

Section 2:

EWS Methodology

EWS DESIGN METHODOLOGY - DAISY.....(809)

Table of Contents.....2-2

Daisy Glossary.....2-5
Basic System Commands.....2-8
DANCE: Daisy Network Connectivity Extractor.....2-17
DRINK: Daisy Resolving Linker.....2-17
SPARC: Simulator Parameter Compiler.....2-17
AGIF: AMCC Interface Format.....2-24
ERC: Engineering Rules Checks.....2-25
FRONT-ANNOTATION.....2-26
BACK-ANNOTATION.....2-26
SIFT: Simulator Intermediate Files Translator.....2-27
SING: Simulation Input Generation.....2-29
FMT_CSD.SING: Format Control File.....2-30
SOM Control File.....2-32
OUTPUTS SECTION.....2-36
SOM.....2-49
TCAL: Timing Calculator.....2-50
DLS.....2-51
AMCCSIMFMT: AMCC Simulation File Formatter.....2-59
AMCCVRC: AMCC Vector Rules Checker.....2-60
Appendix A: Shell Scripts Listings.....2-A-1

DAISY/DNIX DESIGN METHODOLOGY

The following pages are taken from the AMCC EWS Design Methodology seminar for the DAISY Engineering Workstation under the DNIX operating system. No attempt has been made to include all of the course material from the AMCC seminar nor to replace the extensive DAISY reference and design manuals.

This section is designed to walk the user through an overview of the various steps in a design, from schematic capture to final AMCCSIMFMT execution.

AMCC MacroMatrix support software is integrated with the DAISY software to provide a comprehensive set of tools designed to simplify the user interface and allow the user to concentrate on the design rather than on the EWS system itself.

COMMANDS

For those users who are familiar with MAESTRO, the upgrade to DNIX is simplified by the fact that the MAESTRO commands, with minor modifications, work under DNIX. For those users who are familiar with UNIX or UNIX-clones, DNIX is a UNIX-like operating system and supports the lower-case commands as well.

For new users, since all files must be uppercase if they are to transfer to other computers, and since all directories must be uppercase, the MAESTRO commands may be easier to learn first.

DED

The DNIX system offers two graphics editors, DED and the newer DED2. DED and quickscreen editing (see Appendix B in this section) provide a fast graphics capture system. DED2 is oriented to window and mouse or puck operation. Either or both may be used to complete a schematic capture.

AMCC SHELL SCRIPTS

AMCC MacroMatrix support software includes preprogrammed shell scripts (also called shells) which minimize the effort required by the user when executing both the AMCC support software and the DAISY software. These shell scripts are documented in Appendix A at the back of this section.

Where a user does not need to alter these shells, they can be used to call and invoke all required steps up to DLS or DTV simulation. The very last step, AMCCSIMFMT, is called by typing its name and is run after simulation.

When the shells are not adequate for whatever reason, they can be edited or the commands can be entered without capture into a shell. DAISY also allows the user to program function keys with commonly used commands.

The commands are described on the following pages in sufficient detail to allow the user to evaluate the AMCC supplied shells. For further details, consult the DAISY reference manuals shipped with your system.

AMCC GLOSSARY OF EWS TERMS
SUMMARY

DAISY-SPECIFIC

ACE Another editor - not in design center yet
 AGIF AMCC Generic Interface Format
 RUN_AGIF.ERR Error file
 AMCCFRONT.LST Error file
 AMCCSIMFMT AMCC simulation format program
 AMCCSIMFMT.ERR Error file
 CONFIG Configuration file
 DANCE DAISY Network Connectivity Extractor
 DANCE.ERR Concatenated Error file - from AMCC shell
 DED DAISY Drawing EDitor, used for
 schematic capture
 (DED II is also for operation under DNIX)
 DFS ** DAISY Fault Simulator
 DLS DAISY Logical Simulator
 DNIX DAISY UNIX-like operating system
 DRINK DAISY Resolving LINKer
 DRING.ERR DRINK error file - from AMCC shell
 DTV DAISY Timing Verifier
 DTA ** DAISY Testibility Analyzer
 FMT_CSD.SING Format source file
 FMT Default DLS/DTV format file
 IMAGE.SOM Circuit image for simulation,
 from SOM atep
 INCR.DFR DRINK Incremental Report
 n.DRAW Drawing page produced under DED; DED2
 n.DFR Dance report file
 n.DIF Dance Intermediate File
 n.SFR Sift report file
 n.SIF Sift Intermediate File
 RUN-AMCC Super-shell
 RUN_AGIF AGIF shell
 RUN_DD DANCE-DRINK shell
 RUN_ERC ERC shell
 RUN_ANN Front-Annotation shell; Back-Annotation shell
 RUN_SIFT SIFT shell
 RUN_SMAKER SOM_MAKER shell
 RUN_SMT AMCCSIMFMT shell
 RUN_SOM SOM and TCAL shell
 RUN_VRC AMCCVRC shell
 SIFT Simulator Intermediate Files Translator
 (Users choose between MIN, NOM, MIL
 and COM, COM4 or COM5 files)
 SIFT.ERR Error file
 SING Simulation Input Generation Program
 SOM Simulator Object Module Generator
 SOM.ERR SOM control file and error report

SOM_MCF.SING

SPARC Usual name for simulation control file
Simulator Parameter Compiler
(Used by AMCC to create the object
SIFT files)

TCAL Timing Calculator
(part of the Front-Annotation/Back-
Annotation software package, a joint
DAISY-AMCC effort)

TCAL.ERR Error file

TIN ** Test Vector Generating Software

TO.SOM State at time = 0, from SOM step

TREE.DFR DRINK report file

TREE.DNLK DRINK Intermediate File

TREE.SIF Sift Intermediate File

TREE.SFR Sift report file

VLAIF Virtual Logic Analyzer Intermediate Format
(also referred to as remote data-vector
file; can be an output file as well)
This formatted output is required for
AMCCSIMFMT input

*.ERR Transcript error file (AMCC shell)

** not covered in the AMCC DAISY seminar

ANNOTATION FILES:

Front-Annotation:

FNTMIL.DSY	Front-Annotation MILITARY file
FNTCOM.DSY	Front-Annotation COMMERCIAL file
FNTNOM.DSY	Front-Annotation NOMINAL file
FNTMIN.DSY	Front-Annotation MINIMUM file
Intermediate-Annotation (when available):	
IBAMIL.DSY	Intermediate-Annotation MILITARY file
IBACOM.DSY	Intermediate-Annotation COMMERCIAL file
IBANOM.DSY	Intermediate-Annotation NOMINAL file
IBAMIN.DSY	Intermediate-Annotation MINIMUM file
Back-Annotation - Core data only; from AMCC)	
CORMIL.DSY	Back-Annotation MILITARY file
CORCOM.DSY	Back-Annotation COMMERCIAL file
CORNOM.DSY	Back-Annotation NOMINAL file
CORMIN.DSY	Back-Annotation MINIMUM file
Back-Annotation - complete; from COR, CIRCUIT.PKG and AMCCANN software	
BCKMIL.DSY	Back-Annotation MILITARY file
BCKCOM.DSY	Back-Annotation COMMERCIAL file
BCKNOM.DSY	Back-Annotation NOMINAL file
BCKMIN.DSY	Back-Annotation MINIMUM file

AMCC FILES:

AMCCERC.LST	ERC report, error list
AMCCIO.LST	I/O signal list; SSO table
AMCCPKG.LST	Package data report
AMCCVRC.LST	AMCCVRC report and error list
AMCCXREF.LST	Cross-reference listing
CIRCUIT.PKG	AMCC Package Data report
CIRCUIT.SDI	AMCC Formatted Netlist
OUTPUT.DLY	loading data file - submit

INTRODUCTION TO THE DAISY UNDER DNIX

LOGGING ON:

- TYPE YOUR LOGIN CODE (CLASS; DEW; etc.) AS ASSIGNED
- THE LOGIN IS NOT CASE-SPECIFIC
- IF YOU HAVE A PASSWORD DEFINED THEN TYPE THE PASSWORD
 - THE SYSTEM IS NOT SET UP FOR A PASSWORD AT PRESENT
 - A PASSWORD WILL NOT BE USED FOR THE LOGIN "CLASS"
 - PASSWORDS ARE CASE-SPECIFIC

LOGGING OFF:

- IN THE MAIN WINDOW TYPE "LOG"
 - LOGGING OFF DOES NOT CLOSE OPEN WINDOWS
 - CLOSE ALL SUB-WINDOWS BEFORE LOGGING OFF SYSTEM
- IN SUB-WINDOWS TYPE "CTL-E" OR "CTL-D"

INTRODUCTION TO THE DAISY UNDER DNIX

PROMPT:

- THE SYSTEM PROMPT IS "\$" FOR ALL WINDOWS
- UNDER "CLASS" THE PROMPT HAS BEEN DEFINED TO BE "CLASS>" FOR THE MAIN WINDOW
- YOU CAN DEFINE YOUR OWN PROMPT AND CAN DO SO FOR EACH WINDOW (WHY BOTHER?)
- DEFINITION IS IN THE loginfile AND IS FOR
PS1 = ' ' WHICH STANDS FOR "prompt-string-1"

LOGIN FILE:

- ONE EXISTS FOR "CLASS" FOR THE Q3500
- CREATE ONE THE FIRST TIME ON THE SYSTEM
- USE TEC TO CREATE IT
- IT MUST CONTAIN A SUBMIT SUCH AS:
%SUBMIT /AMCC/Qnnn_LIBS/QnnnnSETUP
TO SELECT LBRARY Qnnnn
- EXIT TEC AND LOG OFF!
(executing loginfile {RET} will NOT work)
- LOG BACK ON AND THE LIBRARY IS SELECTED
- THE FILE MUST BE NAMED "loginfile"
- MUST BE LOWERCASE

IC INVADERS!

HOW TO CREATE AND DESTROY WINDOWS

- MULTIPLE WINDOWS SLOW DOWN THE SYSTEM
- TWO WINDOWS OR THREE IS ENOUGH (THE LIMIT DEPENDS ON WHAT THEY ARE DOING)
- IF THE DEFAULT SIZE IS OK, THEN TYPE
{SHIFT}-{ZOOM}
 - THE "\$" PROMPT IN THE WINDOW IS WHERE YOU ARE TYPING
 - YOU ARE STILL IN THE SAME CURRENT CONTEXT
- TO CLOSE - PUT THE CURSOR IN THE WINDOW TO BE CLOSED
AND TYPE
PAN/CTL-E
- CAN TYPE "NW command"
 - OPENS A WINDOW AND FIRES OFF PROCESS
 - WINDOW IS "FOREGROUND" (VISIBLE)
 - CLOSSES WINDOW WHEN DONE
 - USE OF "NW" ALLOWS DIFFERENT WINDOW SIZES TO BE SPECIFIED

INTRODUCTION TO THE DAISY UNDER INIX

- CAN TYPE "command &" TO FIRE OFF A BACKGROUND OPERATION
 - system assigns a process number and returns a prompt to the user - hit <RET> and continue other tasks
 - ASSIGNS A PROCESS NUMBER TO THE TASK
 - DISPLAY THE TASKS IN PROCESS BY TYPING "ps"

EXAMPLE:

```
RUN_DD &
```

- TOGGLE BETWEEN WINDOWS BY THE YELLOW BUTTON ON THE "MOUSE"
 - PUT THE CURSOR ON THE TOP BANNER AND HIT YELLOW
 - NO WINDOW WILL COME UP IF ONE IS NOT THERE
 - TOP AND SMALLER WINDOWS CHANGE PLACE
 - KEEP THE CURSOR OUT OF THE BANNER
 - THE PROCESS WILL HALT
- IF THE CURSOR IS IN THE SELECT STATE (IN A BANNER)
- BLUE BUTTON IS A MENU BUTTON
 - BLUE IS MENU BANNER SELECT (SELECT AS FOR WINDOWS)
 - BLUE IS MENU OPTION SELECT
 - YELLOW CLOSES THE OPTION SELECTED
- LOCATOR OPTION - MOUSE CAN BE SET RELATIVE OR ABSOLUTE IN ITS MOTION
- FRONT/BACK IS THE SAME AS BANNER SELECT
- SHELL WINDOW IS ANOTHER WAY TO OPEN A WINDOW

OUTPUT CONTROL

- A LARGE ZOOMED "H" APPEARS IN THE UPPER CORNER

CURRENT CONTEXT

- APPEARS AT THE TOP OF THE SCREEN

WINDOW PANNING

- DEFAULT WINDOW CAN BE PANNED WITHIN ITS BORDERS
 - MUST BE AT THE BOTTOM TO SEE WHAT YOU ARE TYPING IN
- MAIN (LOGIN) WINDOW DOES NOT PAN OR SCROLL

INTRODUCTION TO THE DAISY UNDER DNIX

TREE STRUCTURE

- AS BEFORE BUT CURRENT CONTEXT IS . INSTEAD OF -
- FILENAMES ARE AS BEFORE
 - A-Z, 0-9 10 CHARACTER PRIMARY, 4 CHARACTER EXTENSION
 - AVOID SPECIAL CHARACTERS, BE MEANINGFUL
 - AVOID DAISY, TEGAS RESERVED WORDS (CONFUSING)
- {NEXT}, {PREVIOUS}, {CHANGE} KEYS ALL WORK AS BEFORE
- DESIGNS CAN BE FLAT, TREE HIERARCHY,
 - NESTED BLOCKS, CELLS

COMMANDS

MAKE A DIRECTORY

```
MKDIR Q700LIB
MKDIR BARREL8
MKDIR APNOTE1
MKDIR CLASSEX
```

INVENTORY THE DIRECTORY OR THOSE AROUND IT

```
INV .                CURRENT
INV . -DE            CURRENT, DEEP
INV . -S -L -DE     CURRENT, DEEP, SORTED, LONG
INV ..              (WAS INV +) ONE UP
```

INITIALIZE A DISK

```
INITDISK -name      WHERE name UP TO 19 CHARACTERS
```

INTRODUCTION TO THE DAISY UNDER DNIX

ERASE FILES, ETC.

ERASE 1.DRAW
ERASE *.BAK

CHECK YOUR CONTEXT FIRST!!!!

DO NOT DO THIS AT SYSTEM LEVEL!

ERASE *.* <---- ALL FILES
ERASE * <---- ALL FILES
THEN ALL DIRECTORIES

YOU GET ONE CHANCE TO SAVE YOURSELF BEFORE
FULL DESTRUCTION

ERASE CLASSEX <---- ERASES CONTENTS
OF DIRECTORY
CLASSEX
ERASE CLASSEX -0 <---- ERASES THE DIRECTORY
CLASSEX

CHANGE A DIRECTORY

CD .. UP TREE
CD Q700 DOWN TREE
CD /USER/CLASS GET WHERE YOU BELONG
CD GOES TO LOGIN DIRECTORY NEAT!

RENAME A FILE, ETC.

RENAME XXX.DRAW TO 3.DRAW
RENAME TEMP TO 1.DRAW
RENAME TEMP_SOM TO SOM_MCF.SING

- AFTER CAPTURE OF THE SCHEMATIC USING
DED OR DED2 ● ● ● ●

DANCE: DAISY NETWORK CONNECTIVITY EXTRACTOR

DANCE PREREQUISITES:

- USES .DRAW FILES PRODUCED UNDER DED OR DED2
- REQUIRES PROFILE, CONTENTS, GLOBAL NAMES,
PARAMETER AND NESTED REFERENCE FILE
(THE LATTER WITH -N OPTION ONLY)
- RECOMMEND USE OF DANCE CONFIGURATION FILE
- PRODUCES n.DFR, n.DIF

n.DRAW ---> DANCE ---> n.DFR
 :--> n.DIF

- SHELL PUTS ALL n.DFR INTO DANCE.ERR
- ALWAYS CHECK ERROR FILES!

INVOKE DANCE BY:

DANCE <drawing-page-name> [option]... {EXECUTE}
<design-path>

The default drawing page name is 1

- needed for P

The default design path is the current context

- needed for B or T

Input-scope:

- P page the default
- B block
- T tree <====

Input:

- U updated pages only <====
- N [<nest-file>] nested (with T input scope only)
 - If not specified, default is Nested Reference file in PROFILE file
- NC no conditional nesteds processed

Message-level:

- M0 lowest message level
- M3 normal setting for errors <===== NO

DEFAULT

Help-display:

- H help display of all legal DANCE syntax elements

DRINK: DAISY RESOLVING LINKER

DRINK PREREQUISITES:

- USES .DIF FILES PRODUCED FROM DANCE
- REQUIRES PROFILE, GLOBAL NAMES, PARAMETER, NESTED REFERENCE FILES
- RECOMMEND USE OF DRINK CONFIGURATION FILE
- PRODUCES TREE.DFR or INCR.DFR
- LINKS THE DRAWING PAGES INTO A SINGLE DESIGN
- RESOLVES INTERPAGE REFERENCES
- TAKES DATA FROM THE n.DIF FILES, RESOLVES THE EXTERNAL REFERENCES AND PRODUCES THE GLOBAL REFERENCES
- THE GLOBAL FILE IS (TREE.DNLK)
- GENERATES A REPORT FILE (TREE.DFR)
- DRINK CONFIGURATION FILE - LISTED IN THE PROFILE FILE
- THE BINARY FILES ARE NAMED 1.DIF, 2.DIF, ETC.
UNLESS LEXICAL MODE IS SET IN THE CONFIGURATION FILE

SPARC: SIMULATOR PARAMETER COMPILER

- BUILDS A GENERIC LIBRARY CONTAINING FUNCTIONAL DESCRIPTIONS OF THE SCHEMATIC COMPONENTS
- THE FILE CONTAINS TECHNOLOGY, TIMING AND FUNCTIONS
- AMCC LIBRARIES HAVE ALREADY BEEN COMPILED
- DESIGNS SUBMITTED TO AMCC MAY NOT CONTAIN ANY COMPONENTS THAT THE DESIGNER CREATED - ONLY AMCC RELEASED MACROS OR AMCC APPROVED PATCHES ARE ALLOWED

SPARC IS BEYOND THE SCOPE OF THE BEGINNER

THE ERCS

- THE AMCC MACROMATRIX SUPPORT SOFTWARE INCLUDES AN EXTENSIVE ENGINEERING RULES CHECKS (ERC) PROGRAM
- THE "ERC" PROGRAM DETECTS ERRORS, SUCH AS:
 - OVER-POPULATED ARRAYS
 - EXCESSIVE CURRENT
 - OVERLOADED MACROS
 - INCORRECT HOOK-UPS
 - INVALID TECHNOLOGY MIXES
 - PIN-CLASS ERRORS
 - INVALID LIBRARY
- THE ERC PROGRAM ISSUES REPORTS THAT ASSIST IN THE FINAL EVALUATION OF A DESIGN, SUCH AS:
 - POPULATION, INCLUDING EXTERNAL PIN COUNT
 - MACRO USAGE, MACRO OCCURRENCE AND POWER
 - FAN-OUT LOADING
- THE ERC PROGRAM SHOULD BE RUN PRIOR TO SIMULATION

- INVOKE THE ERC SHELL BY:

RUN_ERC

or AMCCERC

or choose menu option "1" under the super shell

GENERATES: AMCCERC.LST ERC report
 AMCCIO.LST I/O list
 AMCCXREF.LST cross reference

all files are in the ERC subdirectory

SIFT: SIMULATOR INTERMEDIATE FILES TRANSLATOR

- PROCESSES INFORMATION GENERATED IN DANCE, DRINK PLUS THE SPARC-GENERATED AMCC LIBRARIES
- PREPARES DATA FOR USE BY DLS/MDLS/DTV
- SIFT DETERMINES ON A PAGE BASIS THE RELATIONSHIP BETWEEN COMPONENTS ON A PAGE AND THE INFORMATION IN THE LIBRARY

SIFT PREREQUISITES:

- USES n.DIF FILES PRODUCED FROM DANCE
- USES TREE.DNLK FILE PRODUCED FROM DRINK
- REQUIRES PROFILE, SIFT CONFIGURATION FILE, SPARC LIBRARY FILE(S) {AMCC LIBRARY}, PARAMETER AND NESTED REFERENCE FILES
- PRODUCES n.SIF AND TREE.SIF FILES
- OPTIONALLY PRODUCES n.SFR AND TREE.SFR FILES (NOT USUALLY REQUIRED WHEN RUNNING AN AMCC LIBRARY)

SING: SIMULATION INPUT GENERATION

SING PREREQUISITES:

- USES THE BINARY FILES PRODUCED UNDER
DANCE AND DRINK
- USES A SMALL NUMBER OF USER-WRITTEN PROGRAMS
THAT EXTRACT AND FORMAT INFORMATION
- ALLOWS THE EXTRACTION OF INFORMATION FROM A
DRAWING SUCH AS COMPONENT NAMES AND ATTRIBUTES,
PAGE NUMBER, CONNECTIVITY, PARAMETER VALUES
- PRODUCES INPUT FILES FOR SIMULATORS AND OTHER
DESIGN AUTOMATION TOOLS

SING IS BEYOND THE SCOPE OF A BEGINNER

**

SING "TO DAISY" SHELL:

USES: SING -T -M3 -MCF /AMCC/SOM_MAKER/SOM_MCF

- INVOKE THE SOM_MAKER SHELLS, BY:
 RUN_SMAKER
- OR CHOOSE MENU OPTION "3" UNDER THE SUPER-SHELL

GENERATES: SOM_MAKER.ERR

SOM SOM_MCF.SING -M3

DLS <<1

FORMAT

CARYIN@S16BITADR/3:CARYIN

3DATA@S16BITADR/2:DATA@

3DATA1@S16BITADR/2:DATA1

3DATA1@S16BITADR/2:DATA1@

3DATA11@S16BITADR/2:DATA11

3DATA12@S16BITADR/3:DATA12

3DATA13@S16BITADR/3:DATA13

3DATA14@S16BITADR/3:DATA14

3DATA15@S16BITADR/3:DATA15

3DATA2@S16BITADR/2:DATA2

3DATA3@S16BITADR/2:DATA3

3DATA4@S16BITADR/2:DATA4

3DATA5@S16BITADR/2:DATA5

3DATA6@S16BITADR/2:DATA6

3DATA7@S16BITADR/2:DATA7

3DATA8@S16BITADR/2:DATA8

3DATA9@S16BITADR/2:DATA9

3DATB@S16BITADR/4:DATB@

3DATB1@S16BITADR/4:DATB1

3DATB1@S16BITADR/4:DATB1@

3DATB11@S16BITADR/4:DATB11

3DATB12@S16BITADR/3:DATB12

3DATB13@S16BITADR/3:DATB13

3DATB14@S16BITADR/3:DATB14

3DATB15@S16BITADR/3:DATB15

3DATB2@S16BITADR/4:DATB2

3DATB3@S16BITADR/4:DATB3

3DATB4@S16BITADR/4:DATB4

3DATB5@S16BITADR/4:DATB5

3DATB6@S16BITADR/4:DATB6

3DATB7@S16BITADR/4:DATB7

3DATB8@S16BITADR/4:DATB8

3DATB9@S16BITADR/4:DATB9

3EXTCLK@S16BITADR/1@:EXTCLK

3EXTRST@S16BITADR/1@:EXTRST

3MUXA@S16BITADR/2:MUXA

3MUXB@S16BITADR/4:MUXB

3CAROUT@S16BITADR/1@:CAROUT

3FZERO@S16BITADR/1@:FZERO

3NEXT@S16BITADR/6:NEXT@

3NEXT1@S16BITADR/6:NEXT1

3NEXT1@S16BITADR/8:NEXT1@

3NEXT11@S16BITADR/8:NEXT11

3NEXT12@S16BITADR/9:NEXT12

3NEXT13@S16BITADR/9:NEXT13

3NEXT14@S16BITADR/9:NEXT14

3NEXT15@S16BITADR/9:NEXT15

3NEXT2@S16BITADR/6:NEXT2

3NEXT3@S16BITADR/6:NEXT3

3NEXT4@S16BITADR/7:NEXT4

3NEXT5@S16BITADR/7:NEXT5

3NEXT6@S16BITADR/7:NEXT6

3NEXT7@S16BITADR/7:NEXT7

3NEXT8@S16BITADR/8:NEXT8

3NEXT9@S16BITADR/8:NEXT9

3SUM@S16BITADR/6:SUM@

3SUM1@S16BITADR/6:SUM1

3SUM1@S16BITADR/8:SUM1@

3SUM11@S16BITADR/8:SUM11

3SUM12@S16BITADR/9:SUM12

3SUM13@S16BITADR/9:SUM13

3SUM14@S16BITADR/9:SUM14

3SUM15@S16BITADR/9:SUM15

3SUM2@S16BITADR/6:SUM2

3SUM3@S16BITADR/6:SUM3

3SUM4@S16BITADR/7:SUM4

3SUM5@S16BITADR/7:SUM5

3SUM6@S16BITADR/7:SUM6

3SUM7@S16BITADR/7:SUM7

3SUM8@S16BITADR/8:SUM8

3SUM9@S16BITADR/8:SUM9

3S

PUT FMT

QUIT N

!

FMT CSD.SING

UNEDITED FILE

```

/****
*
* DESIGN PATH /USER/CLASS/S16BITADR   DATE 18 NOV 1990 16:34
*
* COMPANY _____   CIRCUIT_NAME _____
*
* ARRAY _____   PO# _____   REV _____
*
* DESIGNER _____
*
* What tests does this control file support: _____
*
* _____
*
* _____
*
****/

```

```

/**** Configuration section ****/

```

```

$CONFIGURATION
GATE_ACTIVITY_LEVEL := 100;
IMMEDIATE_ACTIVITY_LEVEL := 100;
TIMING_CHECK := 1;

```

```

/**** Signal generator section ****/

```

```

$SIGNAL_GENERATORS
@S16BITADR/3:CARYIN := @0:F0 ;
@S16BITADR/2:DATA0 := @0:F0 ;
@S16BITADR/2:DATA1 := @0:F0 ;
@S16BITADR/2:DATA10 := @0:F0 ;
@S16BITADR/2:DATA11 := @0:F0 ;
@S16BITADR/3:DATA12 := @0:F0 ;
@S16BITADR/3:DATA13 := @0:F0 ;
@S16BITADR/3:DATA14 := @0:F0 ;
@S16BITADR/3:DATA15 := @0:F0 ;
@S16BITADR/2:DATA2 := @0:F0 ;
@S16BITADR/2:DATA3 := @0:F0 ;
@S16BITADR/2:DATA4 := @0:F0 ;
@S16BITADR/2:DATA5 := @0:F0 ;
@S16BITADR/2:DATA6 := @0:F0 ;
@S16BITADR/2:DATA7 := @0:F0 ;
@S16BITADR/2:DATA8 := @0:F0 ;
@S16BITADR/2:DATA9 := @0:F0 ;
@S16BITADR/4:DATB0 := @0:F0 ;
@S16BITADR/4:DATB1 := @0:F0 ;
@S16BITADR/4:DATB10 := @0:F0 ;
@S16BITADR/4:DATB11 := @0:F0 ;
@S16BITADR/3:DATB12 := @0:F0 ;
@S16BITADR/3:DATB13 := @0:F0 ;
@S16BITADR/3:DATB14 := @0:F0 ;
@S16BITADR/3:DATB15 := @0:F0 ;
@S16BITADR/4:DATB2 := @0:F0 ;
@S16BITADR/4:DATB3 := @0:F0 ;
@S16BITADR/4:DATB4 := @0:F0 ;
@S16BITADR/4:DATB5 := @0:F0 ;
@S16BITADR/4:DATB6 := @0:F0 ;
@S16BITADR/4:DATB7 := @0:F0 ;
@S16BITADR/4:DATB8 := @0:F0 ;
@S16BITADR/4:DATB9 := @0:F0 ;
@S16BITADR/10:EXTCLK := @0:F0 ;
@S16BITADR/10:EXTRST := @0:F0 ;
@S16BITADR/2:MUXA := @0:F0 ;
@S16BITADR/4:MUXB := @0:F0 ;

```

SIGNAL GENERATOR EXAMPLES

```

SSIGNAL_GENERATORS
/*
/* INCLUDE ALL PRIMARY INPUTS EITHER IN THIS
/* SECTION OR IN AN "SIGNALS" SECTION, OR USE
/* SECTION FOR INPUT CAP ONLY BE
/* DEFINED IN ONE PLACE, THE OTHER AT ANY TIME
/*
/* TRANSPARENT LOW LATCH DRIVEN BY SAME CLOCK AS RISING-EDGE
/* ACTIVE CLOCK - THEREFORE, DATA CHANGES ON FALLING EDGE
*/
051681TADR/10:EXTCLK == 001F0, 110000:F0, 100000:F1*!; ← Overhaul!
/* FF13, FF#2 RESET BY EXTRST * 1 */
051681TADR/10:EXTRST == 001F0, 020000:F1, 040000:F0;
051681TADR/3:CARVIN == 001F0, 014000:F1, 020000:F0;
051681TADR/2:DATA0 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA1 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA2 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA3 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA10 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA11 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA12 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA13 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA14 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA15 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA2 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA3 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA4 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA5 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA6 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA7 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA8 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/2:DATA9 == 001F0, 060000:F1, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
/* MUX A SELECTS # WHEN S = 1, SELECTS LATCHED DATA BUS WHEN S = 0 */
051681TADR/2:MUXA == 001F0, 020000:F1, 030000:F0;
051681TADR/4:DATA0 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA1 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA2 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA3 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA10 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA11 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA12 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA13 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA14 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/3:DATA15 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA8 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA9 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA5 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA6 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA7 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA8 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
051681TADR/4:DATA9 == 001F0, 000000:F0, 010000:F0, 010000:F0, 010000:F0, 020000:F1, 020000:F0;
/* MUX B SELECTS SUM FEEDBACK WHEN S = 1, SELECTS LATCHED DATA BUS WHEN S = 0 */
051681TADR/4:MUXB == 001F0, 020000:F1, 030000:F0;

```

- SUBMIT SAMPLED SIMULATION OUTPUTS FOR FUNCTIONAL, AT-SPEED AND AC TEST SIMULATIONS

```

/* ----- */
/* OUTPUT FILE SECTION - YOU MUST ADD THIS          */
/* UNTIL AMCC CAN AUTOMATE ITS CREATION            */
/* - IT IS REQUIRED FOR AMCCSIMFMT                  */
/* ----- */

$OUTPUTS

/* PRINT_ON_CHANGE */
/* PUT THIS IN TO CHECK SKEW ON INPUTS */

/* ----- */
/* LIST THE FILE WHERE YOU WANT THE RESULTS        */
/* ----- */

FILE /USER/CLASS/S16BITADR/OUTPUT.LST <-

/* ----- */
/* INPUT SECTION LIST ALL PRIMARY INPUTS HERE    */
/* ----- */

@S16BITADR/1@:EXTCLK, EXTRST,
@S16BITADR/3:CARYIN,
@S16BITADR/3:DATA15, DATA14, DATA13, DATA12,
@S16BITADR/2:DATA11, DATA10, DATA9, DATA8, DATA7, DATA6,
@S16BITADR/2:DATA5, DATA4, DATA3, DATA2, DATA1, DATA0,
@S16BITADR/2:MUXA,
@S16BITADR/3:DATAB15, DATAB14, DATAB13, DATAB12,
@S16BITADR/4:DATAB11, DATAB10, DATAB9, DATAB8, DATAB7, DATAB6,
@S16BITADR/4:DATAB5, DATAB4, DATAB3, DATAB2, DATAB1, DATAB0,
@S16BITADR/4:MUXB,

/* ----- */
/* OUTPUT SECTION LIST ALL PRIMARY OUTPUTS HERE  */
/* ----- */

@S16BITADR/1@:FZERO, CAROUT,
@S16BITADR/9:SUM15, SUM14, SUM13, SUM12,
@S16BITADR/8:SUM11, SUM10, SUM9, SUM8,
@S16BITADR/7:SUM7, SUM6, SUM5, SUM4,
@S16BITADR/6:SUM3, SUM2, SUM1, SUM0,
@S16BITADR/9:NEXT15, NEXT14, NEXT13, NEXT12,
@S16BITADR/8:NEXT11, NEXT10, NEXT9, NEXT8,
@S16BITADR/7:NEXT7, NEXT6, NEXT5, NEXT4,
@S16BITADR/6:NEXT3, NEXT2, NEXT1, NEXT0;

/* ----- */
/* INTERNAL ENABLES LIST HERE (IF ANY)           */
/* ----- */

/* ----- */
/* INCLUDE AN "SEND" STATEMENT                   */
/* ----- */

SEND

```

- FOR THE SAMPLED OUTPUTS SECTION
DATA IS WRITTEN FOR EVERY "VIEW" (RECORDED) STEP BUT ONLY FOR THOSE SIGNALS LISTED IN THE \$OUTPUTS SECTION (LIST RECORDS ALL SIGNALS LISTED IN THE FMT FILE)

- DIFFERENT APPROACH: PRINT_ON_CHANGE

PRINT-ON-CHANGE

```
*****
$OUTPUTS
PRINT_ON_CHANGE
FILE /USER/CLASS/JOHN/DOE/OUT_DATA <-
@DOE/1:VAR1, VAR2, VAR3,
@DOE/2:SIG1, SIG2;
*****
```

- THE USE OR PRINT_ON_CHANGE CAUSES DATA TO BE WRITTEN ANYTIME THAT ONE OF THE MONITORED SIGNALS CHANGES VALUE
- ITEMS IN EITHER THE LIST MAY BE INPUTS, INTERNAL NETS THAT HAVE BEEN NAMED ON THE SCHEMATIC, OUTPUTS
- ITEMS NOT NAMED CAN BE PIECED IN - NOT A GOOD POLICY TO DO THIS - PLAN AHEAD!
- SUBMIT PRINT-ON-CHANGE FILES FOR AT-SPEED AND AC TEST SIMULATION

SAMPLE INPUT SECTION
AND DATA FILE

Date: 28 MAY 86 11:06 File: REMOTE.SING

```

/****
* DESIGN PATH /USER/CLASS/BARREL DATE 85-MAY-1986 14:34
* COMPANY _____ CIRCUIT_NAME _16 BIT BARREL
* ARRAY _01388T_ PO# _____ REV _____
* DESIGNER _____ DEV _____
* TIME _____
* FIELDS _____
* SIZES _____
* What tests does this control file support: _____
* _____ REMOTE DATA FILE _____
* _____
* _____
****/
/** Configuration section ****/
CONFIGURATION
GATE ACTIVITY LEVEL := 100;
IMMEDIATE ACTIVITY_LEVEL := 100;
TIMING_CHECK := 1;
/** Signal generator section ****/
SIGNALS
FILE /USER/CLASS/BARREL16/INPUT.DAT ->
0BARREL16/2:00 *
0BARREL16/2:01 *
0BARREL16/2:02 *
0BARREL16/2:03 *
0BARREL16/2:04 *
0BARREL16/2:05 *
0BARREL16/2:06 *
0BARREL16/2:07 *
0BARREL16/2:08 *
0BARREL16/2:09 *
0BARREL16/2:10 *
0BARREL16/2:11 *
0BARREL16/2:12 *
0BARREL16/2:13 *
0BARREL16/2:14 *
0BARREL16/2:15 *
0BARREL16/2:SEL1 *
0BARREL16/2:SEL2 *
0BARREL16/2:SEL3 *
0BARREL16/2:SEL4 *
SDATA HEADERS
STYPES
I/O
$FORMATS
TIME_VALUE
TOTAL_COLUMNS
8 20
SPACES
D B
FIELDS
TIME
FIELDS
VALUE
SENDS
0000000000 000000000000000000000000
00010000 100000000000000000000000
00020000 100000000000000000000000
00030000 100000000000000000000000
00040000 100000000000000000000000
00050000 100000000000000000000000
00060000 100000000000000000000000
00070000 100000000000000000000000
00080000 100000000000000000000000
00090000 100000000000000000000000
00100000 100000000000000000000000
00110000 100000000000000000000000
00120000 100000000000000000000000
00130000 100000000000000000000000
00140000 100000000000000000000000
00150000 100000000000000000000000
00160000 100000000000000000000000
00170000 011111111111111111111111
00180000 011111111111111111111111
00190000 011111111111111111111111
00200000 011111111111111111111111
00210000 011111111111111111111111
00220000 011111111111111111111111
00230000 011111111111111111111111
00240000 011111111111111111111111
00250000 011111111111111111111111
00260000 011111111111111111111111
00270000 011111111111111111111111
00280000 011111111111111111111111
00290000 011111111111111111111111
00300000 011111111111111111111111
00310000 011111111111111111111111
00320000 011111111111111111111111
00330000 011111111111111111111111
00340000 110110011100001100
00350000 110110011100001101
00360000 110110011100001110
00370000 110110011100001111

```

```

/****
*
* DESIGN PATH /USER/CLASS/S16BITADR   DATE 18 NOV 1990 12:38
* COMPANY  _AMCC_____   CIRCUIT_NAME  16 BIT ADDR
* ARRAY    Q1300S_____   PO#    _____   REV    _____
* DESIGNER  _____DEW_____
*
* What tests does this control file support:  _____
*
* _____FUNCTIONAL VECTOR GENERATION FOR THE _____
*
* _____ADDER - A CASE STUDY_____
****/

```

```

/*-----*/
/**** Configuration section *****/
/*-----*/

```

```

$CONFIGURATION
GATE_ACTIVITY_LEVEL := 100;
IMMEDIATE_ACTIVITY_LEVEL := 100;
TIMING_CHECK := 1;

```

```

/*-----*/
/* INPUT SECTION - YOU CREATE THIS */
/*-----*/

```

SINPUTS

```

/* TELL THE SYSTEM WHERE TO FIND THE DATA */

```

```

FILE /USER/CLASS/S16BITADR/INPUT.DAT ->

```

```

/*-----*/
/* INCLUDE ALL PRIMARY INPUTS EITHER IN THIS */
/* SECTION OR IN AN *$SIGNAL_GENERATORS SECTION */
/* OR USE BOTH SECTIONS - BUT AN INPUT CAN ONLY BE */
/* DEFINED IN ONE PLACE OR THE OTHER AT ANY TIME */
/*-----*/
/* LIST INPUTS IN THE ORDER THAT THEY ARE IN */
/* IN THE REMOTE DATA FILE */
/*-----*/

```

```
@S16BITADR/10:EXTCLK,
```

```
@S16BITADR/10:EXTRST,
```

```
@S16BITADR/3:CARYIN,
```

```

@S16BITADR/2:DATA0,
@S16BITADR/2:DATA1,
@S16BITADR/2:DATA2,
@S16BITADR/2:DATA3,
@S16BITADR/2:DATA4,
@S16BITADR/2:DATA5,
@S16BITADR/2:DATA6,
@S16BITADR/2:DATA7,
@S16BITADR/2:DATA8,
@S16BITADR/2:DATA9,

```

```
/*-----*/  
/* OUTPUT SECTION LIST ALL PRIMARY OUTPUTS HERE */  
/*-----*/
```

```
@S16BITADR/10:FZERO, CAROUT,
```

```
@S16BITADR/9:SUM15, SUM14, SUM13, SUM12,
```

```
@S16BITADR/8:SUM11, SUM10, SUM9, SUM8,
```

```
@S16BITADR/7:SUM7, SUM6, SUM5, SUM4,
```

```
@S16BITADR/6:SUM3, SUM2, SUM1, SUM0,
```

```
@S16BITADR/9:NEXT15, NEXT14, NEXT13, NEXT12,
```

```
@S16BITADR/8:NEXT11, NEXT10, NEXT9, NEXT8,
```

```
@S16BITADR/7:NEXT7, NEXT6, NEXT5, NEXT4,
```

```
@S16BITADR/6:NEXT3, NEXT2, NEXT1, NEXT0;
```

```
/*-----*/  
/* INTERNAL ENABLES LIST HERE (IF ANY) */  
/*-----*/
```

```
/*-----*/  
/* INCLUDE AN "SEND" STATEMENT */  
/*-----*/
```

SEND

```

**
** DESIGN PATH /USER/DICKS/ATOMI DATE 26 JUN 1986 11:05
** COMPANY LITTON ATD _____ CIRCUIT_NAME ATOMI _____
** ARRAY 03500 _____ PO# _____ REV _____
** DESIGNER CHUCK MARLEY/DICK SPEHN _____
** What tests does this control file support: ALL _____
**
**
**

```

```

**** Configuration section ****/
$CONFIGURATION
GATE_ACTIVITY_LEVEL := 100;
IMMEDIATE_ACTIVITY_LEVEL := 100;
TIMING_CHECK := 1;

**** Signal generator section ****/
$SIGNAL_GENERATORS
FILE /USER/DICKS/ATOMI/NEVERSE.LIS ->

```

```

ATOMI/8:RESETN;
ATOMI/8:B16MHZ;
ATOMI/12:ASN;
ATOMI/13:DSN;
ATOMI/13:IRV;
ATOMI/7:FC2;
ATOMI/12:FC1;
ATOMI/12:IFCB;
ATOMI/12:PAD21;
ATOMI/12:PAD20;
ATOMI/7:PAD19;
ATOMI/7:PAD18;
ATOMI/7:PAD17;
ATOMI/7:PAD16;
ATOMI/16:PAD08;

```

Inputs
Section -
with Bidirectional

```

ATOMI/3:ATBD02::CON;
ATOMI/3:ATBD01::CON;
ATOMI/3:ATBD00::CON;

/* Timing section: */
/* Print on change output file section */
$OUTPUTS
/* PRINT ON CHANGE */
FILE /USER/DICKS/ATOMI/FUN_POC.OUT <-
/* INPUT SECTION */

```

```

ATOMI/2:MDEAN;
ATOMI/3:MDEBN;
ATOMI/4:MDEPN;
ATOMI/5:DMUXA;
ATOMI/5:DMUXB;
ATOMI/5:TRID15::CON;
ATOMI/5:TRID14::CON;
ATOMI/5:TRID13::CON;
ATOMI/5:TRID12::CON;
ATOMI/5:TRID11::CON;
ATOMI/5:TRID10::CON;
ATOMI/5:TRID09::CON;
ATOMI/5:TRID08::CON;
ATOMI/5:TRID07::CON;
ATOMI/5:TRID06::CON;
ATOMI/5:TRID05::CON;
ATOMI/5:TRID04::CON;
ATOMI/5:TRID03::CON;
ATOMI/5:TRID02::CON;
ATOMI/5:TRID01::CON;
ATOMI/4:PD31::CON;
ATOMI/4:PD30::CON;
ATOMI/4:PD29::CON;
ATOMI/4:PD28::CON;
ATOMI/4:PD27::CON;
ATOMI/4:PD26::CON;
ATOMI/4:PD25::CON;
ATOMI/4:PD24::CON;
ATOMI/4:PD23::CON;
ATOMI/4:PD22::CON;
ATOMI/4:PD21::CON;
ATOMI/4:PD20::CON;
ATOMI/4:PD19::CON;
ATOMI/4:PD18::CON;
ATOMI/4:PD17::CON;
ATOMI/4:PD16::CON;
ATOMI/2:ATAD15::CON;
ATOMI/2:ATAD14::CON;
ATOMI/2:ATAD13::CON;
ATOMI/2:ATAD12::CON;
ATOMI/2:ATAD11::CON;
ATOMI/2:ATAD10::CON;
ATOMI/2:ATAD09::CON;
ATOMI/2:ATAD08::CON;
ATOMI/2:ATAD07::CON;
ATOMI/2:ATAD06::CON;
ATOMI/2:ATAD05::CON;

```

```

ATOMI/8:RESETN;
ATOMI/8:B16MHZ;
ATOMI/12:ASN;
ATOMI/13:DSN;
ATOMI/13:IRV;
ATOMI/7:FC2;
ATOMI/12:FC1;
ATOMI/12:IFCB;
ATOMI/12:PAD21;
ATOMI/12:PAD20;
ATOMI/7:PAD19;
ATOMI/7:PAD18;
ATOMI/7:PAD17;
ATOMI/7:PAD16;
ATOMI/16:PAD08;

```

INVOKE BY:

SOM control-file [option]... {EXECUTE}

control-file is the name of the SOM control file as defined by the user and this MUST be supplied (no default to SOM_MCF.SING)

NORMAL INVOCATION:

```
SOM SOM_MCF          USES SOM_MCF IN CURRENT CONTEXT
                    AS CONTROL FILE
SOM SOM_MCF.SING -M3 -L SOM.ERR
                    SAME AND DISPLAYS CONTROL FILE
                    AND ERROR MESSAGES; CREATES
                    AN ERROR FILE LISTING THE
                    CONTROL FILE AND POINTING AT THE
                    SUSPECTED ERROR(S)
```

SEE LOGICIAN DESIGN COMPILATION SECTION 7.6
FOR SOM ERROR MESSAGES - TBS

SOM - TCAL SHELL:

SOM <control file name> -M3 -L <error file name>

● INVOKE THE SOM - TCAL SHELL BY:

RUN_SOM <som-ctl-filename> <tcaldelay-filename>

● OR CHOOSE MENU OPTION "4" UNDER THE SUPER-SHELL

GENERATES: SOM.ERR
 TCAL.ERR

DLS

THE SAME UNDER MAESTRO OR DNIX

DTV

THE SAME UNDER MAESTRO AND DNIX

For both:

- USE TEC TO CREATE A SHELL SCRIPT TO KEEP TRACK OF STEPS
- SHELL SCRIPT IS A REQUIRED PART OF THE DESIGN SUBMISSION PACKAGE
- EDIT THE MODE SCREEN FOR THE PROPER MULTIPLIER USE MAX FOR MILITARY AND COMMERCIAL WORST-CASE MAXIMUM AND USE MIN FOR THE NOMINAL OR MINIMUM LIBRARY WORST-CASE MINIMUM.

```

SAMPLE RUN_DLS SHELL:  (USER-ENTERED)
TEC RUN_DLS
RUN_SOM
DLS <<!
GET DLS_FMT          <--- EDITED ON A PREVIOUS PASS
VIEW 9999 10000      (MODE AND FORMAT)
RUN 1000000
START 0
LIST S L1
{ENTER}
QUIT
N
I

```

- CALL ABOVE BY TYPING: RUN_DLS

Note: For Bipolar arrays, BiCMOS arrays, the VIEW step command for DLS for Functional and sampled AC Test simulations should be:

```
VIEW 9999 10000
```

to start sampling at 9999 simulator steps (99.99 ns) and sample every 10000 simulator steps (100ns).

AT-SPEED simulations will sample at a different time (based on the maximum frequency of operation).

The DEFAULT MODE window on the DAISY - THIS MUST BE EDITED

SIMULATION MODE	NOM	ENABLE	TRC?	MAJOR KEY	MINOR KEY
(reserved)					
(reserved)					
(reserved)					
SETUP/HOLD TIME		OFF	OFF	TIME	PATH
MINIMUM PULSE WIDTH		OFF	OFF	TIME	PATH
SIGNAL RELATIONSHIP		OFF	OFF	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH

The EDITED MODE window for MINIMUM worst-case multiplier
 within the timing library selected:
 - USE WITH THE MINIMUM SIFT LIBRARY

SIMULATION MODE	MIN	ENABLE	TRC?	MAJOR KEY	MINOR KEY
(reserved)					
(reserved)					
(reserved)					
SETUP/HOLD TIME		ON	ON	TIME	PATH
MINIMUM PULSE WIDTH		ON	ON	TIME	PATH
SIGNAL RELATIONSHIP		ON	ON	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH
(reserved)		OFF	OFF	TIME	PATH

● AFTER SIMULATION ---

AMCCSIMFMT - AMCC SIMULATION FILE FORMATTER

THE SOM_MCF.SING FILE (WHICH YOU RENAMED)
GENERATES A VLAF OUTPUT FILE (REQUIRED)

THE VLAF OUTPUT FILE IS PROCESSED THROUGH
AMCCSIMFMT TO PRODUCE A FORMATTED FILE THAT
THE AMCC TEST SOFTWARE WILL USE AS DATA
(FUNCTIONAL, AC AND PARAMETRIC)

- AT-SPEED FILES WILL NOT BE USED AS INPUT
TO A TESTER BUT MUST BE SUBMITTED IN THE
SAME FORMAT

AMCCSIMFMT CAN PROCESS SAMPLED OR PRINT_ON_CHANGE
FILES

ALL FILES ARE BINARY

● INVOKE AMCCSIMFMT BY:

AMCCSIMFMT

OR CHOOSE MENU OPTION "5" UNDER THE SUPER-SHELL

GENERATES: USER-NAMED REFORMATTED FILE
AMCCSIMFMT.ERR
