NAME

runadvice – run SysCAD ADVICE program with varying parameters

SYNOPSIS

runadvice [*runfile*] [**-o** *outfile*] [**-p[aram]** *param=p1,p2,...*] [**-V**] [**-D** | **-D=***debuglevel*] [**-?**] [**-CONFIG** *configuration_file*] [**-TMPDIR** *temporary_directory*] [**-ADVICE** *advice_program*] [**-proc** *process,...*] [**-timestep** *step,...*] [**-timelen** *length,...*] [**-temp** *temperature,...*] [**-pow** *voltage,...*]

DESCRIPTION

This program is used to perform ADVICE runs on a circuit with varying parameters of the run. The most common aspects of an ADVICE run that are varied are usually the technology **process**, **temperature**, **timestep**, **timelength**, and **power supply voltage**. This program can actually allow for the varying of any user created parameter in an ADVICE run. This program expects a number of pseudo standardized files to be in the current directory which describe the circuit to be simulated using ADVICE. The files that are expected to be created by the user are:

`run.adv`

`params.adv` (optional)

`control.adv`

`main.adv`

These files contain information that describe the circuit to be simulated along with any necessary subcircuits that ADVICE will need to simulate the "main" circuit. These files, along with others, are described further in a section below.

Argument Key Options

The following keyed options are available for specifying execution behavior of the program.

-? The **-?** is a standard option which causes the program to terminate with a message printed to standard error giving a brief explanation of the program. This is useful when the user forgets the exact syntax of the program's arguments and does not want to consult the manual page.

-V The **-V** option causes the program to write out to standard error the version of the program. The program will then terminate.

-D The **-D** option enables the printing of some debugging information to the standard error. This optional option can (optionally) be followed by a debug level. A specification of the debuglevel is given on the command in the form **-D=***debuglevel*. Specifying this option without the debug level is equivalent to specifying debugging with a debug level of one (1). This option is essentially useless except for the program developer.

-o *output_file* The **-o** key letter option is used to specify an ADVICE output file to receive the results of all ADVICE execution outputs. The output file only is tracked correctly when the user uses the **OUTFILE** variable in the ADVICE run file.

-param *parameter*=*value_list*
This option allows the user to specify (introduce) a variable, denoted by *parameter* above, along with a list of associated values. These values will be used, along with all other parameter values to create enough ADVICE runs to cover all combinations of the values. The *value_list* above takes the form of comma separated list of value items.

-CONFIG *configuration_file*
This option allows the user to specify a configuration file. This file can contain defaults for substitutable parameters as well as some other information. See a sample configuration file for more information about what can be specified in it.

- TMPDIR*temporary_directory*

This option allows the user to specify an alternate temporary directory for use by *runadvice* when creating temporary files.

- ADVICE*ADVICE_program*

This option allows the user to specify an alternate ADVICE program to run instead of the default ('**`runadvice_advice`**') or that specified in the *RUNADVICE_ADVICE* environment variable.

In addition to the options already described, a small set of options are recognized by the program for the added convenience of the user. The following argument option keywords are recognized:

- proc*process_value_list*

This is the same as `-param proc=`*process_value_list* had been given on the command line.

- timestep*timestep_value_list*

This is the same as `-param timestep=`*timestep_value_list* had been given on the command line.

- timelen*timelen_value_list*

This is the same as `-param timelen=`*timelen_value_list* had been given on the command line.

- temp*temperature_value_list*

This is the same as if `-param temp=`*temperature_value_list* had been given on the command line.

- pow*power_supply_voltage_value_list*

This is the same as if `-param pow=`*power_supply_voltage_value_list* had been given on the command line.

ADVICE Circuit Files

There are a number of files expected by this program to describe the circuit to be simulated. These files are listed below. All of the files (ADVICE Circuit Files) discussed below are processed by *runadvice* for substitutable parameter variables.

ADVICE Run File

This file, usually named `run.adv`, contains ADVICE commands to set libraries, read in a circuit file, specify ADVICE output files, and the like. This file is not strictly a circuit file in that no ADVICE circuits (as far as I am aware) are allowed in this file. This file is what is read by ADVICE on its invocation and must contain the proper ADVICE commands appropriate to the user's simulation objective.

This file is specified as *runfile* on the command line. If this is omitted from the command line, then standard input is read as the run file instead. If this file is given as a dash ('- ') character, standard input is read also. A good example of this file would have substitutable parameters for anything that the user wants to be varied by the *runadvice* program. A list of the standard and popular parameter variables are described in a later section. This file is processed for *runadvice* variable parameters.

ADVICE Parameter (`params.adv`) File

This file contains ADVICE commands. An ADVICE parameter file (which is optional) usually contains ADVICE variables specified as an ADVICE line of the form:

```
{ termvolt = 2.1 }
```

In order to have the *runadvice* program vary ADVICE variables, an ADVICE line of the form:

```
{ termvolt = $(TERMVOLT) }
```

would be used. More on the use of the *runadvice* variables is described later. As already alluded to, this file is processed for *runadvice* variable parameters.

ADVICE Control File

This file, made up of ADVICE commands, usually contains an include (realized with the ADVICE **`.use`** command) of the "main" ADVICE circuit file. Other includes are usually located in this file also like for example the inclusion of other necessary subcircuits which are not included into the ADVICE run in any other way. This file is processed for *runadvice* variable parameters.

ADVICE Main Circuit File

This file usually contains the circuit to be simulated proper. Usually, at least the "main" circuit of an ADVICE run is located in this file but other ADVICE subcircuits can be included with the main circuit. ADVICE subcircuits are normally included when this file is created with a connectivity generator such as the one in SysCAD SCHEMA. This file is processed for *runadvice* variable parameters.

ADVICE Output File

An *outfile* argument following the **- o**option key letter specifies the file which is to receive the output of all of the ADVICE runs. All run outputs are placed into this file so that when read into a reader program like ADVPLOT, the individual run outputs appear numbered.

RUNADVICE Variable Parameters

Several of the standard files above, those indicated as such, are processed for parameter substitution. Variable parameters can be introduced into the files by including variable references in the form:

`$(`*variable*`)`

Where *variable* can be any user created variable. These variables are searched for and replaced by a combination of values such that all combinations of the variables given on the command line get a corresponding ADVICE simulation run. Some variable names have special meaning and cannot be used by the user for any other purpose. The variables with special meanings are:

CONTROLCKT	This variable parameter gets replaced with the name of a modified `control.adv` file.
MAINCKT	This variable parameter gets replaced with the name of a modified `main.adv` file.
PARAMS	This variable parameter gets replaced with the name of a modified `params.adv` file.
OUTFILE	This variable parameter gets replaced with the name of a temporary output file whose contents are eventually added to the user specified output file or to the default output file.

In addition to the special *runadvice* variables described so far, the following variables are defaulted internally to the *runadvice* program if the user does not supply them:

PROC	This variable will hold a string specifying a certain technology processing to be used in the simulation. This variable would be used as part of a library specification.
TEMP	This variable will hold the simulation temperature.
TIMESTEP	This variable will hold a time step value like as for a transient analysis.
TIMELEN	This variable will hold a time duration like as for a transient analysis.
POW	This variable will hold a power supply voltage value.

Currently, all parameter names given by the user are converted to uppercase before being used further. In addition, the user should only put uppercase environment variables inside of the ADVICE files processed by *runadvice*.

EXAMPLES

☞ To run a circuit with variations on technology process (identified as **an**, **al**, and **ah**) the following would be executed:

```
runadvice run.adv -param proc=al,an,ah
```

This will produce three ADVICE runs, one for each technology process specified.

☞ To run a circuit with variations on technology process (identified as **an**, **al**, and **ah**) and power supply voltage, execute:

```
runadvice run.adv -param proc=al,an,ah -param pow=3.0,3.3,3.6
```

This will produce nine ADVICE runs.

☞ To run a simulation on a circuit with variations on technology process (identified as **an**, **al**, and **ah**), power supply voltage, and temperature, execute:

```
runadvice run.adv -param proc=al,an,ah -param pow=3.0,3.3,3.6 -param temp=25
```

This will produce eighteen ADVICE runs.

ENVIRONMENT VARIABLES

The following environment variables tailor some aspect of the program to the user's preferences.

TMPDIR This environment variable will direct the program to create temporary files in the directory specified. Any specification of a temporary directory that is given on the command line will override this variable. If this variable is not set, and no directory is given on the command line, the directory `/tmp` will be used for temporary files.

RUNADVICE_CONFIG

This environment variable contains the name of a configuration file to be used by the program for default substitution parameters and some other things. If any configuration file is specified on the invocation command, then that file is used instead of the one in this environment variable.

RUNADVICE_ADVICE

This environment variable contains the name of the ADVICE (or ADVICE like) program to execute. If the ADVICE program is specified on the command line, this variable is ignored.

FILES

run.adv	default ADVICE "run" file
params.adv	default circuit parameter file ; this file usually contains ADVICE parameter assignments
control.adv	circuit control file ; this file usually contains a series of ADVICE "`.use` *file*" lines to include subcircuits not otherwise included in the run
main.adv	default main circuit file ; this file should usually contain an ADVICE circuit beginning with "`.main` *circuit_name*" and ending with a "`.end`".
main.out	default output file ; can be changed with the **-o**option on the command line
runadvice_advice	default ADVICE program executed by this program ; the user's current execution PATH is searched for this program ; this program can be changed either through the environment variable RUNADVICE_ADVICE or on the command line

CAVEATS

Currently, the names of the ADVICE circuit files (`params.adv`, `control.adv`, `main.adv`) cannot be specified on the command line. The names can be changed if the user puts the appropriate control lines in a configuration file. In the future, alternate names for these files may be specified on the command line.

SEE ALSO

eas(1)

SAMPLE FILES

A sample RUNADVICE configuration file is shown below:

```
# CONFIG.RA
```

```
#
# This is a configuration file for the RUNADVICE program.
#
#
# Valid Entries:
#
#     keyword           parameters
#-------------------------------------------------------------------------
#
#     advice            ADVICE program
#
#     export            an environment variable name followed by a value
#
#     default           a parameter name followed by one or more values
#                  in a list separated by commas
#
#     control           the name of the ADVICE 'control.adv' like file
#
#     params            the name of the ADVICE 'params.adv' like file
#
#     main        the name of the ADVICE 'main.adv' like file
#                  containing the "main" circuit to be simulated
#
#     machine           this allows for the specification of available
#                  machines ; only those machines listed are used
#
#
#
# NOTE about how the ADVICE files relate to each other:
#
#     The "run" file possibly specified on the command line
#     must read in a circuit file which will be in the "control"
#     file.  The "control" file can optionally contain a "params"
#     file included with an ADVICE '.use' command.  The "control"
#     file must include the "main" file with an ADVICE '.use'
#     command.  The four files, "run", "control", "params", and "main"
#     are processed for parameters substitutions.
#
#
#


# ADVICE program
#advice           runadvice_advice

export            TMPDIR=/proj/starbase/tmp
export            SEEMEHERE=/proj/starbase/tmp

default           proc=an
default           temp=85
default           timestep=0.1n
default           timelen=5n
default           pow=3.3
```

```
default           vterm=1.2
default           subnch=0.0



# ADVICE files processed (in addition to "run") for parameter substitutions
control           control.adv
params            params.adv
main        main.adv


# machine entries
#
#     These entries have the form:
#
#           keyword           name  load_capability
#
#     where:
#
#           <keyword>         is always 'machine'
#
#           <name>                  is the name of the machine
#
#           <load_capability> is a number that rates the ability
#                             of the machine to run jobs as
#                             compared with the other machines
#

# our System V machines

machine           hodia=2.0
machine           hodid=2.0
machine           hodif=0.8

# our BSD machines

#machine          hodig=0.8
#machine          octet=2.0
```

A sample run file is shown here.

```
* ADVICE run file
*
* ADVICE run file which is executed from the the X Windows menu
*
*
*     The following character strings get substituted for:
*
*           PROC        process
*           TEMP        temperature
*           TIMESTEP    time step intervale
*           TIMELEN           time length
```

```
*           POW         power supply voltage
*
*
* here are some test substitutions:
*     PROC=$(PROC)
*     TEMP=$(TEMP)
*     TIMESTEP=$(TIMESTEP)
*     TIMELEN=$(TIMELEN)
*     POW=$(POW)
*
* and the circuit files:
*     MAINCKT=$(MAINCKT)
*     CONTROLCKT=$(CONTROLCKT)
*
*
*
* general library
*
.lib lib4=genlib.adv
*
*
* straight CMOS, 0.5um, 3.3 Volt
*
.lib lib1=cmos5_33$(PROC).adv
*
*
.use options.adv
*
*
.rd ${CONTROLCKT} ;n
*
.temp $(TEMP)
*
*
.opti rfile=init.adv
*
.opti pivot=1
.opti lvlprt=3
*
.opti itl1=500
.opti ipta=0
.dc op
.file rfile=winit.adv
*
*
.op * > op.results
*
.tran $(TIMESTEP) $(TIMELEN)
*
.file ofile=$(OUTFILE)
*
*
.end
```

A sample control file is shown here.

```
* ADVICE circuit "control" file

* .use skew2.adv
* .use ddlo_p.adv
.use constants.adv
.use $(PARAMS)
.use ${MAINCKT}
```

A sample ADVICE *params* file is shown here.

```
* circuit parameters for ADVICE
*
*
.set { vterm=$(VTERM) }
.set { pow=$(POW) }
.set { subnch=$(SUBNCH) }
*
*
*
```

AUTHOR

David A.D. Morano