option is re-defined as KEEP NONE IN DC. The destination disk mode for KEEP, however, won't be changed. On the other hand, if the IPDC option is OFF or the circuit under analysis contains no #MODEL-level elements, the MCFR analysis will not change any DC analysis parameters.

The response to the #Q FR command is listed below.

```
#q fr

--> FR   ANALYSIS
BEGIN      5.000E+00
END        5.000E+02
LOGARITH   2.500E+01
MAGNITUD
TRUE
VSFREQUE
IPOINT DC
PLOT-TEK  1760      5
OUTPUTS: V 2-0
X-AXIS:  DEFAULT
KEEP NONE
     IN FR       ON D
```

All parameters for FR analysis are the same before and after the MCFR analysis except the KEEP, VBounds (VB) and IBounds (IB). VB and IB settings for the FR analysis are removed whenever the MCFR analysis is executed. Furthermore, the KEEP option for the FR analysis is always set to KEEP NONE IN FR after the MCFR analysis to protect the Monte-Carlo simulation data file from accidental over-write. However, the destination disk mode for writing the data remains at the MCFR setting.

In the DI analysis case, the parameters are reset so that the DI analysis is not ready to run. A query of DI analysis at this stage will have the following response:

```
#q di

--> DI   ANALYSIS
MAGNITUD
PLOT-TEK  1760      5
X-AXIS:  DEFAULT
KEEP NONE
     IN DI        ON D
### WARNING ### SOME INFORMATION MAY BE MISSING OR INVALID
```

The DI USE parameter is always set to empty (un-defined) after an MCFR analysis. This is done since the previous setting for DI analysis is changed by MCFR. The KEEP action, similarly to FR, is set to KEEP NONE IN DI. The disk mode is also defined the same as that for MCFR's KEEP action. Furthermore, the output action is set the same as that for the MCFR analysis.

However, if a user tries to run DI analysis right after the MCFR analysis, the following messages will appear on the terminal.

```
di
    DI
### ERROR ### USE PARAMETER VALUE MISSING
    DI
### WARNING ### MISSING OUTPUTS
```

This forces the user to re-enter the USE and OUTPUTS parameters to run DI analysis right after the MCFR analysis since these parameters are used during the MCFR analysis.

5.3.4 USING THE MCDI FR POST-PROCESSOR

If the user would like to see these statistical results for example 2 in other units, say dB, rather than in voltage (absolute magnitude), or outputs other than V 2-0, the MCDI post-processor can do the job nicely. Five choices of other units are available. They are MAGNITUD, PHASE, REAL, IMAGINAR, and DB. Since all of the ten samples' data is kept on disk already, it is not necessary to run those ten FR analyses again to display any of these five statistical results in dB. With this post-processor, significant computer time can be saved. To see the statistical results in Fig. 5.4 in dB values, the following command line should be entered.

```
MCDI FR USE BOCTOR ON D OUT V 2 DB PL NOIN
```

The computer response to the above input is shown in Fig. 5.5. The command word MCDI tells Watand to run the Monte-Carlo post-processor. Key word FR tells the post-processor to process MCFR analysis results rather than MCDC analysis results. FR also acts as a parameter name to transfer the analysis parameters (i.e., USE BOCTOR ON D OUT V 2 DB PL in the above command) into DI analysis. See section 5.3.5.

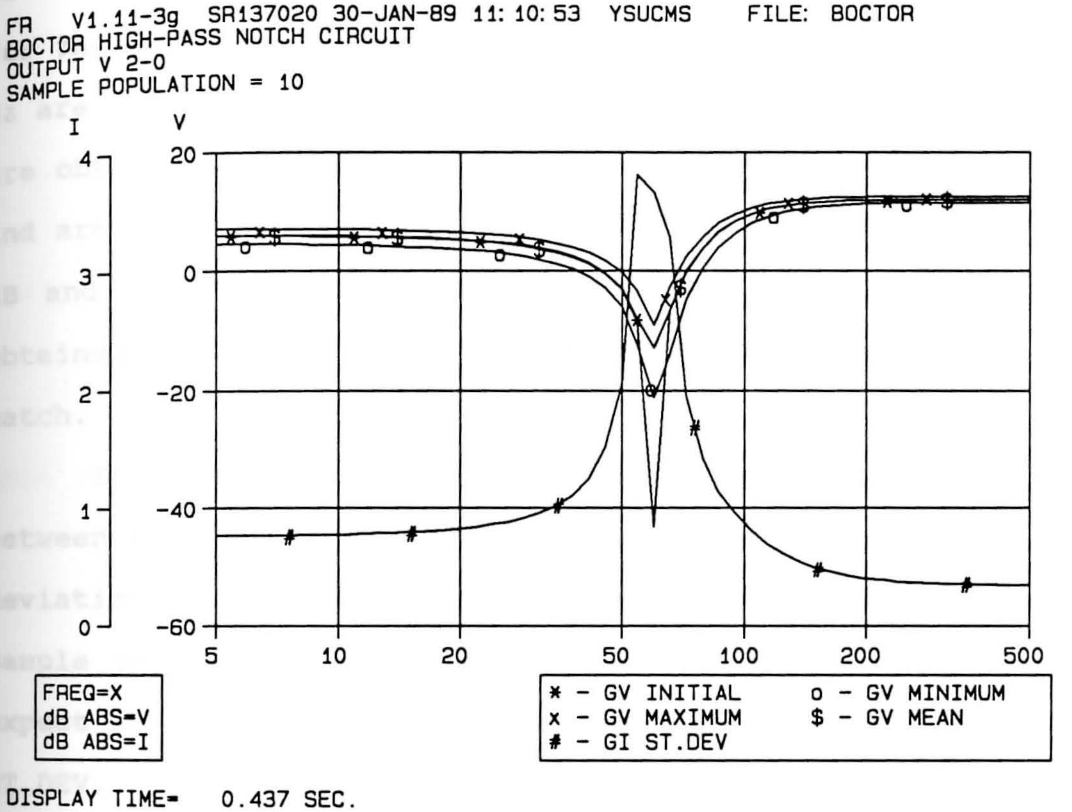

Fig. 5.5 MCFR statistical results of output V 2-0 in dB versus frequency for the Boctor HPN filter based on ten samples. Displayed by MCDI.

NOIN (NO INput statistical results), however, is the key word that tells the post-processor not to display the components' statistical results.

The information in Fig. 5.5 agrees exactly with that of Fig. 5.4 by properly transferring between units, except for the ST.DEV. For example, the MINIMUM and MAXIMUM at 60 Hz are 0.0875 V and 0.3526 V in Fig. 5.4. These two values are obtained by MCDI post-processor with print (PR) action and are shown in Appendix B. In dB values, they are -21.2 dB and -9.1 dB. Comparing these two values with those obtained by reading Fig. 5.5 at 60 Hz, they show an exact match.

It should be noted that there is no relationship between any two ST.DEV values in different units. Standard deviation is a number that indicates how wide spread the sample population is in that unit. The user should not expect to have the same number by doing a unit transfer on ST.DEV. That is,

$$ST.DEV_{dB} \neq 20 * Log\ ST.DEV_{ma}$$

The user is allowed to zoom into a smaller/larger portion of a PL-action plot in the MCDI FR post-processing environment. The input for this is similar to the DI zoom. After an MCDI with PL display, the zoom is activated by entering

MCDI ZO

Fig. 5.6 is the computer response when the zoom window corners are chosen as -30 dB to 10 dB on the V-axis and 20 Hz to 100 Hz on the X-axis. Fig. 5.6 can be zoomed as many times as desired. However, the MCDI zoom option is only allowed in the Monte-Carlo post processing environment with the PL action.

5.3.5 DI ANALYSIS PARAMETER SETTING AFTER MCDI FR

Whenever MCDI FR is used, some parameters of the DI analysis are reset. How the reset affects the DI analysis's parameters depends on what is entered following key word FR in the MCDI input command line. It has the same effect as the following command.

#E DI ... whatever entered following FR.

For example, the last MCDI command entered in section 5.3.4 contains 'FR USE BOCTOR ON D OUT V 2 DB PL' so the DI analysis parameters are reset as the following command:

#E DI USE BOCTOR ON D OUT V 2 DB PL

A query on DI analysis shows this:

```
#q di

--> DI   ANALYSIS
USE BOCTOR    ON D
DB
PLOT-TEK  1760       5
OUTPUTS: V 2-0
X-AXIS:  DEFAULT
KEEP NONE
     IN DI         ON D
```

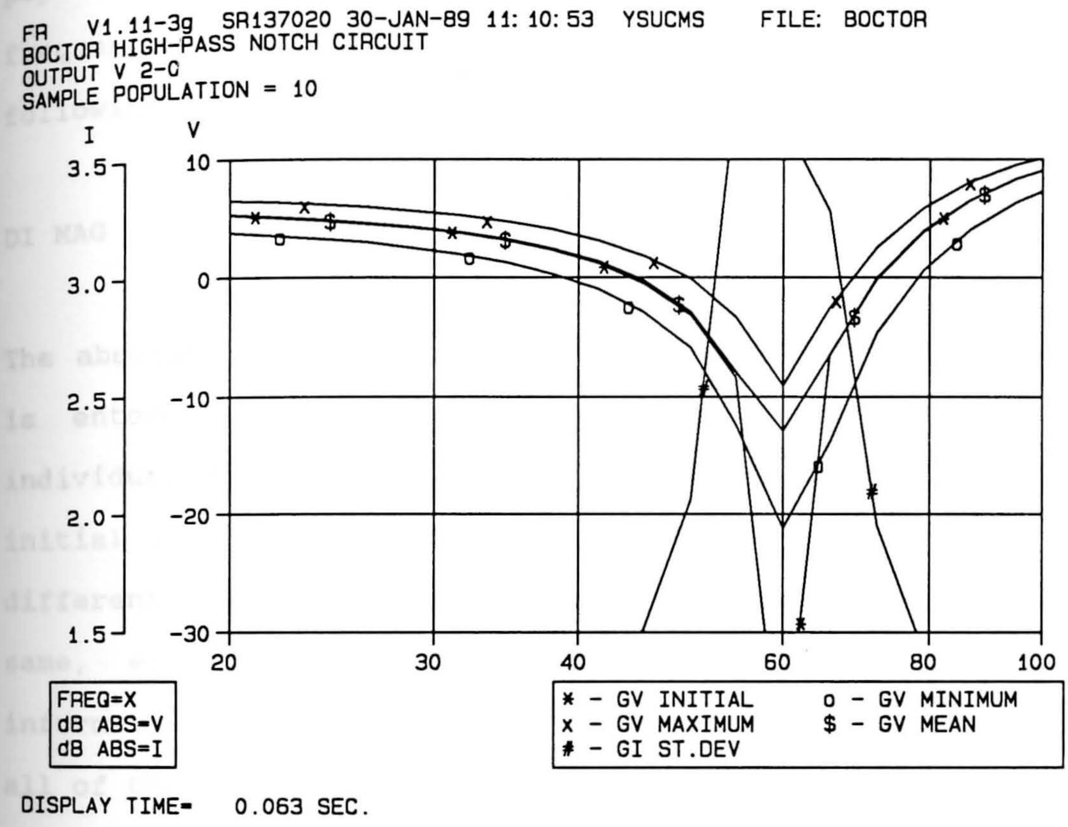

Fig. 5.6 A zoom display of Fig. 5.5.

On the other hand, the MCDI ZO command has no effect on the analysis parameters of the DI analysis.

With these DI settings, it is very convenient for the user to display the frequency responses of the sample population. For example, to display all of the ten frequency responses simulated in section 5.3.2, the following simple command line should be entered.

```
DI MAG
```

The absolute magnitude output is more interesting, so MAG is entered in addition to DI. Fig. 5.7 shows the individual responses of the ten simulation samples plus the initial one. Each notch frequency for each circuit is different from the nominal and their gains are not the same, either. Fig. 5.7 provides visual statistical information for the circuit designer who may want to see all of the actual responses on one graph.

5.3.6 MCFR SIMULATION WITH 100 SAMPLES

Since the input line for the Monte-Carlo simulation is inconvenient to type in for successive analyses, a Wexec macro file could be used. One named BOCTOR WEXEC is listed below:

```
MCFR WRTD R.1 R.2 R.3 R.4 R.5 R.6 C.1 C.2 WRTM OA.741 A0 50
* DSEED 1963924 &POSITION
```

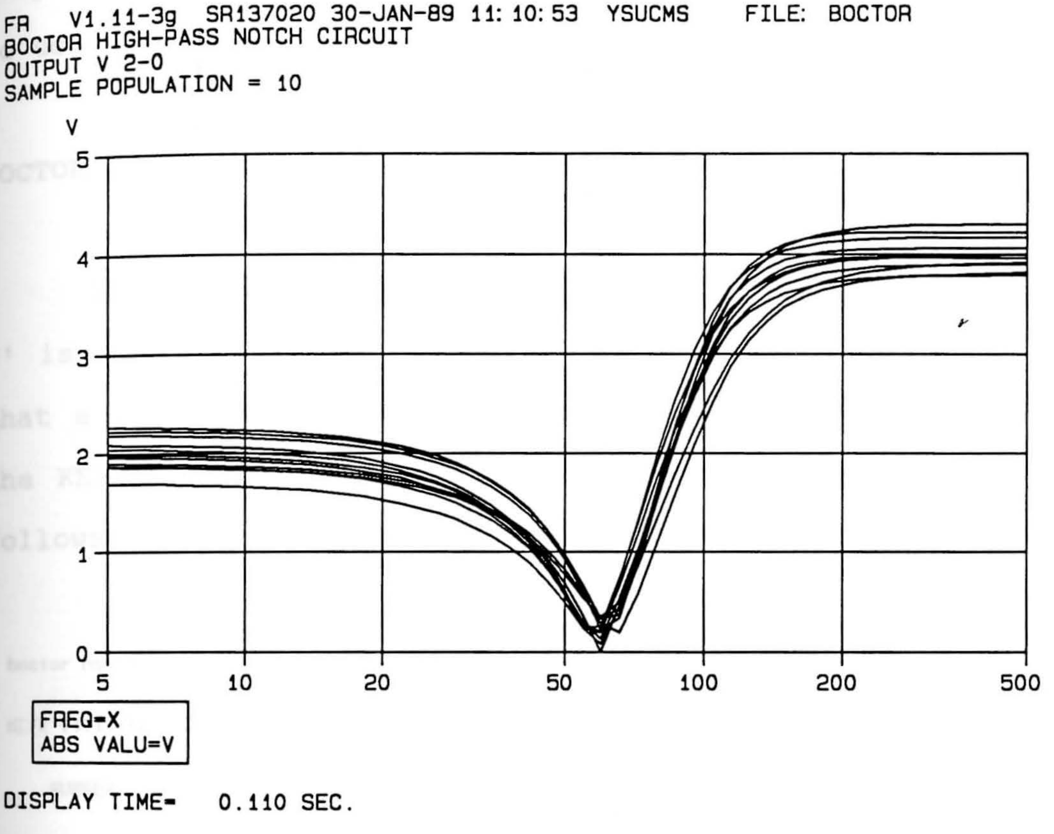

Fig. 5.7 Magnitude versus frequency results for the initial and ten randomly obtained Boctor HPN filter circuits. Displayed by Watand DI analysis.

In this section, the number of the samples is set to 100 to compare with the previous NS=10 results in section 5.3.2. All other inputs and the circuit remain the same. By going through this case the user can see the effects of the sample population on the simulation results. The command to use the BOCTOR macro is

BOCTOR NS 100 IPDC ON FR KEEP OUT ON T

In this input command, 'NS 100 IPDC ON FR KEEP OUT ON T' is substituted for &POSITION in the wexec file. Note that a relatively large disk, T, has been defined to hold the KEEP data. The input line and console output are as follows:

```
boctor ns 100 ipdc on fr keep out on t

MCFR V1.11-3g  SR137020 30-JAN-89 11:42:42  YSUCMS    FILE: BOCTOR

   NUMBER OF SAMPLES =  100              GAUSSIAN  DISTRIBUTION

INPUT                         INITIAL    MINIMUM    MAXIMUM     MEAN      ST.DEV

R.1                          1.250E+04  1.160E+04  1.353E+04  1.254E+04  4.313E+02
R.2                          1.667E+04  1.520E+04  1.798E+04  1.666E+04  5.717E+02
R.3                          5.000E+04  4.676E+04  5.284E+04  4.990E+04  1.469E+03
R.4                          5.000E+04  4.511E+04  5.561E+04  4.998E+04  1.736E+03
R.5                          1.000E+04  9.314E+03  1.094E+04  9.977E+03  3.098E+02
R.6                          2.500E+04  2.279E+04  2.734E+04  2.493E+04  9.796E+02
C.1                          1.500E-07  1.376E-07  1.606E-07  1.490E-07  5.239E-09
C.2                          1.500E-07  1.383E-07  1.622E-07  1.501E-07  4.747E-09
OA.741      A0               2.000E+05  1.217E+05  3.007E+05  1.999E+05  3.565E+04

ABS VALU: V 2-0

Hit ENTER key when ready to display output...
```

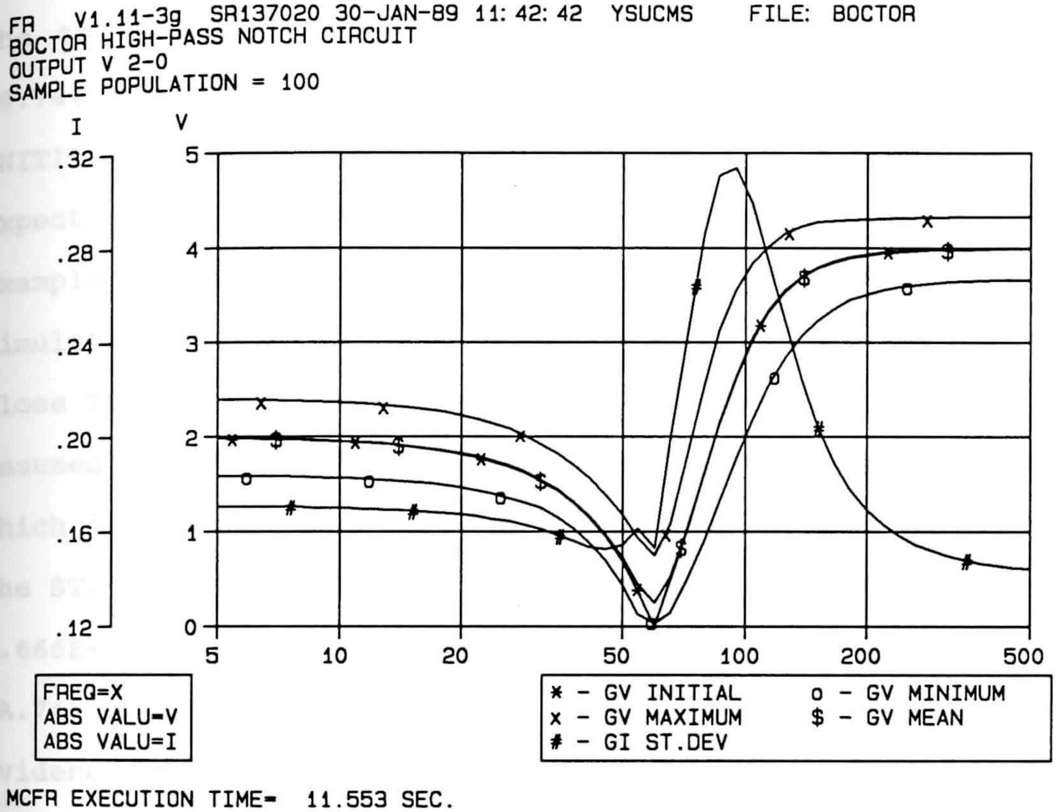

Fig. 5.8 MCFR statistical results of output V 2-0 in magnitude versus frequency for the Boctor HPN filter based on 100 samples.

From the display, the error tolerance interval for every tolerance elements/parameter follow quite well the Gaussian distribution. That is the MAXIMUM and MINIMUM are close to the INITIAL plus/minus tolerance range while the MEAN close to the INITIAL and the ST.DEV close to 1/3 of the tolerance. The Gaussian distribution also says that 99.74% of the sample population should be inside the INITIAL$\pm3\sigma$ range [12]. Therefore, it is appropriate to expect some samples outside the INITIAL$\pm3\sigma$ range. For example, resistor R.4 have MAXIMUM and MINIMUM in this simulation as 55.61 kilohms and 45.11 kilohms which are close to 50$\pm$5 kilohms. Note that the tolerance for R.4 is assumed as 10% or 5 kilohms. The MEAN is 49.98 kilohms which is almost equal to 50 kilohms, the INITIAL value, and the ST.DEV, 1.736E+03, is very close to 1/3 of 5 kilohms or 1.666E+03. Note that the MAXIMUM sampled for parameter OA.741 A0 is slightly beyond the INITIAL+3σ. All these evidences show a good simulation had been performed. The results match to the theory.

With the sample population equal to 100, the mean value is closer to the initial value and the standard deviation is closer to 1/3 of the tolerance than those in the sample population equal to 10 case, as would be expected. Note that the valley of ST.DEV at 60 Hz in Fig. 5.4 is almost smoothed out in Fig. 5.8.

Generally speaking, the greater the sample population the more accurate the statistical simulation results should be. However, a simulation involving a larger sample population will consume more CPU time. For instance, the execution time difference between a simulation of 10 samples at 1.683 seconds and a simulation of 100 samples at 11.553 seconds is noticeable. However, in this case the computer time doesn't increase in proportion to the number of samples. That is 10*1.683 $\neq$ 11.553. The Watand MCFR analysis provides the choice between statistical accuracy and CPU time.

One extra MCFR run is performed to show the frequency responses of the Boctor HPN filter from 50 Hz to 70 Hz. The sample population is set to 50 as a compromise between large NS and being able to see the individual responses. The command is:

```
BOCTOR NS 50 IPDC ON *
FR NO BE 50 END 70 LOG 400 KEEP OUT ON T
```

The outputs are dispalyed by Watand DI analysis and are shown in Fig. 5.9. The variations in notch frequencies of may be observed with this kind of output.

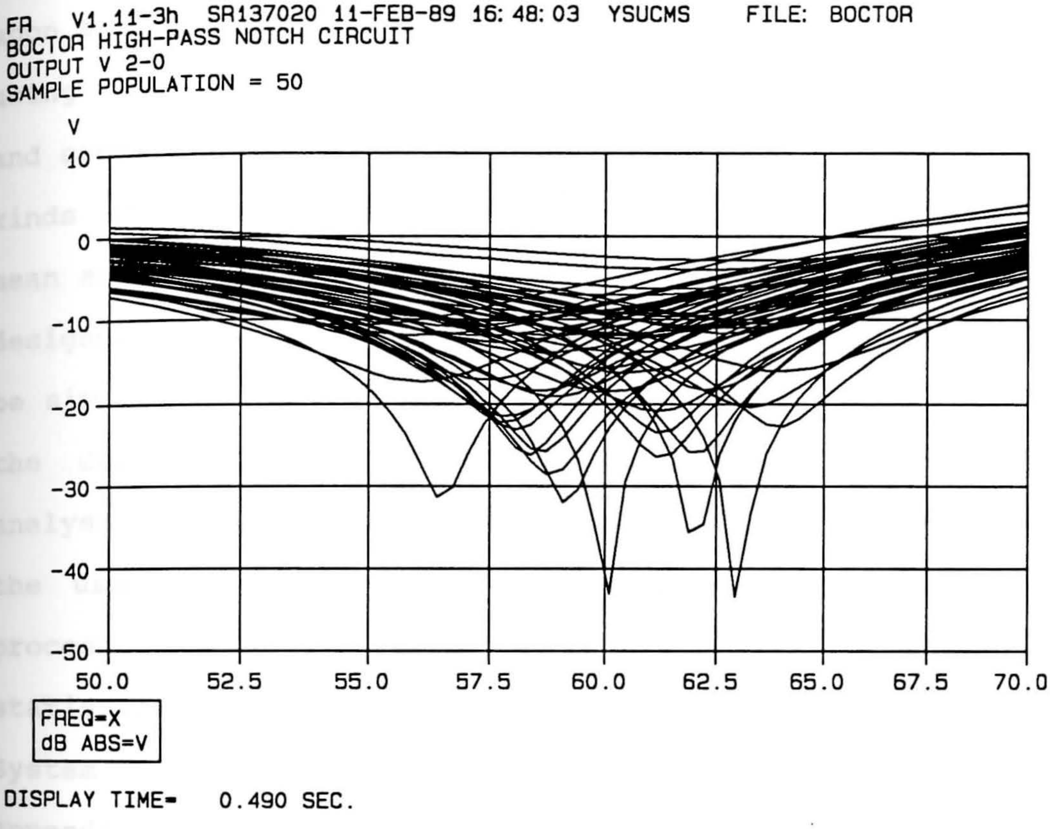

Fig. 5.9 Decibel versus frequency results for the initial and fifty randomly obtained Boctor HPN filter circuits. Displayed by Watand DI analysis.

CHAPTER VI

CONCLUSION

The Monte-Carlo FR analysis offers an effective method to simulate the tolerant circuit components's effect upon the circuit performance in the frequency domain. When using this analysis, almost all of the Watand's powerful and comprehensive options are applicable to the user. Five kinds of statistical results, initial, minimum, maximum, mean and standard deviation, are available for the circuit designer. The number of tolerant elements or parameters to be simulated simultaneously is only limited to the size of the computer memory, instead of 20 in previous MCDC analysis, while the computer memory size can be defined by the user if IBM CMS system is used. The MCDI post-processor can be used to selectively display the statistical results generated by the MCFR or MCDC analyses. System help files have been written and are given in Appendix C.

Three factors were always balanced during the development of the programs. They are the maximum execution speed, the minimum memory request, and the greatest flexibility. The execution speed was increased by avoiding unnecessary or duplicated programming and by using the Watand table storage, instead of disk storage, to store the data whenever possible. Avoiding the storage of error messages in the Fortran subroutines and instead putting them in the Wexec macros where possible, and using Watand's dynamic table memory access/release capability both save some computer memory. The feature allowing a user to redefine the MDRAND subroutine to simulate any kind of statistical distribution for the tolerance elements and parameters was previously created for the MCDC analysis. This gives great flexibility in using the Monte-Carlo analysis and was adopted by the MCFR analysis.

Future development of Monte-Carlo analysis for Watand should be the TC analysis which is operated in the time domain. One could then obtain complete statistical performance of a circuit by employing the Monte-Carlo DC, FR, and TC analyses. Work might also be done by the Watand development group at the University of Waterloo to embed the Monte-Carlo analysis into the Watand package as a built-in analysis.

APPENDIX A

NUMERICAL RESULTS FOR THE BOCTOR FILTER EXAMPLE

In this Appendix, all of the output from the ten FR analyses used in example 2, the Boctor HPN filter, is listed in numerical format. Absolute magnitude and dB values are provided. The data is listed in two sets of five pages each. The first set is labeled A1, A2, etc, and is the magnitude versus frequency sets. After that comes the second set of five pages of dB values versus frequency and is labeled B1, B2, etc.

This output was obtained by using the DI post-processor operating on the Watand KEEP files created by the MCFR analysis as described in section 5.3.5.

FREQ (A1)	V 2-0 mag	FREQ (A2)	V 2-0 mag
5.00000E+00	2.23300E+00	5.00000E+00	1.95998E+00
5.48239E+00	2.23104E+00	5.48239E+00	1.95775E+00
6.01132E+00	2.22869E+00	6.01132E+00	1.95507E+00
6.59128E+00	2.22586E+00	6.59128E+00	1.95183E+00
7.22720E+00	2.22245E+00	7.22720E+00	1.94794E+00
7.92447E+00	2.21835E+00	7.92447E+00	1.94325E+00
8.68900E+00	2.21339E+00	8.68900E+00	1.93759E+00
9.52730E+00	2.20742E+00	9.52730E+00	1.93076E+00
1.04465E+01	2.20020E+00	1.04465E+01	1.92253E+00
1.14543E+01	2.19149E+00	1.14543E+01	1.91258E+00
1.25594E+01	2.18095E+00	1.25594E+01	1.90055E+00
1.37711E+01	2.16820E+00	1.37711E+01	1.88600E+00
1.50998E+01	2.15273E+00	1.50998E+01	1.86836E+00
1.65566E+01	2.13395E+00	1.65566E+01	1.84695E+00
1.81539E+01	2.11110E+00	1.81539E+01	1.82092E+00
1.99054E+01	2.08324E+00	1.99054E+01	1.78920E+00
2.18258E+01	2.04916E+00	2.18258E+01	1.75046E+00
2.39315E+01	2.00735E+00	2.39315E+01	1.70300E+00
2.62404E+01	1.95587E+00	2.62404E+01	1.64466E+00
2.87720E+01	1.89219E+00	2.87720E+01	1.57271E+00
3.15479E+01	1.81303E+00	3.15479E+01	1.48361E+00
3.45915E+01	1.71411E+00	3.45915E+01	1.37280E+00
3.79289E+01	1.58978E+00	3.79289E+01	1.23449E+00
4.15882E+01	1.43268E+00	4.15882E+01	1.06134E+00
4.56005E+01	1.23348E+00	4.56005E+01	8.44475E-01
5.00000E+01	9.81453E-01	5.00000E+01	5.74137E-01
5.48239E+01	6.69357E-01	5.48239E+01	2.45283E-01
6.01132E+01	3.39814E-01	6.01132E+01	1.97240E-01
6.59128E+01	4.44636E-01	6.59128E+01	6.63358E-01
7.22720E+01	9.94756E-01	7.22720E+01	1.19101E+00
7.92447E+01	1.64215E+00	7.92447E+01	1.73215E+00
8.68900E+01	2.28091E+00	8.68900E+01	2.23698E+00
9.52730E+01	2.83230E+00	9.52730E+01	2.66512E+00
1.04465E+02	3.25513E+00	1.04465E+02	2.99922E+00
1.14543E+02	3.55110E+00	1.14543E+02	3.24410E+00
1.25594E+02	3.74604E+00	1.25594E+02	3.41635E+00
1.37711E+02	3.86990E+00	1.37711E+02	3.53478E+00
1.50998E+02	3.94702E+00	1.50998E+02	3.61539E+00
1.65566E+02	3.99444E+00	1.65566E+02	3.67018E+00
1.81539E+02	4.02329E+00	1.81539E+02	3.70756E+00
1.99054E+02	4.04062E+00	1.99054E+02	3.73324E+00
2.18258E+02	4.05082E+00	2.18258E+02	3.75102E+00
2.39315E+02	4.05661E+00	2.39315E+02	3.76346E+00
2.62404E+02	4.05969E+00	2.62404E+02	3.77225E+00
2.87720E+02	4.06114E+00	2.87720E+02	3.77852E+00
3.15479E+02	4.06161E+00	3.15479E+02	3.78304E+00
3.45915E+02	4.06152E+00	3.45915E+02	3.78634E+00
3.79289E+02	4.06112E+00	3.79289E+02	3.78877E+00
4.15882E+02	4.06058E+00	4.15882E+02	3.79058E+00
4.56005E+02	4.05998E+00	4.56005E+02	3.79194E+00
5.00000E+02	4.05938E+00	5.00000E+02	3.79297E+00
5.00000E+02	4.05938E+00	5.00000E+02	3.79297E+00

FREQ (A3)	V 2-0 mag	FREQ (A4)	V 2-0 mag
5.00000E+00	2.19024E+00	5.00000E+00	1.69674E+00
5.48239E+00	2.18826E+00	5.48239E+00	1.69461E+00
6.01132E+00	2.18586E+00	6.01132E+00	1.69204E+00
6.59128E+00	2.18298E+00	6.59128E+00	1.68895E+00
7.22720E+00	2.17951E+00	7.22720E+00	1.68523E+00
7.92447E+00	2.17532E+00	7.92447E+00	1.68075E+00
8.68900E+00	2.17028E+00	8.68900E+00	1.67536E+00
9.52730E+00	2.16420E+00	9.52730E+00	1.66886E+00
1.04465E+01	2.15686E+00	1.04465E+01	1.66103E+00
1.14543E+01	2.14799E+00	1.14543E+01	1.65159E+00
1.25594E+01	2.13728E+00	1.25594E+01	1.64021E+00
1.37711E+01	2.12432E+00	1.37711E+01	1.62647E+00
1.50998E+01	2.10861E+00	1.50998E+01	1.60987E+00
1.65566E+01	2.08954E+00	1.65566E+01	1.58980E+00
1.81539E+01	2.06637E+00	1.81539E+01	1.56551E+00
1.99054E+01	2.03814E+00	1.99054E+01	1.53608E+00
2.18258E+01	2.00365E+00	2.18258E+01	1.50037E+00
2.39315E+01	1.96139E+00	2.39315E+01	1.45696E+00
2.62404E+01	1.90944E+00	2.62404E+01	1.40412E+00
2.87720E+01	1.84529E+00	2.87720E+01	1.33966E+00
3.15479E+01	1.76571E+00	3.15479E+01	1.26088E+00
3.45915E+01	1.66646E+00	3.45915E+01	1.16441E+00
3.79289E+01	1.54197E+00	3.79289E+01	1.04609E+00
4.15882E+01	1.38490E+00	4.15882E+01	9.00857E-01
4.56005E+01	1.18568E+00	4.56005E+01	7.22713E-01
5.00000E+01	9.32290E-01	5.00000E+01	5.04946E-01
5.48239E+01	6.11309E-01	5.48239E+01	2.41349E-01
6.01132E+01	2.23639E-01	6.01132E+01	8.74903E-02
6.59128E+01	3.45183E-01	6.59128E+01	4.53844E-01
7.22720E+01	9.36923E-01	7.22720E+01	8.76285E-01
7.92447E+01	1.60419E+00	7.92447E+01	1.33394E+00
8.68900E+01	2.27334E+00	8.68900E+01	1.80017E+00
9.52730E+01	2.86665E+00	9.52730E+01	2.24295E+00
1.04465E+02	3.33285E+00	1.04465E+02	2.63417E+00
1.14543E+02	3.66424E+00	1.14543E+02	2.95777E+00
1.25594E+02	3.88354E+00	1.25594E+02	3.21147E+00
1.37711E+02	4.02210E+00	1.37711E+02	3.40278E+00
1.50998E+02	4.10709E+00	1.50998E+02	3.54351E+00
1.65566E+02	4.15810E+00	1.65566E+02	3.64564E+00
1.81539E+02	4.18803E+00	1.81539E+02	3.71940E+00
1.99054E+02	4.20507E+00	1.99054E+02	3.77272E+00
2.18258E+02	4.21428E+00	2.18258E+02	3.81146E+00
2.39315E+02	4.21880E+00	2.39315E+02	3.83981E+00
2.62404E+02	4.22054E+00	2.62404E+02	3.86075E+00
2.87720E+02	4.22069E+00	2.87720E+02	3.87636E+00
3.15479E+02	4.21997E+00	3.15479E+02	3.88812E+00
3.45915E+02	4.21881E+00	3.45915E+02	3.89707E+00
3.79289E+02	4.21747E+00	3.79289E+02	3.90394E+00
4.15882E+02	4.21612E+00	4.15882E+02	3.90926E+00
4.56005E+02	4.21482E+00	4.56005E+02	3.91342E+00
5.00000E+02	4.21363E+00	5.00000E+02	3.91670E+00
5.00000E+02	4.21363E+00	5.00000E+02	3.91670E+00

FREQ (A5)	V 2-0 mag	FREQ (A6)	V 2-0 mag
5.00000E+00	1.89644E+00	5.00000E+00	2.04457E+00
5.48239E+00	1.89461E+00	5.48239E+00	2.04222E+00
6.01132E+00	1.89241E+00	6.01132E+00	2.03940E+00
6.59128E+00	1.88976E+00	6.59128E+00	2.03600E+00
7.22720E+00	1.88657E+00	7.22720E+00	2.03191E+00
7.92447E+00	1.88273E+00	7.92447E+00	2.02698E+00
8.68900E+00	1.87810E+00	8.68900E+00	2.02103E+00
9.52730E+00	1.87252E+00	9.52730E+00	2.01386E+00
1.04465E+01	1.86580E+00	1.04465E+01	2.00521E+00
1.14543E+01	1.85768E+00	1.14543E+01	1.99475E+00
1.25594E+01	1.84789E+00	1.25594E+01	1.98211E+00
1.37711E+01	1.83606E+00	1.37711E+01	1.96682E+00
1.50998E+01	1.82176E+00	1.50998E+01	1.94827E+00
1.65566E+01	1.80444E+00	1.65566E+01	1.92577E+00
1.81539E+01	1.78346E+00	1.81539E+01	1.89840E+00
1.99054E+01	1.75798E+00	1.99054E+01	1.86504E+00
2.18258E+01	1.72700E+00	2.18258E+01	1.82428E+00
2.39315E+01	1.68924E+00	2.39315E+01	1.77434E+00
2.62404E+01	1.64311E+00	2.62404E+01	1.71294E+00
2.87720E+01	1.58662E+00	2.87720E+01	1.63718E+00
3.15479E+01	1.51721E+00	3.15479E+01	1.54331E+00
3.45915E+01	1.43168E+00	3.45915E+01	1.42652E+00
3.79289E+01	1.32595E+00	3.79289E+01	1.28068E+00
4.15882E+01	1.19490E+00	4.15882E+01	1.09806E+00
4.56005E+01	1.03227E+00	4.56005E+01	8.69494E-01
5.00000E+01	8.30804E-01	5.00000E+01	5.85810E-01
5.48239E+01	5.83707E-01	5.48239E+01	2.52241E-01
6.01132E+01	2.95490E-01	6.01132E+01	2.66317E-01
6.59128E+01	2.02571E-01	6.59128E+01	7.50655E-01
7.22720E+01	5.90490E-01	7.22720E+01	1.30917E+00
7.92447E+01	1.07393E+00	7.92447E+01	1.88166E+00
8.68900E+01	1.58522E+00	8.68900E+01	2.41261E+00
9.52730E+01	2.07909E+00	9.52730E+01	2.85883E+00
1.04465E+02	2.51549E+00	1.04465E+02	3.20323E+00
1.14543E+02	2.87105E+00	1.14543E+02	3.45265E+00
1.25594E+02	3.14252E+00	1.25594E+02	3.62594E+00
1.37711E+02	3.34054E+00	1.37711E+02	3.74357E+00
1.50998E+02	3.48103E+00	1.50998E+02	3.82259E+00
1.65566E+02	3.57932E+00	1.65566E+02	3.87555E+00
1.81539E+02	3.64780E+00	1.81539E+02	3.91114E+00
1.99054E+02	3.69562E+00	1.99054E+02	3.93517E+00
2.18258E+02	3.72920E+00	2.18258E+02	3.95152E+00
2.39315E+02	3.75300E+00	2.39315E+02	3.96271E+00
2.62404E+02	3.77004E+00	2.62404E+02	3.97044E+00
2.87720E+02	3.78236E+00	2.87720E+02	3.97582E+00
3.15479E+02	3.79137E+00	3.15479E+02	3.97959E+00
3.45915E+02	3.79803E+00	3.45915E+02	3.98226E+00
3.79289E+02	3.80302E+00	3.79289E+02	3.98416E+00
4.15882E+02	3.80678E+00	4.15882E+02	3.98553E+00
4.56005E+02	3.80966E+00	4.56005E+02	3.98652E+00
5.00000E+02	3.81188E+00	5.00000E+02	3.98724E+00
5.00000E+02	3.81188E+00	5.00000E+02	3.98724E+00

FREQ (A7)	V 2-0 mag	FREQ (A8)	V 2-0 mag
5.00000E+00	1.87190E+00	5.00000E+00	2.09217E+00
5.48239E+00	1.86978E+00	5.48239E+00	2.08984E+00
6.01132E+00	1.86724E+00	6.01132E+00	2.08704E+00
6.59128E+00	1.86417E+00	6.59128E+00	2.08366E+00
7.22720E+00	1.86048E+00	7.22720E+00	2.07959E+00
7.92447E+00	1.85604E+00	7.92447E+00	2.07469E+00
8.68900E+00	1.85068E+00	8.68900E+00	2.06878E+00
9.52730E+00	1.84422E+00	9.52730E+00	2.06165E+00
1.04465E+01	1.83643E+00	1.04465E+01	2.05305E+00
1.14543E+01	1.82703E+00	1.14543E+01	2.04266E+00
1.25594E+01	1.81567E+00	1.25594E+01	2.03009E+00
1.37711E+01	1.80194E+00	1.37711E+01	2.01488E+00
1.50998E+01	1.78532E+00	1.50998E+01	1.99645E+00
1.65566E+01	1.76517E+00	1.65566E+01	1.97407E+00
1.81539E+01	1.74072E+00	1.81539E+01	1.94685E+00
1.99054E+01	1.71098E+00	1.99054E+01	1.91368E+00
2.18258E+01	1.67473E+00	2.18258E+01	1.87314E+00
2.39315E+01	1.63046E+00	2.39315E+01	1.82344E+00
2.62404E+01	1.57621E+00	2.62404E+01	1.76231E+00
2.87720E+01	1.50954E+00	2.87720E+01	1.68681E+00
3.15479E+01	1.42731E+00	3.15479E+01	1.59316E+00
3.45915E+01	1.32555E+00	3.45915E+01	1.47644E+00
3.79289E+01	1.19915E+00	3.79289E+01	1.33028E+00
4.15882E+01	1.04173E+00	4.15882E+01	1.14649E+00
4.56005E+01	8.45518E-01	4.56005E+01	9.14849E-01
5.00000E+01	6.01940E-01	5.00000E+01	6.23469E-01
5.48239E+01	3.06492E-01	5.48239E+01	2.63362E-01
6.01132E+01	1.41557E-01	6.01132E+01	2.13892E-01
6.59128E+01	5.44812E-01	6.59128E+01	7.40435E-01
7.22720E+01	1.04598E+00	7.22720E+01	1.34167E+00
7.92447E+01	1.58398E+00	7.92447E+01	1.96481E+00
8.68900E+01	2.11198E+00	8.68900E+01	2.54714E+00
9.52730E+01	2.58300E+00	9.52730E+01	3.03564E+00
1.04465E+02	2.96658E+00	1.04465E+02	3.40781E+00
1.14543E+02	3.25617E+00	1.14543E+02	3.67123E+00
1.25594E+02	3.46311E+00	1.25594E+02	3.84869E+00
1.37711E+02	3.60586E+00	1.37711E+02	3.96474E+00
1.50998E+02	3.70240E+00	1.50998E+02	4.03937E+00
1.65566E+02	3.76710E+00	1.65566E+02	4.08692E+00
1.81539E+02	3.81035E+00	1.81539E+02	4.11702E+00
1.99054E+02	3.83929E+00	1.99054E+02	4.13596E+00
2.18258E+02	3.85873E+00	2.18258E+02	4.14777E+00
2.39315E+02	3.87184E+00	2.39315E+02	4.15504E+00
2.62404E+02	3.88073E+00	2.62404E+02	4.15942E+00
2.87720E+02	3.88678E+00	2.87720E+02	4.16197E+00
3.15479E+02	3.89093E+00	3.15479E+02	4.16336E+00
3.45915E+02	3.89378E+00	3.45915E+02	4.16404E+00
3.79289E+02	3.89575E+00	3.79289E+02	4.16428E+00
4.15882E+02	3.89711E+00	4.15882E+02	4.16426E+00
4.56005E+02	3.89807E+00	4.56005E+02	4.16410E+00
5.00000E+02	3.89873E+00	5.00000E+02	4.16386E+00
5.00000E+02	3.89873E+00	5.00000E+02	4.16386E+00

FREQ (A9)	V 2-0 mag	FREQ (A10)	V 2-0 mag
5.00000E+00	2.27113E+00	5.00000E+00	1.86378E+00
5.48239E+00	2.26904E+00	5.48239E+00	1.86197E+00
6.01132E+00	2.26652E+00	6.01132E+00	1.85979E+00
6.59128E+00	2.26349E+00	6.59128E+00	1.85717E+00
7.22720E+00	2.25984E+00	7.22720E+00	1.85400E+00
7.92447E+00	2.25545E+00	7.92447E+00	1.85020E+00
8.68900E+00	2.25015E+00	8.68900E+00	1.84561E+00
9.52730E+00	2.24376E+00	9.52730E+00	1.84008E+00
1.04465E+01	2.23606E+00	1.04465E+01	1.83341E+00
1.14543E+01	2.22675E+00	1.14543E+01	1.82536E+00
1.25594E+01	2.21551E+00	1.25594E+01	1.81564E+00
1.37711E+01	2.20191E+00	1.37711E+01	1.80389E+00
1.50998E+01	2.18545E+00	1.50998E+01	1.78967E+00
1.65566E+01	2.16548E+00	1.65566E+01	1.77245E+00
1.81539E+01	2.14122E+00	1.81539E+01	1.75154E+00
1.99054E+01	2.11170E+00	1.99054E+01	1.72614E+00
2.18258E+01	2.07569E+00	2.18258E+01	1.69519E+00
2.39315E+01	2.03164E+00	2.39315E+01	1.65740E+00
2.62404E+01	1.97759E+00	2.62404E+01	1.61113E+00
2.87720E+01	1.91103E+00	2.87720E+01	1.55430E+00
3.15479E+01	1.82874E+00	3.15479E+01	1.48425E+00
3.45915E+01	1.72655E+00	3.45915E+01	1.39760E+00
3.79289E+01	1.59911E+00	3.79289E+01	1.29005E+00
4.15882E+01	1.43957E+00	4.15882E+01	1.15627E+00
4.56005E+01	1.23944E+00	4.56005E+01	9.90042E-01
5.00000E+01	9.89074E-01	5.00000E+01	7.85918E-01
5.48239E+01	6.81540E-01	5.48239E+01	5.48064E-01
6.01132E+01	3.45315E-01	6.01132E+01	3.52614E-01
6.59128E+01	3.77874E-01	6.59128E+01	4.98250E-01
7.22720E+01	8.90125E-01	7.22720E+01	9.42463E-01
7.92447E+01	1.50962E+00	7.92447E+01	1.49844E+00
8.68900E+01	2.14241E+00	8.68900E+01	2.08488E+00
9.52730E+01	2.71903E+00	9.52730E+01	2.62982E+00
1.04465E+02	3.19190E+00	1.04465E+02	3.07774E+00
1.14543E+02	3.54727E+00	1.14543E+02	3.40813E+00
1.25594E+02	3.79813E+00	1.25594E+02	3.63213E+00
1.37711E+02	3.96843E+00	1.37711E+02	3.77515E+00
1.50998E+02	4.08162E+00	1.50998E+02	3.86263E+00
1.65566E+02	4.15617E+00	1.65566E+02	3.91424E+00
1.81539E+02	4.20517E+00	1.81539E+02	3.94353E+00
1.99054E+02	4.23744E+00	1.99054E+02	3.95923E+00
2.18258E+02	4.25877E+00	2.18258E+02	3.96679E+00
2.39315E+02	4.27292E+00	2.39315E+02	3.96959E+00
2.62404E+02	4.28235E+00	2.62404E+02	3.96968E+00
2.87720E+02	4.28866E+00	2.87720E+02	3.96832E+00
3.15479E+02	4.29289E+00	3.15479E+02	3.96626E+00
3.45915E+02	4.29573E+00	3.45915E+02	3.96392E+00
3.79289E+02	4.29764E+00	3.79289E+02	3.96157E+00
4.15882E+02	4.29893E+00	4.15882E+02	3.95935E+00
4.56005E+02	4.29979E+00	4.56005E+02	3.95732E+00
5.00000E+02	4.30037E+00	5.00000E+02	3.95551E+00
5.00000E+02	4.30037E+00	5.00000E+02	3.95551E+00

FREQ (B1)	V 2-0 dB	FREQ (B2)	V 2-0 dB
5.00000E+00	6.97776E+00	5.00000E+00	5.84505E+00
5.48239E+00	6.97017E+00	5.48239E+00	5.83516E+00
6.01132E+00	6.96101E+00	6.01132E+00	5.82324E+00
6.59128E+00	6.94997E+00	6.59128E+00	5.80886E+00
7.22720E+00	6.93666E+00	7.22720E+00	5.79151E+00
7.92447E+00	6.92059E+00	7.92447E+00	5.77056E+00
8.68900E+00	6.90117E+00	8.68900E+00	5.74523E+00
9.52730E+00	6.87769E+00	9.52730E+00	5.71458E+00
1.04465E+01	6.84926E+00	1.04465E+01	5.67745E+00
1.14543E+01	6.81479E+00	1.14543E+01	5.63239E+00
1.25594E+01	6.77293E+00	1.25594E+01	5.57760E+00
1.37711E+01	6.72198E+00	1.37711E+01	5.51082E+00
1.50998E+01	6.65980E+00	1.50998E+01	5.42920E+00
1.65566E+01	6.58369E+00	1.65566E+01	5.32909E+00
1.81539E+01	6.49019E+00	1.81539E+01	5.20581E+00
1.99054E+01	6.37477E+00	1.99054E+01	5.05318E+00
2.18258E+01	6.23152E+00	2.18258E+01	4.86303E+00
2.39315E+01	6.05248E+00	2.39315E+01	4.62427E+00
2.62404E+01	5.82679E+00	2.62404E+01	4.32155E+00
2.87720E+01	5.53929E+00	2.87720E+01	3.93299E+00
3.15479E+01	5.16812E+00	3.15479E+01	3.42637E+00
3.45915E+01	4.68079E+00	3.45915E+01	2.75217E+00
3.79289E+01	4.02674E+00	3.79289E+01	1.82976E+00
4.15882E+01	3.12298E+00	4.15882E+01	5.17128E-01
4.56005E+01	1.82267E+00	4.56005E+01	-1.46826E+00
5.00000E+01	-1.62614E-01	5.00000E+01	-4.81969E+00
5.48239E+01	-3.48684E+00	5.48239E+01	-1.22067E+01
6.01132E+01	-9.37518E+00	6.01132E+01	-1.41001E+01
6.59128E+01	-7.03990E+00	6.59128E+01	-3.56505E+00
7.22720E+01	-4.56692E-02	7.22720E+01	1.51829E+00
7.92447E+01	4.30825E+00	7.92447E+01	4.77171E+00
8.68900E+01	7.16216E+00	8.68900E+01	6.99324E+00
9.52730E+01	9.04279E+00	9.52730E+01	8.51433E+00
1.04465E+02	1.02514E+01	1.04465E+02	9.54018E+00
1.14543E+02	1.10072E+01	1.14543E+02	1.02219E+01
1.25594E+02	1.14715E+01	1.25594E+02	1.06713E+01
1.37711E+02	1.17540E+01	1.37711E+02	1.09672E+01
1.50998E+02	1.19254E+01	1.50998E+02	1.11631E+01
1.65566E+02	1.20291E+01	1.65566E+02	1.12938E+01
1.81539E+02	1.20916E+01	1.81539E+02	1.13818E+01
1.99054E+02	1.21290E+01	1.99054E+02	1.14417E+01
2.18258E+02	1.21509E+01	2.18258E+02	1.14830E+01
2.39315E+02	1.21633E+01	2.39315E+02	1.15118E+01
2.62404E+02	1.21699E+01	2.62404E+02	1.15320E+01
2.87720E+02	1.21730E+01	2.87720E+02	1.15464E+01
3.15479E+02	1.21740E+01	3.15479E+02	1.15568E+01
3.45915E+02	1.21738E+01	3.45915E+02	1.15644E+01
3.79289E+02	1.21729E+01	3.79289E+02	1.15700E+01
4.15882E+02	1.21718E+01	4.15882E+02	1.15741E+01
4.56005E+02	1.21705E+01	4.56005E+02	1.15772E+01
5.00000E+02	1.21692E+01	5.00000E+02	1.15796E+01
5.00000E+02	1.21692E+01	5.00000E+02	1.15796E+01

FREQ (B3)	V 2-0 dB	FREQ (B4)	V 2-0 dB
5.00000E+00	6.80985E+00	5.00000E+00	4.59231E+00
5.48239E+00	6.80196E+00	5.48239E+00	4.58137E+00
6.01132E+00	6.79245E+00	6.01132E+00	4.56820E+00
6.59128E+00	6.78099E+00	6.59128E+00	4.55232E+00
7.22720E+00	6.76716E+00	7.22720E+00	4.53317E+00
7.92447E+00	6.75047E+00	7.92447E+00	4.51006E+00
8.68900E+00	6.73031E+00	8.68900E+00	4.48214E+00
9.52730E+00	6.70594E+00	9.52730E+00	4.44840E+00
1.04465E+01	6.67643E+00	1.04465E+01	4.40756E+00
1.14543E+01	6.64066E+00	1.14543E+01	4.35806E+00
1.25594E+01	6.59723E+00	1.25594E+01	4.29799E+00
1.37711E+01	6.54438E+00	1.37711E+01	4.22491E+00
1.50998E+01	6.47991E+00	1.50998E+01	4.13581E+00
1.65566E+01	6.40103E+00	1.65566E+01	4.02685E+00
1.81539E+01	6.30417E+00	1.81539E+01	3.89313E+00
1.99054E+01	6.18467E+00	1.99054E+01	3.72828E+00
2.18258E+01	6.03644E+00	2.18258E+01	3.52395E+00
2.39315E+01	5.85130E+00	2.39315E+01	3.26896E+00
2.62404E+01	5.61812E+00	2.62404E+01	2.94807E+00
2.87720E+01	5.32129E+00	2.87720E+01	2.53988E+00
3.15479E+01	4.93839E+00	3.15479E+01	2.01345E+00
3.45915E+01	4.43591E+00	3.45915E+01	1.32208E+00
3.79289E+01	3.76152E+00	3.79289E+01	3.91357E-01
4.15882E+01	2.82834E+00	4.15882E+01	-9.06884E-01
4.56005E+01	1.47934E+00	4.56005E+01	-2.82069E+00
5.00000E+01	-6.08978E-01	5.00000E+01	-5.93511E+00
5.48239E+01	-4.27478E+00	5.48239E+01	-1.23471E+01
6.01132E+01	-1.30091E+01	6.01132E+01	-2.11608E+01
6.59128E+01	-9.23901E+00	6.59128E+01	-6.86186E+00
7.22720E+01	-5.65926E-01	7.22720E+01	-1.14709E+00
7.92447E+01	4.10512E+00	7.92447E+01	2.50271E+00
8.68900E+01	7.13330E+00	8.68900E+01	5.10628E+00
9.52730E+01	9.14750E+00	9.52730E+01	7.01638E+00
1.04465E+02	1.04563E+01	1.04465E+02	8.41287E+00
1.14543E+02	1.12797E+01	1.14543E+02	9.41929E+00
1.25594E+02	1.17846E+01	1.25594E+02	1.01341E+01
1.37711E+02	1.20891E+01	1.37711E+02	1.06367E+01
1.50998E+02	1.22707E+01	1.50998E+02	1.09887E+01
1.65566E+02	1.23779E+01	1.65566E+02	1.12355E+01
1.81539E+02	1.24402E+01	1.81539E+02	1.14095E+01
1.99054E+02	1.24755E+01	1.99054E+02	1.15331E+01
2.18258E+02	1.24945E+01	2.18258E+02	1.16218E+01
2.39315E+02	1.25038E+01	2.39315E+02	1.16862E+01
2.62404E+02	1.25074E+01	2.62404E+02	1.17334E+01
2.87720E+02	1.25077E+01	2.87720E+02	1.17685E+01
3.15479E+02	1.25062E+01	3.15479E+02	1.17948E+01
3.45915E+02	1.25038E+01	3.45915E+02	1.18148E+01
3.79289E+02	1.25010E+01	3.79289E+02	1.18301E+01
4.15882E+02	1.24983E+01	4.15882E+02	1.18419E+01
4.56005E+02	1.24956E+01	4.56005E+02	1.18511E+01
5.00000E+02	1.24931E+01	5.00000E+02	1.18584E+01
5.00000E+02	1.24931E+01	5.00000E+02	1.18584E+01

FREQ (B5)	V 2-0 dB	FREQ (B6)	V 2-0 dB
5.00000E+00	5.55880E+00	5.00000E+00	6.21202E+00
5.48239E+00	5.55042E+00	5.48239E+00	6.20206E+00
6.01132E+00	5.54032E+00	6.01132E+00	6.19005E+00
6.59128E+00	5.52814E+00	6.59128E+00	6.17557E+00
7.22720E+00	5.51346E+00	7.22720E+00	6.15809E+00
7.92447E+00	5.49575E+00	7.92447E+00	6.13699E+00
8.68900E+00	5.47437E+00	8.68900E+00	6.11147E+00
9.52730E+00	5.44854E+00	9.52730E+00	6.08060E+00
1.04465E+01	5.41729E+00	1.04465E+01	6.04319E+00
1.14543E+01	5.37944E+00	1.14543E+01	5.99779E+00
1.25594E+01	5.33353E+00	1.25594E+01	5.94257E+00
1.37711E+01	5.27775E+00	1.37711E+01	5.87527E+00
1.50998E+01	5.20982E+00	1.50998E+01	5.79300E+00
1.65566E+01	5.12687E+00	1.65566E+01	5.69207E+00
1.81539E+01	5.02526E+00	1.81539E+01	5.56773E+00
1.99054E+01	4.90028E+00	1.99054E+01	5.41375E+00
2.18258E+01	4.74584E+00	2.18258E+01	5.22184E+00
2.39315E+01	4.55383E+00	2.39315E+01	4.98075E+00
2.62404E+01	4.31336E+00	2.62404E+01	4.67486E+00
2.87720E+01	4.00945E+00	2.87720E+01	4.28192E+00
3.15479E+01	3.62092E+00	3.15479E+01	3.76905E+00
3.45915E+01	3.11690E+00	3.45915E+01	3.08558E+00
3.79289E+01	2.45051E+00	3.79289E+01	2.14880E+00
4.15882E+01	1.54664E+00	4.15882E+01	8.12555E-01
4.56005E+01	2.75878E-01	4.56005E+01	-1.21467E+00
5.00000E+01	-1.61002E+00	5.00000E+01	-4.64486E+00
5.48239E+01	-4.67611E+00	5.48239E+01	-1.19637E+01
6.01132E+01	-1.05892E+01	6.01132E+01	-1.14920E+01
6.59128E+01	-1.38685E+01	6.59128E+01	-2.49119E+00
7.22720E+01	-4.57575E+00	7.22720E+01	2.33995E+00
7.92447E+01	6.19503E-01	7.92447E+01	5.49080E+00
8.68900E+01	4.00178E+00	8.68900E+01	7.64974E+00
9.52730E+01	6.35748E+00	9.52730E+01	9.12378E+00
1.04465E+02	8.01247E+00	1.04465E+02	1.01118E+01
1.14543E+02	9.16083E+00	1.14543E+02	1.07630E+01
1.25594E+02	9.94556E+00	1.25594E+02	1.11884E+01
1.37711E+02	1.04763E+01	1.37711E+02	1.14657E+01
1.50998E+02	1.08342E+01	1.50998E+02	1.16472E+01
1.65566E+02	1.10760E+01	1.65566E+02	1.17667E+01
1.81539E+02	1.12406E+01	1.81539E+02	1.18461E+01
1.99054E+02	1.13537E+01	1.99054E+02	1.18993E+01
2.18258E+02	1.14323E+01	2.18258E+02	1.19353E+01
2.39315E+02	1.14876E+01	2.39315E+02	1.19598E+01
2.62404E+02	1.15269E+01	2.62404E+02	1.19768E+01
2.87720E+02	1.15552E+01	2.87720E+02	1.19885E+01
3.15479E+02	1.15759E+01	3.15479E+02	1.19968E+01
3.45915E+02	1.15912E+01	3.45915E+02	1.20026E+01
3.79289E+02	1.16026E+01	3.79289E+02	1.20067E+01
4.15882E+02	1.16112E+01	4.15882E+02	1.20097E+01
4.56005E+02	1.16177E+01	4.56005E+02	1.20119E+01
5.00000E+02	1.16228E+01	5.00000E+02	1.20135E+01
5.00000E+02	1.16228E+01	5.00000E+02	1.20135E+01

FREQ (B7)	V 2-0 dB	FREQ (B8)	V 2-0 dB
5.00000E+00	5.44565E+00	5.00000E+00	6.41195E+00
5.48239E+00	5.43583E+00	5.48239E+00	6.40228E+00
6.01132E+00	5.42399E+00	6.01132E+00	6.39061E+00
6.59128E+00	5.40973E+00	6.59128E+00	6.37655E+00
7.22720E+00	5.39251E+00	7.22720E+00	6.35957E+00
7.92447E+00	5.37174E+00	7.92447E+00	6.33907E+00
8.68900E+00	5.34663E+00	8.68900E+00	6.31430E+00
9.52730E+00	5.31627E+00	9.52730E+00	6.28432E+00
1.04465E+01	5.27951E+00	1.04465E+01	6.24799E+00
1.14543E+01	5.23492E+00	1.14543E+01	6.20391E+00
1.25594E+01	5.18076E+00	1.25594E+01	6.15032E+00
1.37711E+01	5.11482E+00	1.37711E+01	6.08500E+00
1.50998E+01	5.03432E+00	1.50998E+01	6.00517E+00
1.65566E+01	4.93575E+00	1.65566E+01	5.90726E+00
1.81539E+01	4.81457E+00	1.81539E+01	5.78667E+00
1.99054E+01	4.66489E+00	1.99054E+01	5.63738E+00
2.18258E+01	4.47892E+00	2.18258E+01	5.45138E+00
2.39315E+01	4.24618E+00	2.39315E+01	5.21782E+00
2.62404E+01	3.95227E+00	2.62404E+01	4.92162E+00
2.87720E+01	3.57687E+00	2.87720E+01	4.54131E+00
3.15479E+01	3.09039E+00	3.15479E+01	4.04517E+00
3.45915E+01	2.44789E+00	3.45915E+01	3.38430E+00
3.79289E+01	1.57745E+00	3.79289E+01	2.47885E+00
4.15882E+01	3.55091E-01	4.15882E+01	1.18743E+00
4.56005E+01	-1.45754E+00	4.56005E+01	-7.73007E-01
5.00000E+01	-4.40893E+00	5.00000E+01	-4.10370E+00
5.48239E+01	-1.02716E+01	5.48239E+01	-1.15889E+01
6.01132E+01	-1.69814E+01	6.01132E+01	-1.33961E+01
6.59128E+01	-5.27507E+00	6.59128E+01	-2.61026E+00
7.22720E+01	3.90463E-01	7.22720E+01	2.55292E+00
7.92447E+01	3.99497E+00	7.92447E+01	5.86640E+00
8.68900E+01	6.49379E+00	8.68900E+01	8.12106E+00
9.52730E+01	8.24250E+00	9.52730E+01	9.64499E+00
1.04465E+02	9.44513E+00	1.04465E+02	1.06495E+01
1.14543E+02	1.02541E+01	1.14543E+02	1.12962E+01
1.25594E+02	1.07893E+01	1.25594E+02	1.17063E+01
1.37711E+02	1.11402E+01	1.37711E+02	1.19643E+01
1.50998E+02	1.13697E+01	1.50998E+02	1.21263E+01
1.65566E+02	1.15201E+01	1.65566E+02	1.22279E+01
1.81539E+02	1.16193E+01	1.81539E+02	1.22917E+01
1.99054E+02	1.16850E+01	1.99054E+02	1.23315E+01
2.18258E+02	1.17289E+01	2.18258E+02	1.23563E+01
2.39315E+02	1.17584E+01	2.39315E+02	1.23715E+01
2.62404E+02	1.17783E+01	2.62404E+02	1.23807E+01
2.87720E+02	1.17918E+01	2.87720E+02	1.23860E+01
3.15479E+02	1.18011E+01	3.15479E+02	1.23889E+01
3.45915E+02	1.18074E+01	3.45915E+02	1.23903E+01
3.79289E+02	1.18118E+01	3.79289E+02	1.23908E+01
4.15882E+02	1.18149E+01	4.15882E+02	1.23908E+01
4.56005E+02	1.18170E+01	4.56005E+02	1.23904E+01
5.00000E+02	1.18185E+01	5.00000E+02	1.23899E+01
5.00000E+02	1.18185E+01	5.00000E+02	1.23899E+01

FREQ (B9)	V 2-0 dB	FREQ (B10)	V 2-0 dB
5.00000E+00	7.12484E+00	5.00000E+00	5.40791E+00
5.48239E+00	7.11684E+00	5.48239E+00	5.39946E+00
6.01132E+00	7.10720E+00	6.01132E+00	5.38928E+00
6.59128E+00	7.09558E+00	6.59128E+00	5.37701E+00
7.22720E+00	7.08157E+00	7.22720E+00	5.36222E+00
7.92447E+00	7.06466E+00	7.92447E+00	5.34436E+00
8.68900E+00	7.04423E+00	8.68900E+00	5.32279E+00
9.52730E+00	7.01954E+00	9.52730E+00	5.29673E+00
1.04465E+01	6.98965E+00	1.04465E+01	5.26519E+00
1.14543E+01	6.95344E+00	1.14543E+01	5.22697E+00
1.25594E+01	6.90947E+00	1.25594E+01	5.18059E+00
1.37711E+01	6.85600E+00	1.37711E+01	5.12420E+00
1.50998E+01	6.79080E+00	1.50998E+01	5.05547E+00
1.65566E+01	6.71108E+00	1.65566E+01	4.97146E+00
1.81539E+01	6.61324E+00	1.81539E+01	4.86842E+00
1.99054E+01	6.49266E+00	1.99054E+01	4.74150E+00
2.18258E+01	6.34327E+00	2.18258E+01	4.58436E+00
2.39315E+01	6.15695E+00	2.39315E+01	4.38855E+00
2.62404E+01	5.92274E+00	2.62404E+01	4.14262E+00
2.87720E+01	5.62536E+00	2.87720E+01	3.83070E+00
3.15479E+01	5.24303E+00	3.15479E+01	3.43016E+00
3.45915E+01	4.74359E+00	3.45915E+01	2.90768E+00
3.79289E+01	4.07759E+00	3.79289E+01	2.21216E+00
4.15882E+01	3.16468E+00	4.15882E+01	1.26115E+00
4.56005E+01	1.86449E+00	4.56005E+01	-8.69244E-02
5.00000E+01	-9.54230E-02	5.00000E+01	-2.09246E+00
5.48239E+01	-3.33017E+00	5.48239E+01	-5.22338E+00
6.01132E+01	-9.23570E+00	6.01132E+01	-9.05400E+00
6.59128E+01	-8.45307E+00	6.59128E+01	-6.05105E+00
7.22720E+01	-1.01098E+00	7.22720E+01	-5.14714E-01
7.92447E+01	3.57733E+00	7.92447E+01	3.51280E+00
8.68900E+01	6.61806E+00	8.68900E+01	6.38161E+00
9.52730E+01	8.68827E+00	9.52730E+01	8.39851E+00
1.04465E+02	1.00810E+01	1.04465E+02	9.76463E+00
1.14543E+02	1.09979E+01	1.14543E+02	1.06503E+01
1.25594E+02	1.15914E+01	1.25594E+02	1.12032E+01
1.37711E+02	1.19724E+01	1.37711E+02	1.15387E+01
1.50998E+02	1.22166E+01	1.50998E+02	1.17377E+01
1.65566E+02	1.23739E+01	1.65566E+02	1.18530E+01
1.81539E+02	1.24757E+01	1.81539E+02	1.19177E+01
1.99054E+02	1.25421E+01	1.99054E+02	1.19522E+01
2.18258E+02	1.25857E+01	2.18258E+02	1.19688E+01
2.39315E+02	1.26145E+01	2.39315E+02	1.19749E+01
2.62404E+02	1.26336E+01	2.62404E+02	1.19751E+01
2.87720E+02	1.26464E+01	2.87720E+02	1.19721E+01
3.15479E+02	1.26550E+01	3.15479E+02	1.19676E+01
3.45915E+02	1.26607E+01	3.45915E+02	1.19625E+01
3.79289E+02	1.26646E+01	3.79289E+02	1.19574E+01
4.15882E+02	1.26672E+01	4.15882E+02	1.19525E+01
4.56005E+02	1.26690E+01	4.56005E+02	1.19480E+01
5.00000E+02	1.26701E+01	5.00000E+02	1.19440E+01
5.00000E+02	1.26701E+01	5.00000E+02	1.19440E+01

APPENDIX B

STATISTICAL RESULTS OF EXAMPLE 2 IN NUMERICAL FORM

In this appendix, the numerical data (in absolute value and in dB value) of example 2 in Chapter V with sample population equal to 10 is provided. This is Watand output with PRINT output action. Note that the last two frequencies are not exactly equal if digit precision is taken into consideration.

mcdi fr use boctor on d out v 2 pr mag noin

MCFR V1.11-3g SR137020 30-JAN-89 10:36:50 YSUCMS FILE: BOCTOR

ABS VALU: V 2-0

FREQ	INITIAL	MINIMUM	MAXIMUM	MEAN	ST.DEV
5.00000E+00	1.98890E+00	1.69674E+00	2.27113E+00	2.01200E+00	1.76717E-01
5.48239E+00	1.98679E+00	1.69461E+00	2.26904E+00	2.00991E+00	1.76703E-01
6.01132E+00	1.98424E+00	1.69204E+00	2.26652E+00	2.00741E+00	1.76686E-01
6.59128E+00	1.98118E+00	1.68895E+00	2.26349E+00	2.00439E+00	1.76665E-01
7.22720E+00	1.97748E+00	1.68523E+00	2.25984E+00	2.00075E+00	1.76639E-01
7.92447E+00	1.97304E+00	1.68075E+00	2.25545E+00	1.99637E+00	1.76608E-01
8.68900E+00	1.96768E+00	1.67536E+00	2.25015E+00	1.99110E+00	1.76570E-01
9.52730E+00	1.96121E+00	1.66886E+00	2.24376E+00	1.98473E+00	1.76523E-01
1.04465E+01	1.95342E+00	1.66103E+00	2.23606E+00	1.97706E+00	1.76467E-01
1.14543E+01	1.94400E+00	1.65159E+00	2.22675E+00	1.96779E+00	1.76397E-01
1.25594E+01	1.93263E+00	1.64021E+00	2.21551E+00	1.95659E+00	1.76312E-01
1.37711E+01	1.91887E+00	1.62647E+00	2.20191E+00	1.94305E+00	1.76206E-01
1.50998E+01	1.90221E+00	1.60987E+00	2.18545E+00	1.92665E+00	1.76075E-01
1.65566E+01	1.88202E+00	1.58980E+00	2.16548E+00	1.90676E+00	1.75912E-01
1.81539E+01	1.85749E+00	1.56551E+00	2.14122E+00	1.88261E+00	1.75708E-01
1.99054E+01	1.82764E+00	1.53608E+00	2.11170E+00	1.85322E+00	1.75454E-01
2.18258E+01	1.79123E+00	1.50037E+00	2.07569E+00	1.81737E+00	1.75136E-01
2.39315E+01	1.74671E+00	1.45696E+00	2.03164E+00	1.77352E+00	1.74738E-01
2.62404E+01	1.69210E+00	1.40412E+00	1.97759E+00	1.71974E+00	1.74244E-01
2.87720E+01	1.62488E+00	1.33966E+00	1.91103E+00	1.65353E+00	1.73642E-01
3.15479E+01	1.54182E+00	1.26088E+00	1.82874E+00	1.57172E+00	1.72931E-01
3.45915E+01	1.43877E+00	1.16441E+00	1.72655E+00	1.47021E+00	1.72149E-01
3.79289E+01	1.31037E+00	1.04609E+00	1.59911E+00	1.34375E+00	1.71419E-01
4.15882E+01	1.14979E+00	9.00857E-01	1.43957E+00	1.18568E+00	1.71063E-01
4.56005E+01	9.48422E-01	7.22713E-01	1.23944E+00	9.87796E-01	1.71813E-01
5.00000E+01	6.95922E-01	5.04946E-01	9.89074E-01	7.40984E-01	1.75166E-01
5.48239E+01	3.80851E-01	2.41349E-01	6.81540E-01	4.40270E-01	1.82833E-01
6.01132E+01	7.13308E-03	8.74903E-02	3.52614E-01	2.46337E-01	8.53429E-02
6.59128E+01	4.72263E-01	2.02571E-01	7.50655E-01	5.02162E-01	1.67878E-01
7.22720E+01	1.00541E+00	5.90490E-01	1.34167E+00	1.01189E+00	2.12581E-01
7.92447E+01	1.57672E+00	1.07393E+00	1.96481E+00	1.58249E+00	2.43903E-01
8.68900E+01	2.13752E+00	1.58522E+00	2.54714E+00	2.14756E+00	2.67141E-01
9.52730E+01	2.63639E+00	2.07909E+00	3.03564E+00	2.65124E+00	2.78208E-01
1.04465E+02	3.04006E+00	2.51549E+00	3.40781E+00	3.05841E+00	2.76032E-01
1.14543E+02	3.34215E+00	2.87105E+00	3.67123E+00	3.36237E+00	2.63994E-01
1.25594E+02	3.55592E+00	3.14252E+00	3.88354E+00	3.57679E+00	2.47358E-01
1.37711E+02	3.70195E+00	3.34054E+00	4.02210E+00	3.72279E+00	2.30395E-01
1.50998E+02	3.79982E+00	3.48103E+00	4.10709E+00	3.82026E+00	2.15403E-01
1.65566E+02	3.86485E+00	3.57932E+00	4.15810E+00	3.88477E+00	2.03121E-01
1.81539E+02	3.90796E+00	3.64780E+00	4.20517E+00	3.92733E+00	1.93450E-01
1.99054E+02	3.93660E+00	3.69562E+00	4.23744E+00	3.95544E+00	1.85974E-01
2.18258E+02	3.95568E+00	3.72920E+00	4.25877E+00	3.97404E+00	1.80230E-01
2.39315E+02	3.96844E+00	3.75300E+00	4.27292E+00	3.98638E+00	1.75812E-01
2.62404E+02	3.97701E+00	3.77004E+00	4.28235E+00	3.99459E+00	1.72397E-01
2.87720E+02	3.98280E+00	3.77852E+00	4.28866E+00	4.00006E+00	1.69739E-01

3.15479E+02	3.98671E+00	3.78304E+00	4.29289E+00	4.00371E+00	1.67654E-01
3.45915E+02	3.98937E+00	3.78634E+00	4.29573E+00	4.00615E+00	1.66007E-01
3.79289E+02	3.99118E+00	3.78877E+00	4.29764E+00	4.00777E+00	1.64696E-01
4.15882E+02	3.99242E+00	3.79058E+00	4.29893E+00	4.00885E+00	1.63646E-01
4.56005E+02	3.99326E+00	3.79194E+00	4.29979E+00	4.00956E+00	1.62801E-01
5.00000E+02	3.99384E+00	3.79297E+00	4.30037E+00	4.01003E+00	1.62117E-01
5.00000E+02	3.99384E+00	3.79297E+00	4.30037E+00	4.01003E+00	1.62117E-01

DISPLAY TIME= 0.563 SEC.

mcdi fr db noin

MCFR V1.11-3g SR137020 30-JAN-89 10:36:50 YSUCMS FILE: BOCTOR

dB ABS: V 2-0

FREQ	INITIAL	MINIMUM	MAXIMUM	MEAN	ST.DEV
5.00000E+00	5.97228E+00	4.59231E+00	7.12484E+00	6.03862E+00	7.70391E-01
5.48239E+00	5.96303E+00	4.58137E+00	7.11684E+00	6.02955E+00	7.71125E-01
6.01132E+00	5.95189E+00	4.56820E+00	7.10720E+00	6.01864E+00	7.72010E-01
6.59128E+00	5.93846E+00	4.55232E+00	7.09558E+00	6.00547E+00	7.73078E-01
7.22720E+00	5.92226E+00	4.53317E+00	7.08157E+00	5.98959E+00	7.74367E-01
7.92447E+00	5.90270E+00	4.51006E+00	7.06466E+00	5.97042E+00	7.75925E-01
8.68900E+00	5.87907E+00	4.48214E+00	7.04423E+00	5.94727E+00	7.77809E-01
9.52730E+00	5.85049E+00	4.44840E+00	7.01954E+00	5.91926E+00	7.80090E-01
1.04465E+01	5.81589E+00	4.40756E+00	6.98965E+00	5.88535E+00	7.82858E-01
1.14543E+01	5.77393E+00	4.35806E+00	6.95344E+00	5.84424E+00	7.86222E-01
1.25594E+01	5.72296E+00	4.29799E+00	6.90947E+00	5.79430E+00	7.90319E-01
1.37711E+01	5.66092E+00	4.22491E+00	6.85600E+00	5.73351E+00	7.95325E-01
1.50998E+01	5.58519E+00	4.13581E+00	6.79080E+00	5.65933E+00	8.01462E-01
1.65566E+01	5.49248E+00	4.02685E+00	6.71108E+00	5.56852E+00	8.09018E-01
1.81539E+01	5.37852E+00	3.89313E+00	6.61324E+00	5.45692E+00	8.18373E-01
1.99054E+01	5.23780E+00	3.72828E+00	6.49266E+00	5.31914E+00	8.30037E-01
2.18258E+01	5.06303E+00	3.52395E+00	6.34327E+00	5.14805E+00	8.44710E-01
2.39315E+01	4.84441E+00	3.26896E+00	6.15695E+00	4.93411E+00	8.63378E-01
2.62404E+01	4.56851E+00	2.94807E+00	5.92274E+00	4.66420E+00	8.87489E-01
2.87720E+01	4.21642E+00	2.53988E+00	5.62536E+00	4.31991E+00	9.19257E-01
3.15479E+01	3.76069E+00	2.01345E+00	5.24303E+00	3.87451E+00	9.62254E-01
3.45915E+01	3.15983E+00	1.32208E+00	4.74359E+00	3.28769E+00	1.02265E+00
3.79289E+01	2.34790E+00	3.91357E-01	4.07759E+00	2.49547E+00	1.11206E+00
4.15882E+01	1.21235E+00	-9.06884E-01	3.16468E+00	1.38891E+00	1.25482E+00
4.56005E+01	-4.59970E-01	-2.82069E+00	1.86449E+00	-2.37871E-01	1.51044E+00
5.00000E+01	-3.14879E+00	-5.93511E+00	-9.54230E-02	-2.84818E+00	2.06311E+00
5.48239E+01	-8.38490E+00	-1.23471E+01	-3.33017E+00	-7.93693E+00	3.80941E+00
6.01132E+01	-4.29345E+01	-2.11608E+01	-9.05400E+00	-1.28394E+01	3.66777E+00
6.59128E+01	-6.51631E+00	-1.38685E+01	-2.49119E+00	-6.54549E+00	3.27917E+00
7.22720E+01	4.69071E-02	-4.57575E+00	2.55292E+00	-1.05851E-01	1.95451E+00
7.92447E+01	3.95508E+00	6.19503E-01	5.86640E+00	3.87496E+00	1.42607E+00
8.68900E+01	6.59822E+00	4.00178E+00	8.12106E+00	6.56610E+00	1.14893E+00
9.52730E+01	8.42020E+00	6.35748E+00	9.64499E+00	8.41765E+00	9.61767E-01
1.04465E+02	9.65766E+00	8.01247E+00	1.06495E+01	9.67252E+00	8.17550E-01
1.14543E+02	1.04805E+01	9.16083E+00	1.12962E+01	1.05051E+01	7.02689E-01
1.25594E+02	1.10190E+01	9.94556E+00	1.17846E+01	1.10486E+01	6.12676E-01
1.37711E+02	1.13686E+01	1.04763E+01	1.20891E+01	1.14005E+01	5.44085E-01
1.50998E+02	1.15953E+01	1.08342E+01	1.22707E+01	1.16279E+01	4.92995E-01
1.65566E+02	1.17426E+01	1.10760E+01	1.23779E+01	1.17754E+01	4.55431E-01
1.81539E+02	1.18390E+01	1.12406E+01	1.24757E+01	1.18714E+01	4.27918E-01
1.99054E+02	1.19024E+01	1.13537E+01	1.25421E+01	1.19343E+01	4.07708E-01
2.18258E+02	1.19444E+01	1.14323E+01	1.25857E+01	1.19757E+01	3.92754E-01
2.39315E+02	1.19724E+01	1.14876E+01	1.26145E+01	1.20032E+01	3.81578E-01
2.62404E+02	1.19911E+01	1.15269E+01	1.26336E+01	1.20214E+01	3.73135E-01
2.87720E+02	1.20038E+01	1.15464E+01	1.26464E+01	1.20336E+01	3.66686E-01

3.15479E+02	1.20123E+01	1.15568E+01	1.26550E+01	1.20417E+01	3.61707E-01
3.45915E+02	1.20181E+01	1.15644E+01	1.26607E+01	1.20471E+01	3.57824E-01
3.79289E+02	1.20220E+01	1.15700E+01	1.26646E+01	1.20508E+01	3.54769E-01
4.15882E+02	1.20247E+01	1.15741E+01	1.26672E+01	1.20532E+01	3.52346E-01
4.56005E+02	1.20266E+01	1.15772E+01	1.26690E+01	1.20548E+01	3.50411E-01
5.00000E+02	1.20278E+01	1.15796E+01	1.26701E+01	1.20559E+01	3.48855E-01
5.00000E+02	1.20278E+01	1.15796E+01	1.26701E+01	1.20559E+01	3.48855E-01

DISPLAY TIME= 0.593 SEC.

APPENDIX C

HELP FILES FOR MCFR AND MCDI

In this appendix, the help files for the MCFR analysis and the MCDI post-processor are offered. These files join the one for MCDC and appear with the Youngstown State University Electrical Engineering helps.

MCFR ANALYSIS 2/07/89

This user-defined Monte-Carlo analysis command finds minimum, maximum, mean, and standard-deviation values of FR-analyses outputs. Element and model-parameter values may be varied randomly.

```
+--------+-------------------------------------------------------------+
| MCFR   |  WRTD  typ.naml<.Vn> <ml pl>   typ.nam2...                  |
|        |                                                             |
|        |  WRTM  typ.mnaml  pnaml<.Vn> <ml pl>  pnam2 ...            |
|        |          <END>  typ.mnam2 ...                               |
|        |                                                             |
|        |  < FR  fr-analysis specs >           < options >            |
+--------+-------------------------------------------------------------+
```

options: (defaults listed first)

```
NS      10 | nsamp
DSEED    0 | seed
IPDC   OFF | ON
AS     ALL | INI | MIN | MAX | AVG | STD
```

where: (defaults)

WRTD specifies the #DATA-level element values to be randomly varied:

- typ.nam specifies element type and name;
- Vn is an optional value qualifier (n=1);
- m,p are numbers passed to the subroutine which generates random values for typ.nam. (m=0, p=0)

WRTM specifies which #MODEL-level parameters are to be varied:

- typ.mnam specifies element type and model name;
- pnam specifies the parameter of the model to be varied;
- Vn is an optional parameter-value qualifier (n=1);
- m,p are numbers passed to the subroutine which generates random values for pnam (m=0, p=0);
- END may be used to prevent ambiguity between a parameter name and an element type.

FR Any normal FR-analysis parameters may be specified as desired, except that 'KEep NOne', COmplex, and CPolar are not allowed. FR KEEP defaults to 'KEEP ALL IN MCFR ON A' and allows only ALL or OUTPUTS. FR IPOINT uses DC or user name depending upon the IPDC setting and the circuit. (User settings)

NS specifies the number of FR-analysis executions to be performed using random element/model-parameter values. (nsamp=10)

DSEED specifies a seed value for random number generation. For seed =0, a random seed is generated from the CPU time. (seed=0)

IPDC specifies running DC analysis before each FR analysis to generate new DC initial points for FR to use. When the circuit has no models, the setting is ignored and assumed OFF. When IPDC is ON, FR uses IP DC and ignores other ipnames. (OFF)

AS specifies which statistical output mode to display. (ALL)

- ALL - all of the following five modes for one plotted OUtput or one to seven printed OUtputs;
- INI - initial element- and model-value FR analysis results displayed for one to seven specified OUtputs;
- MIN - minimum values generated at each frequency for one to seven outputs;
- MAX - maximum values;
- AVG - average values;
- STD - standard-deviation values.

Notes:

1. MCFR may be used only in the WEXEC environment. For long input lines it may be useful to write a WEXEC macro to call MCFR, and then make alterations in it as needed.

2. At least one WRTD or WRTM specification is required.

3. When IPDC is ON, an initial point named DC must be available. That is, one should run DC analysis before using MCFR IPDC ON.

4. Caution must be exercised when using MCFR FR KEEP ALL. Results will be incorrect when output is requested which must be caluculated using an element value which was statistically altered. For example, if the current in a resistor is requested, it would be calculated as voltage divided by the present resistor value.

5. All FR analyses KEEP data is in complex form. Therefore MCDI, like MCFR, may be used to display any output type except CO and CP. DI may be used with any output type including CO and CP to display all of the kept data at once in printed or plotted form.

6. MCFR uses FR, DC, and DI analyses and may change some user settings.

7. See the MCDC help for information about the random number generator.

Examples:

```
MCFR  NS 20 WRTM N.Q1 BF 20 IS 0.5  WRTD R.B R.E 15 *
    FR OUT I R.L V 20 40 PR IPDC ON
MCFR WRTD R.B 20 V.IN.V3 5 FR KE ALL IN RUN1 PL AS STD
```

MCDI POST-PROCESSOR

2/10/89

This user-defined Monte-Carlo post-processor command reads kept results of a Monte-Carlo analysis, MCDC or MCFR, and displays them according to parameters entered.

```
+--------+-----------------------------------------------------+
| MCDI   |     DC                                              |
|        |     OU  out-specs                                   |
|        |   < NOIN >                                          |
|        |   < USE   fname >                                   |
|        |   < FM    fmode >                                   |
+--------+-----------------------------------------------------+
| MCDI   |     FR  < DI-analysis specs >                       |
|        |   < NOIN >                                          |
|        |   < AS   as-specs >                                 |
+--------+-----------------------------------------------------+
| MCDI   |     ZO                                              |
+--------+-----------------------------------------------------+
```

options: (defaults listed first)

```
NOIN
USE  MCDC | fname
FM   A | fmode
AS   ALL | INI | MIN | MAX | AVG | STD
```

where: (defaults)

DC — indicates that MCDC kept results are to be processed and displayed.

OU — specifies what outputs are to be displayed. OU is required to display MCDC results generated with DC KE ALL, but is ignored otherwise. Valid out-specs are node voltages and auxiliary variables. To identify these, see the display produced by DC PR OU ALL or by #Q #IP(DC). Up to seven outputs may be specified at one time.

USE — specifies the name of MCDC KEEP files to be used. (fname=MCDC).

NOIN — Entering this parameter tells MCDI not to display the input statistical data. (The default is input data displayed.)

FM — specifies the file mode of MCDC KEEP files. (fmode=A).

FR — indicates that MCFR kept results are to be processed and displayed. Any normal DI-analysis parameters may follow this parameter.

AS specifies which statistical output mode to display. (ALL)

- ALL - all of the following five modes for one plotted OUtput or one to seven printed OUtputs;
- INI - initial element- and model-value FR analysis results displayed for one to seven specified OUtputs;
- MIN - minimum values generated at each frequency for one to seven outputs;
- MAX - maximum values;
- AVG - average values;
- STD - standard-deviation values.

ZO specifies zoom action and must be entered alone on the MCDI line. MCDI ZO must follow a previous MCDI FR with PL output action or a previous MCDI ZO. Compare DI ZOom.

Notes:

1. MCDI can be used in conjunction with MCDC and MCFR to display more than seven output values. For example, 'MCDC DC KE ALL' would be used once to generate the results, and then repeated use of MCDI DC can be used to display up to seven outputs at one time. Only 'DC OU ALL' results are available with this approach.

2. MCDI FR can be used to look at different aspects of MCFR results. For example, one can change the AS values and/or the output type (MA, DB, etc.) with subsequent displays. Output action can also be changed to obtain different displays of results, e.g., PR, PL, etc.

3. Caution must be exercised when using MCFR FR KEEP ALL data. Results will be incorrect when output is requested which must be caluculated using an element value which was statistically altered. For example, if the current in a resistor is requested, it would be calculated as voltage divided by the present resistor value.

4. Correct results are always obtained from MCDC and MCFR KEEP OUTPUTS since only the kept outputs can be displayed.

Examples:

```
MCDI DC NOIN
MCDI USE RUN1  OU V 1 V 25 I V.CC DC
MCDI DC OU V 10 V 20 V 30 I V.IN  USE RUN2 FM D
MCDI FR USE RUN1 ON D OUT V 20 30 V C.1 I R.2 PR DB
MCDI NOIN FR AS AVG
MCDI FR VB -6 10 IB 1D-3 5D-3 XB 1ME 2ME PP
```

APPENDIX D

FORTRAN SUBROUTINES

In this appendix, the Fortran subroutines designed for the MCFR analysis and the MCDI FR post-processor are listed in alphabetical order. Programming comments can be seen in each subroutine. The subroutine

MDCHCK is called from the MCFR, MFRDI, and MFRZO macros,

MDINPT is called from the MFRDI macro,

MDINTL is called from the MCFR macro,

MDREAD is called from the MCFR macro,

MFDADS is called by subroutine MFRUSR,

MFRNEW is called by subroutine MFRUSR,

MFRUSR is called from the MCFR macro,

MFSTAS is called from the MCFR, MFRDI, and MFRZO macros,

MFTIME is called from the MCFR, MFRDI, and MFRZO macros.

```
C*********************************************************************
      SUBROUTINE MDCHCK(IN,NIN,OUT,MOUT)
C*********************************************************************
C
C     MONTE CARLO ANALYSIS ROUTINE
C
C     NIN = 1:
C        IN(1)   = 0 => CHECK ENVIRONMENT(WI/WE)
C            OUT(1) =  0, NO ERROR(WE ENVIRONMENT)
C            OUT(1) = -1, ERROR(NOT IN WE ENVIRONMENT)
C        IN(1)  ^= 0 => CHECK READY FLAG, OUTPUT TYPE CODE,
C                     # OF #M ELEMENTS AND ACTION CODE
C            OUT(1) = 0 WATAND ANALYSIS IS READY TO RUN
C                    -1 NOT READY
C            OUT(2) = OUTPUT TYPE CODE(MA/PH/RE/IM/DB)
C            OUT(3) = # OF #M ELEMENTS.
C            OUT(4) = ACTION CODE(NO/PR/PP/PT/PL/PO/PV/CALL)
C            IF MCFR CHECK IP POINT AVAILABILITY.
C            IF MCDI FR WITH ACTION EXCEPT NO AND DI IS READY TO RUN
C            THEN CHECK THE READING ERROR/OUTPUT COMPARABILITY.
C
C    NIN = 2; IN(1) ^= 12
C      IN(1) = WATAND ANALYSIS TYPE CODE. POINTS THE ANALYSIS WHICH
C              GLOBAL PARAMETER WILL BE COPIED INTO THE OTHER ANALYSIS
C      IN(2) = WATAND ANALYSIS TYPE CODE. POINTS THE ANALYSIS WHICH
C              WILL STORE PARAMETERS TRANSFER FROM THE FIRST ANALYSIS
C
C    NIN = 2; IN(1)=12
C      IN(2) =0 : STORE FILE NAME FOR DI US
C      IN(2)^=0 : RESTORE ACTION CODE AND DEFINE FMODE OF KE FOR DI.
C
C     CREATED 03-AUG-1987      J.F. SUEN  EE DEP.  YSU
C
C      UPDATE 12-FEB-1989        D. YON    EE DEP.  YSU
C            CHANGES SO THAT SUBROUTINE CAN BE USED FOR MCFR ANALYSIS.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*8 RDATA,
     *       IPNAME,
     *       IN(2),OUT(4)
C
      INTEGER IDATA(1),
     *        ANALYZ,
     *        WIWEFL,
     *        NPWL,NSRCES,SPPTR,SSPTR,
     *        NIN,MOUT,
     *        ATP1,ATP2,ANPNTR,DIPNTR,ACTION,USPTR
      INTEGER RPTR,IER,EXSAVD
C
      CHARACTER*1 EXBUFR
      CHARACTER*8 CIPNAM
C
```

```
      EQUIVALENCE (RDATA(1),IDATA(1)),
     *            (IPNAME,CIPNAM)
C
      COMMON /MEMORY/ RDATA(1)
     *       /APNTRS/ ANALYZ(30)
     *       /WIWEBL/ WIWEFL
     *       /PPNTRS/ NPWL,NSRCES,SPPTR,SSPTR
     *       /KUXTRA/ EXSAVD(2),EXBUFR(80)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IF (NIN.EQ.2) GOTO 200
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
100   IF (IN(1).NE.0) GOTO 150
C
C...CHECK ENVIRONMENT
      IF (MOUT.LT.1) GOTO 998
      MOUT=1
      OUT(1)=0D0
      IF (WIWEFL.EQ.2) GOTO 999
      OUT(1)=-1D0
      GOTO 999
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...CHECK ANALYSIS READY FLAG, OUTPUT TYPE CODE,
C...LINEAR CODE AND ACTION CODE
150   IF(MOUT.LT.4) GOTO 998
      MOUT=4
      ATP1=IN(1)
      IF (ATP1.LT.0) ATP1=-ATP1
      ANPNTR=ANALYZ(ATP1)
C
C...CHECK INITIAL POINT FOR FR OR TC.
      IF (ATP1.EQ.1.OR.ATP1.EQ.12) GOTO 170
      IPNAME=RDATA(IDATA(ANPNTR+34))
C
C...IGNORE IP CHECKING IF 'IP ZERO'.
      IF (CIPNAM.EQ.'ZERO') GOTO 170
      CALL IPFIND(IPNAME,IPVP,1)
      IF (IPVP.EQ.0) THEN
       CALL ERRMSG(66)
       OUT(1)=-1D0
       MOUT=1
       GOTO 999
      END IF
170   OUT(2)=IDATA(ANPNTR+37)
      OUT(3)=NPWL
      OUT(4)=IDATA(ANPNTR+2)
C
C...IF   (MODE=6 )    AND    (ACTION GT 2) AND   ( NOUT GE 2)
      IF ((IN(1).LT.0D0).AND.(OUT(4).GT.2D0).AND.(IDATA(ANPNTR+7).GE.2))
     * CALL ERRMSG(104131)
```

```
      OUT(1)=0D0
      IF(IDATA(ANPNTR+1).EQ.0) THEN
        OUT(1)=-1D0
      ELSE
        IF (ATP1.EQ.12.AND.OUT(4).NE.1D0) THEN
C
C...CHECK READING ERROR/OUTPUT COMPARABILITY
          RPTR=IDATA(ANPNTR)
          CALL KUINIT(ANPNTR+10,IDATA(ANPNTR+7),ANPNTR+31,RDATA(RPTR),
     *                EXSAVD,EXBUFR,IER)
          CALL KUTERM
          IF (IER.NE.0) OUT(1)=-2D0
        END IF
      END IF
      GOTO 999
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
200   ATP1=IN(1)
      ATP2=IN(2)
      ANPNTR=ANALYZ(ATP1)
C
C...MCFR OR MCDI?
      IF (ATP1.NE.12) GOTO 210
C
C...MCDI FR
      IF (ATP2.EQ.0) THEN
        IF (MOUT.LT.1) GOTO 998
        MOUT=1
        USPTR=0
        RPTR=IDATA(ANPNTR)
        CALL GSTORE(1,USPTR,8,0,IER)
        IF (USPTR.NE.0) RDATA(USPTR+1)=RDATA(RPTR)
        OUT(1)=USPTR
      ELSE
C
C...SET ACTION CODE AND DEFINE FMODE FOR KE
        IDATA(ANPNTR+2)=ATP2
        RDATA(IDATA(IDATA(ANPNTR+35))+1)=RDATA(IDATA(ANPNTR)+1)
        MOUT=0
      END IF
      GOTO 999
C
C...MCFR -- SET DI ANALYSIS PARAMETERS.
210   DIPNTR=ANALYZ(ATP2)
C
C...SET OU
      DO 220 K=2,30
220   IDATA(DIPNTR+K)=IDATA(ANPNTR+K)
C
C...DI US FNAME ON FMODE
      RDATA(IDATA(DIPNTR))=RDATA(IDATA(IDATA(ANPNTR+35)))
      RDATA(IDATA(DIPNTR)+1)=RDATA(IDATA(IDATA(ANPNTR+35))+1)
C
C...MA,PH,RE,IM,DB
      IDATA(DIPNTR+36)=IDATA(ANPNTR+37)
```

```
C
C...KE ON FMODE
      RDATA(IDATA(IDATA(DIPNTR+35))+1)=RDATA(IDATA(IDATA(ANPNTR+35))+1)
998   MOUT=0
999   RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE MDINPT(IN,NIN,RSUL,MOUT)
C*********************************************************************
C
C     MCDI POST-PROCESSOR ROUTINE
C     DISPLAYS INPUT INFORMATION FOR MCDC ANALYSIS
C
C     IN(1) = IDCODE = 1 INPUT DISPLAY ON
C                    = 0 INPUT DISPLAY OFF
C     IN(2) = PCODE  = 1 OUTPUT SPECIFIED
C                    = 0 NO OUTPUT SPECIFIED, (KEPT OU USED)
C     RSUL(1) = NUMBER OF SAMPLES IN MCDC ANALYSIS
C             = -1 ERROR READING FILE
C     RSUL(2) = OUCODE = 1 READING FILES FROM KE OU IN MCDC
C                      = 2 READING FILES FROM KE ALL IN MCDC
C     RSUL(3) = NS, NUMBER OF SAMPLES
C     RSUL(4) = TIME OF TIMER START
C
C     CREATED 03-AUG-1987     J.F. SUEN   EE DEP.  YSU
C
C      UPDATE 01-SEP-1987  P.MUNRO   ELEC ENG DEPT.
C          ADD CALL TO ERRMSG(136) FOR FILE NOT FOUND.
C      UPDATE 12-FEB-1989        D. YON    EE DEP.  YSU
C          GET ANALYSIS TITLE FROM MDKPINFO FILE INSTEAD OF FORTRAN
C          FORMAT. CHANGES SO THAT MCDC AND MCFR CAN ACCOMODATE.
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4  TIME
      REAL*8  RDATA,
     *        LVLID,PRIVID,
     *        WDJTID,
     *        FNAME,
     *        ANAME,
     *        BUFFER(11),IN(2),RSUL(4)
      CHARACTER*8 CRDATA(1),
     *           FILEID(3),INFO,VALS
C
      INTEGER IDATA(1),
     *        EDITID,
     *        WDJTFL,
     *        RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW,
     *        ANALYZ,
     *        NIN,MOUT,RECON,IDCODE,OUCODE,PCODE,
     *        LS,LT,NS,ITOTAL,APNTR
C
      EQUIVALENCE  (RDATA(1),IDATA(1),CRDATA(1))
C
      COMMON /MEMORY/ RDATA(1)
     *       /BLCK07/ LVLID,PRIVID,EDITID
     *       /WDJTBL/ WDJTID,WDJTFL
     *       /FNBLOK/ FNAME
     *       /BINPUT/ RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW
     *       /APNTRS/ ANALYZ(30)
C
```

```
      DATA INFO/'MDKPINFO'/,
     *     VALS/'MDKPVALS'/
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IF(MOUT.LT.4) GOTO 901
      IDCODE=IN(1)
      PCODE=IN(2)
      CALL GTIMER(TIME)
      RSUL(3)=0.0
      RSUL(4)=TIME
C
C...FIND FILE NAME AND FILE TYPE
      APNTR=ANALYZ(12)
      FILEID(1)=CRDATA(IDATA(APNTR))
      FILEID(3)=CRDATA(IDATA(APNTR)+1)
      FILEID(2)=INFO
C
C...READ TITLE INFORMATION
      RECON=0
      CALL IOOPEN(FILEID,1,11*8,RECON,IER)
      IF(IER-1) 10,902,903
10    CALL IOREAD(FILEID,RECON,BUFFER,11*8,IER)
      IF(IER.EQ.2) GOTO 903
      CALL IOCLOS(FILEID,1,IER)
      ITOTAL=BUFFER(1)
      NS=BUFFER(2)
      OUCODE=BUFFER(3)
      RSUL(3)=NS
C
C...RETURN IF DIDN'T SPECIFY OUTPUT AND IS A KE ALL TYPE FILE.
      IF(PCODE.EQ.0.AND.OUCODE.EQ.2) GOTO 990
C
C...RETURN IF INPUT OFF, A KE ALL TYPE FILE, AND MCDC ANALYSIS.
      IF(IDCODE.EQ.0.AND.OUCODE.EQ.2.AND.NIN.EQ.2) GOTO 990
      ANAME=BUFFER(4)
      WRITE(WRUNT,1400) ANAME,LVLID,EDITID,PRIVID,(BUFFER(J),J=9,11),
     *                  WDJTID,FNAME
      IF(IDCODE.EQ.0) GOTO 990
      WRITE(WRUNT,1300) NS,(BUFFER(J),J=5,8)
C
C...READ DISK INPUT INFORMATION
      FILEID(2)=VALS
      RECON=0
      CALL IOOPEN(FILEID,1,8*8,RECON,IER)
      IF(IER-1) 50,902,903
50    WRITE(WRUNT,1100)
      DO 70 I=1,ITOTAL
      CALL IOREAD(FILEID,RECON,BUFFER,8*8,IER)
      IF(IER.EQ.2) GOTO 903
70    WRITE(WRUNT,1200) (BUFFER(J),J=1,3),(BUFFER(J),J=4,8)
      CALL IOCLOS(FILEID,1,IER)
```

```
C
C...FINISH
990   RSUL(1)=NS
      RSUL(2)=OUCODE
      MOUT=4
      GOTO 999
C
C...ERRORS AND RETURN
901   MOUT=0
      GOTO 999
902   CALL ERRMSG(136)
903   RSUL(1)=-1
      MOUT=1
999   RETURN
C
1100  FORMAT(' INPUT                INITIAL    MINIMUM     MAXIMUM
     *    MEAN      ST.DEV'/)
1200  FORMAT(1X,3A8,1P5E11.3)
1300  FORMAT(/5X,'NUMBER OF SAMPLES = ',I4,9X,4A8/)
1400  FORMAT(/1X,A4,1X,A6,A2,1X,A8,1X,2A8,A4,A8,'  FILE: ',A8)
      END
```

```
C***********************************************************************
      SUBROUTINE MDINTL(IN,NIN,RSLT,MOUT)
C***********************************************************************
C
C     MCDC ANALYSIS ROUTINE
C     CHECKS THE 'WRTD' 'WRTM' PART AND DC ANALYSIS SPECIFIES
C
C     RSLT(1) = OUCODE = 1 FOR 'KE OU'
C                      = 2 FOR 'KE ALL'
C                      =-1 ERROR -- 'DC OU ALL'
C                      =-2 ERROR -- KE 'NONE', 'SPECS', OR 'OU ALL'
C     RSLT(2) = # OF WRTD SPECS
C     RSLT(3) = # OF WRTM SPECS
C     RSLT(4) = MCDC STARTING TIME
C
C     CREATED 03-AUG-1987       J.F. SUEN  EE DEP. YSU
C
C     UPDATE 06-APR-1988  P.MUNRO  YSU ELEC ENGIN DEPT.
C            CHANGE KEEP NONE CHECK TO .LE.0 (SEE KPINIT).
C     UPDATE 12-FEB-1989  D. YON  EE DEP.  YSU
C            CHANGE SO THAT SUBROUTINE CAN BE USED FOR MCFR ANALYSIS.
C            CHANGE 'WRTL' TO 'WRTD'.  SEE MDREAD AND MCDC WEXEC.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4 TIME
      REAL*8 RDATA,
     *       IN(1),RSLT(4)
      CHARACTER*8 VALNAM,
     *       CNDC,CNWRT(2)
C
      INTEGER IDATA(1),
     *        LENHED,LINK,FREEH,
     *        MACROH,GLBLPH,GLBLNH,MGLPHP,LCLPH,
     *        ANALYZ,
     *        ATP,APNTR,KEPTR,
     *        IWRT(2),OUCODE,
     *        IT,IV,NIN,MOUT,IER
C
      EQUIVALENCE (RDATA(1),IDATA(1))
C
      COMMON /MEMORY/ RDATA(1)
     *       /MCLIST/ VALNAM(750),LENHED(750),LINK(750),FREEH
     *       /MCHEDS/ MACROH,GLBLPH,GLBLNH,MGLPHP,LCLPH
     *       /APNTRS/ ANALYZ(30)
C
      DATA CNWRT/'WRTD','WRTM'/,CNDC/'DC'/
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...INITIALIZE TIME AND VALUES
      CALL GTIMER(TIME)
      OUCODE=1
      IWRT(1)=0
      IWRT(2)=0
      MOUT=4
      RSLT(4)=TIME
C
```

```
C...CHECK 'WRTD' AND 'WRTM' PART
      DO 60 I=1,2
      IT=LCLPH
10    IF(VALNAM(IT).EQ.CNWRT(I)) GOTO 20
      IT=LINK(IT)
      GOTO 10
20    IF(LENHED(IT).EQ.0) GOTO 50
      IV=LENHED(IT)
30    IF(LENHED(IV).EQ.0) GOTO 40
      IWRT(I)=IWRT(I)+1
40    IF(LINK(IV).EQ.0) GOTO 50
      IV=LINK(IV)
      GOTO 30
50    RSLT(I+1)=IWRT(I)
60    CONTINUE
C
C...GET POINTERS
      ATP=IN(1)
      APNTR=ANALYZ(ATP)
      KEPTR=IDATA(APNTR+35)
C
C...CHECK FOR DC 'OU ALL'
      IF(IDATA(APNTR+9).EQ.1) GOTO 997
C
C...CHECK FOR 'KE NONE'
      IF(KEPTR.LE.0) GOTO 998
C
C...CHECK FOR ANY SPECIFIED OUTPUTS
      IF(IDATA(KEPTR+3).NE.0) GOTO 998
C
C...CHECK FOR 'KE ALL OU'
      IF(IDATA(KEPTR+1).NE.0.AND.IDATA(KEPTR+2).NE.0) GOTO 998
C
C...CHECK FOR 'KE ALL'
      IF(IDATA(KEPTR+1).NE.0) OUCODE=2
      RSLT(1)=OUCODE
      GOTO 999
C
C...ERRORS AND EXIT
997   RSLT(1)=-1
      GOTO 999
998   RSLT(1)=-2
999   RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE MDREAD(IN,NIN,PAR,MOUT)
C***********************************************************************
C
C     MCDC ANALYSIS ROUTINE
C     ENCODES INPUT WRTD AND WRTM VARIABLES
C
C     PAR(1) = IWRTD = # OF LINEAR ELEMENTS
C            = -1    DEFINE STORAGE BLOCK FOR ENCODING ELEMENTS
C     PAR(2) = IWRTM = # OF NON-LINEAR ELEMENTS/PARAMETERS
C     PAR(3) = POINTER TO GSTORE MEMORY.
C     PAR(4) = SIZE OF GSTORE MEMORY BLOCK WHICH IS USED TO STORE
C              WRT ELEMENT/PARAMETER POINTERS.
C     IN(1) = OUCODE
C     IN(2) = # OF WRTD SPECS.
C     IN(3) = # OF WRTM SPECS.
C
C     CREATED  03-AUG-1987    J.F. SUEN   EE DEP. YSU
C
C     UPDATE 12-FEB-1989    , D. YON    EE DEP.  YSU
C          REMOVED THE LIMIT ON MAXIMUM NUMBER OF RANDOM ELEMENTS
C          FROM 20 TO AS MANY AS MEMORY ALLOWED. ALSO SEE MFRNEW.
C          CHANGE WRTL TO WRTD.  CHANGE TO USE WITH MCFR.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4 TDATA(1)
      REAL*8 RDATA,
     *       IN(3),PAR(4),
     *       PNAME,
     *       VCHANG,REALN
C
      INTEGER IDATA(1),
     *       LENHED,LINK,FREEH,
     *       MACROH,GLBLPH,GLBLNH,MGLPHP,LCLPH,
     *       NVLS,
     * IER=0  BTYPE,NTYPE,MTYPE,
     *       UIPTR,
     *       ENAME(13),
     *       LEN,NIN,MOUT,
     *       IT,NMBR,PTEST,INDEX,
     *       CODESZ,CODPTR,NEWPTR,
     *       NCHUSD,NSIGD,IER,TYPE,POSPTR,
     *       ETP,ELN,ETPP,ELNN,IWRTD,IWRTM
C
      CHARACTER ERSUB1(64)
      CHARACTER*4 IPAR(24),BLANK,LETV,DOT,VNAME,ERR1
      CHARACTER*5 ERR12
      CHARACTER*8 WRTD,WRTM,ERRSUB,VALNAM,NAME,END
      CHARACTER*13 ERR3
      CHARACTER*14 ERR2,ERR10
      CHARACTER*15 ERR9
      CHARACTER*17 ERR5
      CHARACTER*26 ERR6,ERR8
      CHARACTER*32 ERR11
      CHARACTER*35 ERR4
```

```
      CHARACTER*37 ERR13
      CHARACTER*42 ERR7
      CHARACTER*64 ERSUB
      CHARACTER*96 CIPAR
C
      EQUIVALENCE (RDATA(1),IDATA(1),TDATA(1)),
     *            (ERRSUB,ERSUB1,ERSUB),(IPAR,CIPAR)
C
      COMMON /MEMORY/ RDATA(1)
     *       /MCLIST/ VALNAM(750),LENHED(750),LINK(750),FREEH
     *       /MCHEDS/ MACROH,GLBLPH,GLBLNH,MGLPHP,LCLPH
     *       /BLCK06/ NVLS(70)
     *       /BLCK00/ BTYPE,NTYPE,MTYPE
     *       /UDELEM/ UIPTR(20)
     *       /ERRSUB/ ERRSUB(8)
C
      DATA BLANK/' '/,LETV/'V'/,DOT/'.'/,
     *     WRTD/'WRTD'/,WRTM/'WRTM'/,END/'END'/
      DATA ERR1  /'TYPE'/,
     *     ERR2  /'NOT RECOGNIZED'/,
     *     ERR3  /'ELEMENT/MODEL'/,
     *     ERR4  /'INVALID WRTD/WRTM SPECIFICATION FOR'/,
     *     ERR5  /'CANNOT BE ALTERED'/,
     *     ERR6  /'V-VALUE MISSING OR INVALID'/,
     *     ERR7  /'DUPLICATE ELEMENT/MODEL PARAMETER NAME FOR'/,
     *     ERR8  /'INPUT INFORMATION TOO LONG'/,
     *     ERR9  /'ERROR'/
     *     ERR10 /'PARAMETER NAME'/,
     *     ERR11 /'MISSING PARAMETER NAME FOR MODEL'/,
     *     ERR12 /'MODEL'/,
     *     ERR13 /'NO STORAGE FOR WRT ELEMENTS" POINTERS'/
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...BEGIN THE LINEAR PART
C...SET INITIAL CONDITION
      IER=0
      CODPTR=0
      CALL GSTORE(CODESZ,CODPTR,4,-4,IER)
      IF (IER.NE.0) GOTO 920
      PAR(3)=CODPTR
      PAR(4)=CODESZ
      PAR(1)=0D0
      PAR(2)=0D0
      IREF=0
      IWRTD=0
      IWRTM=0
      IETP=0
      IMDEL=0
      INDEX=1
      NCOUNT=1
      ICOUNT=0
      IPARM=0
      IFLAG=0
      IER1=0
      IT=LCLPH
```

```
      IF(IN(2).EQ.0D0) GOTO 310
C
C...FIND INPUT PARAMETER 'WRTD'
20    IF(VALNAM(IT).EQ.WRTD) GOTO 30
      IT=LINK(IT)
      GOTO 20
30    IV=LENHED(IT)
      GOTO 60
50    IV=LINK(IV)
60    ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.IN(2)) GOTO 220
      LEN=LENHED(IV)
      IF(LEN.GT.22) GOTO 906
      CIPAR=' '
      DO 70 I=1,17,8
      CALL PACKCH(VALNAM(IV),IPAR(I),8,1,4)
      IF(LEN.LE.(I+7)) GOTO 80
70    IV=LINK(IV)
C
C...CHECK FOR LITERAL OR NUMBER ON INPUT LINE
80    CALL RDCODE(IPAR(1),LEN,NCHUSD,REALN,NSIGD,IER)
      IF(IER.EQ.1) GOTO 150
      IF(IETP.EQ.0) GOTO 901
C
C...INPUT IS A NUMBER
100   IF(NCOUNT.EQ.2) GOTO 110
      TDATA(CODPTR+IREF+3)=REALN
      NCOUNT=2
      GOTO 50
110   TDATA(CODPTR+IREF+4)=REALN
      IETP=0
      NCOUNT=1
      IFLAG=1
      GOTO 50
C
C...INPUT IS A LITERAL
C...CHECK ELEMENT TYPE ETP
150   IF(IPAR(1).EQ.DOT) GOTO 901
      CALL MDPACK(IPAR,ETP,ELN,INDEX,PNAME,POSPTR,1,1,IER)
      IF(ETP.EQ.0) GOTO 901
C
C...CHECK ELEMENT LINE ELN
160   CALL MDPACK(IPAR,ETP,ELN,INDEX,PNAME,POSPTR,1,2,IER)
      IF(ELN.EQ.0) GOTO 902
      IF(ETP.EQ.7.OR.ETP.EQ.8) GOTO 170
      IF(ETP.GT.50) GOTO 170
      NMBR=NVLS(ETP)
      IF(NMBR.EQ.0) GOTO 903
C
C...CHECK THE INDEX VALUE
170   CALL MDPACK(IPAR,ETP,ELN,INDEX,PNAME,POSPTR,1,3,IER)
      IF(IER.EQ.1) GOTO 904
      IF(ETP.EQ.7.OR.ETP.EQ.8) GOTO 190
      IF(ETP.GT.50) GOTO 180
```

```
      IF(INDEX.GT.NMBR) GOTO 904
      GOTO 210
180   L=UIPTR(ETP-BTYPE)
      NMBR=IDATA(L+19)+IDATA(L+20)
      IF(NMBR.EQ.0) GOTO 903
      IF(INDEX.GT.NMBR) GOTO 904
      GOTO 210
C
C...FOR VOLTAGE OR CURRENT SOURCE
190   CALL MDSRCE(VCHANG,ELN,1,INDEX,3,IER)
      IF(IER-1) 210,903,904
C
C...BEGIN SET DATA
210   IF(IETP.EQ.1.OR.IFLAG.EQ.1) IREF=IREF+5
      IF ((IREF+5).GT.CODESZ) GOTO 920
      NEWPTR=CODPTR+IREF
      IDATA(NEWPTR+1)=ELN
      IDATA(NEWPTR+2)=ETP
      TDATA(NEWPTR+3)=0.0
      TDATA(NEWPTR+4)=0.0
      IDATA(NEWPTR+5)=INDEX
      IWRTD=IWRTD+1
      INDEX=1
      IETP=1
      NCOUNT=1
      GOTO 50
C
C...PREPARE OUTPUT
220   MOUT=4
      PAR(1)=IWRTD
C
C...CHECK DUPLICATION ERRORS
      IF(IWRTD.EQ.1) GOTO 300
      L=5*IWRTD-9
      N=L+5
      DO 250 I=1,L,5
      K=I+5
      DO 250 J=K,N,5
      IF(IDATA(CODPTR+I+1).EQ.7D0.OR.IDATA(CODPTR+I+1).EQ.8D0) GOTO 230
      IF((IDATA(CODPTR+I+1).EQ.35D0).OR.(IDATA(CODPTR+I+1).GT.50D0))
     * GOTO 230
      IF(IDATA(CODPTR+I).EQ.IDATA(CODPTR+J)) GOTO 905
      GOTO 250
230   IF((IDATA(CODPTR+I).EQ.IDATA(CODPTR+J)).AND.
     * (IDATA(CODPTR+I+4).EQ.IDATA(CODPTR+J+4))) GOTO 905
250   CONTINUE
C
C...BEGIN THE NON-LINEAR PART
300   IREF=IREF+5
      IT=LCLPH
      ICOUNT=0
      NCOUNT=1
      INDEX=1
310   IF(IN(3).EQ.0D0) GOTO 999
```

```
      IFLAG=0
C
C...CHECK FOR 'WRTM' PARAMETER
320   IF(VALNAM(IT).EQ.WRTM) GOTO 330
      IT=LINK(IT)
      GOTO 320
330   IV=LENHED(IT)
      GOTO 360
350   IV=LINK(IV)
360   ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.IN(3)) GOTO 500
      LEN=LENHED(IV)
      IF(LEN.GT.22) GOTO 906
      CIPAR=' '
      DO 370 I=1,17,8
      CALL PACKCH(VALNAM(IV),IPAR(I),8,1,4)
      IF(LEN.LE.(I+7)) GOTO 380
370   IV=LINK(IV)
C
C...CHECK FOR 'END' PARAMETER
380   IF(VALNAM(IV).NE.END) GOTO 390
      IF(IMDEL.NE.1) GOTO 918
      IF(IPARM.NE.1.AND.IFLAG.EQ.0) GOTO 911
      IF(IPARM.EQ.1) IREF=IREF+5
      IFLAG=0
      IMDEL=0
      IPARM=0
      NCOUNT=1
      GOTO 350
C
C...CHECK FOR NUMBER OR LITERAL ON INPUT LINE
390   CALL RDCODE(IPAR,LEN,NCHUSD,REALN,NSIGD,PTEST)
      IF(PTEST.NE.0) GOTO 420
C
C...INPUT IS A NUMBER
      IF(IMDEL.NE.1) GOTO 918
      IF(IPARM.NE.1) GOTO 911
      IF(NCOUNT.EQ.2) GOTO 410
      TDATA(CODPTR+IREF+3)=REALN
      NCOUNT=2
      GOTO 350
410   TDATA(CODPTR+IREF+4)=REALN
      NCOUNT=1
      IPARM=0
      IFLAG=1
      IREF=IREF+5
      GOTO 350
C
C...INPUT IS A LITERAL
C...CHECK MODEL TYPE ETP
420   ETPP=0
      CALL MDPACK(IPAR,ETPP,ELN,INDEX,PNAME,POSPTR,1,1,IER)
      IF(ETPP.EQ.0) GOTO 450
      IF(ETPP.EQ.33.OR.ETPP.EQ.34.AND.IMDEL.EQ.1) GOTO 450
```

```
      ETP=ETPP
C
C...CHECK THE ELN
      CALL MDPACK(IPAR,ETP,ELN,INDEX,PNAME,POSPTR,2,2,IER)
      IF(ELN.EQ.0) GOTO 902
      IF(IMDEL.EQ.1) GOTO 430
      IMDEL=1
      NCOUNT=1
      GOTO 350
430   IF(IPARM.NE.1) GOTO 911
      IREF=IREF+5
      IPARM=0
      NCOUNT=1
      GOTO 350
C
C...CHECK PARAMETER
450   IF(IMDEL.EQ.0) GOTO 918
      POSPTR=0
      CALL MDPACK(IPAR,ETP,ELNN,INDEX,PNAME,POSPTR,3,2,IER)
      IF(IER.EQ.1) GOTO 914
      IF(IPAR(POSPTR).EQ.BLANK) GOTO 460
      CALL MDPACK(IPAR,ETP,ELNN,INN,PNAME,POSPTR,3,3,IER)
      IF(IER.EQ.1) IER1=1
      GOTO 470
460   INN=1
C
C...CHECK PARAMETER NAME AND INDEX VALUE
470   CALL MDMODL(VCHANG,PNAME,ELN,ETP,INDEX,INN,3,IER)
      IF(IER-1) 480,915,914
480   IF(IER1.EQ.1) GOTO 915
      IF(IMDEL.NE.1) GOTO 918
      IF(IPARM.EQ.1) IREF=IREF+5
      IF ((IREF+5).GT.CODESZ) GOTO 920
      NEWPTR=CODPTR+IREF
      IDATA(NEWPTR+1)=ELN
      IDATA(NEWPTR+2)=ETP
      TDATA(NEWPTR+3)=0.0
      TDATA(NEWPTR+4)=0.0
      IDATA(NEWPTR+5)=INDEX
      IWRTM=IWRTM+1
      IPARM=1
      INDEX=1
      NCOUNT=1
      GOTO 350
C
C...PREPARE THE OUTPUT
500   IF(IPARM.EQ.0.AND.PTEST.EQ.1) GOTO 917
      PAR(2)=IWRTM
      MOUT=4
      IF(IWRTM.EQ.1) GOTO 999
      L=5*(IWRTD+IWRTM)-9
      N=L+5
      M=IWRTD*5+1
```

```
C
C...CHECK DUPLICATION ERRORS
      DO 510 I=M,L,5
      K=I+5
      DO 510 J=K,N,5
      IF ((IDATA(CODPTR+I).EQ.IDATA(CODPTR+J)).AND.
     * (IDATA(CODPTR+I+4).EQ.IDATA(CODPTR+J+4))) GOTO 916
510   CONTINUE
      GOTO 999
C
C...ERRORS
901   ERSUB=ERR1
      CALL PACKCH(IPAR(1),ERSUB1(6),LEN,4,1)
      ERSUB((LEN+7):)=ERR2
      GOTO 997
902   ERSUB=ERR4
      ERSUB(37:50)=ERR3
      CALL PACKCH(IPAR(1),ERSUB1(51),LEN,4,1)
      GOTO 997
903   ERSUB=ERR3
      CALL PACKCH(IPAR(1),ERSUB1(9),LEN,4,1)
      ERSUB((LEN+10):)=ERR5
      GOTO 997
904   CALL PACKCH(IPAR(1),ERSUB1,LEN,4,1)
      ERSUB((LEN+2):)=ERR6
      GOTO 997
905   ELN=IDATA(CODPTR+I)
      CALL PACKNM(ELN,ENAME,LEN,2)
925   ERSUB=ERR7
      CALL PACKCH(ENAME,ERSUB1(44),LEN,4,1)
      GOTO 997
906   ERSUB=ERR8
      GOTO 997
911   ERSUB=ERR9
      CALL PACKCH(IPAR(1),ERSUB1(17),LEN,4,1)
      ERSUB((LEN+18):)=ERR2
      GOTO 997
914   ERSUB=ERR10
      CALL PACKCH(IPAR(1),ERSUB1(16),LEN,4,1)
      ERSUB((LEN+17):)=ERR2
      GOTO 997
915   CALL PACKCH(IPAR(1),ERSUB1,LEN,4,1)
      ERSUB((LEN+2):)=ERR6
      GOTO 997
916   MNP=IDATA(CODPTR+I)-3
      CALL PACKNM(MNP,ENAME,LEN,1)
      GOTO 925
917   ERSUB=ERR11
      CALL PACKCH(IPAR(1),ERSUB1(34),LEN,4,1)
      GOTO 997
918   ERSUB=ERR12
      CALL PACKCH(IPAR(1),ERSUB1(7),LEN,4,1)
      ERSUB((LEN+8):)=ERR2
      GOTO 997
```

```
920    ERSUB=ERR13
997    CALL ERRMSG(108159)
998    PAR(1)=-1
996    MOUT=1
999    IF (PAR(1).LT.0.AND.CODPTR.NE.0) CALL FSTORE(CODESZ,CODPTR,4)
       RETURN
       END
C
C
C***********************************************************************
       SUBROUTINE MDPACK(IPAR,ETP,ELN,INDEX,NAME,POSPTR,MFLAG,TYPE,IER)
C***********************************************************************
C
C      MCDC ANALYSIS ROUTINE CALLED BY MDREAD
C      FINDS ELEMENT TYPE, POSITION LINE, AND QUALIFIER VALUE
C
C      TYPE = 1  FIND ELEMENT TYPE ETP
C           = 2  FIND ELEMENT LINE ELN OR PARAMETER NAME
C           = 3  FIND QUALIFIER VALUE (INDEX)
C      IER = 0 NO ERROR
C          = 1 OR 2 ERROR
C
C      CREATED  03-AUG-1987  J.F. SUEN  EE DEP.  YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       CHARACTER*8 BLANK8,NAME
C
       INTEGER TYPE,
     *         ETP,ELN,POSPTR
       CHARACTER*4 IPAR(24),DIGIT(10),LETV,DOT,BLANK,VNAME
C
       DATA BLANK8/' '/,LETV/'V'/,DOT/'.'/,
     * DIGIT /'0','1','2','3','4','5','6','7','8','9'/
C
       EQUIVALENCE (BLANK8,BLANK)
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       IF(TYPE-2) 10,50,150
C
C...FIND ELEMENT/MODEL TYPE
10     N=0
       I=0
20     I=I+1
       IF(N.GT.4) GOTO 997
       IF(IPAR(I).EQ.DOT) GOTO 30
       N=N+1
       GOTO 20
30     VNAME=BLANK
       CALL PACKCH(IPAR(1),VNAME,N,4,1)
       CALL CHETYP(VNAME,ETP)
       POSPTR=I
       GOTO 998
C
C...FIND ELEMENT/MODEL POSITION LINE
50     I=POSPTR
```

```
      N=0
60    I=I+1
      IF(N.GT.8) GOTO 997
      IF(IPAR(I).EQ.DOT.OR.IPAR(I).EQ.BLANK) GOTO 70
      N=N+1
      GOTO 60
70    NAME=BLANK8
      IF(MFLAG-2) 80,90,100
80    CALL PACKCH(IPAR(I-N),NAME,N,4,1)
      CALL FINDE(ETP,NAME,ELN)
      GOTO 110
90    CALL PACKCH(IPAR(I-N),NAME,N,4,1)
      CALL FINDM(ETP,NAME,ELN)
      GOTO 110
100   CALL PACKCH(IPAR(1),NAME,N,4,1)
      CALL FINDM(ETP,NAME,ELN)
110   POSPTR=I
      GOTO 998
C
C...FIND QUALIFIER VALUE
150   IF(IPAR(POSPTR).EQ.BLANK) GOTO 998
      I=POSPTR+1
      IF(IPAR(I).NE.LETV) GOTO 997
      INDEX=0
      N=0
160   I=I+1
      N=N+1
      IF(N.GT.6) GOTO 997
      IF(IPAR(I).EQ.BLANK.AND.N.NE.1) GOTO 998
      DO 170 J=1,10
      IF(IPAR(I).EQ.DIGIT(J)) GOTO 180
170   CONTINUE
      GOTO 997
180   INDEX=10*INDEX+J-1
      GOTO 160
997   IER=1
      GOTO 999
998   IER=0
999   RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE  MFDADS(PAR,FILEID,ATP)
C***********************************************************************
C
C     MONTE CARLO ANALYSIS SUBROUTINE
C     CALCULATE AND DISPLAY THE ELEMENT/PARAMETERS STATISTICAL DATA
C
C     PAR   = THE DATA TRANSMITTED FROM MACRO ROUTINE.
C           PAR(1) = OUCODE
C           PAR(2) = IPDC ON/OFF CODE
C           PAR(3) = ACTION CODE
C           PAR(5) = NS
C           PAR(6) = DSEED, SEED OF RANDOM NUMBER GENERATOR.
C           PAR(9) = GSTORE MEMORY POINTER
C           PAR(10)= GSTORE MEMORY SIZE
C     INPFL ^= 1  TO STOP DISPLAY OF INPUT INFORMATION.
C     FILEID = THE FILE NAME,FILE TYPE AND DISK NAME IN WHICH
C              THE MC ANALYSIS PARAMETERS WERE STORED.
C     FILEIP = THE FILE NAME,FILE TYPE AND DISK NAME IN WHICH
C              THE STATISTIC RESULTS WERE STORED.
C     ATP    = WATAND ANALYSIS CODE NUMBER.
C     RDATA(NEWPTR+1) = ELEMENT/PARAMETERS INITIAL VALUE
C     RDATA(  "   +2) = SMALLEST VALUE.
C     RDATA(  "   +3) = LARGEST VALUE.
C     RDATA(  "   +4) = AVERAGE
C     RDATA(  "   +5) = ST. DEVIATION
C
C     CREATED 12-FEB-1989        D. YON     EE DEP.  YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*8 RDATA,
     *       LVLID,PRIVID,
     *       DATE,
     *       WDJTID,
     *       PNAME,RANVAL,
     *       VCHANG,BOB,FNAME,
     *       PAR(10),
     *       TITLE(4),
     *       BUFFER(11),
     *       DABS
C
      INTEGER IDATA(1),
     *       NELEMS,NNODES,SIPTR,SRPTR,ETPTR,
     *       RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW,
     *       EDITID,
     *       WDJTFL,
     *       ANALYZ,ATP,
     *       ITOTAL,NS,OUCODE,IWRTD,IWRTM,STATSZ,STAPTR,
     *       CODPTR,NEWPTR,
     *       INPFL,IREP,APNTR,KEPTR,
     *       IER,RPTR,NMBR,ERR,
     *       INN,MNP,ELN,ETP,INDEX,INUMB
C
      CHARACTER NAME1(24)
```

```
      CHARACTER*4 DIGIT(10),BLANK,LETV,DOT,NAME(24),IPARM(8)
      CHARACTER*8 CRDATA(1),
     *            CBUFER(11),
     *            ERRSUB,
     *            MCDC,MCFR,MCTC,INFO,VALS,
     *            FILEID(3),ANAME,RNAME(3)
      CHARACTER*64 ERSUB
      CHARACTER*96 CNAME
C
      EQUIVALENCE (RDATA(1),IDATA(1),CRDATA(1)),
     *            (RNAME,NAME1),
     *            (NAME,CNAME),
     *            (ERRSUB,ERSUB),
     *            (BUFFER(1),CBUFER(1))
C
      COMMON /MEMORY/ RDATA(1)
     *       /CPNTRS/ NELEMS,NNODES,SIPTR,SRPTR,ETPTR
     *       /BINPUT/ RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW
     *            /BLCK07/ LVLID,PRIVID,EDITID
     *       /DATEBL/ DATE(3)
     *       /WDJTBL/ WDJTID,WDJTFL
     *       /FNBLOK/ FNAME
     *            /ERRSUB/ ERRSUB(8)
     *       /APNTRS/ ANALYZ(30)
     *       /MCRDA/ TITLE
     *       /MCIDA/ ITOTAL,NS,OUCODE,IWRTD,IWRTM,STATSZ,STAPTR
C
      DATA DIGIT /'0','1','2','3','4','5','6','7','8','9'/,
     *     BLANK/' '/, LETV/'V'/, DOT/'.'/,
     *       INFO/'MDKPINFO'/,
     *     VALS/'MDKPVALS'/,
     *     MCFR/'MCFR'/,MCDC/'MCDC'/,MCTC/'MCTC'/
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      CODPTR=PAR(9)
      INPFL=1
C
C...NO OUTPUT DISPLAYED IF ACTION CODE = 1
      IF (PAR(3).EQ.1) INPFL=0
C
C...FIND FILE NAME AND FILE TYPE
300   APNTR=ANALYZ(ATP)
      KEPTR=IDATA(APNTR+35)
      FILEID(1)=CRDATA(IDATA(KEPTR))
      FILEID(3)=CRDATA(IDATA(KEPTR)+1)
      FILEID(2)=INFO
      IF (ATP.EQ.1) ANAME=MCDC
      IF (ATP.EQ.3) ANAME=MCFR
      IF (ATP.EQ.4) ANAME=MCTC
C
C...PREPARE OUTPUTS
370   CALL IOOPEN(FILEID,0,11*8,1,IER)
      IF(IER.NE.0) GOTO 997
      BUFFER(1)=ITOTAL
```

```
      BUFFER(2)=NS
      BUFFER(3)=OUCODE
      CBUFER(4)=ANAME
      BUFFER(5)=TITLE(1)
      BUFFER(6)=TITLE(2)
      BUFFER(7)=TITLE(3)
      BUFFER(8)=TITLE(4)
      BUFFER(9)=DATE(1)
      BUFFER(10)=DATE(2)
      BUFFER(11)=DATE(3)
      CALL IOWRIT(FILEID,BUFFER,11*8,IER)
      IF(IER.NE.0) GOTO 997
      CALL IOCLOS(FILEID,0,IER)
      IF(IER.NE.0) GOTO 997
      WRITE(WRUNT,1300) ANAME,LVLID,EDITID,PRIVID,DATE,WDJTID,FNAME
      IF(INPFL.NE.1) GOTO 380
      WRITE(WRUNT,1400) NS,TITLE
      WRITE(WRUNT,1200)
C
380   FILEID(2)=VALS
      CALL IOOPEN(FILEID,0,8*8,1,IER)
      IF(IER.NE.0) GOTO 997
C
C...CALCULATE STATISTICAL RESULTS AND WRITE ON DISK
      DO 750 I=1,ITOTAL
      NEWPTR=STAPTR+5*(I-1)
      ELN=IDATA(CODPTR+1)
      ETP=IDATA(CODPTR+2)
      INDEX=IDATA(CODPTR+5)
C
C...CALCULATE STATISTICAL DATA
      RDATA(NEWPTR+4)=RDATA(NEWPTR+4)/NS
      RDATA(NEWPTR+5)=RDATA(NEWPTR+5)/NS-RDATA(NEWPTR+4)**2
      RDATA(NEWPTR+5)=DABS(RDATA(NEWPTR+5))**0.5
C
C...PUT INITIAL VALUE BACK TO ELEMENT
      IF(I.GT.IWRTD) GOTO 395
      IF(ETP.EQ.7.OR.ETP.EQ.8) GOTO 390
      RPTR=IDATA(SRPTR+ELN)-1
      RDATA(RPTR+INDEX)=RDATA(NEWPTR+1)
      GOTO 500
C
C...FOR VOLTAGE AND CURRENT SOURCE
390   VCHANG=RDATA(NEWPTR+1)
      CALL MDSRCE(VCHANG,ELN,1,INDEX,2,IER)
      GOTO 500
C
C...FOR NON-LINEAR ELEMENT
395   VCHANG=RDATA(NEWPTR+1)
      CALL MDMODL(VCHANG,PNAME,ELN,ETP,INDEX,INN,2,IER)
C
C...DISPLAY THE OUTPUT AND STORE STATISTICAL INPUT VALUES IN DISK
500   CNAME=' '
      IF(I.GT.IWRTD) GOTO 550
```

```
      CALL PACKNM(ELN,NAME,II,2)
      INN=INDEX
      IF(INDEX.EQ.1) GOTO 700
      NAME(II+1)=DOT
      NAME(II+2)=LETV
      L=II+2
      GOTO 600
C
C...FIND THE QUALIFIER VALUES
550   MNP=ELN-3
      CALL PACKNM(MNP,NAME,II,1)
      CALL MDMODL(RANVAL,PNAME,ELN,ETP,INDEX,INN,4,IER)
      CALL PACKCH(PNAME,IPARM(1),8,1,4)
      DO 560 II=1,6
      IF(IPARM(II).EQ.BLANK) GOTO 570
560   NAME(II+14)=IPARM(II)
570   IF(INN.EQ.1) GOTO 700
      NAME(II+14)=DOT
      NAME(II+15)=LETV
      L=II+15
C
C...DECODE THE INDEX NUMBER
600   DO 610 K=1,6
      INUMB=INN/10**(6-K)
      IF(INUMB.NE.0) GOTO 620
610   CONTINUE
620   K=7-K
      BOB=INN
      DO 650 J=1,K
      BOB=BOB/10**(K-J)
      INUMB=BOB
      BOB=(BOB-INUMB)*10**(K-1)
      L=L+1
      DO 630 N=0,9
      IF(INUMB.EQ.N) GOTO 650
630   CONTINUE
650   NAME(L)=DIGIT(N+1)
700   CALL PACKCH(NAME,NAME1(1),24,4,1)
710   IF(INPFL.NE.1) GOTO 720
C
C...WRITE OUTPUT
      WRITE(WRUNT,1100) RNAME,(RDATA(NEWPTR+M),M=1,5)
C
C...WRITE DATA IN DISK
720   CBUFER(1)=RNAME(1)
      CBUFER(2)=RNAME(2)
      CBUFER(3)=RNAME(3)
      BUFFER(4)=RDATA(NEWPTR+1)
      BUFFER(5)=RDATA(NEWPTR+2)
      BUFFER(6)=RDATA(NEWPTR+3)
      BUFFER(7)=RDATA(NEWPTR+4)
      BUFFER(8)=RDATA(NEWPTR+5)
      CALL IOWRIT(FILEID,BUFFER,8*8,IER)
      IF(IER.NE.0) GOTO 997
```

```
750   CODPTR=CODPTR+5
C
C...CLOSE THE DISK
      CALL IOCLOS(FILEID,0,IER)
      IF(IER.EQ.0) GOTO 999
C
C...ERROR
997   CALL ERRMSG(156)
C
999   RETURN
1100  FORMAT(1X,3A8,1P5E11.3)
1200  FORMAT(' INPUT                         INITIAL     MINIMUM     MAXIMUM
     *    MEAN        ST.DEV'/)
1300  FORMAT(/1X,A4,1X,A6,A2,2X,A8,1X,2A8,A4,A8,'    FILE: ',A8)
1400  FORMAT(/5X,'NUMBER OF SAMPLES = ',I4,9X,4A8/)
      END
```

```
C***********************************************************************
      SUBROUTINE MFRNEW(IREP,PAR)
C***********************************************************************
C
C     MONTE CARLO ANALYSIS SUBROUTINE
C     RUNS BEFORE FR/DC ANALYSIS TO CHANGE ELEMENT/PARAMETER VALUES
C
C     IREP = THE WATAND ANALYSIS EXECUTION COUNTER.
C     PAR  = THE DATA TRANSMITTED FROM MACRO ROUTINE.
C            PAR(1) = OUCODE
C            PAR(2) = IPDC ON/OFF CODE
C            PAR(3) = ACTION CODE
C            PAR(5) = NS
C            PAR(6) = DSEED, SEED OF RANDOM NUMBER GENERATOR.
C            PAR(9) = GSTORE MEMORY POINTER
C            PAR(10)= GSTORE MEMORY SIZE
C
C     RDATA(NEWPTR+1) = INITIAL VALUE
C     RDATA(  "   +2) = SMALLEST VALUE.
C     RDATA(  "   +3) = LARGEST VALUE.
C     RDATA(  "   +4) = ACCUMULATED SUM OF EVERY RANDOM VALUE.
C     RDATA(  "   +5) = ACCUMULATED SUM OF EVERY SQUARE OF RANDOM VALUE.
C
C     CREATED 12-FEB-1989       D. YON   EE DEP.  YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4 TIME,TDATA(1)
      REAL*8 RDATA,
     *    DSEED,PNAME,RANVAL,PAR1,PAR2,
     *    INIVL,VCHANG,SEED,
     *       PAR(9)
C
      CHARACTER*8 ERRSUB
      CHARACTER*31 RANERR
      CHARACTER*32 TITLE
      CHARACTER*48 STOERR
      CHARACTER*64 ERSUB
C
      INTEGER IDATA(1)
     *      ,ITOTAL,NS,OUCODE,IWRTL,IWRTM
     *      ,NELEMS,NNODES,SIPTR,SRPTR,ETPTR
     *      ,ATTCD
     *      ,CODPTR,STATSZ,STAPTR,NEWPTR
     *      ,IREP,ISEED
     *      ,IER,RPTR
     *      ,INN,ELN,ETP,INDEX
C
      EQUIVALENCE (RDATA(1),IDATA(1),TDATA(1)),
     *            (ERRSUB,ERSUB)
C
      COMMON /MEMORY/ RDATA(1)
     *      /CPNTRS/ NELEMS,NNODES,SIPTR,SRPTR,ETPTR
     *      /ATTIBL/ ATTCD
     *      /ERRSUB/ ERRSUB(8)
```

```
     *     /MCRDA/ TITLE
     *     /MCIDA/ ITOTAL,NS,OUCODE,IWRTL,IWRTM,STATSZ,STAPTR
C
      DATA STOERR/'NO STORAGE FOR WRT ELEMENTS" STATISTICAL RESULTS'/,
     * RANERR/'UNABLE TO GENERATE RANDOM VALUE'/
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IF(IREP.GT.1) GOTO 50
C
C...READ INPUT DATA
      OUCODE=PAR(1)
      NS=PAR(5)
      DSEED=PAR(6)
      IWRTL=PAR(7)
      IWRTM=PAR(8)
      CODPTR=PAR(9)
      ITOTAL=IWRTM+IWRTL
      STATSZ=5*ITOTAL
      STAPTR=0
      CALL GSTORE(STATSZ,STAPTR,8,-1,IER)
      IF (IER.NE.0) GOTO 910
      IF(DSEED.NE.0D0) GOTO 50
C
C...PICK DSEED NUMBER FROM CPU TIME
      CALL GTIMER(TIME)
      SEED=10.0*TIME
      ISEED=SEED
      SEED=SEED-ISEED
      SEED=SEED*10D8
      ISEED=SEED
      DSEED=ISEED
C
C...BEGIN DECODING THE INPUT DATA
50    CODPTR=PAR(9)
      DO 250 I=1,ITOTAL
      NEWPTR=STAPTR+5*(I-1)
      ELN=IDATA(CODPTR+1)
      ETP=IDATA(CODPTR+2)
      PAR1=TDATA(CODPTR+3)
      PAR2=TDATA(CODPTR+4)
      INDEX=IDATA(CODPTR+5)
      IF(IREP.GT.1) GOTO 160
C
C...FIRST COMING IN SET INITIAL VALUE AND FIND MEAN VALUE
C
C...LINEAR AND NON-LINEAR ELEMENTS
80    IF(I.GT.IWRTL) GOTO 120
      IF(ETP.EQ.7.OR.ETP.EQ.8) GOTO 110
      RPTR=IDATA(SRPTR+ELN)-1
      RDATA(NEWPTR+1)=RDATA(RPTR+INDEX)
      GOTO 150
C
C...VOLTAGE OR CURRENT SOURCE
110   CALL MDSRCE(VCHANG,ELN,1,INDEX,1,IER)
      RDATA(NEWPTR+1)=VCHANG
```

```
      GOTO 150
C
C...CHANGE MODEL VALUES
120   CALL MDMODL(VCHANG,PNAME,ELN,ETP,INDEX,INN,1,IER)
      RDATA(NEWPTR+1)=VCHANG
C
C...SET INITIAL CONDITION
150   RDATA(NEWPTR+4)=0
      RDATA(NEWPTR+5)=0
C
C...CALL RANDOM NUMBER AND SORTING
160   INIVL=RDATA(NEWPTR+1)
      CALL MDRAND(RANVAL,TITLE,INPFL,IREP,ICOUNT,DSEED,INIVL,
     *    PAR1,PAR2,IER)
      IF (IER.NE.0) GOTO 997
      RDATA(NEWPTR+4)=RDATA(NEWPTR+4)+RANVAL
      RDATA(NEWPTR+5)=RDATA(NEWPTR+5)+RANVAL**2
      IF(IREP.GT.1) GOTO 170
      RDATA(NEWPTR+2)=RANVAL
      RDATA(NEWPTR+3)=RANVAL
      GOTO 180
170   IF(RANVAL.LT.RDATA(NEWPTR+2)) RDATA(NEWPTR+2)=RANVAL
      IF(RANVAL.GT.RDATA(NEWPTR+3)) RDATA(NEWPTR+3)=RANVAL
C
C...CHANGE VALUES (LINEAR AND MODEL)
180   IF(I.GT.IWRTL) GOTO 220
      IF(ETP.EQ.7.OR.ETP.EQ.8) GOTO 210
      RPTR=IDATA(SRPTR+ELN)-1
      RDATA(RPTR+INDEX)=RANVAL
      GOTO 250
C
C...CHANGE VOLTAGE OR CURRENT SOURCE
210   CALL MDSRCE(RANVAL,ELN,1,INDEX,2,IER)
      GOTO 250
C
C...CHANGE MODEL PARAMETER VALUE
220   CALL MDMODL(RANVAL,PNAME,ELN,ETP,INDEX,INN,2,IER)
250   CODPTR=CODPTR+5
      GOTO 999
910   ERSUB=STOERR
      GOTO 998
997   ERSUB=RANERR
998   CALL ERRMSG(108159)
      ATTCD=1
999   RETURN
      END
```

```
C***************************************************************************
      SUBROUTINE MFRUSR(IREP,PAR)
C***************************************************************************
C
C     MONTE CARLO ANALYSIS SUBROUTINE.
C     MCFR MAIN ROUTINE FOR #RU CONTROL STATEMENT.
C
C     IREP = THE FR ANALYSIS EXECUTION COUNTER.
C     PAR  = THE DATA TRANSMITTED FROM MCFR WEXEC MACRO.
C            PAR(1) = OUCODE
C            PAR(2) = IPDC ON/OFF CODE
C            PAR(3) = ACTION CODE
C            PAR(5) = NS
C            PAR(6) = DSEED, SEED OF RANDOM NUMBER GENERATOR.
C            PAR(9) = GSTORE MEMORY POINTER
C            PAR(10)= GSTORE MEMORY SIZE
C
C     CREATED 12-FEB-1989        D. YON   EE DEP.  YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*8 PAR(10),FILEID(3)
      INTEGER IREP,
     *        KPFNID,KPFNKP,KPBPTR,KPNPTR,KPMBR,KPNBR,KPRSIZ,KPNLW,
     *          KPNRW,KPFLW,KPIPTR,KPOPTR,KPNOUT,KPALFL,KPOUFL,
     *        ITOTAL,NS,OUCODE,IWRTL,IWRTM,STATSZ,STAPTR,
     *        TKPIPT,
     *        CODESZ,CODPTR
C
      COMMON /KEPBL1/ KPFNID,KPFNKP,KPBPTR,KPNPTR,KPMBR,KPNBR,
     *                KPRSIZ,KPNLW,KPNRW,KPFLW,KPIPTR,KPOPTR,KPNOUT,
     *                KPALFL,KPOUFL
      COMMON /MCIDA/ ITOTAL,NS,OUCODE,IWRTL,IWRTM,STATSZ,STAPTR
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...CHECK ATTENTION CODE.
      IF (IREP.LT.0) GOTO 200
C
C...IF EXECUTION TIMES GREATER THAN NS THEN GOTO 200
      IF (IREP.GT.PAR(5)) GOTO 200
C
C...RENEW CIRCUIT ELEMENTS' VALUES.
      CALL MFRNEW(IREP,PAR)
      GOTO 500
C
C...RESET ALL ALTERED ELEMENTS BACK TO THEIR ORIGINAL VALUES.
C...CALCULATE STATISTIC DATA FOR ELEMENTS ALTERED.
C...RECORD AND DISPLAY THEM.
C...FREE ENCODING ELEMENTS STORAGE BLOCK BEFORE RETURN
200   CALL MFDADS(PAR,FILEID,3)
300   CODESZ=PAR(10)
      CODPTR=PAR(9)
      IF (CODPTR.NE.0) CALL FSTORE(CODESZ,CODPTR,4)
      IF (STAPTR.NE.0) CALL FSTORE(STATSZ,STAPTR,8)
      CODPTR=0
      STAPTR=0
```

```
      IF (IREP.LT.0) GOTO 999
C
C...IF NON-LINEAR CIRCUIT AND IPDC IS ON, DO DC ANALYSIS TO UPDATE
C...IP VECTOR.
500   IF (PAR(2).NE.1) GOTO 999
      TKPIPT=KPIPTR
                    KPIPTR=0
      CALL DCMAIN
      KPIPTR=TKPIPT
999   RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE MFSTAS(IN,NIN,IIER,MOUT)
C***********************************************************************
C
C     MONTE CARLO ANALYSIS SUBROUTINE
C     CALCULATES AND DISPLAYS STATISTICAL RESULTS OF OUTPUT
C
C       IN(1) = MODE, THE MCFR OUTPUT FORMAT CONTROL CODE;
C               1(INI), 2(MIN), 3(MAX), 4(AVG), 5(STD), 6(ALL), AND
C               -6(SECOND CALL FOR ALL).
C       IN(2) = NS, NUMBER OF SAMPLES.
C       IN(3) > 0 NOT MCDI ZO.
C             < 0 MCDI ZO.
C       IN(4) = TIME, MCFR ANALYSIS START TIME(NOT USED HERE).
C
C       IIER  = 0 NO ERROR
C             =-1 ERROR, READING FILE/INCOMPARABLE OUTPUT REQUEST
C                          INSUFFICIENT VIRTUAL STORAGE.
C             EMPTY => INSUFFICIENT MACRO STORAGE.
C             > 0 NUMBER OF SETS AVAILABLE IN WKEEP FILE.
C                 WHEN NKPS NOT EQUAL TO NS+1.
C
C     CREATED 12-FEB-1989      D. YON  EE DEP. YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4 TIME
      REAL*8 RDATA,
     *    VOUT,OUTPCD,
     *    OUTPV,
     *    IN(6),IIER(1),
     *    FREQ,SUM(7),SUMW(7),MEMO(5,7),
     *    DABS,DCOS,DSIN,DSQRT
C
      INTEGER IDATA(1),
     *    KUFNID,KUFNKP,KUBPTR,KUNPTR,KUMBR,KUFBR,KUNBR,KURSIZ,
     *      KUNLR,KUNRR,KUMRR,KUFLR,KUOUPT,KUNOUT,KUCOMP,KUATP,
     *      KUNOD1,KUOPTR,
     *    RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW,
     *    ANALYZ,
     *    ATTCD,
     *    HEAD,ETOUSP,NOUT,ALLCD,XLBLCD,CMPLCD,XANDCD,
     *    RNOUT,NNOUT,NKPS,MODE,ACTION,COUTT,LNTH(11),
     *    NS,SN,IKUPTR,PONTER,STASIZ,NEEDSZ,STAPTR,
     *    NIN,MOUT,
     *    I,J,K,L,M,MX,BADF,IER
      INTEGER APNTR,IPTR,RPTR,EXSAVD
C
      CHARACTER*1 OUTREC,EXBUFR,COLON
      CHARACTER*2 OF
      CHARACTER*5 VSX(2)
      CHARACTER*8 OUTPN(6),MODEN(5),OUT,OUTPM,CBUFER(7)
      CHARACTER*40 ERR1
      CHARACTER*64 ERRSUB
      CHARACTER*132 COUREC
```

```
C
      EQUIVALENCE (RDATA(1),IDATA(1)),
     *            (OUTREC,COUREC)
C
      COMMON /MEMORY/ RDATA(1)
     *       /FRBL/   VOUT(16),OUTPCD
     *       /OUTBL2/ OUTPV(8),OUTPTP(8)
     *       /KUSBL1/ KUFNID,KUFNKP,KUBPTR,KUNPTR,KUMBR,KUFBR,
     *                KUNBR,KURSIZ,KUNLR,KUNRR,KUMRR,KUFLR,KUOUPT,
     *                KUNOUT,KUCOMP,KUATP,KUNOD1,KUOPTR
     *       /BINPUT/ RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW
     *       /APNTRS/ ANALYZ(30)
     *       /ATTIBL/ ATTCD
     *       /OUTBL1/ HEAD(12,7),ETOUSP(3,8),NOUT,ALLCD,XLBLCD,CMPLCD,
     *                XANDCD
     *       /BLISTR/ NOUTRC,OUTREC(132)
     *       /ERRSUB/ ERRSUB
     *       /KUXTRA/ EXSAVD(2),EXBUFR(80)
C
      DATA OUTPN/'ABS VALU','PHASE','REAL','IMAG',' ','dB ABS'/,
     *   VSX/'FREQ ','OMEGA'/,
     *   COLON/':'/,
     *   MODEN/'INITIAL','MINIMUM','MAXIMUM','MEAN','ST.DEV'/,
     *   LNTH/8,5,4,4,0,6,7,7,7,4,6/,
     *   OUT/'OUTPUT'/,
     *   OF/'OF'/,
     *   ERR1/'TRY INCREASE MEMORY/REDUCE FREQ. DENSITY'/
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IF (MOUT.LT.1) GOTO 980
      MODE=IN(1)
      IF (MODE.EQ.-6) GOTO 620
      IIER(1)=0
      NS=IN(2)
      APNTR=ANALYZ(12)
      ACTION=IDATA(APNTR+2)
C
C...IN(3) < 0 ==> MCDI ZO
      IF (IN(3).LT.0) THEN
        IF (MOUT.LT.25) GOTO 980
       MOUT=25
         CALL GTIMER(TIME)
         IIER(MOUT)=TIME
       GOTO 600
      END IF
      COUTT=IDATA(APNTR+36)
      IPTR=APNTR+NFIX
      RPTR=IDATA(APNTR)
      RNOUT=IDATA(APNTR+7)
      NNOUT=RNOUT
      IF (MODE.EQ.6.AND.ACTION.GT.2) NNOUT=1
C
C...READ OUT EXSAVD FOR KPINIT. KUMRR FOR GSTORE.
      CALL KUINIT(APNTR+10,IDATA(APNTR+7),APNTR+31,RDATA(RPTR),EXSAVD,
     *     EXBUFR,IER)
```

```
      CALL KUTERM
      IF(IER.NE.0) GOTO 970
C
C...CHECK STORAGE PROBLEM
      IF (KUNRR.EQ.KUMRR) GOTO 590
      STAPTR=0
      GOTO 950
C
C...STORE PARAMETERS FOR DI OU
590   IF (ACTION.LE.2) GOTO 650
      IF (MODE.NE.6) GOTO 620
      IF (MOUT.LT.24) GOTO 980
      MOUT=24
600   DO 610 I=7,30
610   IIER(I-6)=IDATA(APNTR+I)
      GOTO 999
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C...MFST INITIALIZATION
620   CALL KPINIT(12,IER)
      IF (IER.NE.0) GOTO 999
C
C...RESTORE DI ANALYSIS OU PARAMETERS
      IF (MODE.NE.-6) GOTO 10
      MODE=-MODE
      DO 640 I=7,30
640   IDATA(APNTR+I)=IN(I-2)
      GOTO 10
C
C...GET STORAGE
650   STASIZ=NNOUT*KUMRR*5+KUMRR
      STAPTR=0
      CALL GSTORE(STASIZ,STAPTR,8,-1,IER)
      IF (IER.NE.0) GOTO 950
C
C...READ AND PROCESS DATA OF THE SAME FREQUENCY ON EVERY SET.
10    I=0
      BADF=0
      PONTER=STAPTR
C
C...I IS THE COUNTER FOR FREQUENCY SAMPLED.
30    I=I+1
      DO 40 L=1,NNOUT
      SUM(L)=0D0
40    SUMW(L)=0D0
      CALL KUINIT(APNTR+10,IDATA(APNTR+7),APNTR+31,RDATA(RPTR),EXSAVD,
     *     EXBUFR,IER)
      IF(IER.NE.0) GOTO 960
C
C...CHECK STORAGE PROBLEM
      IF (KUNRR.NE.KUMRR) THEN
       IF (STAPTR.EQ.0) STAPTR=1
       GOTO 950
```

```
      END IF
      NKPS=0
      IKUPTR=KUNPTR+(I-1)*KURSIZ
50    IF (ATTCD.EQ.1) GOTO 960
      KUNPTR=IKUPTR
      KUNBR=I
      CALL KURECE(IER)
      IF (IER.NE.0) GOTO 190
      NKPS=NKPS+1
C
C...DISCARD THIS SAMPLE IF FREQUENCY NOT THE SAME AMONG SETS.
C...REJECT THE RECORD CREATED BY IPDC ON.
      IF (NKPS.EQ.1) FREQ=OUTPV(8)
      IF ((FREQ.EQ.0).OR.(OUTPV(8).NE.FREQ)) THEN
        BADF=BADF+1
        GOTO 180
      END IF
C
C...IF READING THE LAST SET, BYPASS IT FROM STATISTICAL DATA PROCESSING.
      IF (NKPS.GT.NS) GOTO 100
      DO 60 L=1,NNOUT
      SUM(L)=SUM(L)+OUTPV(L)
60    SUMW(L)=SUMW(L)+OUTPV(L)**2
C
C...MEMO(1,X)=INITIAL,MEMO(2,X)=SMALLEST,MEMO(3,X)=LARGEST,
C...MEMO(4,X)=AVERAGE,MEMO(5,X)=ST.DEV.
      IF (NKPS.EQ.1) THEN
        DO 70 L=1,NNOUT
        MEMO(2,L)=OUTPV(L)
70      MEMO(3,L)=OUTPV(L)
      ELSE
        DO 80 L=1,NNOUT
        IF (OUTPV(L).LT.MEMO(2,L)) MEMO(2,L)=OUTPV(L)
        IF (OUTPV(L).GT.MEMO(3,L)) MEMO(3,L)=OUTPV(L)
80      CONTINUE
      END IF
C
C...NEXT SET
100   CALL KUNEXT(IER)
      IF (IER.EQ.0) GOTO 50
C
C...CALCULATE THE CURRENT OUTPUT STATISTICAL RESULTS
C...AT THE CURRENT FREQUENCY.
      SN=NKPS
      IF(NKPS.EQ.NS+1) SN=SN-1
      DO 110 L=1,NNOUT
      MEMO(4,L)=SUM(L)/SN
      MEMO(5,L)=MEMO(4,L)**2-SUMW(L)/SN
      MEMO(5,L)=DSQRT(DABS(MEMO(5,L)))
      MEMO(1,L)=OUTPV(L)
C
C...INITIAL=0 TO INDICATE NO INITIAL OUTPUT AVAILABLE IN WKEEP FILE.
      IF (NKPS.NE.NS+1) MEMO(1,L)=0D0
110   CONTINUE
```

```
C
C...STORE THE STATISTICAL RESULTS
      IF (ACTION.LE.2) THEN
C
C...ENTER THE RESULTS INTO RDATA. THE FORMAT IS:
C...+-----------------------------------------------------------------+
C...|F1| OUTPUT 1              | OUTPUT 2  |...|OUT N|F2| OUTPUT 1     |   |
C...|  |INI|MIN|MAX|AVG|STD|INI|MIN|...|...|...   |  |INI|MIN|MAX...|...|
C...+-----------------------------------------------------------------+
        RDATA(PONTER+1)=FREQ
        PONTER=PONTER+1
        DO 120 L=1,NNOUT
        RDATA(PONTER+1)=MEMO(1,L)
        RDATA(PONTER+2)=MEMO(2,L)
        RDATA(PONTER+3)=MEMO(3,L)
        RDATA(PONTER+4)=MEMO(4,L)
        RDATA(PONTER+5)=MEMO(5,L)
120     PONTER=PONTER+5
C
      ELSE
C
C...TRANSFORM DATA INTO REAL AND IMAGINARY ACCORDING TO OUTPUT TYPE
        GOTO (160,130,160,140,180,150,180), COUTT
C
C...PH
130     IF (MODE.EQ.6) THEN
          DO 132 J=1,5
          VOUT(2*J-1)=DCOS(MEMO(J,1))
132       VOUT(2*J)=DSIN(MEMO(J,1))
        ELSE
          DO 134 J=1,RNOUT
          VOUT(2*J-1)=DCOS(MEMO(MODE,J))
134       VOUT(2*J)=DSIN(MEMO(MODE,J))
        END IF
        GOTO 170
C
C...IM
140     IF (MODE.EQ.6) THEN
          DO 142 J=1,5
          VOUT(2*J-1)=1D0
142       VOUT(2*J)=MEMO(J,1)
        ELSE
          DO 144 J=1,RNOUT
          VOUT(2*J-1)=1D0
144       VOUT(2*J)=MEMO(MODE,J)
        END IF
        GOTO 170
C
C...DB
150     IF (MODE.EQ.6) THEN
          DO 152 J=1,5
152       MEMO(J,1)=10**(MEMO(J,1)/20D0)
        ELSE
          DO 154 J=1,RNOUT
```

```
154     MEMO(MODE,J)=10**(MEMO(MODE,J)/20D0)
      END IF
C
C...MA,RE
160     IF (MODE.EQ.6) THEN
          DO 162 J=1,5
          VOUT(2*J-1)=MEMO(J,1)
162       VOUT(2*J)=0D0
        ELSE
          DO 164 J=1,RNOUT
          VOUT(2*J-1)=MEMO(MODE,J)
164       VOUT(2*J)=0D0
        END IF
C
C...STORE RESULTS INTO RDATA BUFFER
170     CALL KPENTR(FREQ,VOUT,J-1,DMMY,2)
      END IF
C
C...NEXT FREQUENCY
180   CALL KUTERM
      GOTO 30
C
C...JOB DONE. PREPARE OUTPUT FORMAT
190   CALL KUTERM
      COUREC=' '
      IF (MODE.NE.6) GOTO 400
C
C...MODE=6
      OUTPM=OUTPN(COUTT)
      M=LNTH(COUTT)+2
      IF (ACTION.NE.4) GOTO 200
      OUTPM=OUT
      M=8
200   COUREC=OUTPM
      COUREC((M-1):)=COLON
C
C...PERFORM OUTPUT ACTION ACCORDING TO THE ACTION CODE
      IF (ACTION-2) 240,220,250
C
C...PR
220   KUMRR=KUMRR-BADF
      DO 230 L=1,NNOUT
      WRITE (WRUNT,1100) (OUTREC(J),J=1,M),(HEAD(J,L),J=1,12)
      WRITE (WRUNT,1200) VSX(EXSAVD(2))
      PONTER=STAPTR+5*(L-1)+1
      DO 230 J=1,KUMRR
      FREQ=RDATA(STAPTR+(J-1)*5*NNOUT+J)
      WRITE(WRUNT,1300) FREQ,(RDATA(PONTER+I),I=1,5)
230   PONTER=PONTER+5*NNOUT+1
240   CALL FSTORE(STASIZ,STAPTR,8)
      GOTO 800
C
C...PP/PL/PO/PT/PV/CALL
250   WRITE(WRUNT,1100) (OUTREC(J),J=1,M),(HEAD(J,1),J=1,12)
```

```
      CALL KPMARK(1)
      CALL KPTERM
      GOTO 800
C
C...MODE^=6
400   IF (ACTION-2) 240,420,460
C
C...PR
420   M=LNTH(COUTT)+2
      COUREC=OUTPN(COUTT)
      OUTREC(M-1)=COLON
      WRITE(WRUNT,1100) (OUTREC(J),J=1,M)
      WRITE(WRUNT,1600) VSX(EXSAVD(2)),((HEAD(J,L),J=1,12),L=1,NNOUT)
      DO 430 L=1,NNOUT
430   CBUFER(L)=MODEN(MODE)
      WRITE(WRUNT,1700)(CBUFER(L),L=1,NNOUT)
      WRITE(WRUNT,1800)
      NEEDSZ=5*NNOUT
      KUFBR=KUFBR-BADF
      DO 450 I=1,KUFBR
      PONTER=STAPTR+5*(I-1)*NNOUT+I
      FREQ=RDATA(PONTER)
      PONTER=PONTER+MODE-1
450   WRITE(WRUNT,1500) FREQ,(RDATA(PONTER+K),K=1,NEEDSZ,5)
      CALL FSTORE(STASIZ,STAPTR,8)
      GOTO 800
C
C...PP,PL,PO,PT,PV,CALL
460   M=LNTH(MODE+6)+1
      COUREC=MODEN(MODE)
      OUTREC(M)=COLON
      WRITE(WRUNT,1100) (OUTREC(J),J=1,M)
      CALL KPMARK(1)
      CALL KPTERM
C
C...CHECK NUMBER OF SETS READ.
800   IF (NKPS.NE.NS+1) IIER(1)=NKPS
      GOTO 975
C
C...ERROR
950   CALL ERRMSG(108011)
      ERRSUB=ERR1
      CALL ERRMSG(100159)
      IF (STAPTR.EQ.0) GOTO 970
960   CALL KUTERM
      IF (ACTION.LE.2) THEN
        CALL FSTORE(STAPTR,STASIZ,8)
      ELSE
        CALL KPMARK(1)
        CALL KPTERM
      END IF
970   IIER(1)=-1
975   MOUT=1
      GOTO 999
```

```
C
980   MOUT=0
C
999   RETURN
C
1100  FORMAT(/,1X,132A1)
1200  FORMAT(/,1X,T8,A5,T20,'INITIAL',T33,'MINIMUM',T46,'MAXIMUM',
     * T60,'MEAN',T72,'ST.DEV',/)
1300  FORMAT(' ',1X,1P8E13.5)
1400  FORMAT(/,' ','MCFR EXECUTION TIME =',F8.3,' SEC.',/)
1500  FORMAT(1P8E16.5)
1600  FORMAT(/,1X,T6,A5,T22,7(12A1,4X))
1700  FORMAT(1X,T22,7(A7,9X))
1800  FORMAT(1X)
      END
```

```
C***********************************************************************
      SUBROUTINE MFTIME(IN,NIN,DMMY,MOUT)
C***********************************************************************
C
C  MONTE CARLO ANALYSIS SUBROUTINE.
C   PERFORM PP/PL/PO/PT/PV/CALL ACTION AND WRITES TIME STAMP.
C
C     IN(1) = 1, MCFR
C           = 2, MCDI FR
C           = 3, MCDI ZO
C     IN(2) = NS, NOT USED HERE.
C     IN(3) = NOT USED.
C     IN(4) = TSTART, MCFR EXECUTION START TIME.
C     IN(5) = USPTR, POINTER OF FNAME FOR DI USE.
C
C  CREATED 12-FEB-1989    D. YON   EE DEP.  YSU
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*4 TIME
      REAL*8 RDATA,
     *        RINFNT,IINFNT,NFIX,
     *        IN(5),DMMY,TSTART
C
      INTEGER IDATA(1),
     *          ANALYZ,
     *          RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW,
     *           ANIDSW,TSTMSW,WAITSW,
     *           NIN,MOUT,
     *           ANPNTR,DIPNTR,ACTION,USPTR,WHICH
C
      EQUIVALENCE (RDATA(1),IDATA(1))
C
      COMMON /MEMORY/ RDATA(1)
     *     /BLCK04/ RINFNT,IINFNT,NFIX
     *       /APNTRS/ ANALYZ(30)
     *         /BINPUT/ RDUNT,WRUNT,TRMUNT,WRTSW,TRMSW,RDSW,EOFSW
     *       /OPTBL2/ ANIDSW,TSTMSW,WAITSW
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      MOUT=0
      WHICH=IN(1)
      ANPNTR=ANALYZ(3)
      DIPNTR=ANALYZ(12)
      ACTION=IDATA(DIPNTR+2)
      TSTART=IN(4)
C
C...IF PL, PAUSE EXECUTION HERE
20    IF (ACTION.LE.2.OR.WHICH.EQ.3) GOTO 100
      IF (ACTION.NE.6) GOTO 80
      WRITE (TRMUNT,1100)
      READ (TRMUNT,1400,END=50)
```

```
50     REWIND TRMUNT
C
C...TURN ON ANALYSIS TITLE
       ANIDSW=1
80     CALL DISPLA
C
C...TIME STAMP
100    CALL GTIMER(TIME)
       TIME=TIME-TSTART
C
C...MCFR OR MCDI ?
       GOTO (110,200,210), WHICH
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...MCFR ENVIRONMENT.
C...PUT FR ANALYSIS ACTION CODE BACK.
110    IDATA(ANPNTR+2)=ACTION
C
C...SET FR VB/IB TO UNDEFINED.
       DO 120 I=3,6
120    RDATA(IDATA(ANPNTR)+I)=RINFNT
C
C...SET DI NOT READY.
       IDATA(DIPNTR+1)=0
C
C...SET NO FNAME FOR DI US
       RDATA(IDATA(DIPNTR))=RINFNT
C
C...TURN OFF DI OUTPUT SPECIFICATION.
       DO 130 I=7,30
130    IDATA(DIPNTR+I)=0
C
C...WRITE TIME STAMP FOR MCFR.
       WRITE(WRUNT,1200) TIME
       GOTO 999
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C...MCDI ENVIRONMENT
C...PUT DI ANALYSIS ACTION CODE BACK.
200    IDATA(DIPNTR+2)=ACTION
210    USPTR=IN(5)
       IF (USPTR.NE.0.) THEN
         RDATA(IDATA(DIPNTR))=RDATA(USPTR+1)
         CALL FSTORE(1,USPTR,8)
       END IF
       IF (NIN.LE.6) GOTO 230
C
C...RESTORE DI OU
       DO 220 I=7,30
220    IDATA(DIPNTR+I)=IN(I-1)
C
C...WRITE TIME STAMP FOR MCDI FR.
230    WRITE(WRUNT,1300) TIME
C
999    RETURN
```

```
C
1100  FORMAT(1X,'Hit ENTER key when ready to display output...')
1200  FORMAT(1X,'MCFR EXECUTION TIME=',F8.3,' SEC.',/)
1300  FORMAT(1X,'DISPLAY TIME=',F8.3,' SEC.',/)
1400  FORMAT(A1)
      END
```

LIST OF REFERENCES

1. Vlach, Martin, WATAND User's Manual (Version 1.09-08), revised by Bryant, P.R., Strayer, H.J. and Munro, P.C., University of Waterloo, December 15, 1985.

2. Hajj,I.N., Singhal,K., Vlach,J., and Bryant,P.R., "A Program for the Analysis and Design of Linear and Piecewise-Linear Networks, Proceedings of the Sixteenth Midwest Symposium on Circuit Theory, U. of Waterloo, April 12-13, 1973.

3. Munro,P.C., Watand Help Files on YSU Computer System, 1988.

4. Munro,P.C., Watand User's Manual for Youngstown State University, V1.11-3f, December 1988.

5. Boctor, S.A., "A Novel Second-order Canonical RC-active Realization of High-pass-notch filter", Proc. IEEE International Symposium Circuits and Systems, San Francisco, Calif., April 1974.

6. Suen, Janq-Fang, Computer-Aided Design: Monte-Carlo DC Analysis Using Watand, M.S. Thesis, Youngstown State University., August 1987.

7. Munro, P.C., file WUPDATE MEMO on YSU Watand library disk.

8. Becker, P.W. and Jensen, F., Design of Systems and Circuits for Maximum Reliability or Maximum Production Yield. McGraw-Hill, New York, 1977.

9. Inohira, S., Shinmi, T., Nagata, M., Toyabe, T., and Iida, K., "A Statistical Model Including Parameter Matching for Analog Integrated Circuits Simulation", IEEE Transactions on Computer-Aided Design, Vol. CAD-4, No.4, October 1985. 621-628

10. Van Valkenburg, M.E., Analog Filter Design. Holt, Rinehart and Winston, 1982.

11. Hsu, B.R., An Advanced Large-Signal Operational Amplifier Macromodel for Watand Computer Simulation, M.S. Thesis, Youngstown State University, December 1987.

12. Harnett, Donald L., Statistical Methods, Third Edition, Addison-Wesley, June 1982.