Section 1:

Introduction

INTRODUCTION

This design manual provides a summary of the Applied Micro Circuits Corporation (AMCC) Design Methodology for the DAISY Engineering workstation under the DNIX operating system. Volume 1 of this design manual is concerned with the specific design methodology for a specific array series.

Volume 2 of this design manual is composed of the following sections:

- Section 1: Introduction
- Section 2: EWS-Specific Design Methodology
- Section 3: EWS Schematic Rules and Conventions
- Section 4: Vector Submission Rules and Guidelines
- Section 5: Design Validation
- Section 6: Design Submission
- Section 7: MacroMatrix^{R} Installation
  EWS-specific rules
- Section 8: MacroMatrix^{R} User's Guide
  AMCCERC manual
  AMCCVRC manual
  AMCCANN manual
- Section 9: AMCC Glossary
- Section 10: Index

Volume 2, Section 2 of this design manual contains the Engineering-workstation (EWS) design methodology, covering both the EWS-specific operations and the AMCC MacroMatrix^{R} support software. Section 8 contains the MacroMatrix^{R} User's Guide which details the ERC and VRC checks and error messages and probable causes. Section 9 contains the MacroMatrix^{R} Installation and Operations manual, which summarizes the EWS-specific commands required for operation of the AMCC support software from schematic capture through Back-Annotation.

The Design Validation document in Section 5 details the engineering rules checks that must be reviewed prior to design submission. It is the basic outline of the design review AMCC performs prior to circuit acceptance. Fill in or check off items as indicated and submit the entire document as part of the design submission package. Additional copies can be obtained from AMCC in an 8.5x11" format.

The Design Submission Document in Section 6 is to be completed and submitted along with the design submission package. Additional copies can be obtained from AMCC. The document is now generic (not specific to any workstation) with the exception of one DAISY section. AC test submission and design submission in general will be automated through a user-interface in the near future.

The following trademarks are recognized by AMCC throughout this and other design manuals:

- TEGAS V - General Electric Co.
- COPTR - General Electric Co.
- LOGICIAN - Daisy Systems Corp.
- GATEMASTER - Daisy Systems Corp.
- Mentor Graphics
- Macromatrix - AMCC
- VALID - Valid Logic Systems
- LASAR Version 6 - Teradyne, Inc.

DESIGN SUPPORT INTERFACE

Volume 1, Section 1 introduced an overview of the design interface and support offered by AMCC. Figure 1-1 on page 1-7 shows a more detailed examination of the design steps.

Schematic entry can be performed by either the customer or AMCC. Following schematic entry, the netlist must be generated and then converted into AGIF (AMCC Interface Format). The AMCC Engineering Rules Checks software (AMCCERC) are run and debugged prior to preceeding with simulation. Performing AMCCERC early in the design cycle clears the trivial errors, identifies loading problems and generally allows circuit clean-up before time and effort are expended on the simulations that will need to be redone. AMCCERC checks are described in Section 8.

At the same time, Front-Annotation software (AMCCANN) can be executed, using the AGIF file, generating the delay files for later use.

Another step is testability analysis, where the controlability and the observability of the actual design is measured. Testability analysis can be run early in the design cycle, the sooner the better, and the feedback used to modify the circuit.

The designer must develop the functional and at-speed test vectors. Either the customer or AMCC can perform the simulations. When the simulations have been completed for design submission, following the directions in Section 4, Vector Submission Rules and Guidelines, fault grading needs to be performed on the functional vectors. The maximum worst-case sampled functional and AC test simulation output files, in AMCCSIMFMT, must also be processed through the AMCC Vector Rules Checker, AMCCVRC. AMCCVRC is described in Section 8, Appendix B. AMCCSIMFMT is EWS-specific and is described in Section 7.

Fault grading will determine if the functional simulation covers an adequate percentage of the possible single, stuck-at faults. A circuit that has been developed to ensure testability will be easier to test than one which has buried states, deep, uncontrollable nodes and redundancy.

The AGIF file, CIRCUIT.SDI, is part of the design submission and the generation of the AGIF file, the ERCs and Front-Annotation should reflect the actual submitted circuit.

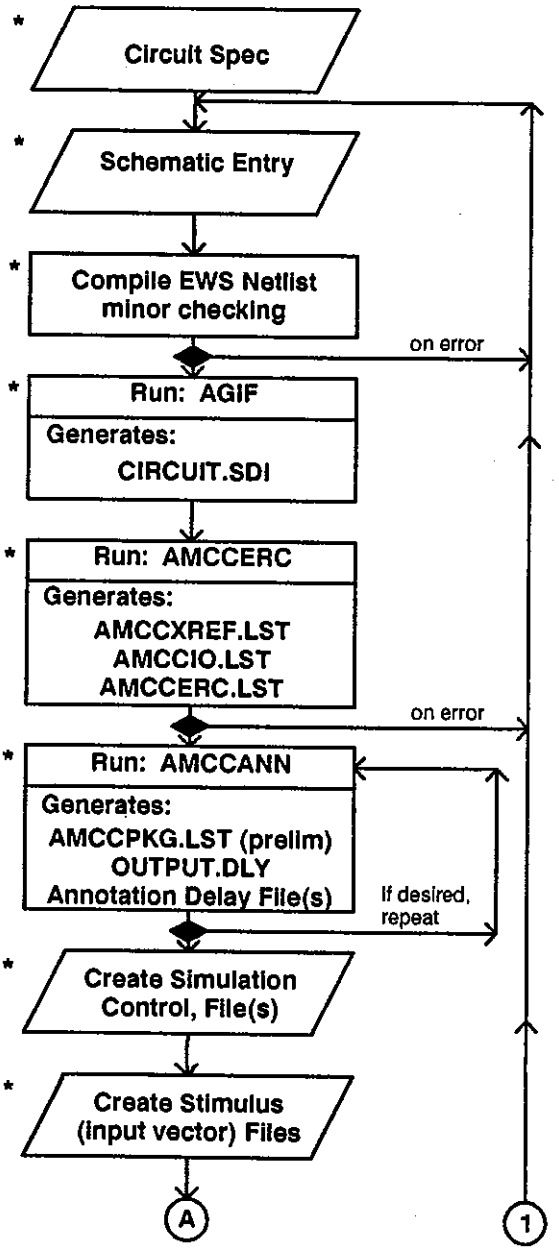
All of the above steps, with the exception of the actual vector generation, are steps that either the customer or AMCC implementation can perform, depending on the contract arrangements.

Following submission, AMCC Implementation will rerun AMCCERC and Front-Annotation using the in-house library. As part of the acceptance review, the at-speed simulation will be re-executed at AMCC.

On approval, the circuit will pass to layout using the AMCC Computer-Aided Design system, AMCCAD. On successful layout, the Back-Annotation actual delay files will be available. The re-execution of the at-speed simulation is performed using the Back-Annotation delay files in place of the Front-Annotation delay files. This can be performed by the customer or AMCC. The approved Back-Annotation at-speed simulation results are considered to be the specification for the circuit.

Finally, on design approval, prototype fabrication can begin.

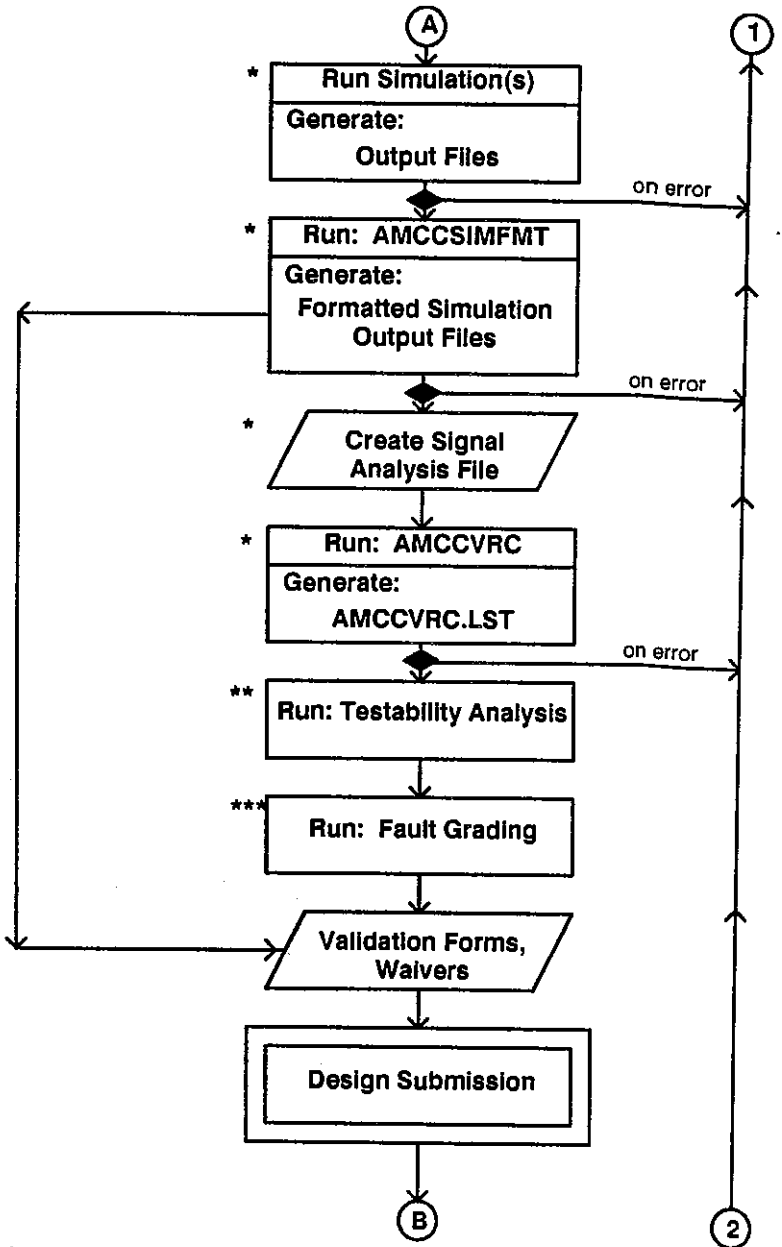
EWS DESIGN FLOW



*AMCC can be contracted to perform these steps.

FIGURE 1-1

EWS DESIGN FLOW (Continued)



* AMCC can be contracted to perform these steps.
** AMCC does not support this function at this time.
*** LASAR 6 only.

FIGURE 1-1

EWS DESIGN FLOW (Continued)

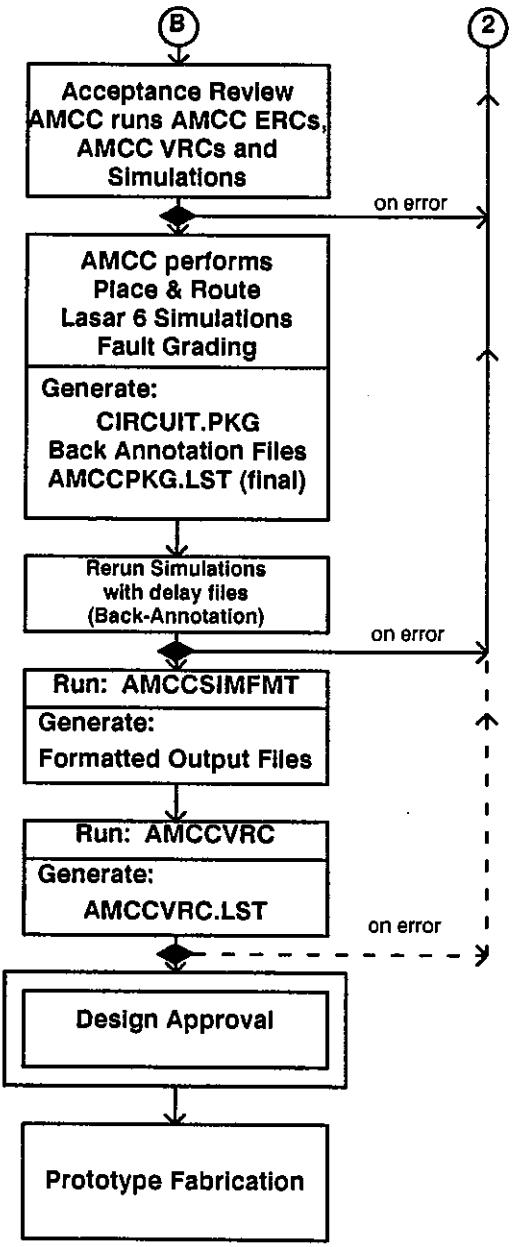


FIGURE 1-1

Section 2:

EWS Methodology

Table of Contents.....2-2

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
Usual name for simulation control file
SPARC 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
T0.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"

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)
  - CLOSES WINDOW WHEN DONE
  - USE OF "NW" ALLOWS DIFFERENT WINDOW SIZES TO BE SPECIFIED

- 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

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
```

MOUNT A F

```
MOUNT /F
```

COPY FILES

```
COPY /F/TCAL_MIL.MCF TO .
COPY SOM_MCF.SING TO /NET/D_T/USER/DEW <--- NOTE
COPY SOM_MCF.SING TO /F/FULL_MCF.SING
COPY . TO /F -B
COPY *.DRAW TO /F/Q700
COPY /F/Q700/*.DRAW TO . -B
COPY 1.DRAW TO 2.DRAW
```

DISMOUNT A DISK

```
DIS
```

TYPE A FILE

```
TYPE SOM_MCF.SING
```

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
```

MOUSE OFF

- DON'T!
  - SOME OPERATIONS REQUIRE THE MOUSE
    (D_T, D_I WILL NEED MICE)
- SHIFT-WRAP WILL TOGGLE THE CURSOR-ACTIVE/INACTIVE

CALL DED

DED 1

DED 15

DED2 1 NEW DED (NOT COVERED HERE)
- SEE BOOK

- CAN USE DED AND DED 2 BACK AND FORTH ON PAGES

DED OR DED1 - THE OLD GRAPHICS EDITOR

- WORKS THE SAME AS BEFORE

- 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!

- CHECKS DRAWING PAGES FOR BASIC DESIGN ERRORS
  - UNUSED PINS
  - DUPLICATE NAMES
  - DUPLICATE OR MISSING PARAMETERS
    (DAISY PARAMETERS)
  - MULTIPLE DRIVES (TWO OR MORE PAGE CONNECTORS HAVE THE SAME NAME)
  - NO DRIVES FOR SIGNALS
    (PAGE CONNECTOR - NO HIERARCHY CONNECTOR)
- COMPILES DATA FROM DRAWING PAGES (n.DRAW FILE) FOR TRANSLATION TO THE INTERMEDIATE FILE (n.DIF FILE) - A BINARY FILE
- GENERATES A REPORT FILE (n.DFR)

- DANCE CONFIGURATION FILE - LISTED IN THE PROFILE FILE

- THE DRAWING PAGES ARE NAMED 1.DRAW, 2.DRAW, ETC. UNLESS LEXICAL MODE IS SET IN THE CONFIGURATION FILE

```
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
```

EXAMPLES

```
DANCE                 PAGE 1.DRAW ONLY
DANCE -M3 -T          ALL PAGES, INDIVIDUAL REPORTS, TREE
                        IN CURRENT CONTEXT
DANCE -M3 -T -ERR     ALL PAGES, CONCATENATED ERR REPORTS
DANCE /USER/CLASS/n   PAGE n ONLY
DANCE -M3 -T -N       ALL PAGES, NESTED MODE, TREE IN
                        CURRENT CONTEXT
```

SEE LOGICIAN DESIGN COMPILATION SECTION 3.5
FOR DANCE ERROR MESSAGES

****
SUMMARY
****

- DANCE -T -N -M3 -E3

  GENERATES n.DFR, n = 1, 2, 3, 4...

- INVOKE THE DANCE-DRINK SHELL BY:
  RUN_DD

- CHOOSE MENU OPTION "1" UNDER THE SUPER-SHELL

BOTH GENERATE: DANCE.ERR
DRINK.ERR

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

```
INVOKE BY:

DRINK <full-link-mode> [option]... {EXECUTE}
      <update-link-mode>

link-file:   default is l.DIF

update-path:  default is top of tree

report:     TREE.DFR for a full link
            INCR.DFR for an update link

Help display:
       -H      provides chart of syntax elements

Support files - use to override the PROFILE file:

        C <configuration-file>
        N <nested-file>
        GLOBAL <global-file>
        PARAM <parameter-parameter>

NORMAL INVOCATION:

     DRINK            ALL PAGES   <====
     DRINK  -M3 -E3   ALL PAGES
                    - MESSAGES REPORTED TO SCREEN
     DRINK  -U        UPDATE LINK ON TREE IN
                       CURRENT CONTEXT

SEE LOGICIAN DESIGN COMPILATION SECTION 4.3
FOR DRINK ERROR MESSAGES

**************************************************

• USE  DRINK -T -M3 -E3
       GENERATES TREE.DFR

• INVOKE THE DANCE-DRINK SHELL BY:
            RUN_DD

• CHOOSE MENU OPTION "1" UNDER THE SUPER-SHELL

GENERATES:   DANCE.ERR
             DRINK.ERR
```

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

RUN AGIF - THE AMCC INTERFACE FORMAT

- BEFORE PROCEEDING WITH THE REST OF THE STEPS USING DAISY SOFTWARE, THE AMCC MACROMATRIX SOFTWARE TO PRODUCE THE AMCC GENERIC INTERFACE FORMAT FILE SHOULD BE RUN

- THE NEXT STEP IS TO RUN AND SUCCESSFULLY PASS THE AMCC ENGINEERING REPORTS AND CHECKS SOFTWARE OR ERCs

- FAILURE IN THE ERCs REQUIRES A RE-ENTRY INTO THE GRAPHICS EDITOR (DED1 OR DED2), THEN AN INCREMENTAL DANCE AND DRINK AND ANOTHER ERC EXECUTION

- ONCE THE ERCs ARE SUCCESSFUL, THE TIMING CALCULATION FRONT-ANNOTATION FILE SHOULD BE GENERATED

****

- INVOKE THE AGIF SHELL BY:

- DIRECT CALL:

  RUN_AGIF

- CHOOSE MENU OPTION "1" UNDER THE SUPER SHELL

GENERATES: RUN_AGIF.ERR
CIRCUIT.SDI
misc. files

****

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
```

****

FRONT-ANNOTATION
BACK-ANNOTATION

- THE AMCC MACROMATRIX SUPPORT SOFTWARE INCLUDES AN ANNOTATION PROGRAM THAT ALLOWS THE SIMULATION TO BE PERFORMED WITH LOADING DELAYS INCLUDED.

- THE FRONT-ANNOTATION DELAY FILE INCLUDES THE LOADING DELAYS ON A NET AS FOLLOWS:
  - ACTUAL FAN-OUT LOAD DELAY
  - ACTUAL WIRE-OR LOAD DELAY
  - STATISTICAL ESTIMATE OF THE METAL LOAD DELAY BASED ON THE NET SIZE
  - ACTUAL OUTPUT CAPACITIVE LOAD DELAY

- AFTER LAYOUT, THE BACK-ANNOTATION SOFTWARE WILL PROVIDE A BACK-ANNOTATION DELAY FILE THAT WOULD BE SUBSTITUTED FOR THE FRONT-ANNOTATION DELAY FILE .

****

- INVOKE THE ANNOTATION SHELL BY:

  RUN_ANN
  or AMCCANN
  or choose menu option "2" under the super shell

GENERATES: RUN_ANN.ERR

PRODUCES: FNTMIN.DSY the delay files
FNTMIL.DSY OR FNTCOM.DSY
(FNTNOM.DSY for your own use)
AMCCPKG.LST REPORT FILE
OUTPUT.DLY (DATA FILE)

****

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)

INVOKE BY:

SIFT [design-path] [option]... {EXECUTE}

NORMAL INVOCATION:

SIFT	PROCESSES FILES FOR TREE IN CURRENT CONTEXT
SIFT -M3	SAME BUT ADDS MESSAGES
SIFT -M3 -L -R	SAME BUT ALSO DISPLAYS CONFIGURATION FILE AND GENERATES TREE.SFR AND n.SFR FILES

SEE LOGICIAN DESIGN COMPILATION SECTION 6.9
FOR SIFT ERROR MESSAGES - TBS

****

SIFT SHELL USES:

SIFT -M3 -LIB $FAMILY/PATCH$1.SLIB .....

- INVOKE THE SIFT SHELL BY:
  RUN_SIFT [option] option = MIN
  NOM
  COM BIPOLAR
  COM4 OR COM5 BICMOS
  MIL
  THE OPTION IS REQUIRED

- OR CHOOSE MENU OPTION "3" UNDER THE SUPER-SHELL
  and follow the prompt (RECOMMENDED)

GENERATES: SIFT.ERR

****

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
****

OPTIONAL STEP

FMT_CSD.SING: FORMAT CONTROL FILE

- THIS IS THE FILE FROM WHICH THE DLS/DTV DEFAULT FORMAT IS CREATED - controls WAVE and LIST formats

---

- THIS FILE MAY BE EDITED USING TEC
- MOVE SIGNALS, ADD SIGNALS (ANY INTERNAL NET NAMED ON THE DRAWING -

- ADD SIGNALS BY DUPLICATING AN EXISTING LINE - THERE ARE SPECIAL CHARACTERS YOU CANNOT SEE OR DECIPHER
- USE TEC AND EXIT WHEN FINISHED
- THEN USE THE COMMAND:

  FMT_CSD.SING

  TO RE-CREATE THE FMT FILE USING THE NEW VERSION OF THE FMT_CSD.SING FILE

- THIS FILE IS A FOR-YOUR-OWN REFERENCE

- NOTE: FMT IS NO LONGER OF INTEREST TO AMCC

- EXIT THE SUPER SHELL BEFORE EDITING FMT_CSD.SING

- NOTE: YOU MUST RUN BOTH THE FMT_CSD.SING COMMAND AND STEP 3 (RUN_SOM) IF YOU EDIT FMT_CSD.SING

```
SOM SOM_MCF.SING -M3
DLS <<!
FORMAT
CARYIN@S16BITADR/3:CARYIN
3DATAØ@S16BITAOR/2:DATAØ
3DATA1@S16BITADR/2:DATA1
3DATA1Ø@S16BITADR/2:DATA1Ø
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
3DATBØ@S16BITADR/4:DATBØ
3DATB1@S16BITADR/4:DATB1
3DATB1Ø@S16BITADR/4:DATB1Ø
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/1Ø:EXTCLK
3EXTRST@S16BITADR/1Ø:EXTRST
3MUXA@S16BITADR/2:MUXA
3MUXB@S16BITADR/4:MUXB
3CAROUT@S16BITADR/1Ø:CAROUT
3FZERO@S16BITADR/1Ø:FZERO
3NEXTØ@S16BITADR/6:NEXTØ
3NEXT1@S16BITADR/6:NEXT1
3NEXT1Ø@S16BITADR/8:NEXT1Ø
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
3SUMØ@S16BITADR/6:SUMØ
3SUM1@S16BITADR/6:SUM1
3SUM1Ø@S16BITADR/8:SUM1Ø
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

SOM CONTROL FILE

- PROVIDES SIGNAL STIMULUS
- PROVIDES OTHER INFORMATION FOR USE BY DTV/DLS/MDLS
- UP TO 9 SECTIONS MAY BE DEFINED:

```
$CAPACITANCE
$CONFIGURATION          * USE DEFAULT
$DATA                   * NOT FOR FUNCTIONAL, AC TEST
                          PARAMETRICS
$INITIALIZE             * NOT FOR FUNCTIONAL, AC TEST
                          PARAMETRICS
$INPUTS                 * FOR DATA FILE
$OUTPUTS                * FOR PRINT_ON_CHANGE
                        * FOR TEGAS TESTPATT FORMAT
                        * FOR OUTPUT FILE DESCRIPTION
$PMX_INFO
$SIGNAL_GENERATORS      * DEFAULT FOR TIME = 0
                        * USE IF SIGNAL NOT IN DATA
                          FILE
$TIMING
```

NOT USED:

```
CAPACITANCE
DATA
INITIALIZE
PMX_INFO
TIMING
```

- EITHER INPUTS OR SIGNAL_GENERATORS MUST BE USED

THE $END MUST BE AT THE END OF THE SOM CONTROL FILE

THE $CONFIGURATION SECTION

- SPECIFY VALUES USED BY THE SIMULATION PROGRAMS
- DEFAULT VALUES NORMALLY UNCHANGED

THE $DATA SECTION

- FOR ROMS, RAMS, PLAS - at-speed simulation only!
- DO NOT USE FOR FUNCTIONAL, AC TEST OR PARAMETRIC SIMULATIONS

THE $INITIALIZE SECTION

- SPECIFY INITIAL VALUES FOR A SIGNAL OR GROUP OF SIGNALS - WHEN IMPOSSIBLE TO DO OTHERWISE
- AMCC REQUIRES THAT A FUNCTIONAL SIMULATION BE INITIALIZED BY VECTORS. APPLIES TO AC TEST AND PARAMETRIC TESTS.

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:

```
  @CLASS/1.SIGNAL := @0:F0;  AT TIME=ZERO,FORCE TO ZERO

@CLASS/1.EXTCLK := @0:F0, [10000:F0, 10000:F1];
                                  100ns CLOCK

  [10:F1,10:F0]**;  INDEFNITE REPEAT
                    RELATIVE - 10 STEPS EACH

  @1000:F1;         ABSOLUTE TIME
                    CANNOT USE IN A REPEAT STEP

  [1000:F1,1000:F0]*20;    20 CYCLES OF 10ns HIGH,
                           10ns LOW, THEN HOLD
                              (Q5000 scale)

 [10000:F1,10000:F0]*100, @2005000:F1;
 absolute must not be less than time passed
```

- RUN SMT TO CONVERT A MAESTRO SOM MCF.SING FILE TO A DNIX SOM_MCF.NEW FILE

• • • BE AT THE SAME TREE NODE!

SIGNAL GENERATOR EXAMPLES

```
$SIGNAL_GENERATORS

/* ---------------------------------------------- */
/* INCLUDE ALL PRIMARY INPUTS EITHER IN THIS      */
/* SECTION OR IN AN "$INPUTS" SECTION, OR USE     */
/* BOTH SECTIONS - BUT AN INPUT CAN ONLY BE       */
/* DEFINED IN ONE PLACE OR THE OTHER AT ANY TIME  */
/* ---------------------------------------------- */

/* TRANSPARENT LOW LATCH DRIVEN BY SAME CLOCK AS RISING-EDGE  */
/* ACTIVE CLOCK    - THEREFORE, DATA CHANGES ON FALLING EDGE  */
/* ---------------------------------------------------------- */

@S16BITADR/10:EXTCLK := 00:F0, [10000:F0, 10000:F1]**;  <- careful!

/* FF13, FFF02 RESET BY EXTRST = 1   */

@S16BITADR/10:EXTRST := 00:F0, @20000:F1, @40000:F0;

@S16BITADR/3:CARYIN := 00:F0, @140000:F1, @220000:F0    ;

@S16BITADR/2:DATA0  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA1  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA10 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA11 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/3:DATA12 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/3:DATA13 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/3:DATA14 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/3:DATA15 := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA2  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA3  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA4  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA5  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;

@S16BITADR/2:DATA6  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA7  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA8  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;
@S16BITADR/2:DATA9  := 00:F0,@60000:F1, @100000:F0, @140000:F1, @180000:F0, @220000:F1, @260000:F0 ;

/* MUX A SELECTS 0 WHEN S = 1, SELECTS LATCHED DATA BUS WHEN S = 0  */

@S16BITADR/2:MUXA := 00:F0, @220000:F1, @300000:F0 ;

[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;
[redacted] := 00:F0, @80000:F1, @120000:F0, [redacted], @240000:F1, @280000:F0 ;

/* MUX B SELECTS SUM FEEDBACK WHEN S = 1, SELECTS LATCHED DATA BUS WHEN S = 0 */

@S16BITADR/4:MUXB := 00:F0, @220000:F1, @300000:F0 ;
```

THE $OUTPUTS SECTION - MANDATORY

- TO USE AMCCSIMFMT, THE OUTPUTS SECTION IS MANDATORY
  NOTE: THIS SECTION IS NOW REQUIRED IN LIEU OF THE FMT FILE

- ALL PRIMARY INPUTS, ALL PRIMARY OUTPUTS, AND ALL 3-STATE AND BIDIRECTIONAL ENABLE SIGNALS MUST BE LISTED IN THE OUTPUT FILE FOR FUNCTIONAL AND AC TEST VECTORS - USE SAME FORMAT FOR AT-SPEED

- FAILURE TO INCLUDE ALL REQUIRED SIGNALS CAN RESULT IN DESIGN SUBMISSION DELAYS

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

- THE 5 SIGNALS ARE WRITTEN IN FIRST ON LEFT, ETC. ORDER IN THE FILE NAMED OUT_DATA IN THE PATH /USER/CLASS/JOHN/DOE

- NOTE PUNCTUATION
  - SEMICOLON ONLY AT THE END OF THE LIST
  - COMMAS AS SEPARATORS IN THE LIST
  - "<-" AT END OF FILE DEFINITION
  - SIGNALS ON THE SAME PAGE CAN BE GROUPED BUT DIFFERENT PAGES REQUIRE A NEW LINE
  - SIGNALS COULD EACH HAVE THEIR OWN LINE
  - ANY SIGNAL NAMED ON THE SCHEMATIC MAY APPEAR IN THE $OUTPUTS SECTION
  - "S" AT THE END OF THE "$OUTPUTS" - COMMON ERROR IS TO FORGET IT OR TO TYPE A "$"

- 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/10: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:DATB15, DATB14, DATB13, DATB12,
@S16BITADR/4:DATB11, DATB10, DATB9, DATB8, DATB7, DATB6,

@S16BITADR/4:DATB5, DATB4, DATB3, DATB2, DATB1, DATB0,
@S16BITADR/4:MUXB,

/* ------------------------------------------------ */
/* 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,
@S16BITAOR/8:NEXT11, NEXT10, NEXT9, NEXT8,
@S16BITAOR/7:NEXT7, NEXT6, NEXT5, NEXT4,
@S16BITAOR/6:NEXT3, NEXT2, NEXT1, NEXT0;

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




/* ------------------------------------------------ */
/* INCLUDE AN "$END" STATEMENT                      */
/* ------------------------------------------------ */


$END
```

SAMPLED FILE OUTPUT SECTION

```
/* ================================================== */

$OUTPUTS

FILE /USER/CLASS/MUX16/FUNCTION.VLAF <-

/* -------------------------------------------------- */
/* LIST ALL PRIMARY INPUTS EXCEPT THERMAL DIODES      */
/* AND VBB MACRO INPUTS                               */
/* -------------------------------------------------- */

@MUX16/2:EXTCLK, EXTRST,
@MUX16/3:SELCT3, SELCT2, SELCT1, SELCTØ,
@MUX16/3:DATØ, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7,
@MUX16/3:DAT8, DAT9, DAT1Ø, DAT11, DAT12, DAT13, DAT14, DAT15,

/* -------------------------------------------------- */
/* LIST ALL PRIMARY OUTPUTS EXCEPT THERMAL DIODES     */
/* -------------------------------------------------- */

@MUX16/2:YOUTPT;

/* -------------------------------------------------- */
/* LIST 3-STATE ENABLES AND BIDIRECTIONAL ENABLES HERE */
/*   IF ANY                                           */
/* -------------------------------------------------- */

$END
```

- 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

THE $INPUTS SECTION - OPTIONAL

- IF INPUT VECTORS FOR THE SIMULATION ARE TO BE SUPPLIED BY A SEPARATE FILE, THIS SECTION DESCRIBES THAT FILE

- SIGNALS IN A SIGNAL GENERATOR SECTION CANNOT ALSO BE IN A "REMOTE" DATA FILE AND VISA VERSA

```
*******************************************************
     $INPUTS
     FILE /USER/CLASS/JOHN/DOE/DATA ->
     @DOE/1:VAR1, VAR2, VAR3,
     @DOE/2:SIG1, SIG2;
*******************************************************
```

- THE 5 SIGNALS ARE IN FIRST ON LEFT, ETC. ORDER IN THE FILE NAMED DATA IN THE PATH /USER/CLASS/JOHN/DOE

- NOTE PUNCTUATION
  - SEMICOLON ONLY AT THE END OF THE LIST
  - COMMAS AS SEPARATORS IN THE LIST
  - "->" AT END OF FILE DEFINITION
  - SIGNALS ON THE SAME PAGE CAN BE GROUPED BUT DIFFERENT PAGES REQUIRE A NEW LINE
  - SIGNALS COULD EACH HAVE THEIR OWN LINE
  - NO SIGNAL IN THE LIST MAY ALSO APPEAR IN THE $SIGNAL_GENERATOR SECTION
  - "S" AT THE END OF THE "$INPUTS" - COMMON ERROR IS TO FORGET IT OR TO TYPE A "$"

SAMPLE INPUT SECTION AND DATA FILE

Date: 28 MAY 86 11:Ø6 File: REMOTE.SING

```
/****
*
* DESIGN PATH /USER/CLASS/BARREL   DATE Ø5-MAY-1986 14:34
*
* COMPANY _____ CIRCUIT_NAME _16 BIT BARREL
*
* ARRAY   _Q1300T__    PO# _____ REV _____
*
* DESIGNER _____DEW_____
*
* 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  ****/

$INPUTS
FILE /USER/CLASS/BARREL16/INPUT.DAT ->

@BARREL16/2:DØ ,
@BARREL16/2:D1 ,
@BARREL16/2:D2 ,
@BARREL16/2:D3 ,
@BARREL16/2:D4 ,
@BARREL16/2:D5 ,
@BARREL16/2:D6 ,
@BARREL16/2:D7 ,
@BARREL16/2:D8 ,
@BARREL16/2:D9 ,
@BARREL16/2:D1Ø ,
@BARREL16/2:D11 ,
@BARREL16/2:D12 ,
@BARREL16/2:D13 ,
@BARREL16/2:D14 ,
@BARREL16/2:D15 ,

@BARREL16/2:SEL1,
@BARREL16/2:SEL2,
@BARREL16/2:SEL3,
@BARREL16/2:SEL4;
```

Date: 28 MAY 86 11:Ø6 File: INPUT.DAT

```
$DATA_HEADER$
$TYPE$
I/O
$FORMAT$
TIME_VALUE
$TOTAL_COLUMNS$
8 2Ø
$BASE$
D B
$FIELD$
TIME
$FIELD$
VALUE
$END$
ØØØØØØØØ 1111111111111111111111111111111111
ØØØ1ØØØØ 1111111111111111111111111111111111
ØØØ2ØØØØ 1111111111111111111111111111111111
ØØØ3ØØØØ 1111111111111111111111111111111111
ØØØ4ØØØØ 1111111111111111111111111111111111
ØØØ5ØØØØ 1111111111111111111111111111111111
ØØØ6ØØØØ 1111111111111111111111111111111111
ØØØ7ØØØØ 1111111111111111111111111111111111
ØØØ8ØØØØ 1111111111111111111111111111111111
ØØØ9ØØØØ 1111111111111111111111111111111111
ØØ1ØØØØØ 1111111111111111111111111111111111
ØØ11ØØØØ 1111111111111111111111111111111111
ØØ12ØØØØ 1111111111111111111111111111111111
ØØ13ØØØØ 1111111111111111111111111111111111
ØØ14ØØØØ 1111111111111111111111111111111111
ØØ15ØØØØ 1111111111111111111111111111111111
ØØ16ØØØØ 1111111111111111111111111111111111
ØØ17ØØØØ 1111111111111111111111111111111111
ØØ18ØØØØ 1111111111111111111111111111111111
ØØ19ØØØØ 1111111111111111111111111111111111
ØØ2ØØØØØ 1111111111111111111111111111111111
ØØ21ØØØØ 1111111111111111111111111111111111
ØØ22ØØØØ 1111111111111111111111111111111111
ØØ23ØØØØ 1111111111111111111111111111111111
ØØ24ØØØØ 1111111111111111111111111111111111
ØØ25ØØØØ 1111111111111111111111111111111111
ØØ26ØØØØ 1111111111111111111111111111111111
ØØ27ØØØØ 1111111111111111111111111111111111
ØØ28ØØØØ 1111111111111111111111111111111111
ØØ29ØØØØ 1111111111111111111111111111111111
ØØ3ØØØØØ 1111111111111111111111111111111111
ØØ31ØØØØ 1111111111111111111111111111111111
ØØ32ØØØØ 1111111111111111111111111111111111
ØØ33ØØØØ ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
ØØ34ØØØØ 11Ø111ØØ1111ØØØØ1ØØ
ØØ35ØØØØ 11Ø111ØØ1111ØØØØ1Ø1
ØØ36ØØØØ 11Ø111ØØ1111ØØØØ111Ø
ØØ37ØØØØ 11Ø111ØØ1111ØØØØ1111
```

TO GENERATE FMT_CSD.SING AND A DRAFTED, UNEDITED SOM_MCF.SING FILE, INVOKE SMAKER:

****

- INVOKE RUN_SMAKER

  OR CHOOSE MENU OPTION 3 UNDER THE SUPER-SHELL

BOTH SHELLS WILL END WITH "TEC SOM_MCF.SING"

- PERFORM THE EDITS DESIRED
- SUPER SHELL WILL REDISPLAY ITS MENU WHEN TEC IS EXITED

****

IF YOU ARE GOING TO CREATE A DATA FILE
- EXIT THE SUPER-SHELL

```
/****
*
* DESIGN PATH /USER/CLASS/S16BITADR   DATE 18 NOV 1990 12:38
*
* COMPANY  __AMCC__________   CIRCUIT_NAME  16 BIT ADDR
*
* ARRAY    Q13ØØS___     PO#  ___________   REV  ______
*
* DESIGNER _____DEW__________________
*
* What tests does this control file support: _________
*
* ____FUNCTIONAL VECTOR GNERATION FOR THE _____________
*
* ______ADDER - A CASE STUDY_____________________________
*
****/

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

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

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

$INPUTS

/*  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/1Ø:EXTCLK,

@S16BITADR/1Ø:EXTRST,

@S16BITADR/3:CARYIN,

@S16BITADR/2:DATAØ,
@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,
```

```
@S16BITADR/2:DATA1Ø,
@S16BITADR/2:DATA11,
@S16BITADR/3:DATA12,
@S16BITADR/3:DATA13,
@S16BITADR/3:DATA14,
@S16BITADR/3:DATA15,

@S16BITADR/2:MUXA,

@S16BITADR/4:DATBØ,
@S16BITADR/4:DATB1,
@S16BITADR/4:DATB2,
@S16BITADR/4:DATB3,
@S16BITADR/4:DATB4,
@S16BITADR/4:DATB5,
@S16BITADR/4:DATB6,
@S16BITADR/4:DATB7,
@S16BITADR/4:DATB8,
@S16BITADR/4:DATB9,
@S16BITADR/4:DATB1Ø,
@S16BITADR/4:DATB11,
@S16BITADR/3:DATB12,
@S16BITADR/3:DATB13,
@S16BITADR/3:DATB14,
@S16BITADR/3:DATB15,

@S16BITADR/4:MUXB:


/* ------------------------------------------------ */
/* 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/1Ø:EXTCLK, EXTRST,

@S16BITADR/3:CARYIN,

@S16BITADR/3:DATA15, DATA14, DATA13, DATA12,
@S16BITADR/2:DATA11, DATA1Ø, DATA9, DATA8, DATA7, DATA6,
@S16BITADR/2:DATA5, DATA4, DATA3, DATA2, DATA1, DATAØ,
@S16BITADR/2:MUXA,

@S16BITADR/3:DATB15, DATB14, DATB13, DATB12,
@S16BITADR/4:DATB11, DATB1Ø, DATB9, DATB8, DATB7, DATB6,
@S16BITADR/4:DATB5, DATB4, DATB3, DATB2, DATB1, DATBØ,
@S16BITADR/4:MUXB,
```

```
/* ------------------------------------------------ */
/* 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 "$END" STATEMENT                      */
/* ------------------------------------------------ */

$END
```

```
$DATA_HEADER$
$TYPE$
I/O
$FORMAT$
TIME_VALUE
$TOTAL_COLUMNS$
6  37
$BASE$
D B
$FIELD$
TIME
$FIELD$
VALUE

/* -------  EECDDDDDDDDDDDDDDDDMDDDDDDDDDDDDDDDM
            XXAAAAAAAAAAAAAAAAAUAAAAAAAAAAAAAAAAU
            TTRTTTTTTTTTTTTTTTTXTTTTTTTTTTTTTTTTX
            CRYAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBB
            LSI11111119876543210 1111119876543210
            KTN543210            543210

------------------------------------------------ */
$END$
000000      10000000000000000000000000000000000000
010000      00000000000000000000000000000000000000
020000      11000000000000000000000000000000000000
030000      01000000000000000000000000000000000000
040000      10000000000000000000000000000000000000
050000      00000000000000000000000000000000000000
060000      10011111111111111110000000000000000000
070000      00011111111111111111000000000000000000
080000      10011111111111111101111111111111111110
090000      00011111111111111110111111111111111110
100000      10000000000000000001111111111111111110
110000      00000000000000000001111111111111111110
120000      10000000000000000000000000000000000000
130000      00000000000000000000000000000000000000
140000      10111111111111111110000000000000000000
150000      00111111111111111110000000000000000000
160000      10111111111111111101111111111111111110
170000      00111111111111111110111111111111111110
180000      10100000000000000001111111111111111110
190000      00100000000000000001111111111111111110
200000      10100000000000000000000000000000000000
210000      00100000000000000000000000000000000000
220000      10011111111111111110000000000000000001
230000      00011111111111111110000000000000000001
240000      10011111111111111111111111111111111111
250000      00011111111111111111111111111111111111
260000      10000000000000000001111111111111111111
270000      00000000000000000001111111111111111111
280000      10000000000000000001000000000000000001
290000      00000000000000000001000000000000000001
```

```
/*                                                                    */
/*                                                                    */
/* DESIGN PATH /USER/DICKS/ATOM1   DATE 26 JUN 1986 11:05             */
/*                                                                    */
/* COMPANY  LITTON ATD_____    CIRCUIT_NAME  ATOM1_____               */
/*                                                                    */
/* ARRAY    Q3500____      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

$INPUTS
FILE /USER/DICKS/ATOM1/NEWTERSE.LIS ->

@ATOM1/0:RESETN,
@ATOM1/0:B16MHZ,
@ATOM1/12:ASN,
@ATOM1/13:DSN,
@ATOM1/13:RW,
@ATOM1/7:FC2,
@ATOM1/12:FC1,
@ATOM1/12:FC0,
@ATOM1/12:PAD21,
@ATOM1/12:PAD20,
@ATOM1/7:PAD19,
@ATOM1/7:PAD18,
@ATOM1/7:PAD17,
@ATOM1/7:PAD16,
@ATOM1/16:PAD01,
```

$Inputs Section — with Bidirectionals

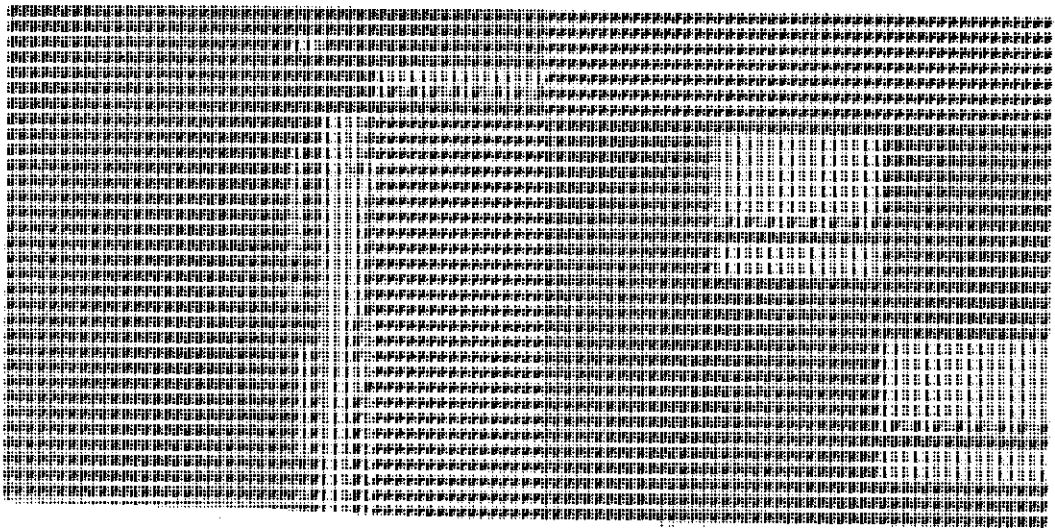
```
@ATOM1/2:MDEAN,
@ATOM1/3:MDEBN,
@ATOM1/4:MDEPN,
@ATOM1/5:DMUXA,
@ATOM1/5:DMUX8,
@ATOM1/5:TRID15::CON,
@ATOM1/5:TRID14::CON,
@ATOM1/5:TRID13::CON,
@ATOM1/5:TRID12::CON,
@ATOM1/5:TRID11::CON,
@ATOM1/5:TRID10::CON,
@ATOM1/5:TRID09::CON,
@ATOM1/5:TRID08::CON,
@ATOM1/5:TRID07::CON,
@ATOM1/5:TRID06::CON,
@ATOM1/5:TRID05::CON,
@ATOM1/5:TRID04::CON,
@ATOM1/5:TRID03::CON,
@ATOM1/5:TRID02::CON,
@ATOM1/5:TRID01::CON,
@ATOM1/5:TRID00::CON,
@ATOM1/4:PD31::CON,
@ATOM1/4:PD30::CON,
@ATOM1/4:PD29::CON,
@ATOM1/4:PD28::CON,
@ATOM1/4:PD27::CON,
@ATOM1/4:PD26::CON,
@ATOM1/4:PD25::CON,
@ATOM1/4:PD24::CON,
@ATOM1/4:PD23::CON,
@ATOM1/4:PD22::CON,
@ATOM1/4:PD21::CON,
@ATOM1/4:PD20::CON,
@ATOM1/4:PD19::CON,
@ATOM1/4:PD18::CON,
@ATOM1/4:PD17::CON,
@ATOM1/4:PD16::CON,
@ATOM1/2:ATAD15::CON,
@ATOM1/2:ATAD14::CON,
@ATOM1/2:ATAD13::CON,
@ATOM1/2:ATAD12::CON,
@ATOM1/2:ATAD11::CON,
@ATOM1/2:ATAD10::CON,
@ATOM1/2:ATAD09::CON,
@ATOM1/2:ATAD08::CON,
@ATOM1/2:ATAD07::CON,
@ATOM1/2:ATAD06::CON,
@ATOM1/2:ATAD05::CON,
```

```
@ATOM1/3:ATBD02::CON,
@ATOM1/3:ATBD01::CON,
@ATOM1/3:ATBD00::CON;

/* Timing section:    */

/* Print on change output file section */
$OUTPUTS
/* PRINT_ON_CHANGE */
FILE /USER/DICKS/ATOM1/FUN_POC.OUT <-

/* INPUT SECTION */

@ATOM1/0:RESETN,
@ATOM1/0:B16MHZ,
@ATOM1/12:ASN,
@ATOM1/13:DSN,
@ATOM1/13:RW,
@ATOM1/7:FC2,
@ATOM1/12:FC1,
@ATOM1/12:FC0,
@ATOM1/12:PAD21,
@ATOM1/12:PAD20,
@ATOM1/7:PAD19,
@ATOM1/7:PAD18,
@ATOM1/7:PAD17,
@ATOM1/7:PAD16,
@ATOM1/16:PAD01,
@ATOM1/16:PAD00,
@ATOM1/12:SIZ1,
@ATOM1/12:SIZ0,
@ATOM1/14:GSAKIN,
@ATOM1/14:GSAK0N,
@ATOM1/14:CSAKIN,
@ATOM1/14:CSAK0N,
@ATOM1/14:TPENN,
@ATOM1/9:HEAD,
@ATOM1/9:BPRI,
@ATOM1/12:RMCPN,
@ATOM1/11:BMAKN,
@ATOM1/2:PADEAN,
@ATOM1/3:PADEBN,
@ATOM1/4:PADEPN,
@ATOM1/2:MDEAN,
@ATOM1/3:MDEBN,
@ATOM1/4:MDEPN,
@ATOM1/5:DMUXA,
```

```
$DATA_HEADER$
$TYPE$
I/O
$FORMAT$
TIME_VALUE
$TOTAL_COLUMNS$
8 99
$BASE$
D B
$FIELD$
TIME
$FIELD$
VALUE
$END$
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
110000
120000
130000
140000
150000
160000
170000
180000
190000
200000
210000
220000
230000
240000
250000
260000
270000
280000
290000
300000
310000
```

Input file

Bidirectionals



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> <tcal-delay-filename>

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

GENERATES: SOM.ERR
TCAL.ERR

****

TCAL: TIMING CALCULATOR

TCAL PREREQUISITES:

- USES AGIF NETLIST PRODUCED FROM TREE.DNLK
  -- CIRCUIT.SDI

- PRODUCES FNTxxx.DSY ONE MIL AND ONE MIN
  OR ONE COM AND ONE MIN FILE
  - NOM file provided for convenience

- MODIFIES DELAYS IN A DTV/DLS INPUT FILE

- USED TO INCORPORATE LAYOUT-DEPENDENT DATA
  INTO THE DELAY TIMES CALCULATION

- USED FOR AMCC FRONT-ANNOTATION (BEFORE LAYOUT)

- USED FOR AMCC BACK-ANNOTATION (AFTER LAYOUT)
INVOKE BY CALLING IT OUT WITH THE RUN_SOM SHELL

SEE LOGICIAN DESIGN COMPILATION SECTION ....
FOR TCAL ERROR MESSAGES

NOTIFY AMCC ON RECEIVING ERROR MESSAGES
AFTER YOU VERIFY THAT THE FRONT-ANNOTATION
CIRCUIT AND THE VERSION YOU ARE NOW TO SIMULATE
ARE THE SAME.

- USE TEC TO EDIT THE FRONT-ANNOTATION FILE UNDER
  APPROVAL OF AMCC

- EDITING OF THE INTERMEDIATE-ANNOTATION FILE
  IS FORBIDDEN

- EDITING OF THE BACK-ANNOTATION FILE IS FORBIDDEN

****

- INVOKE THE SOM - TCAL SHELL BY:

- USE: RUN_SOM <som-ctl-filename> <tcal-delay-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
!
```

- 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).

This is a DNIX FMT window created from an edited FMT_CSD.SING file.
- This can be printed out with the DNIX op sys.
- or (better?) use a copied version of the FMT_CSD.SING to create an I/O list

USE FOR YOU OWN DEBUG - CONTROLS LIST AND WAVE FORMATS
- NOT USED IN DESIGN SUBMISSION

NAME	BASE	POLARITY	STRN	TRC?	SIGNAL_LIST
CLOCK	BIN	+	OFF	ON	@CLASS/2:CLOCK
YOUTPT	BIN	+	OFF	ON	@CLASS/2:YOUTPT
SELCT0	BIN	+	OFF	ON	@CLASS/1:SELCT0
SELCT1	BIN	+	OFF	ON	@CLASS/1:SELCT1
SELCT2	BIN	+	OFF	ON	@CLASS/1:SELCT2
SELCT3	BIN	+	OFF	ON	@CLASS/1:SELCT3
DAT0	BIN	+	OFF	ON	@CLASS/1:DAT0
DAT1	BIN	+	OFF	ON	@CLASS/1:DAT1
DAT2	BIN	+	OFF	ON	@CLASS/1:DAT2
DAT3	BIN	+	OFF	ON	@CLASS/1:DAT3
DAT4	BIN	+	OFF	ON	@CLASS/1:DAT4
DAT5	BIN	+	OFF	ON	@CLASS/1:DAT5
DAT6	BIN	+	OFF	ON	@CLASS/1:DAT6
DAT7	BIN	+	OFF	ON	@CLASS/1:DAT7
DAT8	BIN	+	OFF	ON	@CLASS/1:DAT8
DAT9	BIN	+	OFF	ON	@CLASS/1:DAT9
DAT10	BIN	+	OFF	ON	@CLASS/1:DAT10
DAT11	BIN	+	OFF	ON	@CLASS/1:DAT11
DAT12	BIN	+	OFF	ON	@CLASS/1:DAT12
DAT13	BIN	+	OFF	ON	@CLASS/1:DAT13
DAT14	BIN	+	OFF	ON	@CLASS/1:DAT14
DAT15	BIN	+	OFF	ON	@CLASS/1:DAT15

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

SIMULATION MODE NOM

(reserved)
(reserved)
(reserved)

	ENABLE	TRC?	SORTED BY MAJOR KEY	SORTED BY MINOR KEY
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 MAXIMUM worst-case multiplier
within the timing library selected:
- USE WITH MILITARY OR COMMERCIAL SIFT LIBRARY

SIMULATION MODE MAX

(reserved)
(reserved)
(reserved)

	ENABLE	TRC?	SORTED BY MAJOR KEY	SORTED BY MINOR KEY
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

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

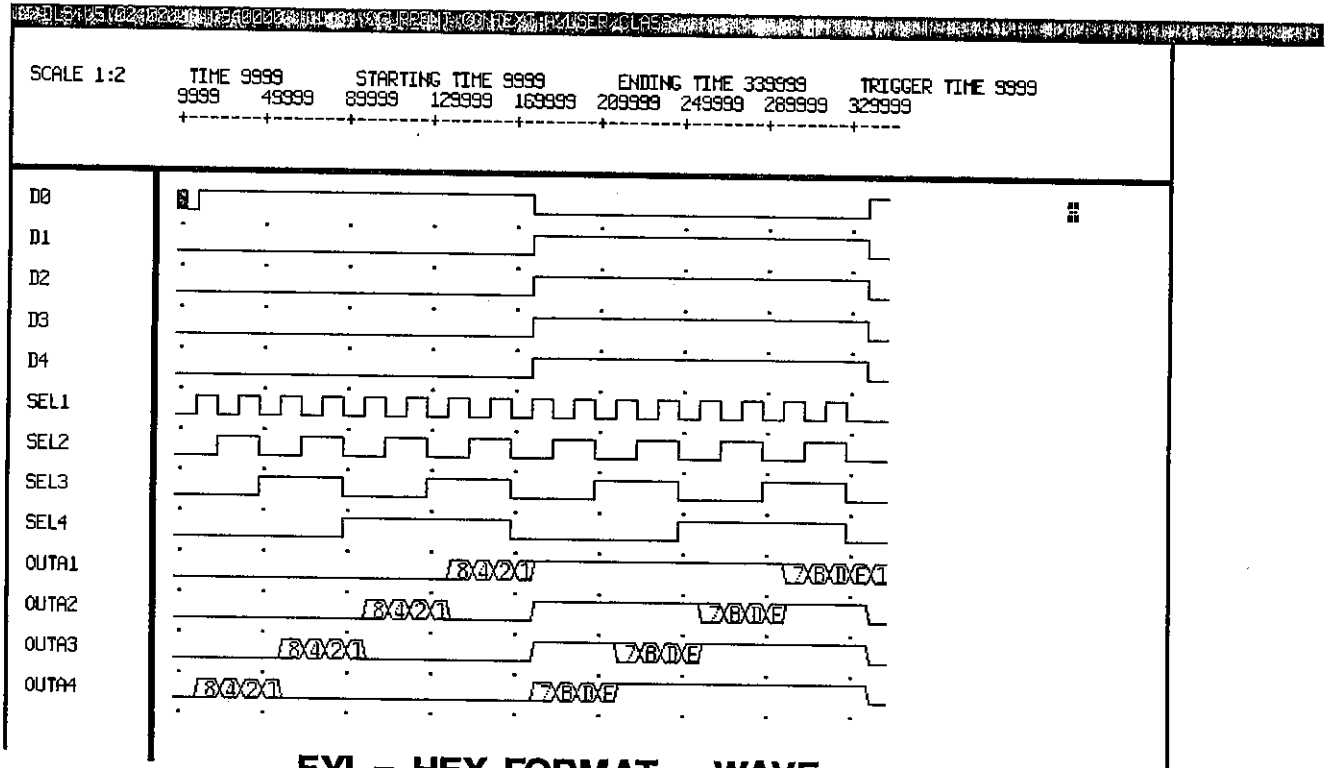
SIMULATION MODE MIN

(reserved)
(reserved)
(reserved)

	ENABLE	TRC?	SORTED BY MAJOR KEY	SORTED BY MINOR KEY
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

```
DLS V5.02.02        CURRENT CONTEXT: /USER/CLASS/MUX16
 ** REPLACE **

NAME           BASE  POLARITY STRN TRC? SIGNAL_LIST

YOUTPT         BIN      +       OFF  ON  @MUX16/2:YOUTPT
DATA           HEX      +       OFF  ON  @MUX16/3:DAT15,DAT14,DAT13,DAT12
DATB           HEX      +       OFF  ON  @MUX16/3:DAT11,DAT10,DAT9,DAT8
DATC           HEX      +       OFF  ON  @MUX16/3:DAT7,DAT6,DAT5,DAT4
DATD           HEX      +       OFF  ON  @MUX16/3:DAT3,DAT2,DAT1,DAT0
EXTCLK         BIN      +       OFF  ON  @MUX16/2:EXTCLK
EXTRST         BIN      +       OFF  ON  @MUX16/2:EXTRST
SELCTA         HEX      +       OFF  ON  @MUX16/3:SELCT3,SELCT2,SELCT1,SELCT0
```

FYI – HEX FORMAT



FYI – HEX FORMAT – WAVE

LABEL BASE POLARITY STRENGTH	TIME	COUNT	DØ BIN + OFF	D1 BIN + OFF	D2 BIN + OFF	D3 BIN + OFF	D4 BIN + OFF	SEL1 BIN + OFF	SEL2 BIN + OFF	SEL3 BIN + OFF	SEL4 BIN + OFF	OUTA1 HEX + OFF	OUTA2 HEX + OFF	OUTA3 HEX + OFF	OUTA4 HEX + OFF
Ø	9999	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
+1	19999	Ø	1	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	8
+2	29999	Ø	1	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	4
+3	39999	Ø	1	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	2
+4	49999	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	1
+5	59999	Ø	1	Ø	Ø	Ø	Ø	1	Ø	1	Ø	Ø	Ø	Ø	8
+6	69999	Ø	1	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	4
+7	79999	Ø	1	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	2
+8	89999	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	1	Ø
+9	99999	Ø	1	Ø	Ø	Ø	Ø	1	Ø	Ø	1	Ø	Ø	8	Ø
+1Ø	1Ø9999	Ø	1	Ø	Ø	Ø	Ø	Ø	1	Ø	1	Ø	Ø	4	Ø
+11	119999	Ø	1	Ø	Ø	Ø	Ø	1	1	Ø	1	Ø	Ø	2	Ø
+12	129999	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	1	Ø
+13	139999	Ø	1	Ø	Ø	Ø	Ø	1	Ø	1	1	Ø	Ø	8	Ø
+14	149999	Ø	1	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	4	Ø
+15	159999	Ø	1	Ø	Ø	Ø	Ø	1	1	1	1	Ø	Ø	2	Ø
+16	169999	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø
+17	179999	Ø	Ø	1	1	1	1	1	Ø	Ø	Ø	F	F	F	7
+18	189999	Ø	Ø	1	1	1	1	Ø	1	Ø	Ø	F	F	F	B
+19	199999	Ø	Ø	1	1	1	1	1	1	Ø	Ø	F	F	F	D
+2Ø	2Ø9999	Ø	Ø	1	1	1	1	Ø	Ø	1	Ø	F	F	F	E
+21	219999	Ø	Ø	1	1	1	1	1	Ø	1	Ø	F	F	7	F
+22	229999	Ø	Ø	1	1	1	1	Ø	1	1	Ø	F	F	B	F
+23	239999	Ø	Ø	1	1	1	1	1	1	1	Ø	F	F	D	F
+24	249999	Ø	Ø	1	1	1	1	Ø	Ø	Ø	1	F	F	E	F
+25	259999	Ø	Ø	1	1	1	1	1	Ø	Ø	1	F	7	F	F
+26	269999	Ø	Ø	1	1	1	1	Ø	1	Ø	1	F	B	F	F
+27	279999	Ø	Ø	1	1	1	1	1	1	Ø	1	F	D	F	F
+28	289999	Ø	Ø	1	1	1	1	Ø	Ø	1	1	F	E	F	F
+29	299999	Ø	Ø	1	1	1	1	1	Ø	1	1	7	F	F	F
+3Ø	3Ø9999	Ø	Ø	1	1	1	1	Ø	1	1	1	B	F	F	F
+31	319999	Ø	Ø	1	1	1	1	1	1	1	1	D	F	F	F
+32	329999	Ø	Ø	1	1	1	1	Ø	Ø	Ø	Ø	E	F	F	F
+33	339999	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø

FYI – HEX FORMAT – LIST

• 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

****

AMCCVRC - AMCC VECTOR RULES CHECKER

- AFTER THE SIMULATION FILE ARE FORMATED, GENERATE A SIGNAL ANALYSIS FILE PER THE RULES IN VOL II, SECTION 8, APPENDIX B.

- THE SIGNAL ANALYSIS FILE STATES THE RELATIONSHIPS BETWEEN CLOCK AND DATA SIGNALS AND IS REQUIRED FOR ANY CLOCKED CIRCUIT

- WHEN THIS FILE IS CREATED, EXECUTE AMCCVRC

- ONLY THE SAMPLED, WORST-CASE MAXIMUM FUNCTIONAL, AC TEST AND PARAMETRIC FILES ARE SCREENED

- AC TEST - IGNORE TOGGLE TEST ERROR MESSAGES

- PARAMETRICS WILL HAVE THEIR OWN TESTS ADDED IN A LATER RELEASE. RUN CHECKS AS LISTED IN SECTION 4-5

****

- INVOKE BY:

  AMCCVRC

  OR CHOOSE MENU OPTION "6" UNDER THE SUPER-SHELL

GENERATES: AMCCVRC.LST
AMCCVRC.ERR

****

APPENDIX A
AMCC-SUPPLIED
SHELL SCRIPTS

APPENDIX A

THE SHELLS:

RUN_AGIF	NETLIST GENERATION
RUN_AMCC	SUPER SHELL
RUN_DD	DANCE AND DRINK
RUN_ERC	AMCC ERC
RUN_ANN	AMCC ANNOTATION
RUN_SIFT	SIFT (TIMING)
RUN_SMAKER	CREATE SOM_MCF.SING AND FMT
RUN_SMT	AMCCSIMFMT
RUN_SOM	INTEGRATION
RUN_VRC	AMCCVRC

THESE ARE SUPPLIED AS PART OF THE AMCC MACROMATRIX SOFTWARE

ONCE A SCHEMATIC IS CAPTURED, TYPE:

RUN_AMCC

THE SUPER SHELL CALLS THE OTHER SHELLS WHICH IN TURN CALL OUT THE VARIOUS COMMANDS TO EXECUTE THE DAISY AND THE AMCC SOFTWARE IN THE PROPER ORDER OF EXECUTION

THE MENU WILL ASK FOR A CHOICE OF 1, 2 OR 3

1 WILL CAUSE RUN_DD DANCE-DRINK TO A DAISY NETLIST
RUN_AGIF REFORMAT TO AMCC AGIF NETLIST
RUN_ERC ERC SOFTWARE

AFTER DEBUG OF ANY ERRORS,

2 WILL CAUSE RUN_ANN ANNOTATION

3 WILL CAUSE RUN_SIFT TIMING LIBRARY SELECTION
RUN_SMAKER SING-TO-DAISY

THIS WILL END WITH: TEC SOM_MCF.SING

AFTER EDITING THE SIMULATION CONTROL FILE, ON EXITING TEC, THE MENU RETURNS

0 WILL CAUSE AN EXIT

IF OTHER EDITING SUCH AS DATA FILE CREATION IS REQUIRED, SELECT 0, ELSE,

4 WILL CAUSE RUN_SOM SOM AND TCAL STEPS

FROM HERE, EXIT USING 0 AND RUN DLS OR DTV AS DESIRED

ALSO, USE TEC TO CREATE THE SIGNAL ANALYSIS FILE

RECALL RUN_AMCC

5 WILL CAUSE RUN_SMT AMCCSIMFMT

6 WILL CAUSE RUN_VRC AMCCVRC

YOU MAY RUN THE SUB-SHELLS BY CALLING THEM INDIVIDUALLY

THE PROPER ORDER IS:

```
     RUN_DD
     RUN_AGIF
     RUN_ERC
DEBUG AS NEEDED

     RUN_ANN
     RUN_SIFT
     RUN_SMAKER

EDIT SIMULATION CONTROL FILE
CREATE A DATA FILE IF DESIRED
     RUN_SOM

FROM HERE, RUN DLS OR DTV AS DESIRED

AND WHEN SIMULATION IS COMPLETE, TYPE
     RUN_SMT

AFTER CREATING THE SIGNAL ANALYSIS FILE, TYPE
     RUN_VRC
```

YOU MAY ALSO RUN THE REQUIRED STEPS BY TYPING IN
THE INDIVIDUAL SPECIFIC COMMANDS

THE COMMANDS:

```
DANCE -T -N -ERR -M3
DRINK -T -M3

RUN_AGIF                    SING-TO-AGIF; USE THIS SHELL

AMCCERC

AMCCANN

 SIFT -M3 -LIB $FAMILY/PATCH$1.SLIB $FAMILY/Q50IO$1.SLIB\
         $FAMILY/Q50GE$1.SLIB $FAMILY/Q50RA$1.SLIB

SING -M3 -MCF /AMCC/SOM_MAKER/SOM_MCF
FMT_CSD.SING
TEC SOM_MCF.SING
    (EDIT THE SIMULATION CONTROL FILE)

SOM SOM_MCF.SING -M3 -L SOM.ERR
TCAL -MCF FNTxxx.DSY -M3

FROM HERE, RUN DLS OR DTV AS DESIRED
       DLS
       GET FMT
       MODE
         (EDIT)
       {ENTER}
       VIEW 9999 10000
       RUN nnnnnnnn
       START 0
       LIST OR WAVE AS DESIRED
       {ENTER}
       QUIT
       N
```

THE $OUTPUTS SECTION FILE DEFINED IN THE SOM_MCF.SING FILE IS THE INPUT TO AMCCSIMFMT

AMCCSIMFMT

THE OUTPUT OF AMCCSIMFMT IS INPUT TO AMCCVRC

AMCCVRC

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_AGIF

echo 'FALSE' > ERROR.ERR
echo '
      ********* RUNNING AGIF NETLISTER ***********
     '
MKDIR ERC 2> /DEV/NULL
AMCCAGIF > ERC/CIRCUIT.SDI 2> /DEV/NULL
if (test -s ERC/CIRCUIT.SDI)
then
   ecno
else
  echo '         *** AGIF SYSTEM ERROR (Contact AMCC) !!!!! ***
        '         **** PROBIBLE CAUSE: EXCCEDED REAL MEMORY ****' > AGIF_SYS.ERR
  echo '
       ********** AGIF SYSTEM ERROR !!! **********
       **                                        **
       ***** PLEASE ENTER "TYPE AGIF_SYS.ERR" *****
      '
  echo 'TRUE' > ERROR.ERR
  exit
fi
if egrep -s 'ERROR' RUN_AGIF.ERR
then
  echo '
       ********* AGIF NETLISTER ERROR !!! *********
       **                                       **
       ***** PLEASE ENTER "TYPE RUN_AGIF.ERR" *****
      '
  echo 'TRUE' > ERROR.ERR
  exit
else
  echo '
       ********* AGIF NETLIST COMPLETE !!! ********
      '
fi
```

Date: 14 SEP 88 14:42 File: /AMCC/SUBMIT/RUN_AMCC

```
menu=9
echo 'FALSE' > ERROR.ERR
while (test $menu -gt Ø)
do
   MAX_MENU.LIB
   echo -n '          Please Enter [Ø - 6] : '
   read menu
   echo
   if (test $menu -gt 6)
   then
      echo '          Only Values Between Ø through 6 are Valid...'
      exit
   fi
   if (test $menu -eq Ø)
   then
      exit
   fi
   if (test $menu -eq 1)
   then
      RUN_DD
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
      RUN_AGIF
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
      RUN_ERC
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
   fi
   if (test $menu -eq 2)
   then
      RUN_ANN
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
   fi
```

```
if (test $menu -eq 3)
then
   if egrep -s 'Q14000' ERC/CIRCUIT.SDI
   then
      echo -n '        Please Enter Product Grade (MIN, NOM, COM4, COM5 or MIL) : ';
   else
      echo -n '        Please Enter Product Grade (MIN, NOM, COM or MIL) : ';
   fi
   read grade
   EXPORT grade
   RUN_SIFT $grade
   if egrep -s 'TRUE' ERROR.ERR
   then
      exit
   fi
   RUN_SMAKER
   if egrep -s 'TRUE' ERROR.ERR
   then
      exit
   fi
   fi
   if (test $menu -eq 4)
   then
      echo -n '        Please Enter SOM Master Control File : ';
      read mcf
      EXPORT mcf
      echo
      echo -n '        Please Enter TCAL Delay File : ';
      read tcal
      EXPORT tcal
      RUN_SOM $mcf $tcal
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
   fi
   if (test $menu -eq 5)
   then
      RUN_SMT
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
   fi
   if (test $menu -eq 6)
   then
      RUN_VRC
      if egrep -s 'TRUE' ERROR.ERR
      then
         exit
      fi
   fi
done
```

RUN_

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_ANN

echo 'FALSE' > ERROR.ERR
echo '
        ************** RUNNING AMCCANN *************
     '
cd ERC
echo -n '         Need to Edit Package Pin Data? (Y or N) : '
read temp
echo ''
if (test $temp = 'Y' -o $temp = 'y')
then
   if (test -f OUTPUT.DLY)
   then
      AMCCANN <<!

!
   else
      AMCCANN <<!


!
   fi
else
   AMCCANN
fi
if egrep -s 'ERROR' AMCCANN.LST
then
  echo '
        ************ AMCCANN ERROR !!!!! ************
        **                                          **
        ****** PLEASE ENTER "TYPE AMCCANN.LST"******
       '
  cd ..
  echo 'TRUE' > ERROR.ERR
  exit
else
  COPY *.DSY TO ../*.DSY -8  2> /DEV/NULL
  rm *.DSY
  cd ..
  echo '
        ************** AMCCANN OK !!!!! *************
       '
fi
```

RUN_DD

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_DD

echo 'FALSE' > ERROR.ERR
echo '
       ****** RUNNING DANCE -T -N -ERR -M3 ********
     '
DANCE -T -N -ERR -M3  2> DANCE.ERR
if egrep -s 'PAGES FLAGGED. HIGHEST SEVERITY WAS Ø' DANCE.ERR
then
  echo '
       ************** DANCE OK !!!!! ***************
     '
else
  echo '
       ************ DANCE ERROR !!!!! *************
       **                                          **
       ****** PLEASE ENTER "TYPE DANCE.ERR" *******
     '
  echo 'TRUE' > ERROR.ERR
fi
echo '
       *********** RUNNING DRINK -T -M3 ***********
     '
DRINK -T -M3  2> DRINK.ERR
if egrep -s 'Ø BLOCKS FLAGGED. HIGHEST SEVERITY WAS Ø' DRINK.ERR
then
  echo '
       ************** DRINK OK !!!!! ***************
     '
else
  echo '
       ************ DRINK ERROR !!!!! *************
       **                                          **
       ****** PLEASE ENTER "TYPE DRINK.ERR" *******
     '
  echo 'TRUE' > ERROR.ERR
fi
```

RUN_ERC

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_ERC

echo 'FALSE' > ERROR.ERR
echo '
      ************** RUNNING AMCCERC ***************
     '
cd ERC
AMCCERC
if egrep -s '^ERROR' AMCCERC.LST
then
  echo '
        ************* AMCCERC ERROR !!!!! ************
        **                                          **
        **** PLEASE ENTER "TYPE ERC/AMCCERC.LST" ***
       '
  cd ..
  echo 'TRUE' > ERROR.ERR
  exit
else
  echo '
        ************** AMCCERC OK !!!!! **************
       '
fi
```

NOTE: /ERC/AMCCERC.LST IS THE ERC REPORT

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_SIFT

echo 'FALSE' > ERROR.ERR
if test -z "$1"
then
  echo '
        ******* NO GRADE VALUE ENTERED !!!!  *******
        **                                  **
        ** ENTER  "RUN_SIFT MIN, NOM, COM or MIL" **
        '
  echo 'TRUE' > ERROR.ERR
  exit
fi
echo "
        *************  RUNNING SIFT  ***************

        FAMILY = /AMCC/Q5000_LIBS       GRADE = $1
     "
SIFT -M3 -LIB /AMCC/Q5000_LIBS/PATCH$1.SLIB /AMCC/Q5000_LIBS/Q50I1$1.SLIB \
              /AMCC/Q5000_LIBS/Q50G1$1.SLIB /AMCC/Q5000_LIBS/Q50G2$1.SLIB \
              /AMCC/Q5000_LIBS/Q50G3$1.SLIB /AMCC/Q5000_LIBS/Q50M1$1.SLIB \
              /AMCC/Q5000_LIBS/Q50G1$1H.SLIB /AMCC/Q5000_LIBS/Q50G2$1H.SLIB \
              /AMCC/Q5000_LIBS/Q50G3$1H.SLIB /AMCC/Q5000_LIBS/Q50M1$1H.SLIB \
              /AMCC/Q5000_LIBS/Q50I1$1H.SLIB 2> SIFT.ERR
if egrep -s 'WRITING TREE SIF FILE' SIFT.ERR
then
  echo '
        ************* SIFT OK !!!!! ****************
        '
else
  echo '
        ************ SIFT ERROR !!!!! **************
        **                                  **
        ****** PLEASE ENTER "TYPE SIFT.ERR" ********
        '
  echo 'TRUE' > ERROR.ERR
  exit
fi
```

RUN_SMAKER

```
Date:  14 SEP 88 14:43   File:  /AMCC/SUBMIT/RUN_SMAKER

echo 'FALSE' > ERROR.ERR
echo '
      ********** RUNNING SING TO SOM  ************
     '
SING -M3 -MCF /AMCC/SOM_MAKER/SOM_MCF 2> SOM_MAKER.ERR
if egrep -s 'SING ERROR' SOM_MAKER.ERR
  then
  echo '
        ********  SING TO DAISY ERROR !!!!! *********
        **                                        **
        ***** PLEASE ENTER "TYPE SOM_MAKER.ERR" *****
       '
  echo 'TRUE' > ERROR.ERR
  exit
else
  echo '
        ********** SING TO DAISY OK !!!!! ***********
       '
fi
echo '
        *********** RUNNING FMT_CSD.SING ************
     '
FMT_CSD.SING 2>> SOM_MAKER.ERR
if egrep -s 'No user errors' SOM_MAKER.ERR
  then
  echo '
        **********  FMT FORMAT OK !!!!!  ************
       '
else
  echo '
        *********  FMT FORMAT ERROR !!!!! ***********
        **                                        **
        ***** PLEASE ENTER "TYPE SOM_MAKER.ERR" *****
       '
  echo 'TRUE' > ERROR.ERR
  exit
fi
TEC SOM_MCF.SING
```

NOTE: THIS SHELL LEAVES YOU IN THE TEC EDITOR

RUN_SOM

```
Date:  14 SEP 88 14:42   File:  /AMCC/SUBMIT/RUN_SOM

echo 'FALSE' > ERROR.ERR
if test -z "$1"
then
  echo '
        **** NO SOM.MCF or TCAL.MCF ENTERED !!!! ***
        **                                        **
        ***** ENTER "RUN_SOM somfile tcalfile" *****
       '
  echo 'TRUE' > ERROR.ERR
  exit
fi
if test -z "$2"
then
  echo '
        ********** NO TCAL.MCF ENTERED !!!! **********
        **                                        **
        ***** ENTER "RUN_SOM somfile tcalfile" *****
       '
  echo 'TRUE' > ERROR.ERR
  exit
fi
echo '
        ************ RUNNING SOM and TCAL *************
     '
SOM $1 -M3 -L SOM.ERR
if egrep -s 'No user errors' SOM.ERR
  then
  echo '
        ************  SOM FORMAT OK !!!!!  ************
       '
else
  echo '
        **********  SOM FORMAT ERROR !!!!! ************
        **                                        **
        ******* PLEASE ENTER "TYPE SOM.ERR" *********
       '
  echo 'TRUE' > ERROR.ERR
  exit
fi
TCAL -MCF $2 -M3 2> TCAL.ERR
if egrep -s '.' TCAL.ERR
then
  echo '
        **********  TCAL FORMAT ERROR !!!!! **********
        **                                        **
        ******* PLEASE ENTER "TYPE TCAL.ERR" ********
       '
  echo 'TRUE' > ERROR.ERR
  exit
else
  echo '
        ************  TCAL FORMAT OK !!!!!  **********
       '
fi
```

```
Date:  14 SEP 88 14:43   File:  /AMCC/SUBMIT/RUN_SMT

#
# This shell script will run AMCCSIMFMT
#
AMCCSIMFMT
```

```
Date:  14 SEP 88 14:43   File:  /AMCC/SUBMIT/RUN_VRC

#
# This shell script will run AMCCVRC and delete any junk files left over
#
# History:          * Feb 19, 1988 - original implementation
#
#
#
AMCCVRC
rm ASF*.DAT
```

Application Note 1.DNIX

Introduction to the DAISY

(809)

Table of Contents

Figures:

INTRODUCTION TO THE DAISY

Q5000 16:1 MUX MIXED MODE EXAMPLE

The following provides a simple introduction to design with the AMCC Q5000 Series Logic Array on a DAISY EWS (Engineering Workstation) under the DNIX operating system, using the DED character graphics editor or DED2 character graphics editor. The problem is to design a 16:1 MUX which has TTL input and ECL output with a single +5V power supply. This requires the use of the TTL and ECL sections of the Q5000 macro library. The output is to pass through a D flip/flop. A parametric test gate tree has been added, along with extra power and ground and two thermal diodes.

The circuit selected demonstrates: 1) the use of mixed-mode +5V ECL/TTL I/O; 2) the use of the ECL output with a latch; 3) proper naming of wires and macros; 4) the use of extensive notes on the drawing page to document the design; 5) the use of the FOD fan-out derating net parameter; 6) the use of the SWGROUP switch group macro parameter; 7) the chip macro parameters; and 8) the logic required for the optional parametric test vectors. This is a flat design.

Logging in requires a user-name and may require a password. The AMCC seminar students use:

```
USER: CLASS
CLASS>
```

CLASS> is the prompt for all actions at the command level, set in the AMCC supplied loginfile.

If the /USER/CLASS user directory is not programmed, the system will come up at the top of the tree, i.e., at /, with a message to that effect. The user must make a directory:

```
MKDIR /USER/CLASS
```

and then change to it:

```
CD /USER/CLASS
```

From there, copy in an existing loginfile from another directory, if available:

```
COPY /NET/D_a/USER/CLASS/loginfile TO .
```

If none is available, use the text editor and create your own, for example:

```
TEC loginfile
%SUBMIT /AMCC/Qnnn_LIBS/QnnnnSETUP
 ... and whatever else you want

 there may be a networked datapath that
 needs to be in here - see your system
 manager
```

Activate the loginfile by logging off and logging back on or by:

```
%SUBMIT loginfile
```

Always verify that: 1) you are in the correct directory, in the correct current context; and 2) that the correct library has been referenced in the loginfile. There is another way to do this but it sometimes has problems.

Before starting schematic capture, any EWS user should make two checks: 1) that there is sufficient space for a new circuit (space on the hard disk); and 2) that the directory area to be used is either empty or loaded with known files. Check on the system disk memory available by typing:

```
SPACE
```

Display an inventory of the directory by typing one of these commands (there are other options):

```
INV . -S                    [ S for sorted ]
INV . -L                    [ L for long   ]
INV /USER/CLASS -L -DE      [ DE for deep nest ]
```

The loginfile at AMCC is currently preprogrammed to select the Q5000 library, as shown above. (Other libraries may be used by altering the loginfile to reflect that library.) AMCC provides preprogrammed shell scripts. The users may choose to add/modify/delete shell scripts at their own location on their own systems at any time - move them to another directory first! Beginners should stay with the provided shell scripts until they understand the actual commands.

Following selection of the library, and the rest of the "housekeeping", the drawing editor DED is invoked by typing:

```
DED1:   DED 1           [ DED n ]          DED I
DED2:   DED2 1          [ DED2 n ]         DED II
```

See the DAISY DED II introduction booklet for the comprable DED2 commands. DED1 will be operable under DNIX 5.02 but the user should begin to switch to the new editor in preparation for the ACE system. This application note was written for DED1.

As soon as the editor is ready, a blank page is presented (if there is no page 1.DRAW already in the directory). For a new page, a border should be placed on the page and edited.

When beginning a multiple-page drawing, plan ahead. If a page will not be created by replicating an existing page, it will always start with a border. Create several blank pages by copying a page with a border, comments, etc. into several page files.

A border is placed one grid point up diagonally from the lower left corner. For any circuit, the border is: /AMCCPAGEB and is a component in the library. If the cursor is positioned, type:

```
DED1:   CO /AMCCPAGEB{EXTRACT}...{EXECUTE}
```

The page should be given a unique six-character alphameric name (such as PAGE1), and then notes should be added in the lower right-hand block to document the date, page number, drawn by and a description of what is on the page. Multiple notes are created by {EXTRACT}ing notes from other notes. Multiple names are not allowed but {EXTRACT} can be used to make a copy of a name.

```
Name:
DED1:   NA {SELECT}name{DEF}{PLACE}...{EXECUTE}
DED1:   NA {EXTRACT}{PLACE}...{EXECUTE}
```

This last version copies the same name from one place to another.

```
Note:
DED1:   NOT {SELECT}any text{DEF}{PLACE}...{EXECUTE}
DED1:   NOT {EXTRACT}any text{DEF}{PLACE}...{EXECUTE}
DED1:   NOT {EXTRACT}{PLACE}...{EXECUTE}
```

This last version copies the same note from one place to another.

A name or a note is deleted by opening the command through {SELECT} and then typing D{MACRO}EXECUTE}. A name or nore is edited by opening the command through {SELECT}, typing the new name or text and then typing {DEF}{PLACE}{EXECUTE}.

The body of the drawing itself is then created by first fetching the basic macros and positioning them on the page. Use a paper sketch to design the initial layout for a circuit, to minimize wasted partial macros, minimize cross-overs, clarify where replication of macros, blocks and pages will save design entry time, and to identify the macros that will be used. (Don't do seat-of-the-pants design at the screen! It leads to frustration and error. Plan ahead.)

For a new design, the first macro on page one should be a chip macro. Once it is properly parameterized, this macro will characterize the array as to its grade (military or commercial), its power supply, the type of ECL (10K or 100K) and power supply, and identify the product with a unique name and number. Parameters are added with a simplified form of the **PARAMETER** command.

For this design, the chip macro is Q5000TTTL10K (this design fits into a Q1300T).

```
DED1:   CO /Q5000TTTL10K{EXTRACT}{PLACE}...{EXECUTE}
```

Three parameters need to be attached to the chip macro, PRODUCT_NAME, DEVICE_NUMBER, PRODUCT_GRADE. A fourth, POWER_SUPPLY, does not apply to the I/O mode chosen (TTL10K is a single power-supply of +5V only). Position the cursor on the first of these parameters, located in the lower left side of the macro symbol, and type:

```
DED1:  PARA {SELECT}param-value{DEF}{PLACE}...{EXECUTE}
```

The {PLACE} can be skipped if you positioned on the first character of the parameter name on the graphic symbol. Repeat for all of the parameters listed on the chip macro.

The value possible for PRODUCT_GRADE is either MIL or COM. The default power supplies are listed on the chip macro. The only time the POWER_SUPPLY may be altered is for the STD ECL circuits, or MIXED ECL/TTL, where 5VREF, STD4 (-4.5V) or STD5 (-5.2V) can be specified. AMCC customer service assigns the PRODUCT_NAME and DEVICE_NUMBER.

Regardless of technology, the input pins on a chip macro are tied to global ground.

If the pin is a power pin (VCC), aim the wire up. If the pin is a ground pin, aim the wire down. The EWS and the software does not care - this is a human readable convention. The output of the chip macro is wired to a terminator or it is wired to a page connector. For this design the terminator is used. Use /LWTERM and connect

it to the macro output pin with a wire. Do not allow the pins of any two macros to touch as the system may reboot and you will loose the page.

Add any extra power or ground macros desired. For all AMCC Series arrays, the extra power and ground macros are ITPWR, ITGND and IEVCC. Regardless of technology, the extra power and ground macros, like the chip macros, have all input pins tied to GND. Draw a wire out and down and name the wire GND.

For the VCC or ITPWR pins, the wire is drawn up by convention. (IEVCC is a power pin in a +5V REF ECL system.) The output of the power or ground macro is terminated.

The basic wire command is:

W {MARK}...{EXECUTE}

Use {MARK} to draw bends. Using {MARK} twice without moving will cause DAISY to try to straighten out the wire. A partial wire requires specific connection steps befoe it can be picked up and continued. Dot-connects are formed by anding one wire on top of another (not at the end). There are on-page connectors that allow a cumbersom wire to be "broken" on a page. These should be used with care to avoid unintelligible drawings.

Refer to the DED command summary for further wiring instruction. DED2 mouse wiring operations are sometimes preferrable. Any design begun in DED2 can switch back and forth between editors without leaving the DED environment. Any drawing begun in DED1 may be invoked from DED2.

Name the macros, including the chip macro (CHIP00). Power and ground macros have a naming convention as do static drivers. Try to follow AMCC naming conventions.

As an aid to debug and to avoid duplicate names which can be catastrophic, put the page number into the macro names. MX0000 on page four would become MX4000.

Proceed to the second drawing page by typing:

DED1: {NEXT}{EXECUTE}Y{EXECUTE}

Bring up a border on the new page, name and note it and begin to capture the schematic. To save time, when a border is brought up and commented for page one, copy it (use the SAVE command) to all following pages (SAVE 2; SAVE 3, SAVE 4, etc.).

All interface and logic Macros are components and are called by:

DED1: CO /macro name{EXTRACT}{PLACE}...{EXECUTE}

The 2{ZOOM} feature allows the entire drawing page to be seen and is a great assist in initial page layout. It will not detail the macros and you cannot wire while in 2{ZOOM}. (You can, but it won't be right.) Return to 1{ZOOM} or one step beyond (1{ZOOM}{ZOOM}) to do the wiring. Use the {VIDEO GRID} key to obtain a dotted background but turn it off (it is a toggle) when doing printout. Changing pages will reset ZOOM to the original 1{ZOOM} setting.

Use the {REDRAW} key to reconnect the displayed lines after names, notes and moves. What is on the screen is what is plotted. You can plot 1{ZOOM} and even higher - higher will provide interesting artwork but it will not be valid as a circuit schematic.

Following macro placement, the page and chip connectors and terminators are added. After these, the wires are added, and then the wires and macros are named and notes are added.

The DAISY connector names are decoded as:

R right	L left
W wire	B bundle or bus
H hierarchical	P page
I input	O output
CON connector	
PI page in	IP intrapage

There are page-to-page connectors (/RWPICON, /RWPOCON), bundled wire connectors (/RBHICON, /RBHOCON, /RBPICON, /RBPOCON), and bidirectional connectors (/RWPBCON, /RWHBCON, /RBPBCON, /RBHBCON). There are others.

The DAISY library also provides the terminators /LWTERM, /RWTERM, /UWTERM and /DWTERM. Other terminators, for bundles, bidirectionals, etc., also exist. Avoid using /UWTERM and /DWTERM since they resemble the ground symbol and this is visually confusing. Terminators are added by first fetching them from the library, and then wiring the terminator to the desired wire stub. Treat terminators as components, except for the fact they need not be named.

Wires are used to tie unused macro inputs to global ground. A wire is added to the input pin and drawn out and down. It is named GND and the symbol appears in place of the name. (Same as for the chip macro.)

DED1: NA {SELECT}GND{DEF}{PLACE}{EXECUTE}

Bundles are useful to route bundles of wires. The BU command is used in place of W in the routing of the bundle. Instead of a name, give a bundle a contents parameter (/CONTS) that contains the identification of what is in the bundle. There is a 64 character limit on the value of the /CONTS parameter. The /CONTS parameter is assigned by the full **PARAMETER** command:

PARA {SELECT}/CONTS{EXTRACT}xxxx{DEF}{PLACE}...{EXECUTE}

Wires coming from a bundle are individually named. DAT0, DAT1, ABUS12 are wires to the bundle with the /CONTS parameter DAT(0:15),ABUS(0:12).

All macros should be named in a flat design or in a hierarchy non-nested design, and all off-chip and off-page connections must be named. Internal wires may be named if you will be interested in them during debug of the design. Internal wires that MUST be named are the enable signals into 3-state output macros or bidirectional macros.

Internal wires (nets) in critical paths must be named or the default name must be made visible (via NA {SELECT}{PLACE}..{EXECUTE}) on the schematic. This is for use in Front- and Back-Annotation analysis. If you are going to take the trouble to make it visible - change its name! It takes little additional effort and speeds the design analysis process. Naming an internal signal allows it to be easily placed in the FMT file for wave and list display later on. Any signal that might help a debug problem should be named.

Hint: since the Front-Annotation file sorts the net names, begin all nets in a critical path with the same first three letters. The nets will sort near each other, making analysis easier. A full Q5000 design could easily have 900 nets.

The macro naming rules are by AMCC convention (see the EWS Schematic Rules and Conventions) including Lxxx for a latch, Bxxx for a buffer, Mxxx for a MUX, Dxxx for a driver, and so on. Macro instance names should be 1 to 6 characters long. Wire names should be 1 to 8 characters long, and should follow AMCC rules (alphameric only); no special characters. All names should be meaningful, which implies at least 3 characters. Names can begin with a number or a letter. As mentioned before, AMCC recommends that one character of the name be reserved for page identification to avoid interpage name duplication.

The rules are also designed to help avoid using a pin-name or other confusing names. They are designed to allow ease of transfer between various support programs. All names must be unique on a page and for the design.

For example, if AOUT appears on page 1.DRAW, it can only appear on page 1.DRAW one time as a chip/page input or chip/page output. A name can be extracted to appear somewhere else along the same wire or wirenet. A name can appear on several intrapage connectors (same-page connector - used to break a long wire for visual clarity). Each time the same wire name appears it is attached to the wire-net with that name.

The positioning of the wire names for on- and off-chip connectors: The names are attached to the wires but placed to the left of the input (/RWHICON) connectors and to the right of the output (/RWHOCON) connectors, or placed just above the wires.

All DAISY components may be rotated. AMCC prefers that rotation not be used since AMCC may run a graphic UPDATE of all submitted schematics against the in-house working library to verify the data. UPDATE does not rotate but puts the macros on the page in unrotated form, breaking connections and destroying the page. Structured design flow is left to right across the page and this flow should be maintained.

At any time during the creation of the drawing page, the page may be saved by typing:

```
DED1:   SAVE {EXECUTE}
DED1:   SAVE n {EXECUTE}          [ where n is a digit ]
```

Perform SAVE once every 15-20 minutes or when a complex structure has been entered or edited.

The use of "SAVE" with a file reference (usually a number for an n.DRAW page) allows pages to be copied while in the editor.

At any time during the creation of the drawing page the page may be plotted by typing:

```
    PL {EXECUTE}.              <--- direct connect
    PL /NET/D_n{MACRO}{EXECUTE}  <----- thru another
```

The second version is for plotting through another ETHERNET node at AMCC.

If there are no more pages or the page is complete or the user is interrupted, then DED is exited by:

```
DED1:   EXIT {EXECUTE}     save
DED1:   QUIT {EXECUTE}     do not save
```

The use of "QUIT" instead of "EXIT" will cause the current workspace page to be lost and the contents of the hard disk file will remain unchanged. If a file has been SAVEd then a QUIT is reasonable; or if an edit session has been abortive, then QUIT is preferred. QUIT will prompt the user to verify that drawing page edits are to be discarded.

Use of the menu and the mouse. Type the blue button to find the menus and select "WINDOW MANAGER".

Use the blue button to select "Plot Screen". Do not use "Plot Window". By toggling the yellow button twice, the entire screen is plotted. The yellow button can also be used to block out that part of the screen that is desired and a partial plot made.

Other selections available are "Shell Window". A shell window can be opened by {SHIFT}{ZOOM}. After use, it is deleted by {CTL}{E}. The cursor must be in the window for the command line to function.

Flip a window backward and forward in the window stack by selecting the yellow button on the banner for the window.

The use of multiple windows will slow down the system due to overhead. Limit the windows to two and use no sub windows when running simulation. To speed simulation, some users turn off NCP (sign off the net).

The next pages show the circuit created, a 16:1 MUX with two thermal diodes and with a gate-tree to allow parametric testing. The parametric tree inputs were taken from the unused pins on the input macros (all were YN), allowing the testing logic to be added without adding time delay to the actual function. This may not always be possible.

The parametric logic will need to be tested with functional vectors - the sample shown herein is 100% fault-coverage of the function but not necessarily of the test logic. Any expansion of the vector set needed to fault-cover the gate tree is left as a student exercise.

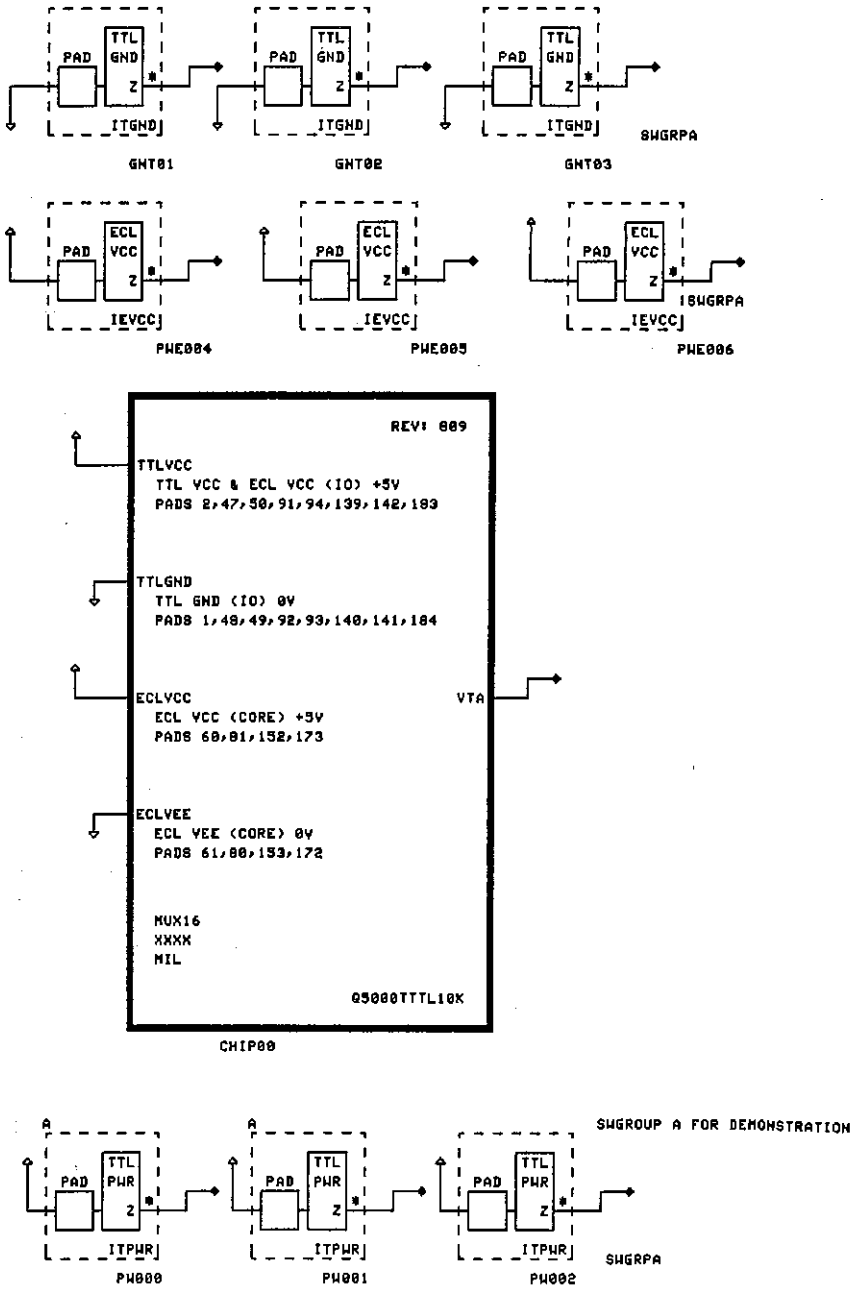

PAD
TTL GND
Z
ITGND
GNT01
GNT02
GNT03
SWGRPA
ECL VCC
IEVCC
PWE004
PWE005
PWE006
REV: 009
TTLVCC
TTL VCC & ECL VCC (IO) +5V
PADS 2,47,50,91,94,139,142,183
TTLGND
TTL GND (IO) 0V
PADS 1,48,49,92,93,140,141,184
ECLVCC
ECL VCC (CORE) +5V
PADS 60,81,152,173
VTA
ECLVEE
ECL VEE (CORE) 0V
PADS 61,80,153,172
MUX16
XXXX
MIL
Q5000TTTL10K
CHIP00
SWGROUP A FOR DEMONSTRATION
TTL PWR
ITPWR
PW000
PW001
PW002
SWGRPA

FIGURE 1

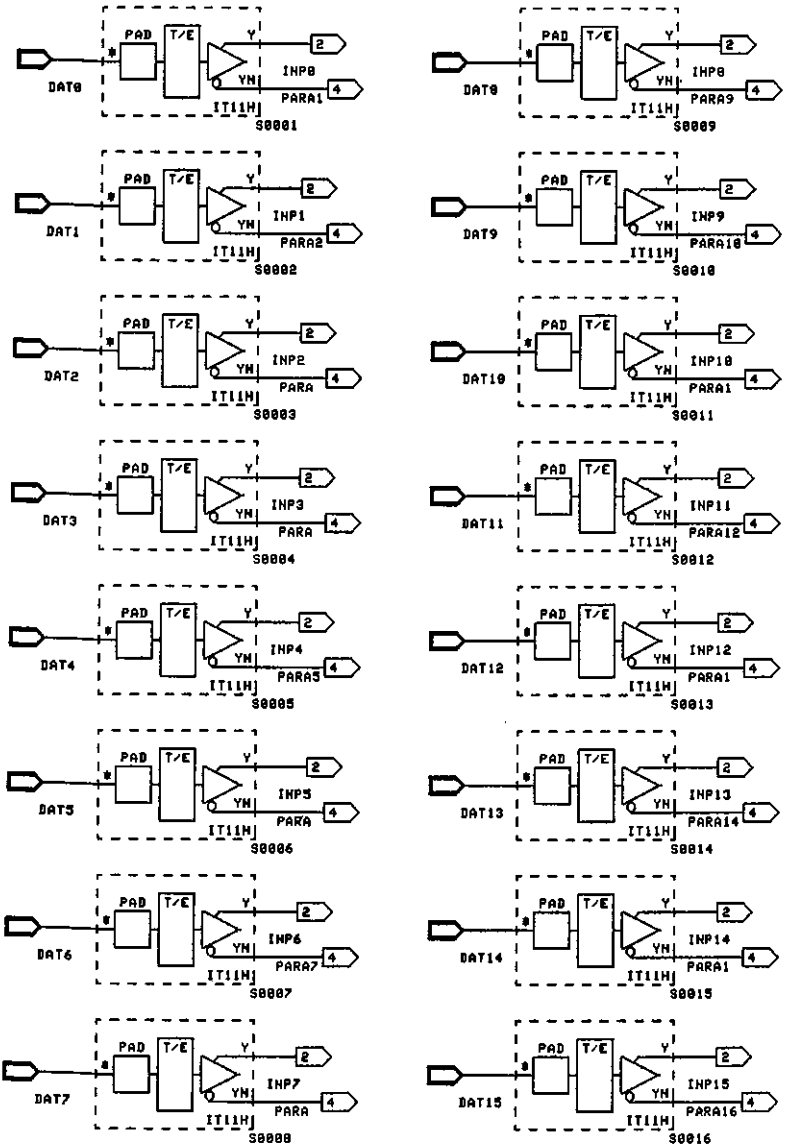

DAT0
PAD
T/E
Y
YN
2
INP0
PARA1
4
IT11H
S0001
DAT1
INP1
PARA2
S0002
DAT2
INP2
PARA
S0003
DAT3
INP3
S0004
DAT4
INP4
PARA5
S0005
DAT5
INP5
S0006
DAT6
INP6
PARA7
S0007
DAT7
INP7
S0008
DAT8
INP8
PARA9
S0009
DAT9
INP9
PARA10
S0010
DAT10
INP10
PARA1
S0011
DAT11
INP11
PARA12
S0012
DAT12
INP12
S0013
DAT13
INP13
PARA14
S0014
DAT14
INP14
S0015
DAT15
INP15
PARA16
S0016

FIGURE 2

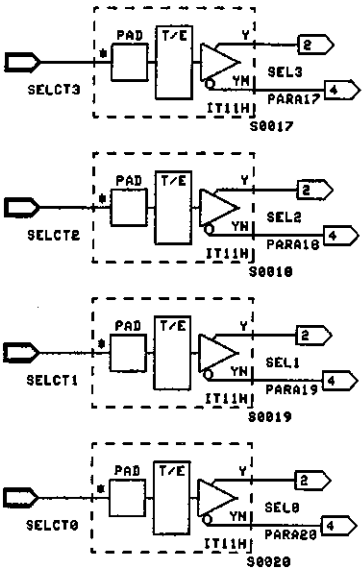


FIGURE 2 CONTINUED

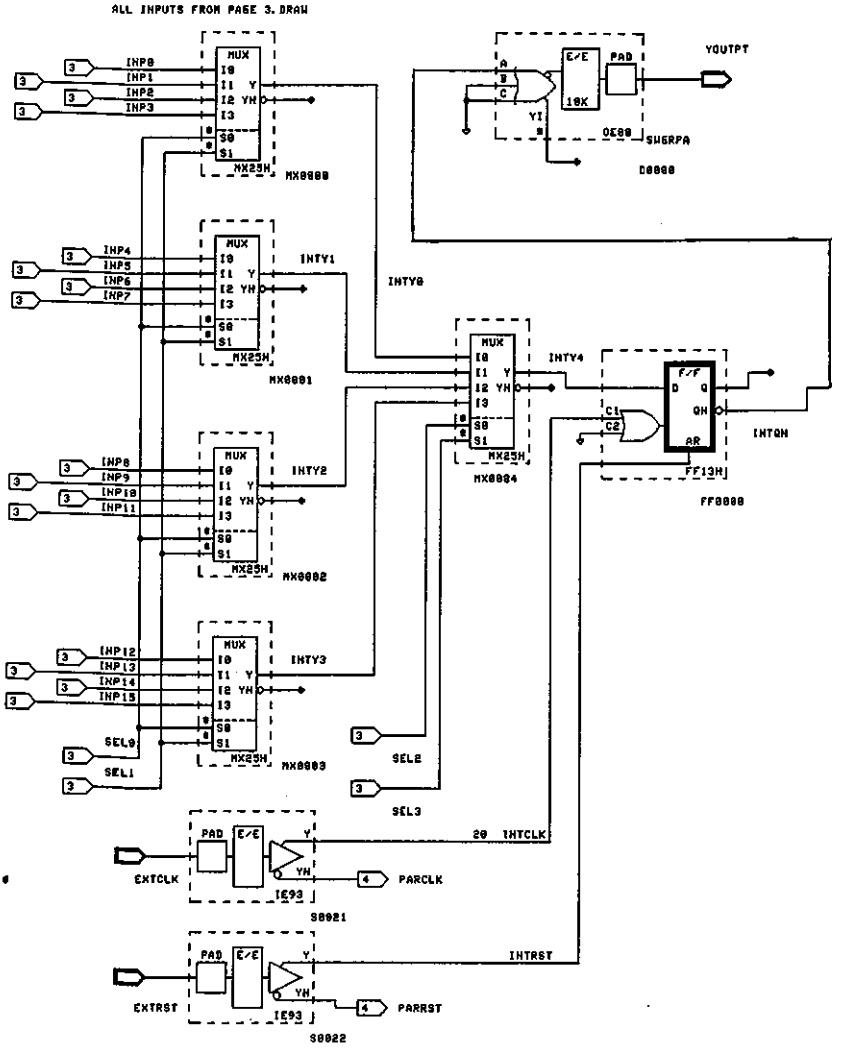

ALL INPUTS FROM PAGE 3.DRAW
INP0
INP1
INP2
INP3
MUX
I0
I1 Y
I2 YH
I3
S0
S1
MX25H
MX0000
INP4
INP5
INP6
INP7
INTY1
MX0001
INP8
INP9
INP10
INP11
INTY2
MX0002
INP12
INP13
INP14
INP15
INTY3
MX0003
SEL0
SEL1
SEL2
SEL3
INTY0
INTY4
MX0004
E/E
PAD
YOUTPT
A
B
C
YI
10K
0E00
SWERPA
D0000
F/F
D
Q
QH
AR
C1
C2
INTQH
FF13H
FF0000
EXTCLK
E93
S0021
PARCLK
INTCLK
EXTRST
S0022
PARRST
INTRST

FIGURE 3

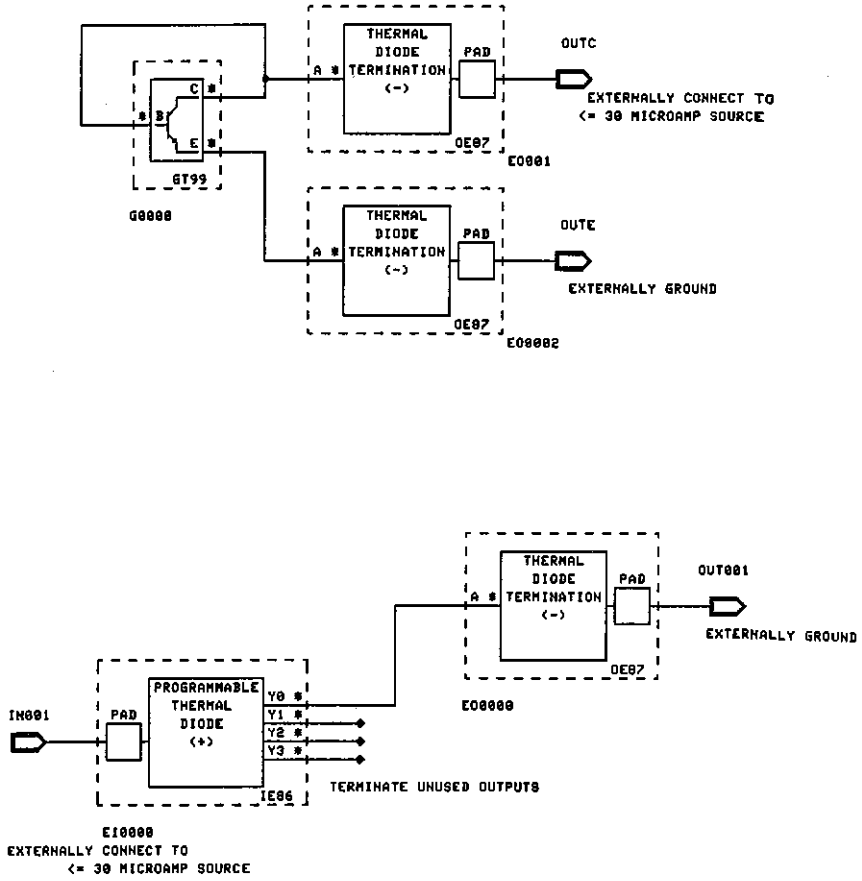

THERMAL
DIODE
TERMINATION
(-)
PAD
OUTC
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE
OE07
E00001
C
B
E
GT99
G0000
THERMAL
DIODE
TERMINATION
(-)
PAD
OUTE
EXTERNALLY GROUND
OE07
E00002
THERMAL
DIODE
TERMINATION
(-)
PAD
OUT001
EXTERNALLY GROUND
OE07
E00000
PROGRAMMABLE
THERMAL
DIODE
(+)
PAD
IN001
Y0
Y1
Y2
Y3
IE06
TERMINATE UNUSED OUTPUTS
EI0000
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE

FIGURE 4

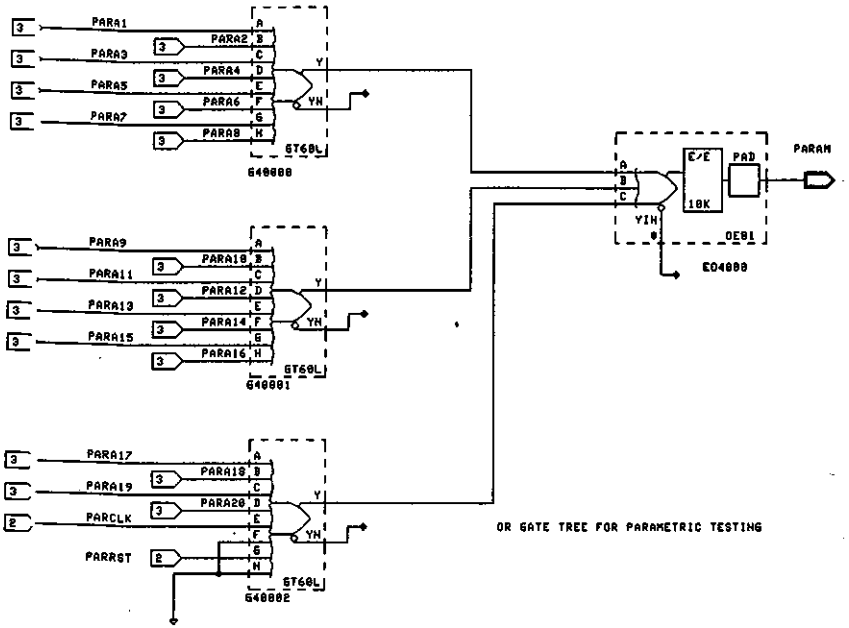

PARA1
PARA2
PARA3
PARA4
PARA5
PARA6
PARA7
PARA8
GT60L
G40000
PARA9
PARA10
PARA11
PARA12
PARA13
PARA14
PARA15
PARA16
GT60L
G40001
PARA17
PARA18
PARA19
PARA20
PARCLK
PARRST
GT60L
G40002
C/E
PAD
10K
YIH
OEBI
E04000
PARAM
OR GATE TREE FOR PARAMETRIC TESTING

FIGURE 5

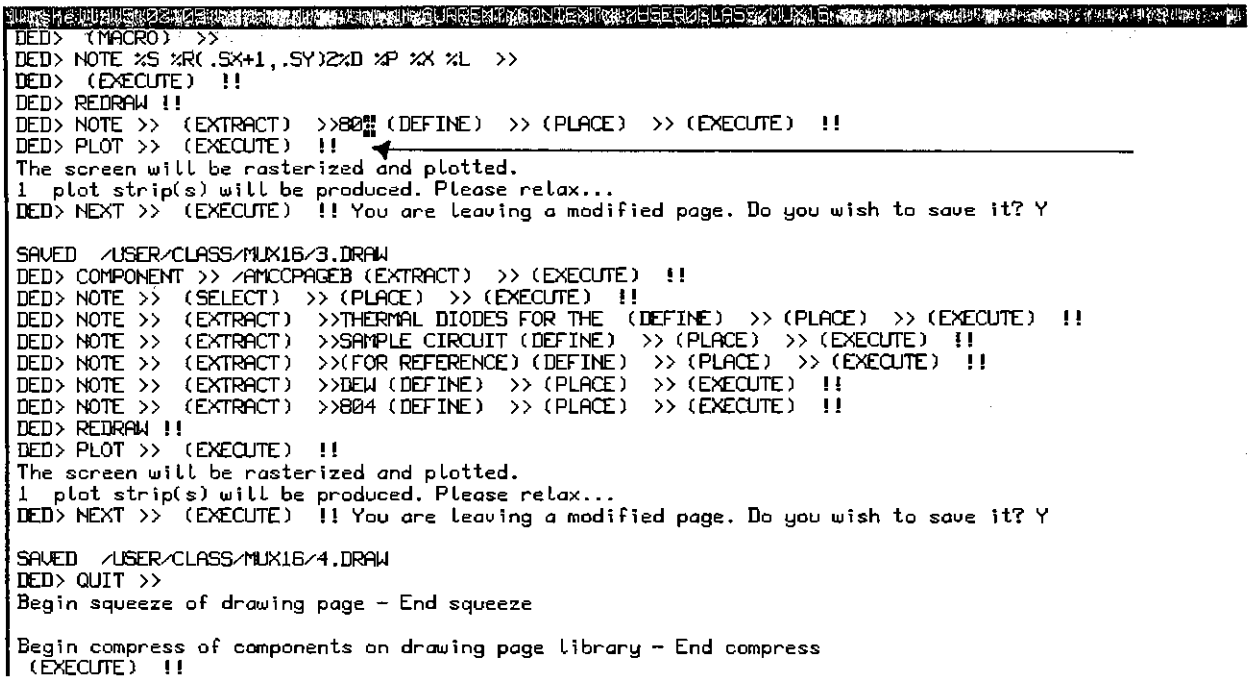


FIGURE 6

AFTER DED - LOG OF ACTIONS ON THE SCREEN

To copy drawing pages when outside of one of the DED editors, copy one page to another by:

```
COPY filename TO filename
```

When inside any DED editor simply save the page:

```
SAVE n (RETURN)
```

Initialize a floppy by:

```
INITDISK -DEMO    ..... ONLY IF NOT INITIALIZED
                        (or you are wiping the file)
```

Use an initialized or previously used floppy by:

```
MOUNT /F          ..... ALWAYS MOUNT THE FLOPPY
```

Note: Never delete the /F directory. If you do not mount a floppy before executing a copy command where /F is the destination, the file(s) are put on the main hard disk in the /F directory. Do not erase /F if you do not know what is in the directory (if the files are not your own).

Erase the /TEMP directory contents periodically. These are created during print or plot. These files are DAISY software temp files. A large /TEMP directory can cause system failures.

```
   COPY 1.DRAW TO /F       ..... WHATEVER
or COPY 1.DRAW TO /F -B
   COPY /F/* to . -B -DE .... reloading from floppy
   COPY * TO /F          ..... overkill; gets BAK, etc.
   COPY * TO /F -B -DE
   DIS                   .....WHEN DONE, DISMOUNT
```

Note: Backup copies of drawings should be kept on a floppy disk or tape. Make a back-up everyday. Keeping copies on the disk in another directory will slow down the system and waste memory. If there is a hard disk failure, you would loose both directories anyway. If you are working at the AMCC design center, ask about storage on the EAGLE hard disk.

Do not leave a floppy on-line during execution. Mount it, use it and dismount. Floppy drives are shared drives at AMCC.

The circuit shown in Figure 1 is flat (all pages in one directory area, no design tree), is not nested, has no parameterized components other than the chip macro, and does not use bundles, blocks or cells, although these features are available on the DAISY. It is a simple design suitable for instructional use.

Hierarchical design rules are covered in Application Note 2. Nested design rules are covered in Application Note 3.

When setting up the floppy, use a master index to keep track of what has been saved. One trick that AMCC likes you to use is the creation of a master index for your design files by typing:

```
INV /F -S -L -DE > /F/INDEX.INV
```

The file INDEX.INV contains a complete inventory of what has been stored on the floppy. The first execution creates an empty INDEX.INV file. You can also do this within a directory as a means of indexing that directory.

The next step is to DANCE the design (DAISY Network Connectivity Extractor). All pages in a design may be DANCEd at once or they may be processed individually. Call DANCE by:

```
DANCE                 one page
DANCE -M3 -T          all pages, tree, individual
                            short reports
DANCE -M3 -T -N       all pages, nested mode,
DANCE -M3 -T -ERR     concatenated reports
```

M3 is a message level, and is usually sufficiently detailed. You can change from M3 to M5 for more detail, or to M1 for less. DANCE will flag errors to the screen and to files x.DFR. The files n.DFR will be produced for this circuit for the first three versions of the command; file ERR.DFR will be produced by the concatenated request. By omitting R, the report will only contain error messages. This is preferred to keep memory usage down.

```
TYPE 1.DFR              use {OUTPUT CONTROL} to read
TYPE 1.DFR > /DEV/LP to line printer (plotter)
TYPE 1.DFR > /NET/D_a/DEV/LP  plot through AMCC
                        remote node to plotter
TYPE ERR.DFR            concatenated report
TYPE ERR.DFR > /DEV/LP -HEA  type with header
```

Use this last option (-HEA) with TYPE to keep a header on all outputs. It saves time during submission assembly.

Whenever there is a question about a DAISY command, use HELP and select the command you need help with. Figure 2 shows the screen during the HELP command with the system uppercase command TYPE selected for explanation.

```
CLASS> RUN_DD

        ****** RUNNING DANCE -T -N -ERR -M3 ********

        ************ DANCE OK !!!!! ***************

        *********** RUNNING DRINK -T -M3 ***********

        ************ DRINK OK !!!!! ***************
```

```
Date:  25 MAY 86 Ø8:44   File:  DANCE.ERR

LOGICIAN DANCE VERSION V5.Ø2.Ø2

DANCE :   BEGIN BLOCK /USER/CLASS/MUX16
DANCE :     BEGIN PAGE 1
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 2.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 3.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 4.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :   END BLOCK

DANCE COMPLETE:     4 PAGES FLAGGED. HIGHEST SEVERITY WAS Ø
```

```
Date:  25 MAY 86 Ø8:44   File:  DRINK.ERR

LOGICIAN DRINK VERSION V5.Ø2.Ø2

DRINK:  BEGIN COMPLETE LINK
DRINK:    TRAVERSING TREE
DRINK:      BEGIN BLOCK /USER/CLASS/MUX16
DRINK:        COLLECTING RECORDS
DRINK:        PROCESSING BLOCK DATA
DRINK:      END BLOCK     Ø ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DRINK:    PROCESSING TREE DATA
DRINK:      LINKING ENTRIES
DRINK:      BLOCK TAB LOADED
DRINK:      PAGE  TAB LOADED
DRINK:      RESERVING MAP
DRINK:      SORTING RECORDS
DRINK:      MERGING RECORDS

DRINK COMPLETE:     Ø BLOCKS FLAGGED. HIGHEST SEVERITY WAS Ø
```

FIGURE 7: RUN_DD, DANCE.ERR, DRINK.ERR

DANCE can be executed for one drawing page as the next is being edited. Move off the page to be danced, open a shell window and execute the DANCE command for the completed page. This is a preferred way of working since commonly made errors that DANCE could catch are detected early before they become ingrained bad habits. Anytime an individual page is edited, it can be re-DANCED by itself.

If DANCE does not produce any errors, the next step is to run DRINK (DAISY Resolving Linker). DANCE acts as a compiler, while DRINK acts as the corresponding linker. DRINK is run against the entire circuit. (There are sophisticated alternatives for update mode that will not be covered here.) Call it by:

```
DRINK                 short error-only report
DRINK -T -M3 -E3      messages to the screen
```

If there are errors in file TREE.DFR then you must return to DED to the page or pages flagged and make the corrections. The page(s) must repass through DED, DANCE and DRINK until there are no errors. Note: DRINK can be executed on a partial circuit; DAISY will simulate a partial circuit, MacroMatrix will execute on a partial circuit. A design cannot be SUBMITTED until all errors are removed.

DANCE and DRINK both can be executed by typing the shell execution command:

```
RUN_DD                (recommended)
                        (part of Super-Shell)
```

RUN_DD will produce DANCE.ERR (a concatenated report of all pages) and DRINK.ERR (same as TREE.DFR). Scan DANCE.ERR before proceeding with the ERCs and other software. There should be two errors per page (the border has no pins and is not connected to anything - a pinless macro). If more than two errors, go fix the page. (Note: The Super Shell will continue to execute with level 0 errors - it currently only halts on level 1 errors.)

If this process has been done before and you think all errors are removed, or you want to fire off processes while you do something else, call the AMCC super-shell and run DANCE, DRINK, AGIF, and ERC.

```
RUN_AMCC              (The Super-Shell)
1
```

If there are no errors from DRINK, and you have not run the Super-Shell, run the netlist transfer file generation software by first executing:

RUN_AGIF (part of Super-Shell)

This routine changes the DAISY-produced netlist into the AMCC generic interface format file (AGIF) (using SING) that will be transmitted to the VAX for layout and other software access. It will also create files in a new sub-directory /ERC.

The principal file that is created is CIRCUIT.SDI, the AMCC-formatted netlist. This file is a design-submission file. It is an input file for: ERCs, Front-, Intermediate- and Back-Annotation, AMCCSIMFMT, AMCCVRC, place and route, LASAR6 simulation and fault-grading and the tester programs. Do not type this file out - it is very large and is encoded.

The AMCC ERC (Engineering Rules Check) software must be run before proceeding with DAISY simulation. The DAISY system doesn't care about this (it is a parallel process) but you should. Why waste time performing and evaluating simulations when there are fundamental interconnection or population errors in the design? Running the ERCs first saves design time.

The ERC software is system-resident and is executed by moving down to the ERC subdirectory created by RUN_AGIF and typing:

AMCCERC (need to change directory)

The AMCC shell script **RUN_ERC** can be run <u>without</u> moving down to the ERC subdirectory. RUN_ERC calls AMCCERC.

RUN_ERC (part of Super-Shell)
(preferred execution)

The AMCC ERC program is a set of routines that will flag various errors such as fan-out exceeded, unconnected pins, improper wire-ORs, invalid names, grounded outputs, duplicate names, and other similar errors. Type out the reports to the screen by:

```
TYPE ERC/AMCCERC.LST
TYPE ERC/AMCCIO.LST
TYPE ERC/AMCCXREF.LST
```

If there are ERC-flagged errors, you must return to DED to fix the pages flagged. This is one reason why individual steps might be more efficient than the Super-Shell for the first pass through the drawing.

Before the next step can be run, the changed pages must be re-DANCED, the circuit re-DRINKed, SING-TO AGIF rerun

and ERC re-executed with no errors. Any time an error is found, the process must be restarted. The ERC software should pick up most of the more commonly encountered oversights and slips made in a design schematic capture.

The ERCs assume that catastrophic DANCE and DRINK errors are removed. If there is an incorrect pin-wire interconnect (contact not really made) it can survive the ERCs but fail in simulation. Unless you are running a partial circuit, always remove DANCE and DRINK errors BEFORE running ERCs. If you are running a partial circuit, remove those errors pertinent to the pages that are captured.

After the ERC software is executed without error, execute the Front-Annotation software while in the /ERC subdirectory by typing:

```
AMCCANN          (need to change directory)
```

Again, the AMCC shell script RUN_ANN can be run without moving to the ERC subdirectory.

```
RUN_ANN
```

Or by using the Super-Shell

```
RUN_AMCC              (part of Super-Shell)
2                     (preferred execution)
```

In the delay files produced, each net is identified by name (user-defined or default) and is followed by six numbers representing the min, typ and max net delay for both rising and falling edges. Only one file is referenced in the simulation at one time. These Front-Annotation delay files provide the internal net interconnect delays due to fan-out, wire-ORs and metal loading. The metal load delay is estimated.

The output net delays due to system and package pin capacitance loading is also computed. The program will prompt for a response which can be as simple as defaulting all values or as intricate as specifying different system and package capacitance loading for each of the primary output signals in the circuit. The package type selected determines the default values for the package pin capacitance.

The default system load is 15pF for TTL and 5pF for ECL. The output macros for the AMCC arrays in the Q14000, Q5000, Q20000 and future arrays are specified under no load.

```
CLASS809> RUN_AMCC

APPLIED MICRO CIRCUITS CORPORATION

        1: Part One:   Run DANCE, DRINK, AGIF and ERCs

        2: Part Two:   Run AMCCANN

        3: Part Three: Run SIFT and SOM_MAKER

        4: Part Four:  Run SOM and TCAL

        5: Part Five:  Run AMCCSIMFMT

        6: Part Six:   Run AMCCVRC

        0: Exit Program:

        Please Enter [0 - 6] : >2

        ************** RUNNING AMCCANN *************

        Need to Edit Package Pin Data? (Y or N) ; >Y

AMCC Delay Annotation VERSION  3.30
Loading Netlist ...
Welcome to the output loading system.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 1
```

Running AMCCANN

FIGURE 8

```
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 1
          0) OTHER.
          1) 224 PGA cavity down
          2) 132 leaded chip carrier cavity up.
          3) 196 leaded chip carrier cavity down.
          4) 149 PGA cavity down.
          5) 149 PGA cavity up.
Type the number of the package that this design will be using.
1
224-PGA-CD
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 2
The current default value for package pin capacitance is 4.9 pf.
Enter <Retn> for no change or enter a new value:
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 3
The current default value for TTL system capacitance is  1.5e+01 pf.
Enter <Retn> for no change or enter a new value: 20
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:
```

FIGURE 8 CONTINUED

```
MW.PASCAL V5.[illegible] CURRENT CONTEXT: /USER/CLASS/MW/
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 4
The current default value for ECL system capacitace is 5.0 pf.
Enter <Retn> for no change or enter a new value: 6
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 5
OUT001        OUTC          OUTE          PARAM         YOUTPT

Enter the signal name or signal names separated by spaces.
YOUTPT
Enter the new value (pf): 12
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 6
OUT001        OUTC          OUTE          PARAM         YOUTPT

Enter the signal name or signal names separated by spaces.
PARAM
Enter the new value (pf): 8
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 
```

FIGURE 8 CONTINUED

```
MW_PASCAL V5.1.2                    CURRENT CONTEXT: /USER/CLASS/MUX1
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 7
DAT0        DAT1        DAT10       DAT11       DAT12
DAT13       DAT14       DAT15       DAT2        DAT3
DAT4        DAT5        DAT6        DAT7        DAT8
DAT9        EXTCLK      EXTRST      IN001       OUT001
OUTC        OUTE        PARAM       SELCT0      SELCT1
SELCT2      SELCT3      YOUTPT
Enter the signal name or signal names separated by spaces.
OUTC
Enter the new value (Ohms): 50
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 8
DAT0        DAT1        DAT10       DAT11       DAT12
DAT13       DAT14       DAT15       DAT2        DAT3
DAT4        DAT5        DAT6        DAT7        DAT8
DAT9        EXTCLK      EXTRST      IN001       OUT001
OUTC        OUTE        PARAM       SELCT0      SELCT1
SELCT2      SELCT3      YOUTPT
Enter the signal name or signal names separated by spaces.
PARAM YOUTPT
Enter the new value (MHz): 100
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:

Exiting and writing AMCCPKG.LST.
Processing MIN Front Annotation Delay File ...
Processing NOM Front Annotation Delay File ...
```

FIGURE 8 CONTINUED

AMCCPKG.LST

FIGURE 9

```
*****************************************************************
*          AMCC Output Loading System                           *
*          Version  1.ØØ                                        *
*****************************************************************
   Netlister version number =                    7Ø8
   SDI version number =                          2.1Ø
   Netlist generation date =                     2Ø SEP 1988
   Netlist generation time =                      9:58
   Engineering workstation type =                DAISY/DNIX
   Engineering workstation path name =           /USER/CLASS/MUX16
   Product Name =                                MUX16
   Product Number =                              XXXX
   Product Grade =                               MIL
   EWS Library =                                 Q5ØØØ
   EWS Library Rev =                             8Ø9
   Macro Parameter family =                      Q5ØØØ
   The ARRAY type =                              Q5ØØØT
   Chip Macro Name =                             Q5ØØØTTTL1ØK
   Circuit Family =                              Q5ØØØ
   Circuit Technology =                          P
   ECL Level =                                   1ØK

Package name = 224-PGA-CD
```

package selected

Item #	Signal Name	Instance Name	Macro Name	PAD	Frq MHz	ECL R	System pf	Package Capacitance Min pf	Package Capacitance Typ pf	Package Capacitance Max pf
1	DATØ	SØØØ1	IT11H							
2	DAT1	SØØØ2	IT11H							
3	DATIØ	SØØ11	IT11H							
15	DAT8	SØØØ9	IT11H							
16	DAT9	SØØ1Ø	IT11H							
17	EXTCLK	SØØ21	IE93		1ØØ.Ø					
18	EXTRST	SØØ22	IE93							
19	INØØ1	EIØØØØ	IE86							
2Ø	OUTØØ1	EOØØØØ	OE87				6.Ø	3.8Ø	4.4Ø	4.9Ø
21	OUTC	EOØØ1	OE87			5Ø.Ø	6.Ø	3.8Ø	4.4Ø	4.9Ø
22	OUTE	EOØØØ2	OE87				6.Ø	3.8Ø	4.4Ø	4.9Ø
23	PARAM	EO4ØØØ	OE81		1ØØ.Ø		6.Ø	8.ØØ	8.ØØ	8.ØØ
24	SELCTØ	SØØ2Ø	IT11H							
25	SELCT1	SØØ19	IT11H							
26	SELCT2	SØØ18	IT11H							
27	SELCT3	SØØ17	IT11H							
28	YOUTPT	DJØØØ	OE8Ø		1ØØ.Ø	55.Ø	12.Ø	12.ØØ	12.ØØ	12.ØØ
29	ITGND	GNTØ3	ITGND							

Using the interface, the frequency for any primary input or output may be specified (commentary documentation) and the ECL resistive loading for any ECL output may be specified (also commentary). Resistive loads should be specified when they do match that assumed for the macro (25 or 50 ohms). Frequency should be entered for any TTL I/O toggling faster than 50MHz and any ECL I/O toggling faster than 100MHz.

AMCCANN (AMCC Annotation) can be run as many times as required to fine-tune the simulation. The file OUTPUT.DLY is the data file created in the first session and then edited by successive executions. A previously entered specific pin capacitance or load capacitance for a signal is deleted (reverting to the default value) by giving its name and typing an * for the new value.

When AMCCANN is completed, it will have generated the Front-Annotation files FNTMIL.DSY or FNTCOM.DSY, FNTNOM.DSY and FNTMIN.DSY as well as the report file AMCCPKG.LST and the data file OUTPUT.DLY. All of these files are to be submitted.

Note: The BiCMOS library has two commercial timing libraries, one for COMMERCIAL circuits running with a -4.5V power supply (COM4) and one for all other COMMERCIAL circuits (COM5). The Front-Annotation file is still named FNTCOM.DSY and provides correct timing based on the ARRAY_FAMILY, PRODUCT_GRADE and the POWER_SUPPLY parameters.

After the ERC software and the Front-Annotation software has successfully run, the next step is to run a simulation to verify that the basic logic is correct.

When the proper loginfile, such as the one shown earlier, is used, the PROFILE file is correctly set for the library.

THE LIBRARY MUST BE INSTALLED AS AMCC INTENDS FOR PROPER OPERATION.

Run SIFT (Simulator Intermediate Files Translator) to append the appropriate timing data to the macros. These are the intrinsic delays for the paths through the macros themselves. SIFT can be run for MINIMUM, NOMINAL, COMMERCIAL or MILITARY timing. The worst-case multipliers are those that support the Front- and Back-Annotation software (1.35 * typical = COM and 1.45 * typical = MIL for the Q5000).

Run SIFT using the AMCC supplied shell:

RUN_SIFT xxx where xxx = NOM, MIL or COM, COM4 or COM5

Note: for large designs, SIFT may run overnight.

Within the SIFT library, there are three multipliers:

BIPOLAR

DLS/DTV	MODE ->	RANGE MIN	TYP	MAX	FRONT-ANNOTATION
SIFT Timing Library	NOM	0.90	1.00	1.10	FNTNOM.DSY
	COM	1.11	1.23	1.35	FNTCOM.DSY
	MIL	1.19	1.32	1.45	FNTMIL.DSY
	MIN	0.70	0.78	0.86	FNTMIN.DSY

BICMOS*

COM4 uses multipliers for -4.5V supply
- use with FNTCOM.DSY

COM5 uses multipliers for -5.2V or +5V supply
- use with FNTCOM.DSY

* see the tables in Volume I of the Design Manual

These numbers are based on a 20% maximum worst-case processing variation across the chip, with 10% the more usual variation. These multipliers are the same multipliers used in the FNTxxx.DSY files.

The use of MIN, TYP or MAX within a timing library is specified in the MODE format under the DAISY Logical Simulator (DLS) or the DAISY Timing Verifier (DTV).

SIFT is also run under the Super-Shell by typing:

```
RUN_AMCC                (Super-Shell)
3
```

AMCC recommends that the Super-Shell be used after the initial debug pass has been made since the menu will prompt you for necessary parameters and options for the commands.

Normally, a simulation will be run using MINIMUM data and then rerun using COMMERCIAL or MILITARY data. If the results are not functionally identical, a timing dependency (problem) is indicated which requires further evaluation or a re-design. AMCC requires both of these simulations (MINIMUM, and MAXIMUM MIL OR COM) be performed and submitted if they are different. Otherwise, submit just the maximum worst-case simulation results.

All submitted simulations should be run under timing checks "ON" and all timing errors removed. In special cases, an AMCC waiver may be required.

Following SIFT, the SING TO DAISY process is initiated. SING is the Simulator Input Generator used to extract information from the netlist. AMCC usually combines this into the following routines: 1) generate a default time=0 simulation control file, SOM_MCF.SING; 2) generate a default FMT file for DLS/DTV use; and open the SOM_MCF.SING file for editing under TEC.

```
SING -T -M3 -MCF /AMCC/SOM_MAKER/SOM_MCF
FMT_CSD.SING
TEC SOM_MCF.SING      (all part of Super-Shell)
```

At the end of the last command (and the end of menu selection 3 for the Super-Shell), the text-editor TEC is open to the simulation control file, SOM_MCF.SING. The time-zero SOM_MCF.SING file comes up with all input (all primary input signals) shown forced to zero at time zero (= 0:F0;). The user must edit this file to include the simulation stimuli by using the DAISY text editor, TEC. Stimuli can be described using the signal generator approach or by using a remote data file.

The example at the end of the text shows a SOM_MCF.SING file after it has been edited. This one uses the $SIGNAL_GENERATORS section to input values. TIMING_CHECK := 1; is present so this file can be used to run timing checks under both DLS and DTV.

AMCC prefers that all files be commented for identification (company, circuit, designer, date, rev level, what is being tested, etc.).

After all edits to the SOM_MCF.SING file have been made, TEC is exited by:

```
{ENTER}EXIT{EXECUTE}          (exit TEC)
```

The SOM CTL file must include an $OUTPUTS section. The output file referenced in the $OUTPUTS section is created by the system and not by the user. The user must describe which signals go in the file in what order. AMCC requires that all primary inputs, all primary outputs, all primary bidirectionals and all internal 3-state enable or bidirectional enable signals be listed in that file in that order.

The files produced during simulation include the LIST file, the WAVE, the VLAF output file. The VLAF output file is the file that becomes the input file to

```
Date:  25 MAY 86 11:17   File:  FMT_CSD.SING

SOM SOM_MCF.SING -M3
DLS <<!
FORMAT
0AT@@MUX16/3:DAT@
3DAT1@MUX16/3:DAT1
3DAT1@@MUX16/3:DAT1@
3DAT11@MUX16/3:DAT11
3DAT12@MUX16/3:DAT12
3DAT13@MUX16/3:DAT13
3DAT14@MUX16/3:DAT14
3DAT15@MUX16/3:DAT15
3DAT2@MUX16/3:DAT2
3DAT3@MUX16/3:DAT3
3DAT4@MUX16/3:DAT4
3DAT5@MUX16/3:DAT5
3DAT6@MUX16/3:DAT6
3DAT7@MUX16/3:DAT7
3DAT8@MUX16/3:DAT8
3DAT9@MUX16/3:DAT9
3EXTCLK@MUX16/2:EXTCLK
3EXTRST@MUX16/2:EXTRST
3IN@@1@MUX16/4:IN@@1
3SELCT@@MUX16/3:SELCT@
3SELCT1@MUX16/3:SELCT1
3SELCT2@MUX16/3:SELCT2
3SELCT3@MUX16/3:SELCT3
3OUT@@1@MUX16/4:OUT@@1
3OUTC@MUX16/4:OUTC
3OUTE@MUX16/4:OUTE
3YOUTPT@MUX16/2:YOUTPT
3S
PUT FMT
QUIT N
!
```

FIGURE 10: FMT_CSD.SING used to build FMT

FMT is used by DLS

AMCCSIMFMT. AMCC does not need to see the FMT file, LIST files, WAVE plots or the VLAF file.

The first three (FMT controls LIST and WAVE contents) are useful for debugging. For example, the user can reference internal signals in the LIST file while the final $OUTPUTS section is not allowed to show them.

The Super-Shell step 3 will end with the text editor open. When TEC is exited, the shell menu will return. If you need to create a remote data file, exit the Super-Shell and open the data file under TEC. If this is the case, exit the Super-Shell by:

```
0                          (exit Super-Shell)
```

If the input stimulus is to be provided by a remote data file, the TEC must be reopened to create whatever input file name was referenced in the simulation control file. To open, use TEC input-filename.

```
TEC DEMO.DAT               (example)
TEC filename.extension     user-defined filename
```

The data files are created from scratch by the designer; there is no fill-in-the-blank formatted file to edit as there is for the SOM_MCF.SING file. The data file must be created following a specified time-value, with comments and blank lines allowed. If comments and blank line appear after the END statement, then run AMCCFILUTL to prepare the file for execution. The source file is commented, the executable file is not. Maintanence is performed using the source file.

ALWAYS RENAME AN EDITED SOM_MCF.SING FILE. The SOM_MCF.SING file is destroyed in favor of a default version any time SING is rerun or anytime Super-Shell step 3 is executed.

AMCC prefers to have meaningful names used on the control files such as FUNCTION.SING, ATSPEED.SING, ACR4PROP.SING. Do not forget to include these in the submission index.

Once the data file and the SOM_MCF.SING file are completed, then run SOM, the Simulator Object Module Generator. This can be executed with the Front-Annotation file, and should be. The AMCC supplied shell is an easy way to do this.

SOM the control file only - for debug:

```
SOM SOM_MCF.SING -M3 -L SOM.ERR
```

```
/****
*
* DESIGN PATH /USER/CLASS/MUX16   DATE 16-SEP-1988 14:5Ø
*
* COMPANY  ________________  CIRCUIT_NAME  ____________
*
* ARRAY  __________  PO#  ____________  REV  ______
*
* DESIGNER  ____________________________
*
* What tests does this control file support:  __________
*
* ____________________________________________________
*
* ____________________________________________________
*
****/

/***   Configuration section    ****/

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

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

$SIGNAL_GENERATORS
@MUX16/3:DATØ := @Ø:FØ ;
@MUX16/3:DAT1 := @Ø:FØ ;
@MUX16/3:DAT1Ø := @Ø:FØ ;
@MUX16/3:DAT11 := @Ø:FØ ;
@MUX16/3:DAT12 := @Ø:FØ ;
@MUX16/3:DAT13 := @Ø:FØ ;
@MUX16/3:DAT14 := @Ø:FØ ;
@MUX16/3:DAT15 := @Ø:FØ ;
@MUX16/3:DAT2 := @Ø:FØ ;
@MUX16/3:DAT3 := @Ø:FØ ;
@MUX16/3:DAT4 := @Ø:FØ ;
@MUX16/3:DAT5 := @Ø:FØ ;
@MUX16/3:DAT6 := @Ø:FØ ;
@MUX16/3:DAT7 := @Ø:FØ ;
@MUX16/3:DAT8 := @Ø:FØ ;
@MUX16/3:DAT9 := @Ø:FØ ;
@MUX16/2:EXTCLK := @Ø:FØ ;
@MUX16/2:EXTRST := @Ø:FØ ;
@MUX16/4:INØØ1 := @Ø:FØ ;
@MUX16/3:SELCTØ := @Ø:FØ ;
@MUX16/3:SELCT1 := @Ø:FØ ;
@MUX16/3:SELCT2 := @Ø:FØ ;
@MUX16/3:SELCT3 := @Ø:FØ ;
```

SOM_MCF.SING

FIGURE 11

SOM and TCAL combined:

```
RUN_SOM SOM_MCF.SING FNTMIL.DSY
                          (part of Super-Shell)
```

Caution: RUN_SOM looks for the FNTxxx.DSY files in the top level directory. Always run DANCE, DRINK, SOM and SING from the top of the design tree.

SOM must be executed anytime the SOM_MCF.SING file is edited to link the new information to the simulation database.

```
RUN_AMCC  (if you aren't in the shell)
4                          select SOM
<control_file_name>        control file name
FNTxxx.DSY                 Annotation file name
0                          exit shell
```

When the steps represented in the AMCC Super-Shell are completed, the next step is to run either DLS, the DAISY Logic Simulator, or DTV, the DAISY Timing Verifier.

DLS is called by:

```
DLS
    GET FMT    using the default - use unless YOU want
               something different; leave no spaces
               after FMT
     MODE      set mode
    {edit for MAX}   for military or commercial;
    {edit for MIN}   for minimum
     {ENTER}
    PUT DLS_FMT      save the FMT and MODE windows
    VIEW 9999 10000  proper view step for Q5000 Series;
                        Q3500 Series; Q14000 Series
    RUN 1000000      run as long as you need to run
    START 0           position data in display buffer
    WAVE              plot by "Plot Screen" mouse menu
     {ENTER}
    LIST  S DEMO     listing saved as file "DEMO"
    {ENTER}
    RESTART 0
    VIEW 99 100      closer look - not for vectors
    RUN 200000           that will be submitted
    WAVE
      {ENTER}
    QUIT{EXECUTE}    don't save
  N{EXECUTE}
```

After DLS is run to satisfaction (the vector set in the example is 100% fault coverage), the vectors that will be submitted for use in test are generated using the AMCCSIMFMT program. This program will take the file produced by the $OUTPUTS section and reformat it. The

```
CLASS> RUN_SOM FUNCTION.SING FNTMIL.DSY

        ********** RUNNING SOM and TCAL **********

Daisy Simulator Object Module(SOM) Generator 5.2.9  (SOM V5.02.02        )25-May-86 15:32:00   FUNCTION.SING

 Copyright (C) 1984 DAISY SYSTEMS  Incorporated.

        **********  SOM FORMAT OK !!!!!  **********

TCAL:  TIMING CALCULATION, V5.02.02

TCAL:  INITIALIZATION
TCAL:      LOADING DATA FROM SOM FILE
TCAL:  PARSING INPUT FILE
TCAL:      PARSING CONTROL SECTION
TCAL:      PARSING DELAY SECTIONS


***** PARSING INPUT FILE COMPLETED.
***** TOTAL NUMBER OF ERRORS FOUND = 0

TCAL:  PARSING INPUT FILE COMPLETED, NO ERROR FOUND.


TCAL:      PARSING SIFT FILES

TCAL: NULL INPUT FROM SIFT FILES.

TCAL:
TCAL:  PROCESSING DELAY CALCULATION
TCAL:
  SIGNAL NAME:
      NOM RISE  MIN RISE  MAX RISE  NOM FALL  MIN FALL  MAX FALL

    @MUX16/2:INTQN(#86) :
           13, 12, 14, 26, 23, 28
    @MUX16/2:INTCLK(#94) :
           26, 23, 28, 26, 23, 28
    @MUX16/2:INTRST(#97) :
           26, 23, 28, 26, 23, 28
```

FIGURE 12: RUN_SOM

```
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP5(#603) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP6(#605) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP7(#607) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP8(#609) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP9(#611) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP10(#613) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP11(#615) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP12(#617) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP13(#619) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP14(#621) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP15(#623) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL3(#625) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL2(#627) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL1(#629) :
            36, 33, 40, 72, 65, 80
    @MUX16/3:SEL0(#631) :
            36, 33, 40, 72, 65, 80
    @MUX16/4:G0000.C(#652) :
            0, 0, 0, 0, 0, 0
    @MUX16/4:G0000.E(#653) :
            0, 0, 0, 0, 0, 0
    @MUX16/4:XSIG5(#654) :
            0, 0, 0, 0, 0, 0


***** DELAY CALCULATION COMPLETED.
***** TOTAL NUMBER OF ERRORS FOUND = 0

TCAL:  DELAY CALCULATION COMPLETED, NO ERROR
TCAL:
TCAL:  UPDATING DT TABLE
TCAL:      UNLOADING DATA TO SOM FILE
TCAL:  TERMINATION

        ********** TCAL FORMAT OK !!!!! **********

CLASS>
```

FIGURE 13: END OF RUN_SOM - TCAL

DLS output file format must be binary for a proper submission. An example AMCCSIMFMT output is shown with the example.

When 3-state macros are included in a design, it is often desirable to see the high-Z state in the listing during debug. The DLS/DTV format can be edited by:

```
FO
  edit          use switch key to toggle options
{ENTER}
PUT DLS_FMT       save for later reference
```

The FORMAT file and the MODE file can be printed out using the PLOT_SCREEN menu command. They can also be saved to a file for printing using the clipsheet options and they can be saved on disk and recalled in another run. All formats (FORMAT, MODE, ACQUIRE, BREAK, TRIGGER) are saved at the same time using:

```
PUT <filename>
```

Another feature that is useful is the PRINT_ON_CHANGE file which can be created. The PRINT_ON_CHANGE file can be requested by editing the SOM_MCF.SING control file $OUTPUTS section to provide the instruction to collect data only on the change of monitored signals, rather than collect sampled data as before. Only the output file is affected.

The format for the LIST and WAVE are tied to the FORMAT file and will not vary. The format for the output file is what is listed in the $OUTPUTS section of the simulation control file. Only those signals listed in the control file $OUTPUTS section will be monitored or recorded.

When the output file is the one desired, run AMCCSIMFMT by typing:

```
AMCCSIMFMT
```

and responding to the prompts.

You can also call the program using the Super-Shell:

```
RUN_AMCC
5               calls AMCCSIMFMT
```

You will be asked for the name of the output file (as listed in the $OUTPUTS section of the control file), the new name you want to use, what format you want with parenthesis (choose "1") and if you want spaces between columns or groups of columns. The spacing is up to you.

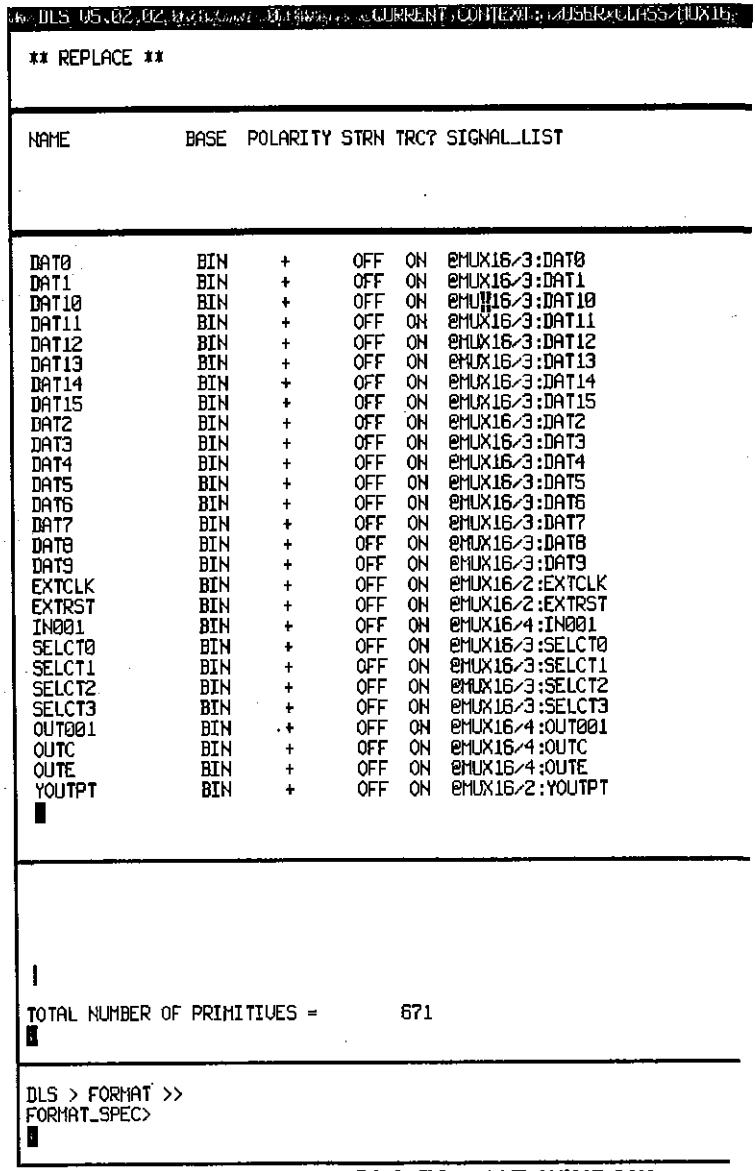


FIGURE 14: DLS FORMAT WINDOW

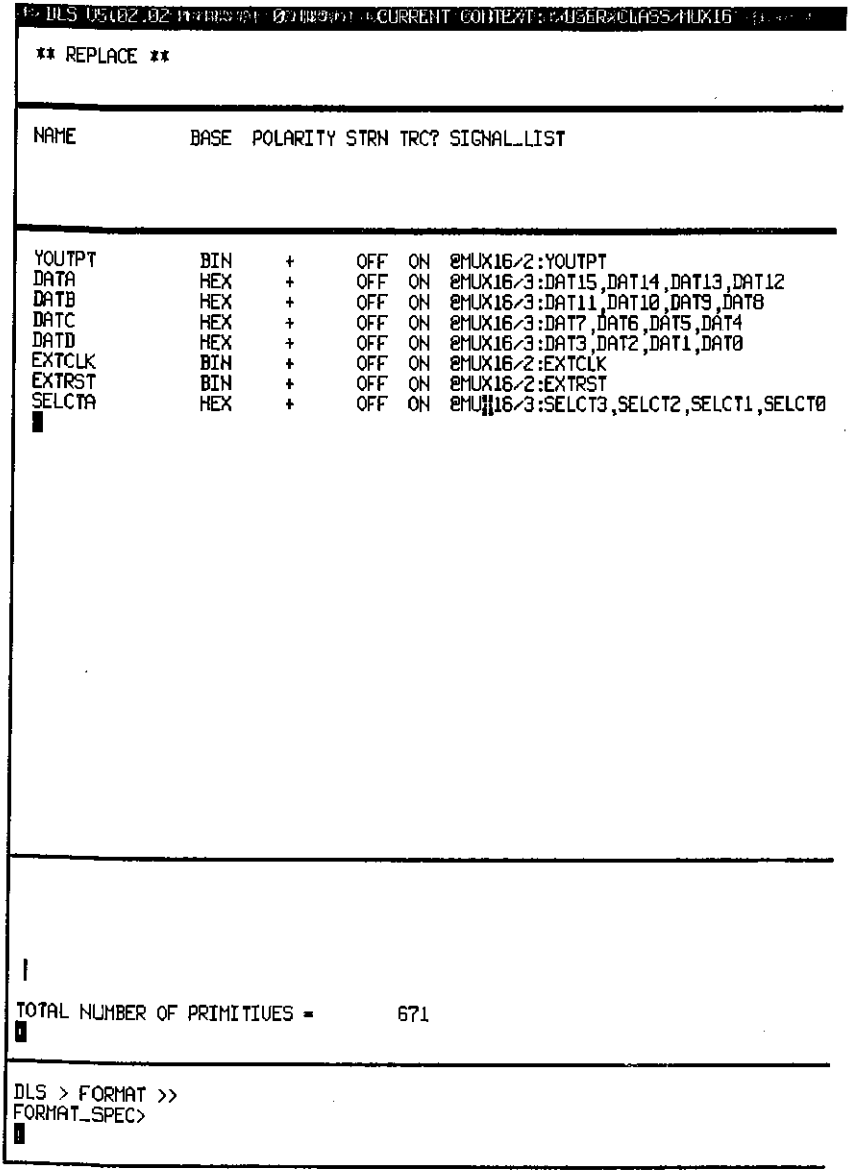


FIGURE 15: Edited DLS FORMAT WINDOW – HEX

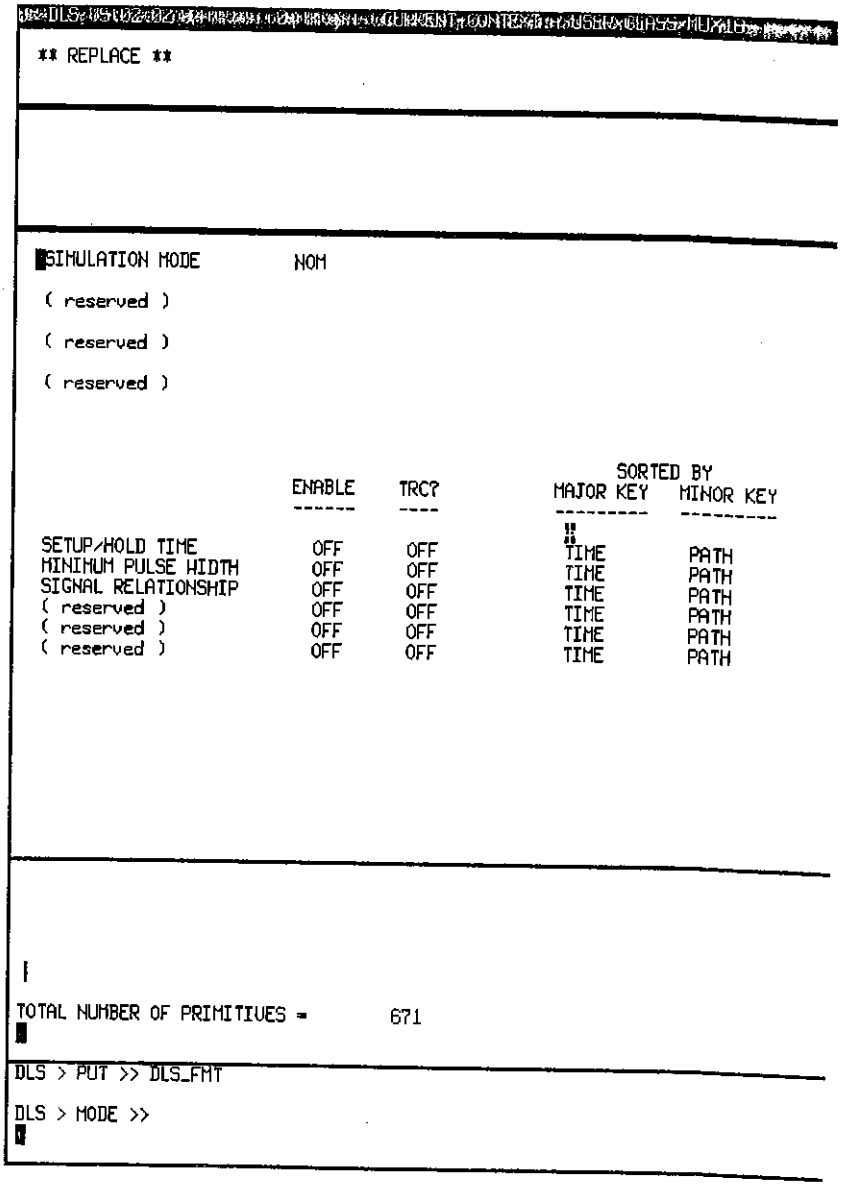


FIGURE 16: DLS MODE WINDOW – NOMINAL

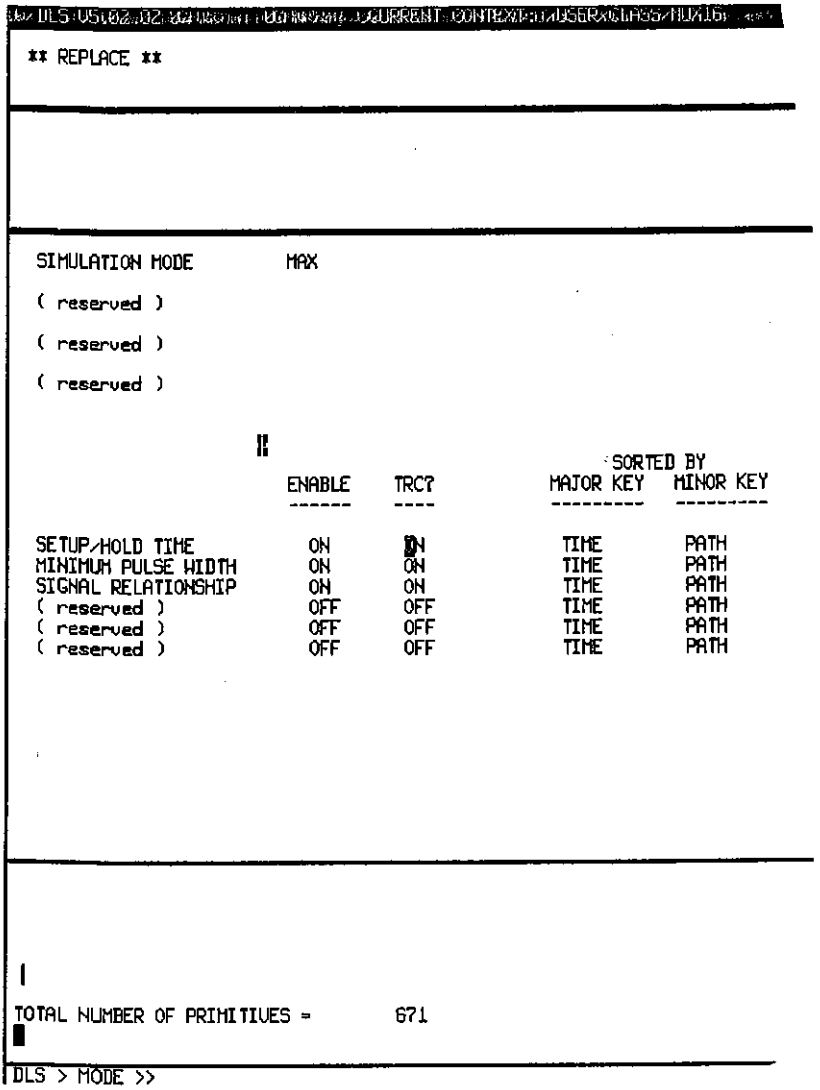


FIGURE 17: DLS MODE WINDOW – Edited for MAXIMUM

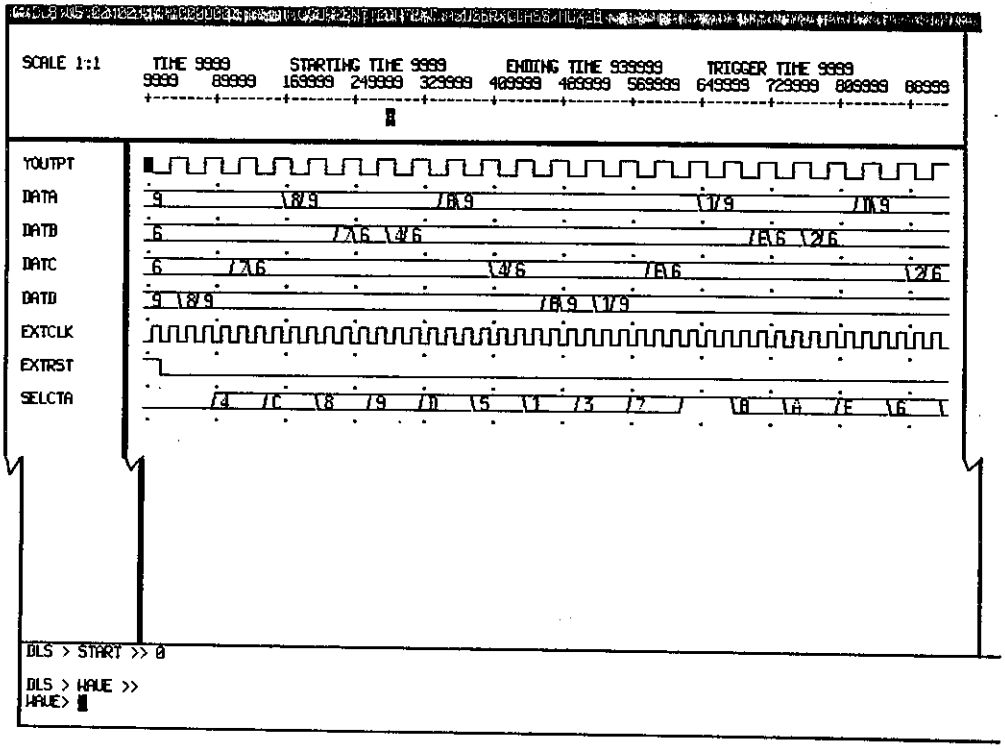
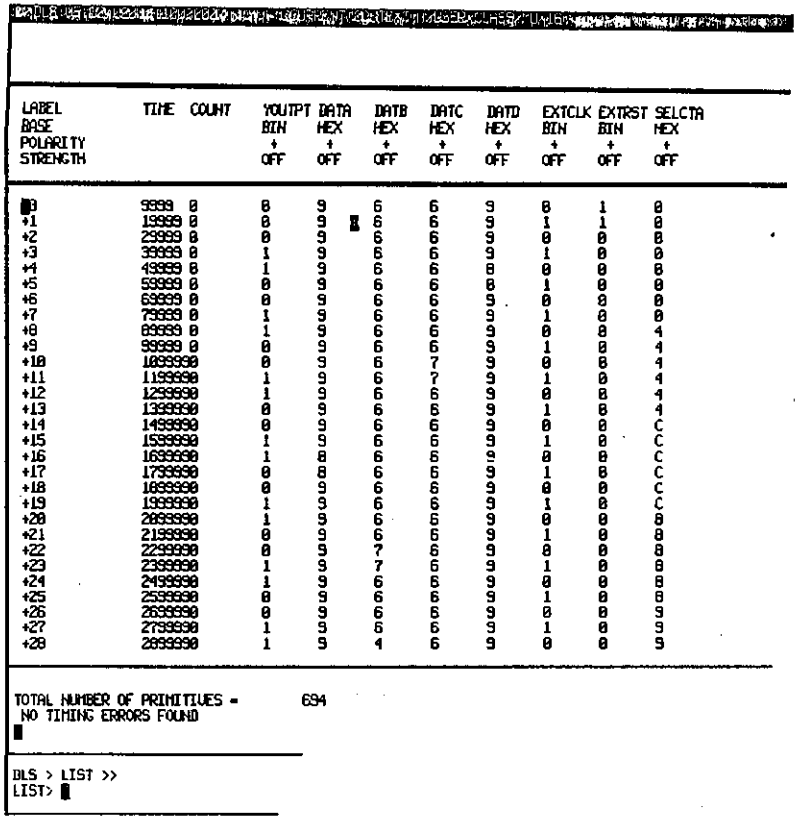


FIGURE 18: DLS WAVE - HEX FORMAT



LABEL BASE POLARITY STRENGTH	TIME	COUNT	YOUTPT BIN + OFF	DATA HEX + OFF	DATB HEX + OFF	DATC HEX + OFF	DATD HEX + OFF	EXTCLK BIN + OFF	EXTRST BIN + OFF	SELCTA HEX + OFF
0	9999	0	0	9	6	6	9	0	1	0
+1	19999	0	0	9	6	6	9	1	1	0
+2	29999	0	0	9	6	6	9	0	0	8
+3	39999	0	1	9	6	6	9	1	0	0
+4	49999	0	1	9	6	6	8	0	0	8
+5	59999	0	0	9	6	6	8	1	0	0
+6	69999	0	0	9	6	6	9	0	0	0
+7	79999	0	1	9	6	6	9	1	0	0
+8	89999	0	1	9	6	6	9	0	0	4
+9	99999	0	0	9	6	6	9	1	0	4
+10	1099998		0	9	6	7	9	0	0	4
+11	1199998		1	9	6	7	9	1	0	4
+12	1299998		1	9	6	6	9	0	0	4
+13	1399998		0	9	6	6	9	1	0	4
+14	1499998		0	9	6	6	9	0	0	C
+15	1599998		1	9	6	6	9	1	0	C
+16	1699998		1	8	6	6	9	0	0	C
+17	1799998		0	8	6	6	9	1	0	C
+18	1899998		0	9	6	6	9	0	0	C
+19	1999998		1	9	6	6	9	1	0	C
+20	2099998		1	9	6	6	9	0	0	8
+21	2199998		0	9	6	6	9	1	0	8
+22	2299998		0	9	7	6	9	0	0	8
+23	2399998		1	9	7	6	9	1	0	8
+24	2499998		1	9	6	6	9	0	0	8
+25	2599998		0	9	6	6	9	1	0	8
+26	2699998		0	9	6	6	9	0	0	9
+27	2799998		1	9	6	6	9	1	0	9
+28	2899998		1	9	4	6	9	0	0	9

TOTAL NUMBER OF PRIMITIVES = 694
NO TIMING ERRORS FOUND

DLS > LIST >>
LIST>

FIGURE 19: DLS LIST WITH HEX FORMAT

Date: 25 MAY 86 13:34 File: SAM

LABEL BASE POLARITY STRENGTH	TIME	COUNT	YOUTPT BIN + OFF	DATA HEX + OFF	DAT8 HEX + OFF	DATC HEX + OFF	DAT0 HEX + OFF	EXTCLK BIN + OFF	EXTRST BIN + OFF	SELCTA HEX + OFF
0	9999	0	0	9	6	6	9	0	1	0
+1	19999	0	0	9	6	6	9	1	1	0
+2	29999	0	0	9	6	6	9	0	0	0
+3	39999	0	1	9	6	6	9	1	0	0
+4	49999	0	1	9	6	6	8	0	0	0
+5	59999	0	0	9	6	6	8	1	0	0
+6	69999	0	0	9	6	6	9	0	0	0
+7	79999	0	1	9	6	6	9	1	0	0
+8	89999	0	1	9	6	6	9	0	0	1
+9	99999	0	0	9	6	6	9	1	0	1
+10	109999	0	0	9	6	7	9	0	0	1
+11	119999	0	1	9	6	7	9	1	0	1
+12	129999	0	1	9	6	6	9	0	0	1
+13	139999	0	0	9	6	6	9	1	0	1
+14	149999	0	0	9	6	6	9	0	0	0
+15	159999	0	1	9	6	6	9	1	0	0
+16	169999	0	1	8	6	6	9	0	0	0
+17	179999	0	0	8	6	6	9	1	0	0
+18	189999	0	0	9	6	6	9	0	0	0
+19	199999	0	1	9	6	6	9	1	0	0
+20	209999	0	1	9	6	6	9	0	0	0
+21	219999	0	0	9	6	6	9	1	0	0
+22	229999	0	0	9	7	6	9	0	0	0
+23	239999	0	1	9	7	6	9	1	0	0
+24	249999	0	1	9	6	6	9	0	0	0
+25	259999	0	0	9	6	6	9	1	0	0
+26	269999	0	0	9	6	6	9	0	0	0
+27	279999	0	1	9	6	6	9	1	0	0
+28	289999	0	1	9	4	6	9	0	0	0
+29	299999	0	0	9	4	6	9	1	0	0
+30	309999	0	0	9	6	6	9	0	0	0
+31	319999	0	1	9	6	6	9	1	0	0
+32	329999	0	1	9	6	6	9	0	0	0
+33	339999	0	0	9	6	6	9	1	0	0
+34	349999	0	0	8	6	6	9	0	0	0
+35	359999	0	1	8	6	6	9	1	0	0
+36	369999	0	1	9	6	6	9	0	0	0
+37	379999	0	0	9	6	6	9	1	0	0
+38	389999	0	0	9	6	6	9	0	0	0
+39	399999	0	1	9	6	6	9	1	0	0
+40	409999	0	1	9	6	4	9	0	0	0
+41	419999	0	0	9	6	4	9	1	0	0
+42	429999	0	0	9	6	6	9	0	0	0
+43	439999	0	1	9	6	6	9	1	0	0
+44	449999	0	1	9	6	6	9	0	0	1
+45	459999	0	0	9	6	6	9	1	0	1
+46	469999	0	0	9	6	6	8	0	0	1
+47	479999	0	1	9	6	6	8	1	0	1
+48	489999	0	1	9	6	6	9	0	0	1
+49	499999	0	0	9	6	6	9	1	0	1
+50	509999	0	0	9	6	6	9	0	0	0
+51	519999	0	1	9	6	6	9	1	0	0
+52	529999	0	1	9	6	6	1	0	0	0
+53	539999	0	0	9	6	6	1	1	0	0
+54	549999	0	0	9	6	6	9	0	0	0
+55	559999	0	1	9	6	6	9	1	0	0
+56	569999	0	1	9	6	6	9	0	0	0
+57	579999	0	0	9	6	6	9	1	0	0

FIGURE 20: PARTIAL "SAVED" LIST FILE

USE:

DLS>LIST S SAM

```
Date:  25 MAY 86 13:35   File:  FUNCTION.VLAF

$DATA_HEADER$
$TYPE$
I/O
$FORMAT$
TIME_VALUE
$TOTAL_COLUMN$
10 23
$BASE$
D B
$FIELD$ 1
@MUX16/2:EXTCLK 1
$FIELD$ 2
@MUX16/2:EXTRST 2
$FIELD$ 3
@MUX16/3:SELCT3 3
$FIELD$ 4
@MUX16/3:SELCT2 4
$FIELD$ 5
@MUX16/3:SELCT1 5
$FIELD$ 6
@MUX16/3:SELCT0 6
$FIELD$ 7
@MUX16/3:DAT7 7
$FIELD$ 8
@MUX16/3:DAT6 8
$FIELD$ 9
@MUX16/3:DAT5 9
$FIELD$ 10
@MUX16/3:DAT4 10
$FIELD$ 11
@MUX16/3:DAT3 11
$FIELD$ 12
@MUX16/3:DAT2 12
$FIELD$ 13
@MUX16/3:DAT1 13
$FIELD$ 14
@MUX16/3:DAT0 14
$FIELD$ 15
@MUX16/3:DAT9 15
$FIELD$ 16
@MUX16/3:DAT8 16
$FIELD$ 17
@MUX16/3:DAT7 17
$FIELD$ 18
@MUX16/3:DAT6 18
$FIELD$ 19
@MUX16/3:DAT5 19
$FIELD$ 20
@MUX16/3:DAT4 20
$FIELD$ 21
@MUX16/3:DAT3 21
$FIELD$ 22
@MUX16/3:DAT2 22
$FIELD$ 23
@MUX16/2:YOUTPT 23
$END$
#          1
9999       0
19999      1
29999      0
39999      1
49999      0
59999      1
69999      0
79999      1
89999      0
99999      1
109999     0
119999     1
129999     0
139999     1
149999     0
159999     1
169999     0
179999     1
189999     0
199999     1
209999     0
219999     1
229999     0
239999     1
249999     0
259999     1
269999     0
279999     1
289999     0
299999     1
309999     0
319999     1
329999     0
339999     1
349999     0
359999     1
369999     0
379999     1
389999     0
399999     1
409999     0
419999     1
429999     0
439999     1
449999     0
459999     1
469999     0
479999     1
489999     0
499999     1
509999     0
519999     1
529999     0
539999     1
549999     0
559999     1
569999     0
579999     1
589999     0
599999     1
609999     0
619999     1
629999     0
639999     1
649999     0
659999     1
669999     0
679999     1
689999     0
699999     1
709999     0
719999     1
729999     0
739999     1
749999     0
759999     1
769999     0
779999     1
789999     0
799999     1
809999     0
819999     1
829999     0
839999     1
849999     0
859999     1
869999     0
879999     1
889999     0
899999     1
909999     0
919999     1
929999     0
939999     1
949999     0
959999     1
969999     0
979999     1
989999     0
999999     1
```

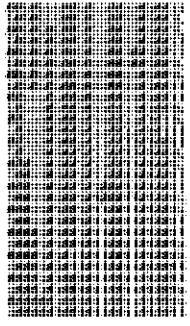


FIGURE 21: OUTPUT VLAF FILE

FROM DLS RUN

```
CLASS> AMCCSIMFMT
VLAIF Conversion    Rev[1.0]

    Enter VLAIF (input) file name: FUNCTION.VLAF

    Enter SIM (output) file name: OUTPUT.FUN


    Choose from menu -

       1 : VLAIF to SIM FORMAT (NO parenthesis).
       2 : VLAIF to SIM FORMAT (with parenthesis).

    Enter choice [1 or 2]: 1


    Do you want DATA seperated in columns [Y / N] ? N


    <<< Processing VLAIF Nets  >>>
    <<< Processing VLAIF Vectors  >>>
    >>>  Time ZERO (0) not output from VLAIF file.
    >>>  (100) lines of Vectors output.

    >>> AMCCSIMFMT conversion completed with NO error(s).
CLASS> TYPE OUTPUT.FUN
```

FIGURE 22: AMCCSIMFMT sample execution

Date: 25 MAY 86 13:36 File: OUTPUT.FUN

```
1***CIRCUIT IDENTIFICATION =

                EESSSSDDDDDDDDDDDDDDDDY
                XXEEEEAAAAAAAAAAAAAAAAO
                TTLLLLTTTTTTTTTTTTTTTTU
                CRCCCCØ123456789111111T
                LSTTTT          Ø12345P
                KT321Ø                T

         TIME

 9999           Ø1ØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 19999          11ØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 29999          ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 39999          1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
 49999          ØØØØØØØØØ1Ø11ØØ11Ø1ØØ11
 59999          1ØØØØØØØØ1Ø11ØØ11Ø1ØØ1Ø
 69999          ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 79999          1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
 89999          ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ11
 99999          1ØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 1Ø9999         ØØØ1ØØ1ØØ1111ØØ11Ø1ØØ1Ø
 119999         1ØØ1ØØ1ØØ1111ØØ11Ø1ØØ11
 129999         [illegible]
 139999         [illegible]
 149999         [illegible]
 159999         [illegible]
 169999         [illegible]
 179999         [illegible]
 189999         [illegible]
 199999         [illegible]
 2Ø9999         [illegible]
 219999         [illegible]
 229999         [illegible]
 239999         [illegible]
 249999         [illegible]
 259999         [illegible]
 269999         [illegible]
 279999         [illegible]
 289999         [illegible]
 299999         [illegible]
 3Ø9999         [illegible]
 319999         [illegible]
 329999         [illegible]
 339999         [illegible]
 349999         [illegible]
 359999         [illegible]
 369999         ØØ11Ø11ØØ1Ø11ØØ11Ø1ØØ11
 379999         1Ø11Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 389999         ØØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 399999         1ØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ11
 4Ø9999         ØØØ1Ø11ØØ1ØØ1ØØ11Ø1ØØ11
 419999         1ØØ1Ø11ØØ1ØØ1ØØ11Ø1ØØ1Ø
 429999         ØØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 439999         1ØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ11
 449999         ØØØØØ11ØØ1Ø11ØØ11Ø1ØØ11
 459999         1ØØØØ11ØØ1Ø11ØØ11Ø1ØØ1Ø
 469999         ØØØØØ111Ø1Ø11ØØ11Ø1ØØ1Ø
 479999         1ØØØØ111Ø1Ø11ØØ11Ø1ØØ11
 489999         ØØØØØ11ØØ1Ø11ØØ11Ø1ØØ11
 499999         1ØØØØ11ØØ1Ø11ØØ11Ø1ØØ1Ø
 5Ø9999         ØØØØ111ØØ1Ø11ØØ11Ø1ØØ1Ø
 519999         1ØØØ111ØØ1Ø11ØØ11Ø1ØØ11
 529999         ØØØØ111ØØØØ11ØØ11Ø1ØØ11
 539999         1ØØØ111ØØØØ11ØØ11Ø1ØØ1Ø
 549999         ØØØØ111ØØ1Ø11ØØ11Ø1ØØ1Ø
 559999         1ØØØ111ØØ1Ø11ØØ11Ø1ØØ11
 569999         ØØØ1111ØØ1Ø11ØØ11Ø1ØØ11
 579999         1ØØ1111ØØ1Ø11ØØ11Ø1ØØ1Ø
 589999         ØØØ1111ØØ1Ø111Ø11Ø1ØØ1Ø
 599999         1ØØ1111ØØ1Ø111Ø11Ø1ØØ11
 6Ø9999         ØØØ1111ØØ1Ø11ØØ11Ø1ØØ11
 619999         1ØØ1111ØØ1Ø11ØØ11Ø1ØØ1Ø
 629999         ØØ11111ØØ1Ø11ØØ11Ø1ØØ1Ø
 639999         1Ø11111ØØ1Ø11ØØ11Ø1ØØ11
 649999         ØØ11111ØØ1Ø11ØØ11Ø1ØØØ1
 659999         1Ø11111ØØ1Ø11ØØ11Ø1ØØØØ
 669999         ØØ11111ØØ1Ø11ØØ11Ø1ØØ1Ø
 679999         1Ø11111ØØ1Ø11ØØ11Ø1ØØ11
 689999         ØØ1Ø111ØØ1Ø11ØØ11Ø1ØØ11
 699999         1Ø1Ø111ØØ1Ø11ØØ11Ø1ØØ1Ø
 7Ø9999         ØØ1Ø111ØØ1Ø11ØØ1111ØØ1Ø
 719999         1Ø1Ø111ØØ1Ø11ØØ1111ØØ11
 729999         ØØ1Ø111ØØ1Ø11ØØ11Ø1ØØ11
 739999         1Ø1Ø111ØØ1Ø11ØØ11Ø1ØØ1Ø
 749999         ØØ1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 759999         1Ø1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 769999         ØØ1Ø1Ø1ØØ1Ø11ØØ1ØØ1ØØ11
 779999         1Ø1Ø1Ø1ØØ1Ø11ØØ1ØØ1ØØ1Ø
 789999         ØØ1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 799999         1Ø1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 8Ø9999         ØØ111Ø1ØØ1Ø11ØØ11Ø1ØØ11
 819999         1Ø111Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 829999         ØØ111Ø1ØØ1Ø11ØØ11Ø1Ø11Ø
 839999         1Ø111Ø1ØØ1Ø11ØØ11Ø1Ø111
 849999         ØØ111Ø1ØØ1Ø11ØØ11Ø1ØØ11
 859999         1Ø111Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 869999         ØØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 879999         1ØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ11
 889999         ØØØ11Ø1ØØ1Ø1ØØØ11Ø1ØØ11
 899999         1ØØ11Ø1ØØ1Ø1ØØØ11Ø1ØØ1Ø
 9Ø9999         ØØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 919999         1ØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ11
 929999         ØØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 939999         1ØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 949999         ØØØØ1Ø1Ø11Ø11ØØ11Ø1ØØ1Ø
 959999         1ØØØ1Ø1Ø11Ø11ØØ11Ø1ØØ11
 969999         ØØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 979999         1ØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 989999         ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 999999         1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
```

FIGURE 23: OUTPUT FROM AMCCSIMFMT

When the AMCCSIMFMT program is complete, execute AMCCVRC, the AMCC Vector Rules Checker. This program operates on the AMCCSIMFMT output file. All VRC errors must be removed before submission. Create a signal analysis file using TEC and then call AMCCVRC by typing

```
AMCCVRC
```

and responding to the prompts.

AMCCVRC is also called by the Super-Shell:

```
RUN_AMCC
6
```

Sample session for AMCCVRC:

```
TEC signal_control_file_name
     {edit}                (create signal
                                 analysis file)
{ENTER}EXIT{EXECUTE}
AMCCVRC
ERC/CIRCUIT.SDI    (netlist - subdirectory)
AMCCSIMFMT_file_name
    {select test or tests}
signal_control_file       (for race test)
```

AMCCVRC is only run against the maximum worst-case sampled functional simulation output. It produces AMCCVRC.LST and must be error-free for submission. Note: RESET, SET will usually generate Vector SSO errors that can be documented for what they are. These are the only allowed "errors". Consult AMCC if you have others.

After the logic has been verified via the functional simulation, run the at-speed simulation. This will require a smaller VIEW step than was used with the functional simulation to pick up the resolution. The timing step should reflect the speed at which the circuit will be operated (hence the term "at-speed" simulation). The simulation should be run at the Maximum specified operating frequency.

The "at-speed" simulation is used to check timing. At-speed simulation can be performed with either DLS or DTV. Using DTV allows an uncertainty range analysis. (Verify if DTV is operational on your system before scheduling its use in the design process.)

If the resolution of the waveform is good enough, the T and {ENTER} keys can be used to find the propagation delay between two edges when a waveform is on the screen. Put the cursor on one and strike T . Put the cursor on

the other and strike {ENTER}. The time will be displayed.

Note: AMCC requires sampled (uniform step) vectors for functional, at-speed and AC test simulations and requires PRINT_ON_CHANGE results for at-speed and AC Test. The designer should refer to Section 4, Vector Submission Rules and Guidelines for further information on vector requirements.

```
Date:  20 SEP 88 12:12   File:  RUN_DLS

DLS <<!
GET DLS_FMT
VIEW 9999 10000
RUN 1030000
QUIT
N
!
```

```
Date:  20 SEP 88 12:12   File:  RUN_ASDLS

DLS <<!
GET DLS_FMT
VIEW  499 500
RUN 50000
QUIT
N
!
```

FIGURE 24

PARAMETRIC TESTS

This circuit was constructed with a gate tree to allow the development of a parametric vector set using the preferred AMCC methodology. The SOM control file previously created for the function vectors can be copied into a new area (PARAMETER.SING was used as the file name) and the signal generators edited to allow one and only one input at a time to switch from "1" to "0" and back to "1".

Because the usued output on the input macro was used, the input signal is inverted. The logic gate used requires all input signals start at "1". The output "PARAM" starts at "0" and switches with each change. All inputs are exercised, including clock, reset and enables, if any.

The output format for the simulation output file is the same as for the functional simulation output file. The sampling is the same as for the functional simulation (100ns).

Figures A-20 and A-21 in Appendix A show the control file and the AMCCSIMFMT output file. The output file is SIMFMTPR.01

The simulation was run using the commands:

```
RUN_SOM PARAMETER.SING FNTMIL.DSY
DLS
GET FMT
VIEW 9999 10000
RUN 450000
QUIT
N
AMCCSIMFMT
FUNCTION.VLAF
SIMFMTPR.01
1
Y
22
TYPE SIMFMTPR.01 > /DEV/LP -HEA
```

THE BEST APPROACH (FLAT DESIGN):

1. Capture page or pages

   - Run DANCE on each page after proceeding to the next page - use a window
     DANCE n -M3
   - Check individual pages as they run and correct errors and rerun DANCE
     TYPE n.DFR in a window
   - When enough captured (the critical path, other areas of concern, etc.), then run DRINK
     DRINK -M3
   - NOTE: NOT ALL DANCE ERRORS MUST BE REMOVED FOR DRINK TO RUN SO BE CERTAIN ALL n.DFR FILES ARE 2 ERRORS ONLY (THE BORDER ERRORS)
   - NOTE: NOT ALL DRINK ERRORS MUST BE REMOVED FOR ERCS TO RUN

2. When DANCE and DRINK are error free or as you want them (partial circuits will still have errors) then run individual shells (on a large circuit this is faster):
   RUN_AGIF
   RUN_ERC
   RUN_ANN

   - Use TEC to add comments to FNTMIL.DSY or FNTCOM.DSY as well as FNTMIN.DSY

3. Now run the super-shell:
   RUN_AMCC
   and select "3" SIFT
   Answer the prompts

4. Step 3 leaves you in TEC so edit the SOM_MCF.SING file and create any data file needed

5. Run or continue to run the super shell:
   RUN_AMCC
   and select "4" SOM
   Answer the prompts

6. Exit the shell and run DLS or DTV

7. Run AMCCSIMFMT

8. Use TEC to create the signal analysis file

9. Run AMCCVRC

Application Note 1.DNIX

Introduction to the DAISY

(809)

Table of Contents

Figures:

INTRODUCTION TO THE DAISY

Q5000 16:1 MUX MIXED MODE EXAMPLE

The following provides a simple introduction to design with the AMCC Q5000 Series Logic Array on a DAISY EWS (Engineering Workstation) under the DNIX operating system, using the DED character graphics editor or DED2 character graphics editor. The problem is to design a 16:1 MUX which has TTL input and ECL output with a single +5V power supply. This requires the use of the TTL and ECL sections of the Q5000 macro library. The output is to pass through a D flip/flop. A parametric test gate tree has been added, along with extra power and ground and two thermal diodes.

The circuit selected demonstrates: 1) the use of mixed-mode +5V ECL/TTL I/O; 2) the use of the ECL output with a latch; 3) proper naming of wires and macros; 4) the use of extensive notes on the drawing page to document the design; 5) the use of the FOD fan-out derating net parameter; 6) the use of the SWGROUP switch group macro parameter; 7) the chip macro parameters; and 8) the logic required for the optional parametric test vectors. This is a flat design.

Logging in requires a user-name and may require a password. The AMCC seminar students use:

```
USER: CLASS
CLASS>
```

CLASS> is the prompt for all actions at the command level, set in the AMCC supplied loginfile.

If the /USER/CLASS user directory is not programmed, the system will come up at the top of the tree, i.e., at /, with a message to that effect. The user must make a directory:

```
MKDIR /USER/CLASS
```

and then change to it:

```
CD /USER/CLASS
```

From there, copy in an existing loginfile from another directory, if available:

```
COPY /NET/D_a/USER/CLASS/loginfile TO .
```

If none is available, use the text editor and create your own, for example:

```
TEC loginfile
%SUBMIT /AMCC/Qnnn_LIBS/QnnnnSETUP
 ... and whatever else you want

 there may be a networked datapath that
 needs to be in here - see your system
 manager
```

Activate the loginfile by logging off and logging back on or by:

```
%SUBMIT loginfile
```

Always verify that: 1) you are in the correct directory, in the correct current context; and 2) that the correct library has been referenced in the loginfile. There is another way to do this but it sometimes has problems.

Before starting schematic capture, any EWS user should make two checks: 1) that there is sufficient space for a new circuit (space on the hard disk); and 2) that the directory area to be used is either empty or loaded with known files. Check on the system disk memory available by typing:

```
SPACE
```

Display an inventory of the directory by typing one of these commands (there are other options):

```
INV . -S                    [ S for sorted ]
INV . -L                    [ L for long   ]
INV /USER/CLASS -L -DE      [ DE for deep nest ]
```

The loginfile at AMCC is currently preprogrammed to select the Q5000 library, as shown above. (Other libraries may be used by altering the loginfile to reflect that library.) AMCC provides preprogrammed shell scripts. The users may choose to add/modify/delete shell scripts at their own location on their own systems at any time - move them to another directory first! Beginners should stay with the provided shell scripts until they understand the actual commands.

Following selection of the library, and the rest of the "housekeeping", the drawing editor DED is invoked by typing:

```
DED1:   DED 1           [ DED n ]          DED I
DED2:   DED2 1          [ DED2 n ]         DED II
```

See the DAISY DED II introduction booklet for the comprable DED2 commands. DED1 will be operable under DNIX 5.02 but the user should begin to switch to the new editor in preparation for the ACE system. This application note was written for DED1.

As soon as the editor is ready, a blank page is presented (if there is no page 1.DRAW already in the directory). For a new page, a border should be placed on the page and edited.

When beginning a multiple-page drawing, plan ahead. If a page will not be created by replicating an existing page, it will always start with a border. Create several blank pages by copying a page with a border, comments, etc. into several page files.

A border is placed one grid point up diagonally from the lower left corner. For any circuit, the border is: /AMCCPAGEB and is a component in the library. If the cursor is positioned, type:

```
DED1:   CO /AMCCPAGEB{EXTRACT}...{EXECUTE}
```

The page should be given a unique six-character alphameric name (such as PAGE1), and then notes should be added in the lower right-hand block to document the date, page number, drawn by and a description of what is on the page. Multiple notes are created by {EXTRACT}ing notes from other notes. Multiple names are not allowed but {EXTRACT} can be used to make a copy of a name.

```
Name:
DED1:   NA {SELECT}name{DEF}{PLACE}...{EXECUTE}
DED1:   NA {EXTRACT}{PLACE}...{EXECUTE}
```

This last version copies the same name from one place to another.

```
Note:
DED1:   NOT {SELECT}any text{DEF}{PLACE}...{EXECUTE}
DED1:   NOT {EXTRACT}any text{DEF}{PLACE}...{EXECUTE}
DED1:   NOT {EXTRACT}{PLACE}...{EXECUTE}
```

This last version copies the same note from one place to another.

A name or a note is deleted by opening the command through {SELECT} and then typing D{MACRO}EXECUTE}. A name or nore is edited by opening the command through {SELECT}, typing the new name or text and then typing {DEF}{PLACE}{EXECUTE}.

The body of the drawing itself is then created by first fetching the basic macros and positioning them on the page. Use a paper sketch to design the initial layout for a circuit, to minimize wasted partial macros, minimize cross-overs, clarify where replication of macros, blocks and pages will save design entry time, and to identify the macros that will be used. (Don't do seat-of-the-pants design at the screen! It leads to frustration and error. Plan ahead.)

For a new design, the first macro on page one should be a chip macro. Once it is properly parameterized, this macro will characterize the array as to its grade (military or commercial), its power supply, the type of ECL (10K or 100K) and power supply, and identify the product with a unique name and number. Parameters are added with a simplified form of the **PARAMETER** command.

For this design, the chip macro is Q5000TTTL10K (this design fits into a Q1300T).

```
DED1:   CO /Q5000TTTL10K{EXTRACT}{PLACE}...{EXECUTE}
```

Three parameters need to be attached to the chip macro, PRODUCT_NAME, DEVICE_NUMBER, PRODUCT_GRADE. A fourth, POWER_SUPPLY, does not apply to the I/O mode chosen (TTL10K is a single power-supply of +5V only). Position the cursor on the first of these parameters, located in the lower left side of the macro symbol, and type:

```
DED1:  PARA {SELECT}param-value{DEF}{PLACE}...{EXECUTE}
```

The {PLACE} can be skipped if you positioned on the first character of the parameter name on the graphic symbol. Repeat for all of the parameters listed on the chip macro.

The value possible for PRODUCT_GRADE is either MIL or COM. The default power supplies are listed on the chip macro. The only time the POWER_SUPPLY may be altered is for the STD ECL circuits, or MIXED ECL/TTL, where 5VREF, STD4 (-4.5V) or STD5 (-5.2V) can be specified. AMCC customer service assigns the PRODUCT_NAME and DEVICE_NUMBER.

Regardless of technology, the input pins on a chip macro are tied to global ground.

If the pin is a power pin (VCC), aim the wire up. If the pin is a ground pin, aim the wire down. The EWS and the software does not care - this is a human readable convention. The output of the chip macro is wired to a terminator or it is wired to a page connector. For this design the terminator is used. Use /LWTERM and connect

it to the macro output pin with a wire. Do not allow the pins of any two macros to touch as the system may reboot and you will loose the page.

Add any extra power or ground macros desired. For all AMCC Series arrays, the extra power and ground macros are ITPWR, ITGND and IEVCC. Regardless of technology, the extra power and ground macros, like the chip macros, have all input pins tied to GND. Draw a wire out and down and name the wire GND.

For the VCC or ITPWR pins, the wire is drawn up by convention. (IEVCC is a power pin in a +5V REF ECL system.) The output of the power or ground macro is terminated.

The basic wire command is:

W {MARK}...{EXECUTE}

Use {MARK} to draw bends. Using {MARK} twice without moving will cause DAISY to try to straighten out the wire. A partial wire requires specific connection steps befoe it can be picked up and continued. Dot-connects are formed by anding one wire on top of another (not at the end). There are on-page connectors that allow a cumbersom wire to be "broken" on a page. These should be used with care to avoid unintelligible drawings.

Refer to the DED command summary for further wiring instruction. DED2 mouse wiring operations are sometimes preferrable. Any design begun in DED2 can switch back and forth between editors without leaving the DED environment. Any drawing begun in DED1 may be invoked from DED2.

Name the macros, including the chip macro (CHIP00). Power and ground macros have a naming convention as do static drivers. Try to follow AMCC naming conventions.

As an aid to debug and to avoid duplicate names which can be catastrophic, put the page number into the macro names. MX0000 on page four would become MX4000.

Proceed to the second drawing page by typing:

DED1: {NEXT}{EXECUTE}Y{EXECUTE}

Bring up a border on the new page, name and note it and begin to capture the schematic. To save time, when a border is brought up and commented for page one, copy it (use the SAVE command) to all following pages (SAVE 2; SAVE 3, SAVE 4, etc.).

All interface and logic Macros are components and are called by:

DED1: CO /macro name{EXTRACT}{PLACE}...{EXECUTE}

The 2{ZOOM} feature allows the entire drawing page to be seen and is a great assist in initial page layout. It will not detail the macros and you cannot wire while in 2{ZOOM}. (You can, but it won't be right.) Return to 1{ZOOM} or one step beyond (1{ZOOM}{ZOOM}) to do the wiring. Use the {VIDEO GRID} key to obtain a dotted background but turn it off (it is a toggle) when doing printout. Changing pages will reset ZOOM to the original 1{ZOOM} setting.

Use the {REDRAW} key to reconnect the displayed lines after names, notes and moves. What is on the screen is what is plotted. You can plot 1{ZOOM} and even higher - higher will provide interesting artwork but it will not be valid as a circuit schematic.

Following macro placement, the page and chip connectors and terminators are added. After these, the wires are added, and then the wires and macros are named and notes are added.

The DAISY connector names are decoded as:

R right	L left
W wire	B bundle or bus
H hierarchical	P page
I input	O output
CON connector	
PI page in	IP intrapage

There are page-to-page connectors (/RWPICON, /RWPOCON), bundled wire connectors (/RBHICON, /RBHOCON, /RBPICON, /RBPOCON), and bidirectional connectors (/RWPBCON, /RWHBCON, /RBPBCON, /RBHBCON). There are others.

The DAISY library also provides the terminators /LWTERM, /RWTERM, /UWTERM and /DWTERM. Other terminators, for bundles, bidirectionals, etc., also exist. Avoid using /UWTERM and /DWTERM since they resemble the ground symbol and this is visually confusing. Terminators are added by first fetching them from the library, and then wiring the terminator to the desired wire stub. Treat terminators as components, except for the fact they need not be named.

Wires are used to tie unused macro inputs to global ground. A wire is added to the input pin and drawn out and down. It is named GND and the symbol appears in place of the name. (Same as for the chip macro.)

DED1: NA {SELECT}GND{DEF}{PLACE}{EXECUTE}

Bundles are useful to route bundles of wires. The BU command is used in place of W in the routing of the bundle. Instead of a name, give a bundle a contents parameter (/CONTS) that contains the identification of what is in the bundle. There is a 64 character limit on the value of the /CONTS parameter. The /CONTS parameter is assigned by the full **PARAMETER** command:

PARA {SELECT}/CONTS{EXTRACT}xxxx{DEF}{PLACE}...{EXECUTE}

Wires coming from a bundle are individually named. DAT0, DAT1, ABUS12 are wires to the bundle with the /CONTS parameter DAT(0:15),ABUS(0:12).

All macros should be named in a flat design or in a hierarchy non-nested design, and all off-chip and off-page connections must be named. Internal wires may be named if you will be interested in them during debug of the design. Internal wires that MUST be named are the enable signals into 3-state output macros or bidirectional macros.

Internal wires (nets) in critical paths must be named or the default name must be made visible (via NA {SELECT}{PLACE}..{EXECUTE}) on the schematic. This is for use in Front- and Back-Annotation analysis. If you are going to take the trouble to make it visible - change its name! It takes little additional effort and speeds the design analysis process. Naming an internal signal allows it to be easily placed in the FMT file for wave and list display later on. Any signal that might help a debug problem should be named.

Hint: since the Front-Annotation file sorts the net names, begin all nets in a critical path with the same first three letters. The nets will sort near each other, making analysis easier. A full Q5000 design could easily have 900 nets.

The macro naming rules are by AMCC convention (see the EWS Schematic Rules and Conventions) including Lxxx for a latch, Bxxx for a buffer, Mxxx for a MUX, Dxxx for a driver, and so on. Macro instance names should be 1 to 6 characters long. Wire names should be 1 to 8 characters long, and should follow AMCC rules (alphameric only); no special characters. All names should be meaningful, which implies at least 3 characters. Names can begin with a number or a letter. As mentioned before, AMCC recommends that one character of the name be reserved for page identification to avoid interpage name duplication.

The rules are also designed to help avoid using a pin-name or other confusing names. They are designed to allow ease of transfer between various support programs. All names must be unique on a page and for the design.

For example, if AOUT appears on page 1.DRAW, it can only appear on page 1.DRAW one time as a chip/page input or chip/page output. A name can be extracted to appear somewhere else along the same wire or wirenet. A name can appear on several intrapage connectors (same-page connector - used to break a long wire for visual clarity). Each time the same wire name appears it is attached to the wire-net with that name.

The positioning of the wire names for on- and off-chip connectors: The names are attached to the wires but placed to the left of the input (/RWHICON) connectors and to the right of the output (/RWHOCON) connectors, or placed just above the wires.

All DAISY components may be rotated. AMCC prefers that rotation not be used since AMCC may run a graphic UPDATE of all submitted schematics against the in-house working library to verify the data. UPDATE does not rotate but puts the macros on the page in unrotated form, breaking connections and destroying the page. Structured design flow is left to right across the page and this flow should be maintained.

At any time during the creation of the drawing page, the page may be saved by typing:

```
DED1:   SAVE {EXECUTE}
DED1:   SAVE n {EXECUTE}          [ where n is a digit ]
```

Perform SAVE once every 15-20 minutes or when a complex structure has been entered or edited.

The use of "SAVE" with a file reference (usually a number for an n.DRAW page) allows pages to be copied while in the editor.

At any time during the creation of the drawing page the page may be plotted by typing:

```
    PL {EXECUTE}.              <--- direct connect
    PL /NET/D_n{MACRO}{EXECUTE}  <----- thru another
```

The second version is for plotting through another ETHERNET node at AMCC.

If there are no more pages or the page is complete or the user is interrupted, then DED is exited by:

```
DED1:   EXIT {EXECUTE}     save
DED1:   QUIT {EXECUTE}     do not save
```

The use of "QUIT" instead of "EXIT" will cause the current workspace page to be lost and the contents of the hard disk file will remain unchanged. If a file has been SAVEd then a QUIT is reasonable; or if an edit session has been abortive, then QUIT is preferred. QUIT will prompt the user to verify that drawing page edits are to be discarded.

Use of the menu and the mouse. Type the blue button to find the menus and select "WINDOW MANAGER".

Use the blue button to select "Plot Screen". Do not use "Plot Window". By toggling the yellow button twice, the entire screen is plotted. The yellow button can also be used to block out that part of the screen that is desired and a partial plot made.

Other selections available are "Shell Window". A shell window can be opened by {SHIFT}{ZOOM}. After use, it is deleted by {CTL}{E}. The cursor must be in the window for the command line to function.

Flip a window backward and forward in the window stack by selecting the yellow button on the banner for the window.

The use of multiple windows will slow down the system due to overhead. Limit the windows to two and use no sub windows when running simulation. To speed simulation, some users turn off NCP (sign off the net).

The next pages show the circuit created, a 16:1 MUX with two thermal diodes and with a gate-tree to allow parametric testing. The parametric tree inputs were taken from the unused pins on the input macros (all were YN), allowing the testing logic to be added without adding time delay to the actual function. This may not always be possible.

The parametric logic will need to be tested with functional vectors - the sample shown herein is 100% fault-coverage of the function but not necessarily of the test logic. Any expansion of the vector set needed to fault-cover the gate tree is left as a student exercise.

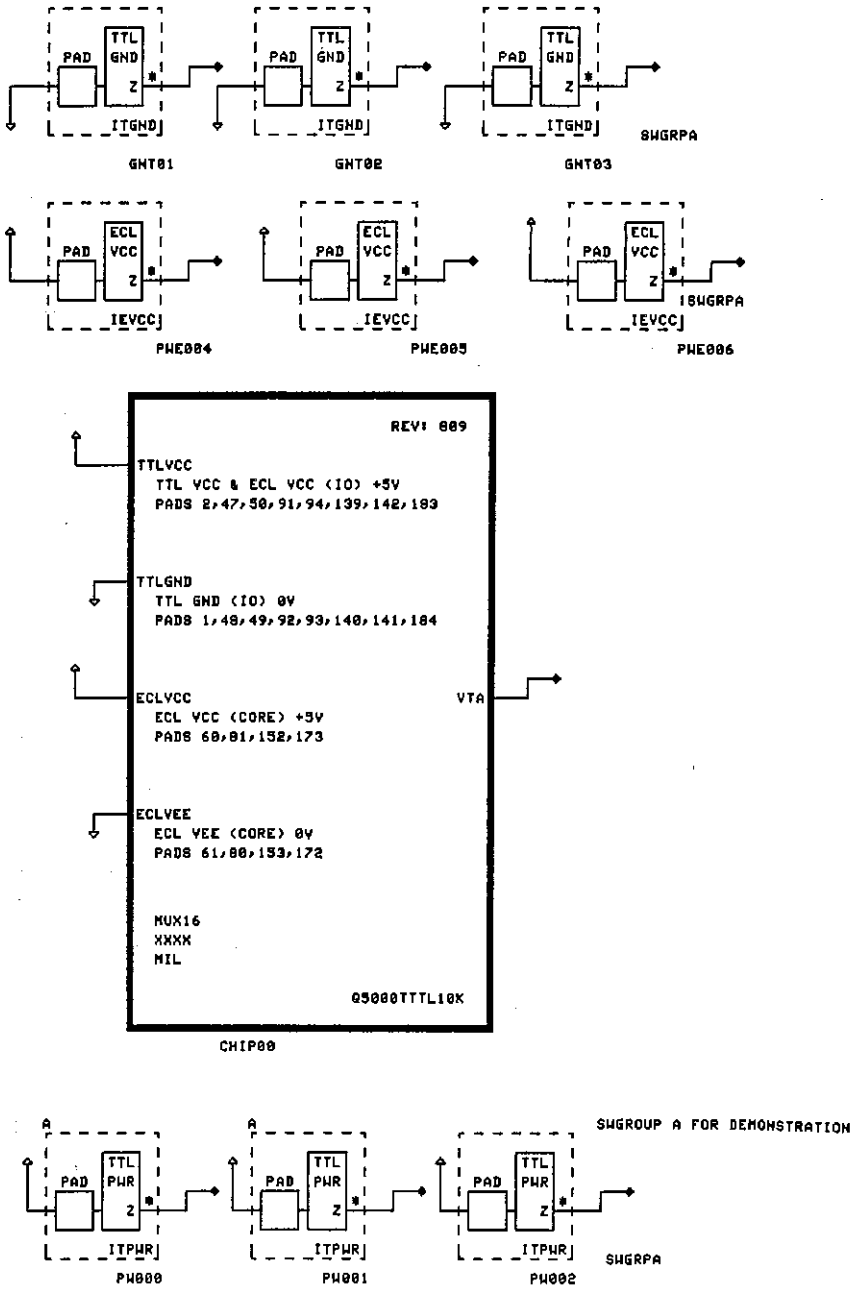

PAD
TTL GND
Z
ITGND
GNT01
GNT02
GNT03
SWGRPA
ECL VCC
IEVCC
PWE004
PWE005
PWE006
REV: 009
TTLVCC
TTL VCC & ECL VCC (IO) +5V
PADS 2,47,50,91,94,139,142,183
TTLGND
TTL GND (IO) 0V
PADS 1,48,49,92,93,140,141,184
ECLVCC
ECL VCC (CORE) +5V
PADS 60,81,152,173
VTA
ECLVEE
ECL VEE (CORE) 0V
PADS 61,80,153,172
MUX16
XXXX
MIL
Q5000TTTL10K
CHIP00
SWGROUP A FOR DEMONSTRATION
TTL PWR
ITPWR
PW000
PW001
PW002
SWGRPA

FIGURE 1

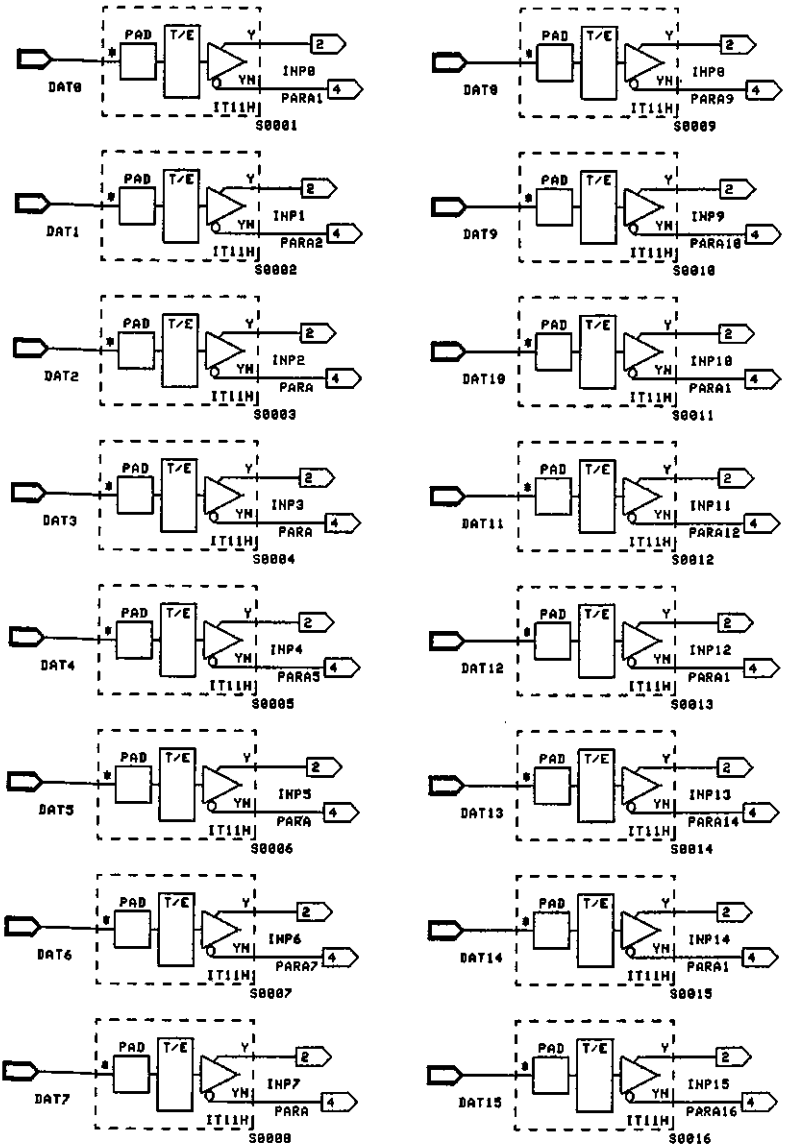

DAT0
PAD
T/E
Y
YN
2
INP0
PARA1
4
IT11H
S0001
DAT1
INP1
PARA2
S0002
DAT2
INP2
PARA
S0003
DAT3
INP3
S0004
DAT4
INP4
PARA5
S0005
DAT5
INP5
S0006
DAT6
INP6
PARA7
S0007
DAT7
INP7
S0008
DAT8
INP8
PARA9
S0009
DAT9
INP9
PARA10
S0010
DAT10
INP10
PARA1
S0011
DAT11
INP11
PARA12
S0012
DAT12
INP12
S0013
DAT13
INP13
PARA14
S0014
DAT14
INP14
S0015
DAT15
INP15
PARA16
S0016

FIGURE 2

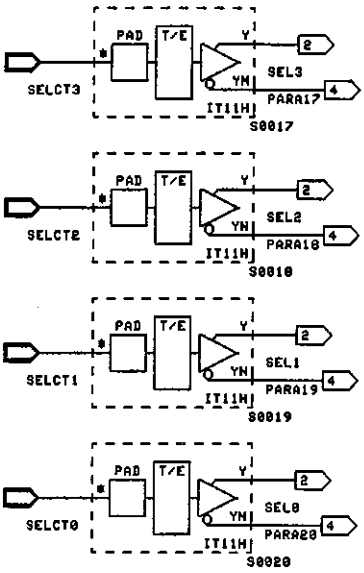


FIGURE 2 CONTINUED

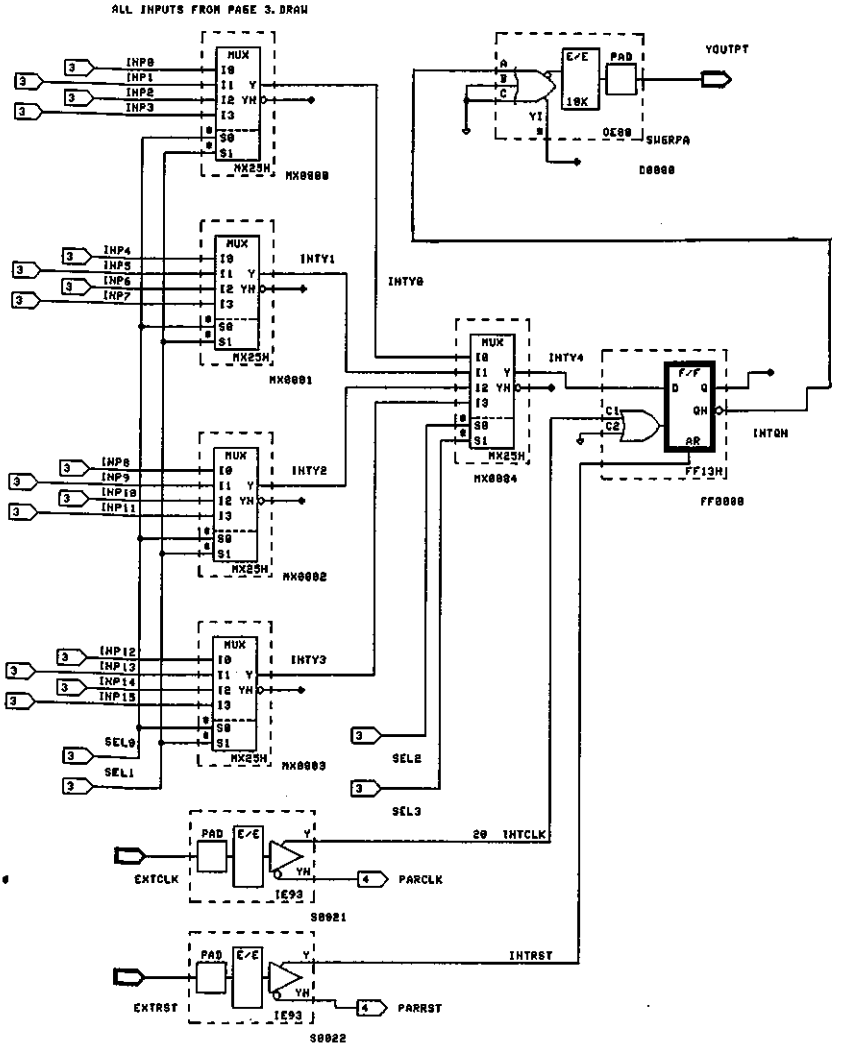

ALL INPUTS FROM PAGE 3.DRAW
INP0
INP1
INP2
INP3
MUX
I0
I1 Y
I2 YH
I3
S0
S1
MX25H
MX0000
INP4
INP5
INP6
INP7
INTY1
MX0001
INP8
INP9
INP10
INP11
INTY2
MX0002
INP12
INP13
INP14
INP15
INTY3
MX0003
SEL0
SEL1
SEL2
SEL3
INTY0
INTY4
MX0004
E/E
PAD
YOUTPT
A
B
C
YI
10K
0E00
SWERPA
D0000
F/F
D
Q
QH
AR
C1
C2
INTQH
FF13H
FF0000
EXTCLK
E93
S0021
PARCLK
INTCLK
EXTRST
S0022
PARRST
INTRST

FIGURE 3

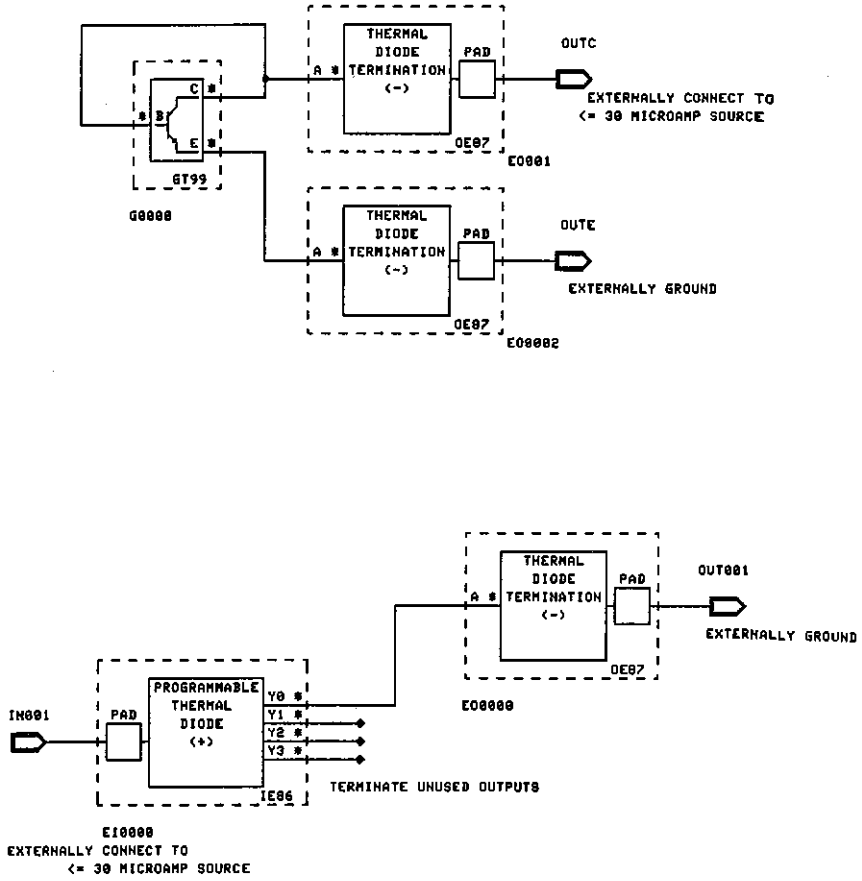

THERMAL
DIODE
TERMINATION
(-)
PAD
OUTC
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE
OE07
E00001
C
B
E
GT99
G0000
THERMAL
DIODE
TERMINATION
(-)
PAD
OUTE
EXTERNALLY GROUND
OE07
E00002
THERMAL
DIODE
TERMINATION
(-)
PAD
OUT001
EXTERNALLY GROUND
OE07
E00000
PROGRAMMABLE
THERMAL
DIODE
(+)
PAD
IN001
Y0
Y1
Y2
Y3
IE06
TERMINATE UNUSED OUTPUTS
EI0000
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE

FIGURE 4

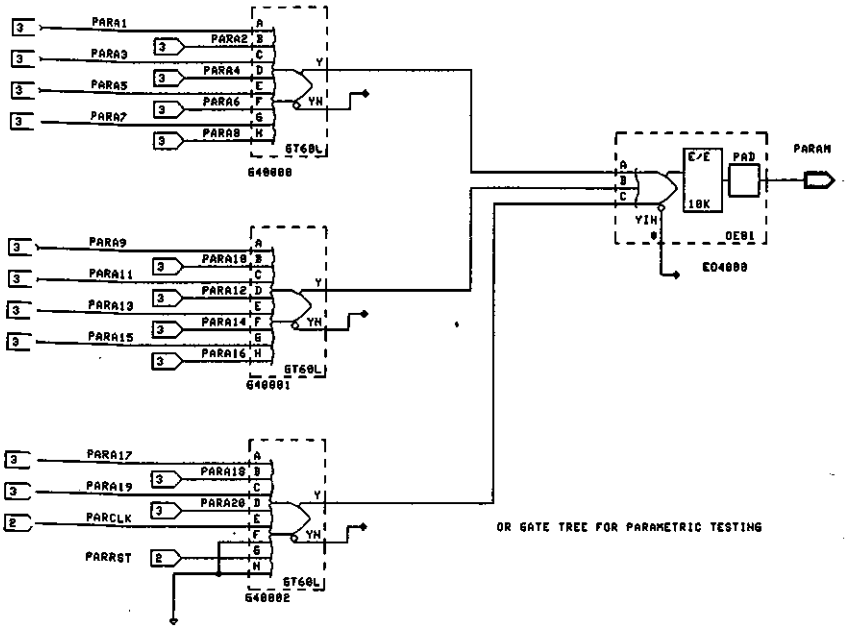

PARA1
PARA2
PARA3
PARA4
PARA5
PARA6
PARA7
PARA8
GT60L
G40000
PARA9
PARA10
PARA11
PARA12
PARA13
PARA14
PARA15
PARA16
GT60L
G40001
PARA17
PARA18
PARA19
PARA20
PARCLK
PARRST
GT60L
G40002
C/E
PAD
10K
YIH
OEBI
E04000
PARAM
OR GATE TREE FOR PARAMETRIC TESTING

FIGURE 5

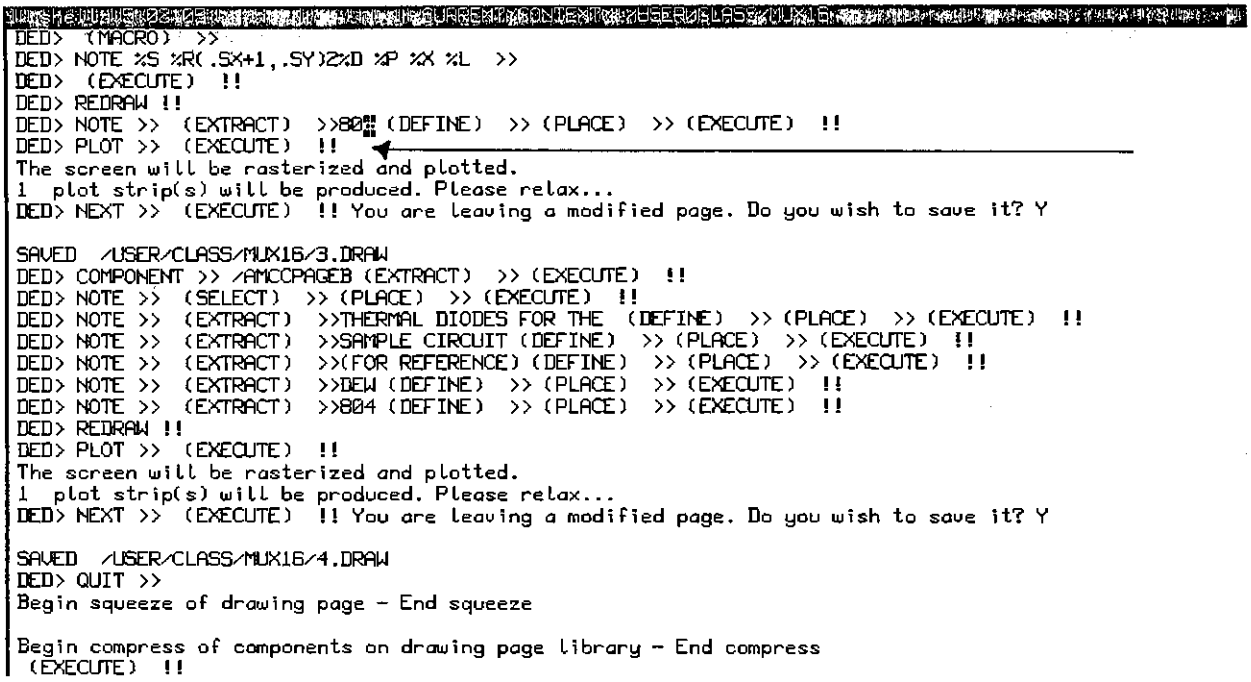


FIGURE 6

AFTER DED - LOG OF ACTIONS ON THE SCREEN

To copy drawing pages when outside of one of the DED editors, copy one page to another by:

```
COPY filename TO filename
```

When inside any DED editor simply save the page:

```
SAVE n (RETURN)
```

Initialize a floppy by:

```
INITDISK -DEMO    ..... ONLY IF NOT INITIALIZED
                        (or you are wiping the file)
```

Use an initialized or previously used floppy by:

```
MOUNT /F          ..... ALWAYS MOUNT THE FLOPPY
```

Note: Never delete the /F directory. If you do not mount a floppy before executing a copy command where /F is the destination, the file(s) are put on the main hard disk in the /F directory. Do not erase /F if you do not know what is in the directory (if the files are not your own).

Erase the /TEMP directory contents periodically. These are created during print or plot. These files are DAISY software temp files. A large /TEMP directory can cause system failures.

```
   COPY 1.DRAW TO /F       ..... WHATEVER
or COPY 1.DRAW TO /F -B
   COPY /F/* to . -B -DE .... reloading from floppy
   COPY * TO /F          ..... overkill; gets BAK, etc.
   COPY * TO /F -B -DE
   DIS                   .....WHEN DONE, DISMOUNT
```

Note: Backup copies of drawings should be kept on a floppy disk or tape. Make a back-up everyday. Keeping copies on the disk in another directory will slow down the system and waste memory. If there is a hard disk failure, you would loose both directories anyway. If you are working at the AMCC design center, ask about storage on the EAGLE hard disk.

Do not leave a floppy on-line during execution. Mount it, use it and dismount. Floppy drives are shared drives at AMCC.

The circuit shown in Figure 1 is flat (all pages in one directory area, no design tree), is not nested, has no parameterized components other than the chip macro, and does not use bundles, blocks or cells, although these features are available on the DAISY. It is a simple design suitable for instructional use.

Hierarchical design rules are covered in Application Note 2. Nested design rules are covered in Application Note 3.

When setting up the floppy, use a master index to keep track of what has been saved. One trick that AMCC likes you to use is the creation of a master index for your design files by typing:

```
INV /F -S -L -DE > /F/INDEX.INV
```

The file INDEX.INV contains a complete inventory of what has been stored on the floppy. The first execution creates an empty INDEX.INV file. You can also do this within a directory as a means of indexing that directory.

The next step is to DANCE the design (DAISY Network Connectivity Extractor). All pages in a design may be DANCEd at once or they may be processed individually. Call DANCE by:

```
DANCE                 one page
DANCE -M3 -T          all pages, tree, individual
                            short reports
DANCE -M3 -T -N       all pages, nested mode,
DANCE -M3 -T -ERR     concatenated reports
```

M3 is a message level, and is usually sufficiently detailed. You can change from M3 to M5 for more detail, or to M1 for less. DANCE will flag errors to the screen and to files x.DFR. The files n.DFR will be produced for this circuit for the first three versions of the command; file ERR.DFR will be produced by the concatenated request. By omitting R, the report will only contain error messages. This is preferred to keep memory usage down.

```
TYPE 1.DFR              use {OUTPUT CONTROL} to read
TYPE 1.DFR > /DEV/LP to line printer (plotter)
TYPE 1.DFR > /NET/D_a/DEV/LP  plot through AMCC
                        remote node to plotter
TYPE ERR.DFR            concatenated report
TYPE ERR.DFR > /DEV/LP -HEA  type with header
```

Use this last option (-HEA) with TYPE to keep a header on all outputs. It saves time during submission assembly.

Whenever there is a question about a DAISY command, use HELP and select the command you need help with. Figure 2 shows the screen during the HELP command with the system uppercase command TYPE selected for explanation.

```
CLASS> RUN_DD

        ****** RUNNING DANCE -T -N -ERR -M3 ********

        ************ DANCE OK !!!!! ***************

        *********** RUNNING DRINK -T -M3 ***********

        ************ DRINK OK !!!!! ***************
```

```
Date:  25 MAY 86 Ø8:44   File:  DANCE.ERR

LOGICIAN DANCE VERSION V5.Ø2.Ø2

DANCE :   BEGIN BLOCK /USER/CLASS/MUX16
DANCE :     BEGIN PAGE 1
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 2.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 3.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :     BEGIN PAGE 4.DRAW
DANCE :       EXTRACTING PRIMITIVES
DANCE :       COMPILING PRIMITIVES
DANCE :       COMPILING CONNECTIVITY
DANCE :       GENERATING LISTINGS
DANCE :     END PAGE.     2 ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DANCE :   END BLOCK

DANCE COMPLETE:     4 PAGES FLAGGED. HIGHEST SEVERITY WAS Ø
```

```
Date:  25 MAY 86 Ø8:44   File:  DRINK.ERR

LOGICIAN DRINK VERSION V5.Ø2.Ø2

DRINK:  BEGIN COMPLETE LINK
DRINK:    TRAVERSING TREE
DRINK:      BEGIN BLOCK /USER/CLASS/MUX16
DRINK:        COLLECTING RECORDS
DRINK:        PROCESSING BLOCK DATA
DRINK:      END BLOCK     Ø ERRORS FLAGGED. HIGHEST SEVERITY WAS Ø.
DRINK:    PROCESSING TREE DATA
DRINK:      LINKING ENTRIES
DRINK:      BLOCK TAB LOADED
DRINK:      PAGE  TAB LOADED
DRINK:      RESERVING MAP
DRINK:      SORTING RECORDS
DRINK:      MERGING RECORDS

DRINK COMPLETE:     Ø BLOCKS FLAGGED. HIGHEST SEVERITY WAS Ø
```

FIGURE 7: RUN_DD, DANCE.ERR, DRINK.ERR

DANCE can be executed for one drawing page as the next is being edited. Move off the page to be danced, open a shell window and execute the DANCE command for the completed page. This is a preferred way of working since commonly made errors that DANCE could catch are detected early before they become ingrained bad habits. Anytime an individual page is edited, it can be re-DANCED by itself.

If DANCE does not produce any errors, the next step is to run DRINK (DAISY Resolving Linker). DANCE acts as a compiler, while DRINK acts as the corresponding linker. DRINK is run against the entire circuit. (There are sophisticated alternatives for update mode that will not be covered here.) Call it by:

```
DRINK                 short error-only report
DRINK -T -M3 -E3      messages to the screen
```

If there are errors in file TREE.DFR then you must return to DED to the page or pages flagged and make the corrections. The page(s) must repass through DED, DANCE and DRINK until there are no errors. Note: DRINK can be executed on a partial circuit; DAISY will simulate a partial circuit, MacroMatrix will execute on a partial circuit. A design cannot be SUBMITTED until all errors are removed.

DANCE and DRINK both can be executed by typing the shell execution command:

```
RUN_DD                (recommended)
                        (part of Super-Shell)
```

RUN_DD will produce DANCE.ERR (a concatenated report of all pages) and DRINK.ERR (same as TREE.DFR). Scan DANCE.ERR before proceeding with the ERCs and other software. There should be two errors per page (the border has no pins and is not connected to anything - a pinless macro). If more than two errors, go fix the page. (Note: The Super Shell will continue to execute with level 0 errors - it currently only halts on level 1 errors.)

If this process has been done before and you think all errors are removed, or you want to fire off processes while you do something else, call the AMCC super-shell and run DANCE, DRINK, AGIF, and ERC.

```
RUN_AMCC              (The Super-Shell)
1
```

If there are no errors from DRINK, and you have not run the Super-Shell, run the netlist transfer file generation software by first executing:

RUN_AGIF (part of Super-Shell)

This routine changes the DAISY-produced netlist into the AMCC generic interface format file (AGIF) (using SING) that will be transmitted to the VAX for layout and other software access. It will also create files in a new sub-directory /ERC.

The principal file that is created is CIRCUIT.SDI, the AMCC-formatted netlist. This file is a design-submission file. It is an input file for: ERCs, Front-, Intermediate- and Back-Annotation, AMCCSIMFMT, AMCCVRC, place and route, LASAR6 simulation and fault-grading and the tester programs. Do not type this file out - it is very large and is encoded.

The AMCC ERC (Engineering Rules Check) software must be run before proceeding with DAISY simulation. The DAISY system doesn't care about this (it is a parallel process) but you should. Why waste time performing and evaluating simulations when there are fundamental interconnection or population errors in the design? Running the ERCs first saves design time.

The ERC software is system-resident and is executed by moving down to the ERC subdirectory created by RUN_AGIF and typing:

AMCCERC (need to change directory)

The AMCC shell script **RUN_ERC** can be run <u>without</u> moving down to the ERC subdirectory. RUN_ERC calls AMCCERC.

RUN_ERC (part of Super-Shell)
(preferred execution)

The AMCC ERC program is a set of routines that will flag various errors such as fan-out exceeded, unconnected pins, improper wire-ORs, invalid names, grounded outputs, duplicate names, and other similar errors. Type out the reports to the screen by:

```
TYPE ERC/AMCCERC.LST
TYPE ERC/AMCCIO.LST
TYPE ERC/AMCCXREF.LST
```

If there are ERC-flagged errors, you must return to DED to fix the pages flagged. This is one reason why individual steps might be more efficient than the Super-Shell for the first pass through the drawing.

Before the next step can be run, the changed pages must be re-DANCED, the circuit re-DRINKed, SING-TO AGIF rerun

and ERC re-executed with no errors. Any time an error is found, the process must be restarted. The ERC software should pick up most of the more commonly encountered oversights and slips made in a design schematic capture.

The ERCs assume that catastrophic DANCE and DRINK errors are removed. If there is an incorrect pin-wire interconnect (contact not really made) it can survive the ERCs but fail in simulation. Unless you are running a partial circuit, always remove DANCE and DRINK errors BEFORE running ERCs. If you are running a partial circuit, remove those errors pertinent to the pages that are captured.

After the ERC software is executed without error, execute the Front-Annotation software while in the /ERC subdirectory by typing:

```
AMCCANN          (need to change directory)
```

Again, the AMCC shell script RUN_ANN can be run without moving to the ERC subdirectory.

```
RUN_ANN
```

Or by using the Super-Shell

```
RUN_AMCC              (part of Super-Shell)
2                     (preferred execution)
```

In the delay files produced, each net is identified by name (user-defined or default) and is followed by six numbers representing the min, typ and max net delay for both rising and falling edges. Only one file is referenced in the simulation at one time. These Front-Annotation delay files provide the internal net interconnect delays due to fan-out, wire-ORs and metal loading. The metal load delay is estimated.

The output net delays due to system and package pin capacitance loading is also computed. The program will prompt for a response which can be as simple as defaulting all values or as intricate as specifying different system and package capacitance loading for each of the primary output signals in the circuit. The package type selected determines the default values for the package pin capacitance.

The default system load is 15pF for TTL and 5pF for ECL. The output macros for the AMCC arrays in the Q14000, Q5000, Q20000 and future arrays are specified under no load.

```
CLASS809> RUN_AMCC

APPLIED MICRO CIRCUITS CORPORATION

        1: Part One:   Run DANCE, DRINK, AGIF and ERCs

        2: Part Two:   Run AMCCANN

        3: Part Three: Run SIFT and SOM_MAKER

        4: Part Four:  Run SOM and TCAL

        5: Part Five:  Run AMCCSIMFMT

        6: Part Six:   Run AMCCVRC

        0: Exit Program:

        Please Enter [0 - 6] : >2

        ************** RUNNING AMCCANN *************

        Need to Edit Package Pin Data? (Y or N) ; >Y

AMCC Delay Annotation VERSION  3.30
Loading Netlist ...
Welcome to the output loading system.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 1
```

Running AMCCANN

FIGURE 8

```
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 1
          0) OTHER.
          1) 224 PGA cavity down
          2) 132 leaded chip carrier cavity up.
          3) 196 leaded chip carrier cavity down.
          4) 149 PGA cavity down.
          5) 149 PGA cavity up.
Type the number of the package that this design will be using.
1
224-PGA-CD
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 2
The current default value for package pin capacitance is 4.9 pf.
Enter <Retn> for no change or enter a new value:
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 3
The current default value for TTL system capacitance is  1.5e+01 pf.
Enter <Retn> for no change or enter a new value: 20
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:
```

FIGURE 8 CONTINUED

```
MW_PASCAL_V5 ... CURRENT CONTEXT: /USER/CLASS/MW
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 4
The current default value for ECL system capacitace is 5.0 pf.
Enter <Retn> for no change or enter a new value: 6
(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 5
OUT001       OUTC         OUTE         PARAM        YOUTPT

Enter the signal name or signal names separated by spaces.
YOUTPT
Enter the new value (pf): 12
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 6
OUT001       OUTC         OUTE         PARAM        YOUTPT

Enter the signal name or signal names separated by spaces.
PARAM
Enter the new value (pf): 8
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:
```

FIGURE 8 CONTINUED

```
MW_PASCAL V5.1.2                    CURRENT CONTEXT: /USER/CLASS/MUX1
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 7
DAT0        DAT1        DAT10       DAT11       DAT12
DAT13       DAT14       DAT15       DAT2        DAT3
DAT4        DAT5        DAT6        DAT7        DAT8
DAT9        EXTCLK      EXTRST      IN001       OUT001
OUTC        OUTE        PARAM       SELCT0      SELCT1
SELCT2      SELCT3      YOUTPT
Enter the signal name or signal names separated by spaces.
OUTC
Enter the new value (Ohms): 50
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform: 8
DAT0        DAT1        DAT10       DAT11       DAT12
DAT13       DAT14       DAT15       DAT2        DAT3
DAT4        DAT5        DAT6        DAT7        DAT8
DAT9        EXTCLK      EXTRST      IN001       OUT001
OUTC        OUTE        PARAM       SELCT0      SELCT1
SELCT2      SELCT3      YOUTPT
Enter the signal name or signal names separated by spaces.
PARAM YOUTPT
Enter the new value (MHz): 100
Enter the signal name or signal names separated by spaces.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:

Exiting and writing AMCCPKG.LST.
Processing MIN Front Annotation Delay File ...
Processing NOM Front Annotation Delay File ...
```

FIGURE 8 CONTINUED

AMCCPKG.LST

FIGURE 9

```
*****************************************************************
*          AMCC Output Loading System                           *
*          Version  1.ØØ                                        *
*****************************************************************
   Netlister version number =                    7Ø8
   SDI version number =                          2.1Ø
   Netlist generation date =                     2Ø SEP 1988
   Netlist generation time =                      9:58
   Engineering workstation type =                DAISY/DNIX
   Engineering workstation path name =           /USER/CLASS/MUX16
   Product Name =                                MUX16
   Product Number =                              XXXX
   Product Grade =                               MIL
   EWS Library =                                 Q5ØØØ
   EWS Library Rev =                             8Ø9
   Macro Parameter family =                      Q5ØØØ
   The ARRAY type =                              Q5ØØØT
   Chip Macro Name =                             Q5ØØØTTTL1ØK
   Circuit Family =                              Q5ØØØ
   Circuit Technology =                          P
   ECL Level =                                   1ØK

Package name = 224-PGA-CD
```

package selected

Item #	Signal Name	Instance Name	Macro Name	PAD	Frq MHz	ECL R	System pf	Package Capacitance Min pf	Package Capacitance Typ pf	Package Capacitance Max pf
1	DATØ	SØØØ1	IT11H							
2	DAT1	SØØØ2	IT11H							
3	DATIØ	SØØ11	IT11H							
15	DAT8	SØØØ9	IT11H							
16	DAT9	SØØ1Ø	IT11H							
17	EXTCLK	SØØ21	IE93		1ØØ.Ø					
18	EXTRST	SØØ22	IE93							
19	INØØ1	EIØØØØ	IE86							
2Ø	OUTØØ1	EOØØØØ	OE87				6.Ø	3.8Ø	4.4Ø	4.9Ø
21	OUTC	EOØØ1	OE87			5Ø.Ø	6.Ø	3.8Ø	4.4Ø	4.9Ø
22	OUTE	EOØØØ2	OE87				6.Ø	3.8Ø	4.4Ø	4.9Ø
23	PARAM	EO4ØØØ	OE81		1ØØ.Ø		6.Ø	8.ØØ	8.ØØ	8.ØØ
24	SELCTØ	SØØ2Ø	IT11H							
25	SELCT1	SØØ19	IT11H							
26	SELCT2	SØØ18	IT11H							
27	SELCT3	SØØ17	IT11H							
28	YOUTPT	DJØØØ	OE8Ø		1ØØ.Ø	55.Ø	12.Ø	12.ØØ	12.ØØ	12.ØØ
29	ITGND	GNTØ3	ITGND							

Using the interface, the frequency for any primary input or output may be specified (commentary documentation) and the ECL resistive loading for any ECL output may be specified (also commentary). Resistive loads should be specified when they do match that assumed for the macro (25 or 50 ohms). Frequency should be entered for any TTL I/O toggling faster than 50MHz and any ECL I/O toggling faster than 100MHz.

AMCCANN (AMCC Annotation) can be run as many times as required to fine-tune the simulation. The file OUTPUT.DLY is the data file created in the first session and then edited by successive executions. A previously entered specific pin capacitance or load capacitance for a signal is deleted (reverting to the default value) by giving its name and typing an * for the new value.

When AMCCANN is completed, it will have generated the Front-Annotation files FNTMIL.DSY or FNTCOM.DSY, FNTNOM.DSY and FNTMIN.DSY as well as the report file AMCCPKG.LST and the data file OUTPUT.DLY. All of these files are to be submitted.

Note: The BiCMOS library has two commercial timing libraries, one for COMMERCIAL circuits running with a -4.5V power supply (COM4) and one for all other COMMERCIAL circuits (COM5). The Front-Annotation file is still named FNTCOM.DSY and provides correct timing based on the ARRAY_FAMILY, PRODUCT_GRADE and the POWER_SUPPLY parameters.

After the ERC software and the Front-Annotation software has successfully run, the next step is to run a simulation to verify that the basic logic is correct.

When the proper loginfile, such as the one shown earlier, is used, the PROFILE file is correctly set for the library.

THE LIBRARY MUST BE INSTALLED AS AMCC INTENDS FOR PROPER OPERATION.

Run SIFT (Simulator Intermediate Files Translator) to append the appropriate timing data to the macros. These are the intrinsic delays for the paths through the macros themselves. SIFT can be run for MINIMUM, NOMINAL, COMMERCIAL or MILITARY timing. The worst-case multipliers are those that support the Front- and Back-Annotation software (1.35 * typical = COM and 1.45 * typical = MIL for the Q5000).

Run SIFT using the AMCC supplied shell:

RUN_SIFT xxx where xxx = NOM, MIL or COM, COM4 or COM5

Note: for large designs, SIFT may run overnight.

Within the SIFT library, there are three multipliers:

BIPOLAR

DLS/DTV	MODE ->	RANGE MIN	TYP	MAX	FRONT-ANNOTATION
SIFT Timing Library	NOM	0.90	1.00	1.10	FNTNOM.DSY
	COM	1.11	1.23	1.35	FNTCOM.DSY
	MIL	1.19	1.32	1.45	FNTMIL.DSY
	MIN	0.70	0.78	0.86	FNTMIN.DSY

BICMOS*

COM4 uses multipliers for -4.5V supply
- use with FNTCOM.DSY

COM5 uses multipliers for -5.2V or +5V supply
- use with FNTCOM.DSY

* see the tables in Volume I of the Design Manual

These numbers are based on a 20% maximum worst-case processing variation across the chip, with 10% the more usual variation. These multipliers are the same multipliers used in the FNTxxx.DSY files.

The use of MIN, TYP or MAX within a timing library is specified in the MODE format under the DAISY Logical Simulator (DLS) or the DAISY Timing Verifier (DTV).

SIFT is also run under the Super-Shell by typing:

```
RUN_AMCC                (Super-Shell)
3
```

AMCC recommends that the Super-Shell be used after the initial debug pass has been made since the menu will prompt you for necessary parameters and options for the commands.

Normally, a simulation will be run using MINIMUM data and then rerun using COMMERCIAL or MILITARY data. If the results are not functionally identical, a timing dependency (problem) is indicated which requires further evaluation or a re-design. AMCC requires both of these simulations (MINIMUM, and MAXIMUM MIL OR COM) be performed and submitted if they are different. Otherwise, submit just the maximum worst-case simulation results.

All submitted simulations should be run under timing checks "ON" and all timing errors removed. In special cases, an AMCC waiver may be required.

Following SIFT, the SING TO DAISY process is initiated. SING is the Simulator Input Generator used to extract information from the netlist. AMCC usually combines this into the following routines: 1) generate a default time=0 simulation control file, SOM_MCF.SING; 2) generate a default FMT file for DLS/DTV use; and open the SOM_MCF.SING file for editing under TEC.

```
SING -T -M3 -MCF /AMCC/SOM_MAKER/SOM_MCF
FMT_CSD.SING
TEC SOM_MCF.SING      (all part of Super-Shell)
```

At the end of the last command (and the end of menu selection 3 for the Super-Shell), the text-editor TEC is open to the simulation control file, SOM_MCF.SING. The time-zero SOM_MCF.SING file comes up with all input (all primary input signals) shown forced to zero at time zero (= 0:F0;). The user must edit this file to include the simulation stimuli by using the DAISY text editor, TEC. Stimuli can be described using the signal generator approach or by using a remote data file.

The example at the end of the text shows a SOM_MCF.SING file after it has been edited. This one uses the $SIGNAL_GENERATORS section to input values. TIMING_CHECK := 1; is present so this file can be used to run timing checks under both DLS and DTV.

AMCC prefers that all files be commented for identification (company, circuit, designer, date, rev level, what is being tested, etc.).

After all edits to the SOM_MCF.SING file have been made, TEC is exited by:

```
{ENTER}EXIT{EXECUTE}          (exit TEC)
```

The SOM CTL file must include an $OUTPUTS section. The output file referenced in the $OUTPUTS section is created by the system and not by the user. The user must describe which signals go in the file in what order. AMCC requires that all primary inputs, all primary outputs, all primary bidirectionals and all internal 3-state enable or bidirectional enable signals be listed in that file in that order.

The files produced during simulation include the LIST file, the WAVE, the VLAF output file. The VLAF output file is the file that becomes the input file to

```
Date:  25 MAY 86 11:17   File:  FMT_CSD.SING

SOM SOM_MCF.SING -M3
DLS <<!
FORMAT
0AT@@MUX16/3:DAT@
3DAT1@MUX16/3:DAT1
3DAT1@@MUX16/3:DAT1@
3DAT11@MUX16/3:DAT11
3DAT12@MUX16/3:DAT12
3DAT13@MUX16/3:DAT13
3DAT14@MUX16/3:DAT14
3DAT15@MUX16/3:DAT15
3DAT2@MUX16/3:DAT2
3DAT3@MUX16/3:DAT3
3DAT4@MUX16/3:DAT4
3DAT5@MUX16/3:DAT5
3DAT6@MUX16/3:DAT6
3DAT7@MUX16/3:DAT7
3DAT8@MUX16/3:DAT8
3DAT9@MUX16/3:DAT9
3EXTCLK@MUX16/2:EXTCLK
3EXTRST@MUX16/2:EXTRST
3IN@@1@MUX16/4:IN@@1
3SELCT@@MUX16/3:SELCT@
3SELCT1@MUX16/3:SELCT1
3SELCT2@MUX16/3:SELCT2
3SELCT3@MUX16/3:SELCT3
3OUT@@1@MUX16/4:OUT@@1
3OUTC@MUX16/4:OUTC
3OUTE@MUX16/4:OUTE
3YOUTPT@MUX16/2:YOUTPT
3S
PUT FMT
QUIT N
!
```

FIGURE 10: FMT_CSD.SING used to build FMT

FMT is used by DLS

AMCCSIMFMT. AMCC does not need to see the FMT file, LIST files, WAVE plots or the VLAF file.

The first three (FMT controls LIST and WAVE contents) are useful for debugging. For example, the user can reference internal signals in the LIST file while the final $OUTPUTS section is not allowed to show them.

The Super-Shell step 3 will end with the text editor open. When TEC is exited, the shell menu will return. If you need to create a remote data file, exit the Super-Shell and open the data file under TEC. If this is the case, exit the Super-Shell by:

```
0                          (exit Super-Shell)
```

If the input stimulus is to be provided by a remote data file, the TEC must be reopened to create whatever input file name was referenced in the simulation control file. To open, use TEC input-filename.

```
TEC DEMO.DAT               (example)
TEC filename.extension     user-defined filename
```

The data files are created from scratch by the designer; there is no fill-in-the-blank formatted file to edit as there is for the SOM_MCF.SING file. The data file must be created following a specified time-value, with comments and blank lines allowed. If comments and blank line appear after the END statement, then run AMCCFILUTL to prepare the file for execution. The source file is commented, the executable file is not. Maintanence is performed using the source file.

ALWAYS RENAME AN EDITED SOM_MCF.SING FILE. The SOM_MCF.SING file is destroyed in favor of a default version any time SING is rerun or anytime Super-Shell step 3 is executed.

AMCC prefers to have meaningful names used on the control files such as FUNCTION.SING, ATSPEED.SING, ACR4PROP.SING. Do not forget to include these in the submission index.

Once the data file and the SOM_MCF.SING file are completed, then run SOM, the Simulator Object Module Generator. This can be executed with the Front-Annotation file, and should be. The AMCC supplied shell is an easy way to do this.

SOM the control file only - for debug:

```
SOM SOM_MCF.SING -M3 -L SOM.ERR
```

```
/****
*
* DESIGN PATH /USER/CLASS/MUX16   DATE 16-SEP-1988 14:50
*
* 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
@MUX16/3:DAT0 := @0:F0 ;
@MUX16/3:DAT1 := @0:F0 ;
@MUX16/3:DAT10 := @0:F0 ;
@MUX16/3:DAT11 := @0:F0 ;
@MUX16/3:DAT12 := @0:F0 ;
@MUX16/3:DAT13 := @0:F0 ;
@MUX16/3:DAT14 := @0:F0 ;
@MUX16/3:DAT15 := @0:F0 ;
@MUX16/3:DAT2 := @0:F0 ;
@MUX16/3:DAT3 := @0:F0 ;
@MUX16/3:DAT4 := @0:F0 ;
@MUX16/3:DAT5 := @0:F0 ;
@MUX16/3:DAT6 := @0:F0 ;
@MUX16/3:DAT7 := @0:F0 ;
@MUX16/3:DAT8 := @0:F0 ;
@MUX16/3:DAT9 := @0:F0 ;
@MUX16/2:EXTCLK := @0:F0 ;
@MUX16/2:EXTRST := @0:F0 ;
@MUX16/4:IN001 := @0:F0 ;
@MUX16/3:SELCT0 := @0:F0 ;
@MUX16/3:SELCT1 := @0:F0 ;
@MUX16/3:SELCT2 := @0:F0 ;
@MUX16/3:SELCT3 := @0:F0 ;
```

SOM_MCF.SING

FIGURE 11

SOM and TCAL combined:

```
RUN_SOM SOM_MCF.SING FNTMIL.DSY
                          (part of Super-Shell)
```

Caution: RUN_SOM looks for the FNTxxx.DSY files in the top level directory. Always run DANCE, DRINK, SOM and SING from the top of the design tree.

SOM must be executed anytime the SOM_MCF.SING file is edited to link the new information to the simulation database.

```
RUN_AMCC  (if you aren't in the shell)
4                          select SOM
<control_file_name>        control file name
FNTxxx.DSY                 Annotation file name
0                          exit shell
```

When the steps represented in the AMCC Super-Shell are completed, the next step is to run either DLS, the DAISY Logic Simulator, or DTV, the DAISY Timing Verifier.

DLS is called by:

```
DLS
    GET FMT    using the default - use unless YOU want
               something different; leave no spaces
               after FMT
     MODE      set mode
    {edit for MAX}   for military or commercial;
    {edit for MIN}   for minimum
     {ENTER}
    PUT DLS_FMT      save the FMT and MODE windows
    VIEW 9999 10000  proper view step for Q5000 Series;
                        Q3500 Series; Q14000 Series
    RUN 1000000      run as long as you need to run
    START 0           position data in display buffer
    WAVE              plot by "Plot Screen" mouse menu
     {ENTER}
    LIST  S DEMO     listing saved as file "DEMO"
    {ENTER}
    RESTART 0
    VIEW 99 100      closer look - not for vectors
    RUN 200000           that will be submitted
    WAVE
      {ENTER}
    QUIT{EXECUTE}    don't save
  N{EXECUTE}
```

After DLS is run to satisfaction (the vector set in the example is 100% fault coverage), the vectors that will be submitted for use in test are generated using the AMCCSIMFMT program. This program will take the file produced by the $OUTPUTS section and reformat it. The

```
CLASS> RUN_SOM FUNCTION.SING FNTMIL.DSY

        ********** RUNNING SOM and TCAL **********

Daisy Simulator Object Module(SOM) Generator 5.2.9  (SOM V5.02.02        )25-May-86 15:32:00   FUNCTION.SING

 Copyright (C) 1984 DAISY SYSTEMS  Incorporated.

        **********  SOM FORMAT OK !!!!!  **********

TCAL:  TIMING CALCULATION, V5.02.02

TCAL:  INITIALIZATION
TCAL:      LOADING DATA FROM SOM FILE
TCAL:  PARSING INPUT FILE
TCAL:      PARSING CONTROL SECTION
TCAL:      PARSING DELAY SECTIONS


***** PARSING INPUT FILE COMPLETED.
***** TOTAL NUMBER OF ERRORS FOUND = 0

TCAL:  PARSING INPUT FILE COMPLETED, NO ERROR FOUND.


TCAL:      PARSING SIFT FILES

TCAL: NULL INPUT FROM SIFT FILES.

TCAL:
TCAL:  PROCESSING DELAY CALCULATION
TCAL:
  SIGNAL NAME:
      NOM RISE  MIN RISE  MAX RISE  NOM FALL  MIN FALL  MAX FALL

    @MUX16/2:INTQN(#86) :
           13, 12, 14, 26, 23, 28
    @MUX16/2:INTCLK(#94) :
           26, 23, 28, 26, 23, 28
    @MUX16/2:INTRST(#97) :
           26, 23, 28, 26, 23, 28
```

FIGURE 12: RUN_SOM

```
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP5(#603) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP6(#605) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP7(#607) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP8(#609) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP9(#611) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP10(#613) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP11(#615) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP12(#617) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP13(#619) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP14(#621) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:INP15(#623) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL3(#625) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL2(#627) :
            13, 12, 14, 26, 23, 28
    @MUX16/3:SEL1(#629) :
            36, 33, 40, 72, 65, 80
    @MUX16/3:SEL0(#631) :
            36, 33, 40, 72, 65, 80
    @MUX16/4:G0000.C(#652) :
            0, 0, 0, 0, 0, 0
    @MUX16/4:G0000.E(#653) :
            0, 0, 0, 0, 0, 0
    @MUX16/4:XSIG5(#654) :
            0, 0, 0, 0, 0, 0


***** DELAY CALCULATION COMPLETED.
***** TOTAL NUMBER OF ERRORS FOUND = 0

TCAL:  DELAY CALCULATION COMPLETED, NO ERROR
TCAL:
TCAL:  UPDATING DT TABLE
TCAL:      UNLOADING DATA TO SOM FILE
TCAL:  TERMINATION

        ********** TCAL FORMAT OK !!!!! **********

CLASS>
```

FIGURE 13: END OF RUN_SOM - TCAL

DLS output file format must be binary for a proper submission. An example AMCCSIMFMT output is shown with the example.

When 3-state macros are included in a design, it is often desirable to see the high-Z state in the listing during debug. The DLS/DTV format can be edited by:

```
FO
  edit          use switch key to toggle options
{ENTER}
PUT DLS_FMT       save for later reference
```

The FORMAT file and the MODE file can be printed out using the PLOT_SCREEN menu command. They can also be saved to a file for printing using the clipsheet options and they can be saved on disk and recalled in another run. All formats (FORMAT, MODE, ACQUIRE, BREAK, TRIGGER) are saved at the same time using:

```
PUT <filename>
```

Another feature that is useful is the PRINT_ON_CHANGE file which can be created. The PRINT_ON_CHANGE file can be requested by editing the SOM_MCF.SING control file $OUTPUTS section to provide the instruction to collect data only on the change of monitored signals, rather than collect sampled data as before. Only the output file is affected.

The format for the LIST and WAVE are tied to the FORMAT file and will not vary. The format for the output file is what is listed in the $OUTPUTS section of the simulation control file. Only those signals listed in the control file $OUTPUTS section will be monitored or recorded.

When the output file is the one desired, run AMCCSIMFMT by typing:

```
AMCCSIMFMT
```

and responding to the prompts.

You can also call the program using the Super-Shell:

```
RUN_AMCC
5               calls AMCCSIMFMT
```

You will be asked for the name of the output file (as listed in the $OUTPUTS section of the control file), the new name you want to use, what format you want with parenthesis (choose "1") and if you want spaces between columns or groups of columns. The spacing is up to you.

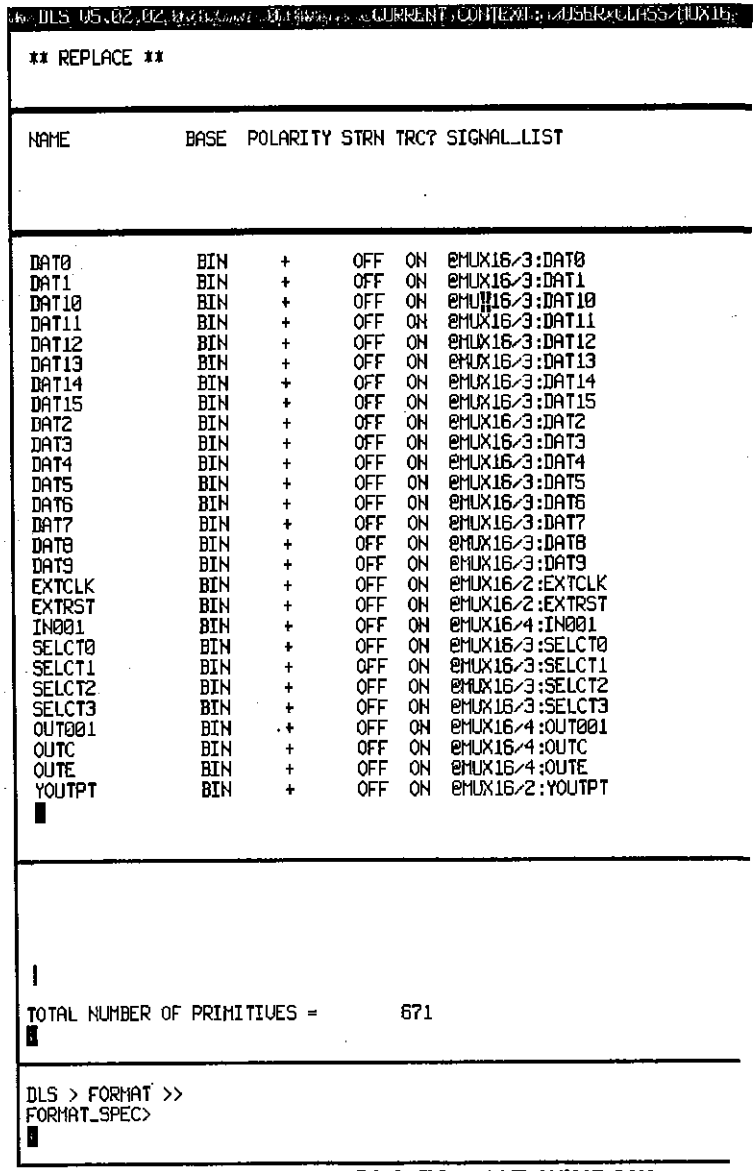


FIGURE 14: DLS FORMAT WINDOW

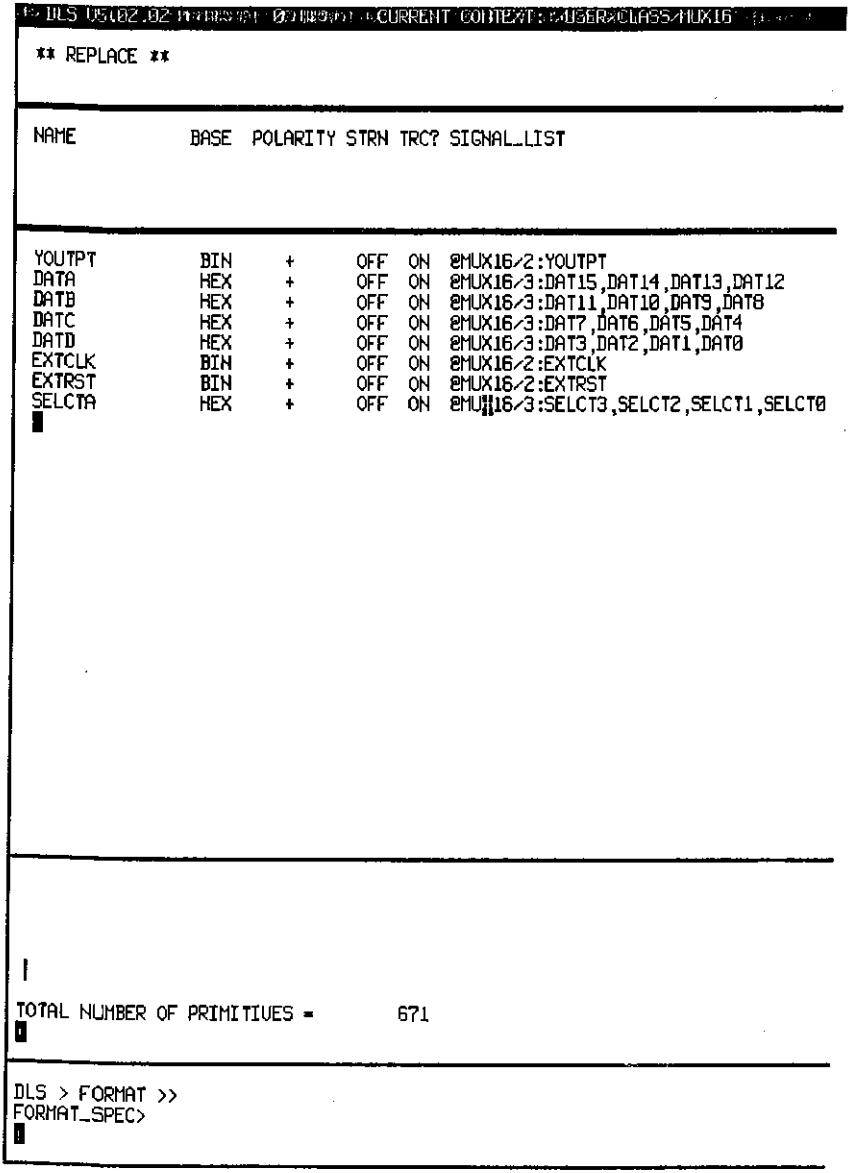


FIGURE 15: Edited DLS FORMAT WINDOW – HEX

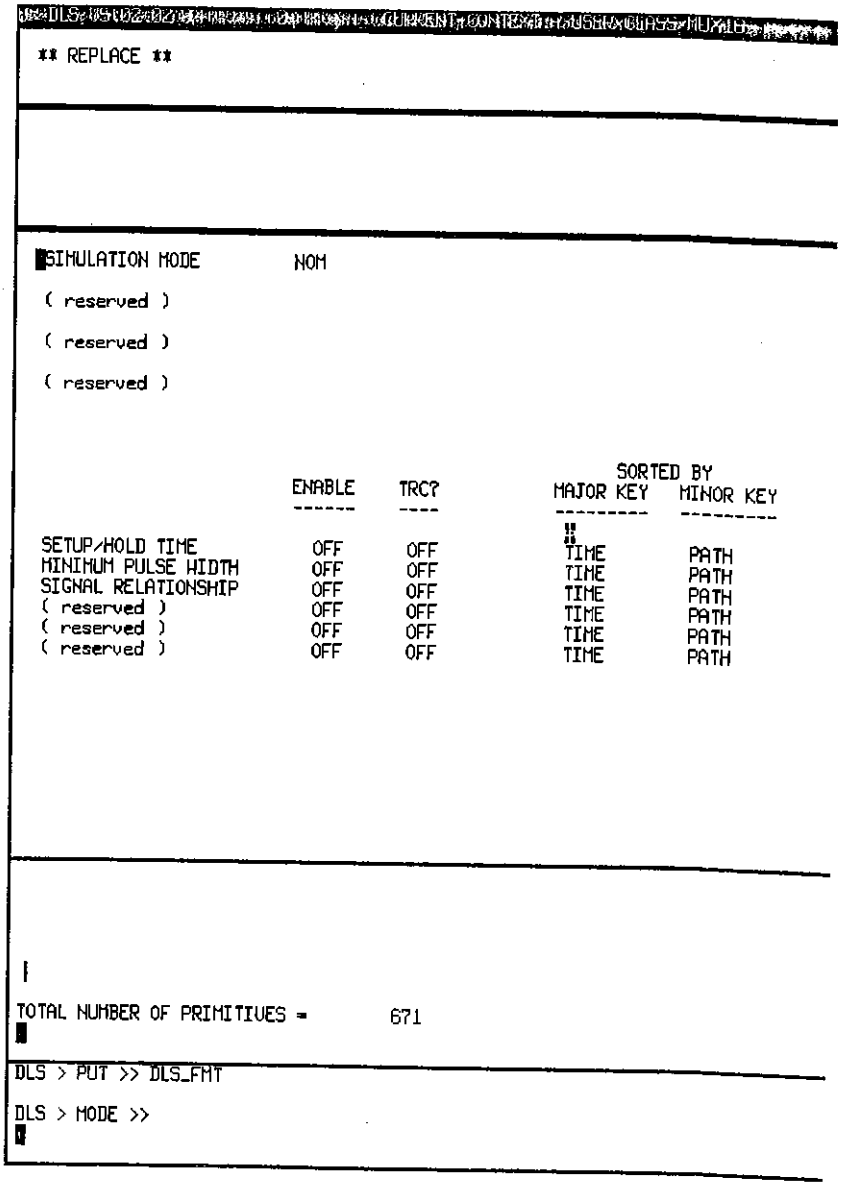


FIGURE 16: DLS MODE WINDOW – NOMINAL

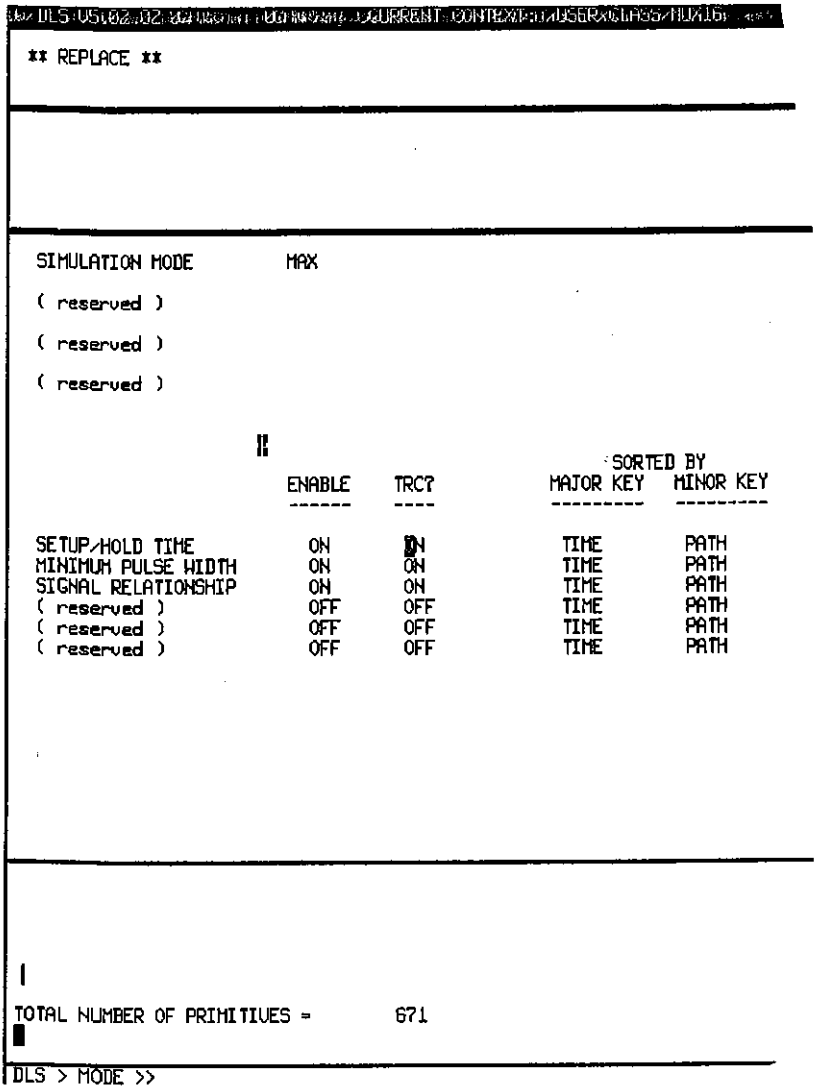


FIGURE 17: DLS MODE WINDOW – Edited for MAXIMUM

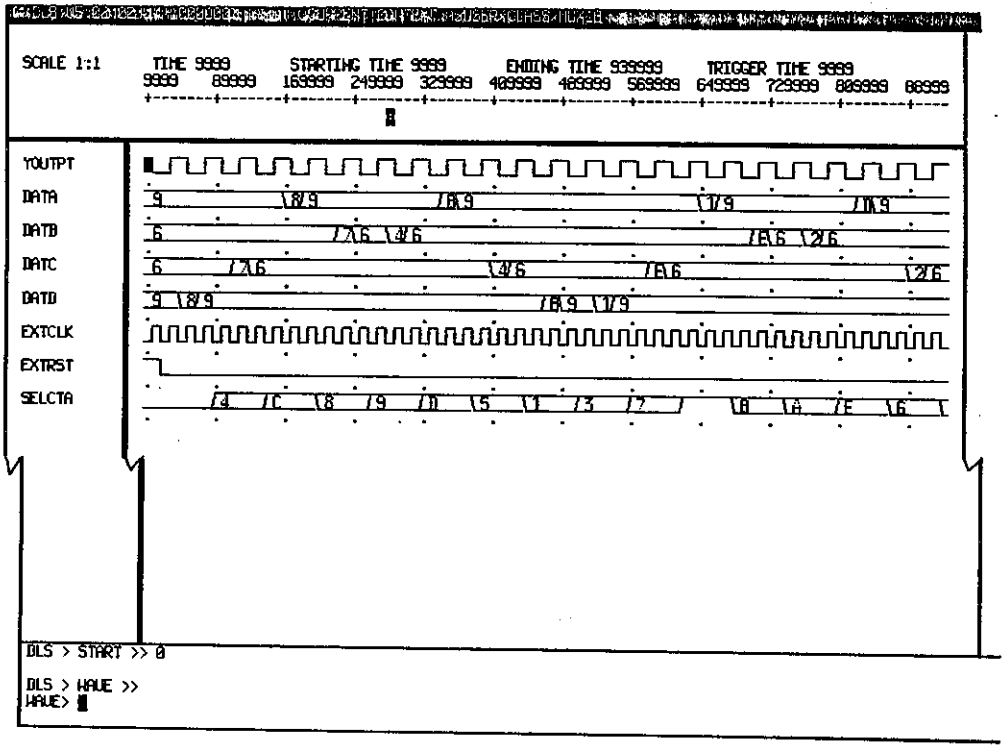
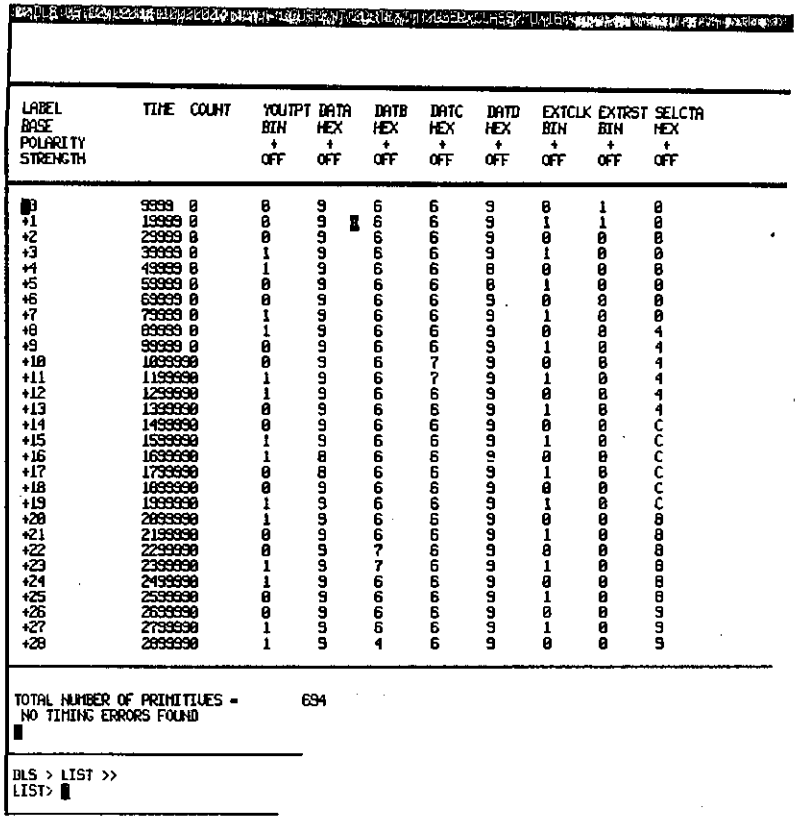


FIGURE 18: DLS WAVE - HEX FORMAT



LABEL BASE POLARITY STRENGTH	TIME	COUNT	YOUTPT BIN + OFF	DATA HEX + OFF	DATB HEX + OFF	DATC HEX + OFF	DATD HEX + OFF	EXTCLK BIN + OFF	EXTRST BIN + OFF	SELCTA HEX + OFF
0	9999	0	0	9	6	6	9	0	1	0
+1	19999	0	0	9	6	6	9	1	1	0
+2	29999	0	0	9	6	6	9	0	0	8
+3	39999	0	1	9	6	6	9	1	0	0
+4	49999	0	1	9	6	6	8	0	0	8
+5	59999	0	0	9	6	6	8	1	0	0
+6	69999	0	0	9	6	6	9	0	0	0
+7	79999	0	1	9	6	6	9	1	0	0
+8	89999	0	1	9	6	6	9	0	0	4
+9	99999	0	0	9	6	6	9	1	0	4
+10	1099998		0	9	6	7	9	0	0	4
+11	1199998		1	9	6	7	9	1	0	4
+12	1299998		1	9	6	6	9	0	0	4
+13	1399998		0	9	6	6	9	1	0	4
+14	1499998		0	9	6	6	9	0	0	C
+15	1599998		1	9	6	6	9	1	0	C
+16	1699998		1	8	6	6	9	0	0	C
+17	1799998		0	8	6	6	9	1	0	C
+18	1899998		0	9	6	6	9	0	0	C
+19	1999998		1	9	6	6	9	1	0	C
+20	2099998		1	9	6	6	9	0	0	8
+21	2199998		0	9	6	6	9	1	0	8
+22	2299998		0	9	7	6	9	0	0	8
+23	2399998		1	9	7	6	9	1	0	8
+24	2499998		1	9	6	6	9	0	0	8
+25	2599998		0	9	6	6	9	1	0	8
+26	2699998		0	9	6	6	9	0	0	9
+27	2799998		1	9	6	6	9	1	0	9
+28	2899998		1	9	4	6	9	0	0	9

TOTAL NUMBER OF PRIMITIVES = 694
NO TIMING ERRORS FOUND

DLS > LIST >>
LIST>

FIGURE 19: DLS LIST WITH HEX FORMAT

Date: 25 MAY 86 13:34 File: SAM

LABEL BASE POLARITY STRENGTH	TIME	COUNT	YOUTPT BIN + OFF	DATA HEX + OFF	DAT8 HEX + OFF	DATC HEX + OFF	DAT0 HEX + OFF	EXTCLK BIN + OFF	EXTRST BIN + OFF	SELCTA HEX + OFF
0	9999	0	0	9	6	6	9	0	1	0
+1	19999	0	0	9	6	6	9	1	1	0
+2	29999	0	0	9	6	6	9	0	0	0
+3	39999	0	1	9	6	6	9	1	0	0
+4	49999	0	1	9	6	6	8	0	0	0
+5	59999	0	0	9	6	6	8	1	0	0
+6	69999	0	0	9	6	6	9	0	0	0
+7	79999	0	1	9	6	6	9	1	0	0
+8	89999	0	1	9	6	6	9	0	0	1
+9	99999	0	0	9	6	6	9	1	0	1
+10	109999	0	0	9	6	7	9	0	0	1
+11	119999	0	1	9	6	7	9	1	0	1
+12	129999	0	1	9	6	6	9	0	0	1
+13	139999	0	0	9	6	6	9	1	0	1
+14	149999	0	0	9	6	6	9	0	0	0
+15	159999	0	1	9	6	6	9	1	0	0
+16	169999	0	1	8	6	6	9	0	0	0
+17	179999	0	0	8	6	6	9	1	0	0
+18	189999	0	0	9	6	6	9	0	0	0
+19	199999	0	1	9	6	6	9	1	0	0
+20	209999	0	1	9	6	6	9	0	0	0
+21	219999	0	0	9	6	6	9	1	0	0
+22	229999	0	0	9	7	6	9	0	0	0
+23	239999	0	1	9	7	6	9	1	0	0
+24	249999	0	1	9	6	6	9	0	0	0
+25	259999	0	0	9	6	6	9	1	0	0
+26	269999	0	0	9	6	6	9	0	0	0
+27	279999	0	1	9	6	6	9	1	0	0
+28	289999	0	1	9	4	6	9	0	0	0
+29	299999	0	0	9	4	6	9	1	0	0
+30	309999	0	0	9	6	6	9	0	0	0
+31	319999	0	1	9	6	6	9	1	0	0
+32	329999	0	1	9	6	6	9	0	0	0
+33	339999	0	0	9	6	6	9	1	0	0
+34	349999	0	0	8	6	6	9	0	0	0
+35	359999	0	1	8	6	6	9	1	0	0
+36	369999	0	1	9	6	6	9	0	0	0
+37	379999	0	0	9	6	6	9	1	0	0
+38	389999	0	0	9	6	6	9	0	0	0
+39	399999	0	1	9	6	6	9	1	0	0
+40	409999	0	1	9	6	4	9	0	0	0
+41	419999	0	0	9	6	4	9	1	0	0
+42	429999	0	0	9	6	6	9	0	0	0
+43	439999	0	1	9	6	6	9	1	0	0
+44	449999	0	1	9	6	6	9	0	0	1
+45	459999	0	0	9	6	6	9	1	0	1
+46	469999	0	0	9	6	6	8	0	0	1
+47	479999	0	1	9	6	6	8	1	0	1
+48	489999	0	1	9	6	6	9	0	0	1
+49	499999	0	0	9	6	6	9	1	0	1
+50	509999	0	0	9	6	6	9	0	0	0
+51	519999	0	1	9	6	6	9	1	0	0
+52	529999	0	1	9	6	6	1	0	0	0
+53	539999	0	0	9	6	6	1	1	0	0
+54	549999	0	0	9	6	6	9	0	0	0
+55	559999	0	1	9	6	6	9	1	0	0
+56	569999	0	1	9	6	6	9	0	0	0
+57	579999	0	0	9	6	6	9	1	0	0

FIGURE 20: PARTIAL "SAVED" LIST FILE

USE:

DLS>LIST S SAM

```
Date:  25 MAY 86 13:35   File:  FUNCTION.VLAF

$DATA_HEADER$
$TYPE$
I/O
$FORMAT$
TIME_VALUE
$TOTAL_COLUMN$
10 23
$BASE$
D B
$FIELD$ 1
@MUX16/2:EXTCLK 1
$FIELD$ 2
@MUX16/2:EXTRST 2
$FIELD$ 3
@MUX16/3:SELCT3 3
$FIELD$ 4
@MUX16/3:SELCT2 4
$FIELD$ 5
@MUX16/3:SELCT1 5
$FIELD$ 6
@MUX16/3:SELCT0 6
$FIELD$ 7
@MUX16/3:DAT7 7
$FIELD$ 8
@MUX16/3:DAT6 8
$FIELD$ 9
@MUX16/3:DAT5 9
$FIELD$ 10
@MUX16/3:DAT4 10
$FIELD$ 11
@MUX16/3:DAT3 11
$FIELD$ 12
@MUX16/3:DAT2 12
$FIELD$ 13
@MUX16/3:DAT1 13
$FIELD$ 14
@MUX16/3:DAT0 14
$FIELD$ 15
@MUX16/3:DAT9 15
$FIELD$ 16
@MUX16/3:DAT8 16
$FIELD$ 17
@MUX16/3:DAT7 17
$FIELD$ 18
@MUX16/3:DAT6 18
$FIELD$ 19
@MUX16/3:DAT5 19
$FIELD$ 20
@MUX16/3:DAT4 20
$FIELD$ 21
@MUX16/3:DAT3 21
$FIELD$ 22
@MUX16/3:DAT2 22
$FIELD$ 23
@MUX16/2:YOUTPT 23
$END$
#          1
9999       0
19999      1
29999      0
39999      1
49999      0
59999      1
69999      0
79999      1
89999      0
99999      1
109999     0
119999     1
129999     0
139999     1
149999     0
159999     1
169999     0
179999     1
189999     0
199999     1
209999     0
219999     1
229999     0
239999     1
249999     0
259999     1
269999     0
279999     1
289999     0
299999     1
309999     0
319999     1
329999     0
339999     1
349999     0
359999     1
369999     0
379999     1
389999     0
399999     1
409999     0
419999     1
429999     0
439999     1
449999     0
459999     1
469999     0
479999     1
489999     0
499999     1
509999     0
519999     1
529999     0
539999     1
549999     0
559999     1
569999     0
579999     1
589999     0
599999     1
609999     0
619999     1
629999     0
639999     1
649999     0
659999     1
669999     0
679999     1
689999     0
699999     1
709999     0
719999     1
729999     0
739999     1
749999     0
759999     1
769999     0
779999     1
789999     0
799999     1
809999     0
819999     1
829999     0
839999     1
849999     0
859999     1
869999     0
879999     1
889999     0
899999     1
909999     0
919999     1
929999     0
939999     1
949999     0
959999     1
969999     0
979999     1
989999     0
999999     1
```

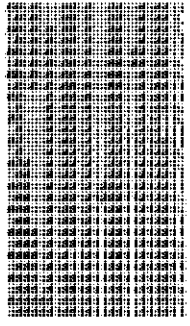


FIGURE 21: OUTPUT VLAF FILE

FROM DLS RUN

```
CLASS> AMCCSIMFMT
VLAIF Conversion    Rev[1.0]

    Enter VLAIF (input) file name: FUNCTION.VLAF

    Enter SIM (output) file name: OUTPUT.FUN


    Choose from menu -

       1 : VLAIF to SIM FORMAT (NO parenthesis).
       2 : VLAIF to SIM FORMAT (with parenthesis).

    Enter choice [1 or 2]: 1


    Do you want DATA seperated in columns [Y / N] ? N


    <<< Processing VLAIF Nets  >>>
    <<< Processing VLAIF Vectors  >>>
    >>>  Time ZERO (0) not output from VLAIF file.
    >>>  (100) lines of Vectors output.

    >>> AMCCSIMFMT conversion completed with NO error(s).
CLASS> TYPE OUTPUT.FUN
```

FIGURE 22: AMCCSIMFMT sample execution

```
Date:  25 MAY 86 13:36    File:  OUTPUT.FUN

1***CIRCUIT IDENTIFICATION =



                       EESSSSDDDDDDDDDDDDDDDDY
                       XXEEEEAAAAAAAAAAAAAAAAO
                       TTLLLLTTTTTTTTTTTTTTTTU
                       CRCCCCØ123456789111111T
                       LSTTTT          Ø12345P
                       KT321Ø                T

              TIME

 9999                  Ø1ØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 19999                 11ØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 29999                 ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 39999                 1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
 49999                 ØØØØØØØØØ1Ø11ØØ11Ø1ØØ11
 59999                 1ØØØØØØØØ1Ø11ØØ11Ø1ØØ1Ø
 69999                 ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 79999                 1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
 89999                 ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ11
 99999                 1ØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 1Ø9999                ØØØ1ØØ1ØØ1111ØØ11Ø1ØØ1Ø
 119999                1ØØ1ØØ1ØØ1111ØØ11Ø1ØØ11
 129999                [illegible]
 139999                [illegible]
 149999                [illegible]
 159999                [illegible]
 169999                [illegible]
 179999                [illegible]
 189999                [illegible]
 199999                [illegible]
 2Ø9999                [illegible]
 219999                [illegible]
 229999                [illegible]
 239999                [illegible]
 249999                [illegible]
 259999                [illegible]
 269999                [illegible]
 279999                [illegible]
 289999                [illegible]
 299999                [illegible]
 3Ø9999                [illegible]
 319999                [illegible]
 329999                [illegible]
 339999                [illegible]
 349999                [illegible]
 359999                [illegible]
 369999                ØØ11Ø11ØØ1Ø11ØØ11Ø1ØØ11
 379999                1Ø11Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 389999                ØØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 399999                1ØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ11
 4Ø9999                ØØØ1Ø11ØØ1ØØ1ØØ11Ø1ØØ11
 419999                1ØØ1Ø11ØØ1ØØ1ØØ11Ø1ØØ1Ø
 429999                ØØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ1Ø
 439999                1ØØ1Ø11ØØ1Ø11ØØ11Ø1ØØ11
 449999                ØØØØØ11ØØ1Ø11ØØ11Ø1ØØ11
 459999                1ØØØØ11ØØ1Ø11ØØ11Ø1ØØ1Ø
 469999                ØØØØØ111Ø1Ø11ØØ11Ø1ØØ1Ø
 479999                1ØØØØ111Ø1Ø11ØØ11Ø1ØØ11
 489999                ØØØØØ11ØØ1Ø11ØØ11Ø1ØØ11
 499999                1ØØØØ11ØØ1Ø11ØØ11Ø1ØØ1Ø
 5Ø9999                ØØØØ111ØØ1Ø11ØØ11Ø1ØØ1Ø
 519999                1ØØØ111ØØ1Ø11ØØ11Ø1ØØ11
 529999                ØØØØ111ØØØØØ1ØØ11Ø1ØØ11
 539999                1ØØØ111ØØØØØ1ØØ11Ø1ØØ1Ø
 549999                ØØØØ111ØØ1Ø11ØØ11Ø1ØØ1Ø
 559999                1ØØØ111ØØ1Ø11ØØ11Ø1ØØ11
 569999                ØØØ1111ØØ1Ø11ØØ11Ø1ØØ11
 579999                1ØØ1111ØØ1Ø11ØØ11Ø1ØØ1Ø
 589999                ØØØ1111ØØ1Ø111Ø11Ø1ØØ1Ø
 599999                1ØØ1111ØØ1Ø111Ø11Ø1ØØ11
 6Ø9999                ØØØ1111ØØ1Ø11ØØ11Ø1ØØ11
 619999                1ØØ1111ØØ1Ø11ØØ11Ø1ØØ1Ø
 629999                ØØ11111ØØ1Ø11ØØ11Ø1ØØ1Ø
 639999                1Ø11111ØØ1Ø11ØØ11Ø1ØØ11
 649999                ØØ11111ØØ1Ø11ØØ11Ø1ØØØ1
 659999                1Ø11111ØØ1Ø11ØØ11Ø1ØØØØ
 669999                ØØ11111ØØ1Ø11ØØ11Ø1ØØ1Ø
 679999                1Ø11111ØØ1Ø11ØØ11Ø1ØØ11
 689999                ØØ1Ø111ØØ1Ø11ØØ11Ø1ØØ11
 699999                1Ø1Ø111ØØ1Ø11ØØ11Ø1ØØ1Ø
 7Ø9999                ØØ1Ø111ØØ1Ø11ØØ1111ØØ1Ø
 719999                1Ø1Ø111ØØ1Ø11ØØ1111ØØ11
 729999                ØØ1Ø111ØØ1Ø11ØØ11Ø1ØØ11
 739999                1Ø1Ø111ØØ1Ø11ØØ11Ø1ØØ1Ø
 749999                ØØ1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 759999                1Ø1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 769999                ØØ1Ø1Ø1ØØ1Ø11ØØ1ØØ1ØØ11
 779999                1Ø1Ø1Ø1ØØ1Ø11ØØ1ØØ1ØØ1Ø
 789999                ØØ1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 799999                1Ø1Ø1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 8Ø9999                ØØ111Ø1ØØ1Ø11ØØ11Ø1ØØ11
 819999                1Ø111Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 829999                ØØ111Ø1ØØ1Ø11ØØ11Ø1Ø11Ø
 839999                1Ø111Ø1ØØ1Ø11ØØ11Ø1Ø111
 849999                ØØ111Ø1ØØ1Ø11ØØ11Ø1ØØ11
 859999                1Ø111Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 869999                ØØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 879999                1ØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ11
 889999                ØØØ11Ø1ØØ1Ø1ØØØ11Ø1ØØ11
 899999                1ØØ11Ø1ØØ1Ø1ØØØ11Ø1ØØ1Ø
 9Ø9999                ØØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 919999                1ØØ11Ø1ØØ1Ø11ØØ11Ø1ØØ11
 929999                ØØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 939999                1ØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 949999                ØØØØ1Ø1Ø11Ø11ØØ11Ø1ØØ1Ø
 959999                1ØØØ1Ø1Ø11Ø11ØØ11Ø1ØØ11
 969999                ØØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ11
 979999                1ØØØ1Ø1ØØ1Ø11ØØ11Ø1ØØ1Ø
 989999                ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1Ø
 999999                1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ11
```

FIGURE 23: OUTPUT FROM AMCCSIMFMT

When the AMCCSIMFMT program is complete, execute AMCCVRC, the AMCC Vector Rules Checker. This program operates on the AMCCSIMFMT output file. All VRC errors must be removed before submission. Create a signal analysis file using TEC and then call AMCCVRC by typing

```
AMCCVRC
```

and responding to the prompts.

AMCCVRC is also called by the Super-Shell:

```
RUN_AMCC
6
```

Sample session for AMCCVRC:

```
TEC signal_control_file_name
     {edit}                (create signal
                                 analysis file)
{ENTER}EXIT{EXECUTE}
AMCCVRC
ERC/CIRCUIT.SDI    (netlist - subdirectory)
AMCCSIMFMT_file_name
    {select test or tests}
signal_control_file       (for race test)
```

AMCCVRC is only run against the maximum worst-case sampled functional simulation output. It produces AMCCVRC.LST and must be error-free for submission. Note: RESET, SET will usually generate Vector SSO errors that can be documented for what they are. These are the only allowed "errors". Consult AMCC if you have others.

After the logic has been verified via the functional simulation, run the at-speed simulation. This will require a smaller VIEW step than was used with the functional simulation to pick up the resolution. The timing step should reflect the speed at which the circuit will be operated (hence the term "at-speed" simulation). The simulation should be run at the Maximum specified operating frequency.

The "at-speed" simulation is used to check timing. At-speed simulation can be performed with either DLS or DTV. Using DTV allows an uncertainty range analysis. (Verify if DTV is operational on your system before scheduling its use in the design process.)

If the resolution of the waveform is good enough, the T and {ENTER} keys can be used to find the propagation delay between two edges when a waveform is on the screen. Put the cursor on one and strike T . Put the cursor on

the other and strike {ENTER}. The time will be displayed.

Note: AMCC requires sampled (uniform step) vectors for functional, at-speed and AC test simulations and requires PRINT_ON_CHANGE results for at-speed and AC Test. The designer should refer to Section 4, Vector Submission Rules and Guidelines for further information on vector requirements.

```
Date:  20 SEP 88 12:12   File:  RUN_DLS

DLS <<!
GET DLS_FMT
VIEW 9999 10000
RUN 1030000
QUIT
N
!
```

```
Date:  20 SEP 88 12:12   File:  RUN_ASDLS

DLS <<!
GET DLS_FMT
VIEW  499 500
RUN 50000
QUIT
N
!
```

FIGURE 24

PARAMETRIC TESTS

This circuit was constructed with a gate tree to allow the development of a parametric vector set using the preferred AMCC methodology. The SOM control file previously created for the function vectors can be copied into a new area (PARAMETER.SING was used as the file name) and the signal generators edited to allow one and only one input at a time to switch from "1" to "0" and back to "1".

Because the usued output on the input macro was used, the input signal is inverted. The logic gate used requires all input signals start at "1". The output "PARAM" starts at "0" and switches with each change. All inputs are exercised, including clock, reset and enables, if any.

The output format for the simulation output file is the same as for the functional simulation output file. The sampling is the same as for the functional simulation (100ns).

Figures A-20 and A-21 in Appendix A show the control file and the AMCCSIMFMT output file. The output file is SIMFMTPR.01

The simulation was run using the commands:

```
RUN_SOM PARAMETER.SING FNTMIL.DSY
DLS
GET FMT
VIEW 9999 10000
RUN 450000
QUIT
N
AMCCSIMFMT
FUNCTION.VLAF
SIMFMTPR.01
1
Y
22
TYPE SIMFMTPR.01 > /DEV/LP -HEA
```

THE BEST APPROACH (FLAT DESIGN):

1. Capture page or pages

   - Run DANCE on each page after proceeding to the next page - use a window
     DANCE n -M3
   - Check individual pages as they run and correct errors and rerun DANCE
     TYPE n.DFR in a window
   - When enough captured (the critical path, other areas of concern, etc.), then run DRINK
     DRINK -M3
   - NOTE: NOT ALL DANCE ERRORS MUST BE REMOVED FOR DRINK TO RUN SO BE CERTAIN ALL n.DFR FILES ARE 2 ERRORS ONLY (THE BORDER ERRORS)
   - NOTE: NOT ALL DRINK ERRORS MUST BE REMOVED FOR ERCS TO RUN

2. When DANCE and DRINK are error free or as you want them (partial circuits will still have errors) then run individual shells (on a large circuit this is faster):
   RUN_AGIF
   RUN_ERC
   RUN_ANN

   - Use TEC to add comments to FNTMIL.DSY or FNTCOM.DSY as well as FNTMIN.DSY

3. Now run the super-shell:
   RUN_AMCC
   and select "3" SIFT
   Answer the prompts

4. Step 3 leaves you in TEC so edit the SOM_MCF.SING file and create any data file needed

5. Run or continue to run the super shell:
   RUN_AMCC
   and select "4" SOM
   Answer the prompts

6. Exit the shell and run DLS or DTV

7. Run AMCCSIMFMT

8. Use TEC to create the signal analysis file

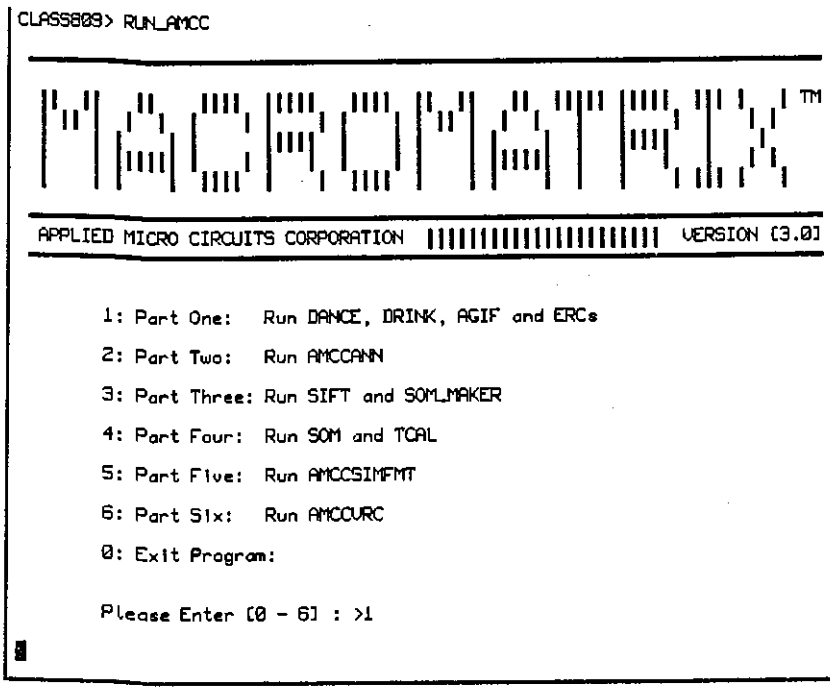
9. Run AMCCVRC

APNOTE 1
APPENDIX A
TRANSCRIPT OF CIRCUIT

(809)

Figures:

Note: for AC TEST files, see Volume II, Section 4



DANCE, DRINK, AGIF, ERCs

FIGURE A1

APPLIED MICRO CIRCUITS CORPORATION ▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌ VERSION [3.0]

```
        1: Part One:   Run DANCE, DRINK, AGIF and ERCs

        2: Part Two:   Run AMCCANN

        3: Part Three: Run SIFT and SOM_MAKER

        4: Part Four:  Run SOM and TCAL

        5: Part Five:  Run AMCCSIMFMT

        6: Part Six:   Run AMCCVRC

        0: Exit Program:

        Please Enter [0 - 6] : >1

        ****** RUNNING DANCE -T -N -ERR -M3 ********

        ************** DANCE OK !!!!! ****************

        ************ RUNNING DRINK -T -M3 ************

        ************** DRINK OK !!!!! ****************

        ********* RUNNING AGIF NETLISTER ************


        ********* AGIF NETLIST COMPLETE !!! ********

        ************** RUNNING AMCCERC ***************
AMCC MacroMatrix ERCs VERSION  3.40
Loading Netlist ...
 Signal I/O List written to AMCCIO.LST
 Signal/Component Cross-Reference written to AMCCXREF.LST
Checking Population.
I/O Statistics Check.
```

FIGURE A1

Continued

```
shell:05.02:08                    CURRENT CONTEXT: /USER/CLASS/MUX16

          ************ RUNNING AMCCERC ************

AMCC MacroMatrix ERCs VERSION  3.40
Loading Netlist ...
 Signal I/O List written to AMCCIO.LST
 Signal/Component Cross-Reference written to AMCCXREF.LST
Checking Population.
I/O Statistics Check.
Checking Valid Names.
Checking Pin Class.
Checking Fan-Out.
Checking Internal Pin Count.
Checking Pin Hookup and Unused Pins.
Checking Bipolar Macro Occurrence and Power Dissipation.
Checking Circuit Technology.
All Done.

          ************ AMCCERC OK !!!!! ************
```

FIGURE A1

Continued

```
Date:  19 SEP 88 14:32    File:  ERC/AMCCERC.LST


*************************************************************
*          AMCC Schematic Data Interface                    *
*          Revision [1.Ø]                                   *
*************************************************************


   Netlister version number =              7Ø8
   SDI version number =                    2.1Ø
   Netlist generation date =               19 SEP 1988
   Netlist generation time =               14:25
   Engineering workstation type =          DAISY/DNIX
   Engineering workstation path name =     /USER/CLASS/MUX16
   Product Name =                          MUX16
   Product Number =                        XXXX
   Product Grade =                         MIL
   EWS Library =                           Q5ØØØ
   EWS Library Rev =                       8Ø4
   Macro Parameter family =                Q5ØØØ
   The ARRAY type =                        Q5ØØØT
   Chip Macro Name =                       Q5ØØØTTTL1ØK
   Circuit Family =                        Q5ØØØ
   Circuit Technology =                    P
   ECL Level =                             1ØK


*************************************************************
*          SIGNAL I/O LIST                                  *
*          REVISION 1.Ø                                     *
*************************************************************


   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

AMCCERC.LST

FIGURE A2

```
**************************************************************
*          POPULATION ERC                                    *
*          VERSION  2.8Ø                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.

                      Circuit   Available
---------------------------------------
SUM_INPUT:             23.Ø              PADS
SUM_OUTPUT:             5.Ø              PADS
SUM_X_TTL_PWR:          3.Ø              PADS
SUM_X_TTL_GND:          3.Ø              PADS
SUM_X_ECL_GND:          3.Ø              PADS
---------------------------------------
SUM_TOTAL_I/O          37.Ø     16Ø.Ø    CELLS
SUM_INTERNAL            9.5     352.Ø    CELLS
---------------------------------------
TOTAL ARRAY PADS       61.Ø     184.Ø    PADS
---------------------------------------

Cell utilization is:                    2.7 %

Total fixed power pins:                  12
Total fixed ground pins:                 12
Total added power pins:                   6
Total added ground pins:                  3

Total input signals:                     23
Total output signals:                     5
Total bidirectional signals:              Ø
```

FIGURE A2 CONTINUED

```
**********************************************************
*          I/O Statistics ERC                            *
*          VERSION  1.3Ø                                 *
**********************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL
```

	number of Objects in the circuit	Object type
Inputs		
TTL	2Ø	PAD
ECL	2	PAD
SubTotal	22	PAD
Outputs		
ECL1ØK	2	PAD
Thermal Diode Connections	4	PAD
SubTotal	6	PAD

```
**********************************************************
*          VALID NAME CHECK ERC                          *
*          VERSION  2.7Ø                                 *
**********************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.

**********************************************************
*          PIN CLASS ERC                                 *
*          VERSION  2.7Ø                                 *
**********************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

FIGURE A2 CONTINUED

```
**************************************************************
*                                                            *
*          FAN-OUT CHECK ERC                                 *
*          VERSION  2.8Ø                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

FAN-OUT LOADING TABLE FORMAT:

COMPONENT	MACRO NAME	SIGNAL	LIMIT	LOAD	% DERATING	PAGE
GØØØØ	GT99	3S28	9	2		4
GØØØØ	GT99	3S29	9	1		4
G4ØØØØ	GT6ØL	3S3Ø	4	1		4
G4ØØØ1	GT6ØL	3S24	4	1		4
EIØØØØ	IE86	3S36	1	1		4
G4ØØØ2	GT6ØL	3S37	4	1		3
SØØ01	IT11H	INPØ	9	1		3
SØØØ2	IT11H	INP1	9	1		3
SØØ11	IT11H	INP1Ø	9	1		3
SØØ12	IT11H	INP11	9	1		3
SØØ13	IT11H	INP12	9	1		3
SØØ14	IT11H	INP13	9	1		3
SØØ15	IT11H	INP14	9	1		3
SØØ16	IT11H	INP15	9	1		3
SØØØ3	IT11H	INP2	9	1		3
SØØØ4	IT11H	INP3	9	1		3
SØØØ5	IT11H	INP4	9	1		3
SØØØ6	IT11H	INP5	9	1		3
SØØØ7	IT11H	INP6	9	1		3
SØØØ8	IT11H	INP7	9	1		3
SØØØ9	IT11H	INP8	9	1		3
SØØ1Ø	IT11H	INP9	9	1		3
SØØ21	IE93	INTCLK	7	1	2Ø	2
FFØØØØ	FF13H	INTQN	9	1		2
SØØ22	IE93	INTRST	9	1		2
MXØØØØ	MX25H	INTYØ	9	1		2
MXØØØ1	MX25H	INTY1	9	1		2
MXØØØ2	MX25H	INTY2	9	1		2
MXØØØ3	MX25H	INTY3	9	1		2
MXØØØ4	MX25H	INTY4	9	1		3
SØØØ1	IT11H	PARA1	9	1		3
SØØ1Ø	IT11H	PARA1Ø	9	1		3
SØØ11	IT11H	PARA11	9	1		3
SØØ12	IT11H	PARA12	9	1		3
SØØ13	IT11H	PARA13	9	1		3
SØØ14	IT11H	PARA14	9	1		3
SØØ15	IT11H	PARA15	9	1		3
SØØ16	IT11H	PARA16	9	1		3
SØØ17	IT11H	PARA17	9	1		3
SØØ18	IT11H	PARA18	9	1		3
SØØ19	IT11H	PARA19	9	1		3
SØØØ2	IT11H	PARA2	9	1		3
SØØ2Ø	IT11H	PARA2Ø	9	1		3
SØØØ3	IT11H	PARA3	9	1		3
SØØØ4	IT11H	PARA4	9	1		3
SØØØ5	IT11H	PARA5	9	1		3
SØØØ6	IT11H	PARA6	9	1		3
SØØØ7	IT11H	PARA7	9	1		3
SØØØ8	IT11H	PARA8	9	1		3
SØØØ9	IT11H	PARA9	9	1		2
SØØ21	IE93	PARCLK	9	1		2
SØØ22	IE93	PARRST	9	1		3
SØØ2Ø	IT11H	SELØ	9	4		3
SØØ19	IT11H	SEL1	9	4		3
SØØ18	IT11H	SEL2	9	1		3
SØØ17	IT11H	SEL3	9	1		3

FIGURE A2 CONTINUED

```
*************************************************************
*          INTERNAL PIN COUNT ERC                           *
*          VERSION  2.8Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


                CIRCUIT    AVAILABLE
-----------------------------------------
SUM_INT_PINS:      119         3572
-----------------------------------------

Internal pin count is within bounds.
This array is routable.


   No errors found.



*************************************************************
*          PIN HOOKUP & UNUSED PINS CHECK ERC               *
*          VERSION  2.8Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

FIGURE A2 CONTINUED

```
**************************************************************
*          MACRO OCCURRENCE AND POWER DISSIPATION            *
*          VERSION  2.90                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5000
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


        WORST-CASE INTERNAL CURRENT:   19.85 mA
        MAXIMUM INTERNAL CURRENT SPECIFICATION IS: 1383.00 mA


   No errors found.
```

MACRO NAME	# USED	SPECS ICC mA	SPECS IEE mA	TOTALS ICC mA	TOTALS IEE mA
FF13H	1	2.75	0.00	2.75	0.00
GT60L	3	1.26	0.00	3.78	0.00
GT99	1	0.00	0.00	0.00	0.00
IE96	1	0.00	0.00	0.00	0.00
IE93	2	1.35	0.00	2.70	0.00
IEVCC	3	0.00	0.00	0.00	0.00
IT11H	20	1.35	0.00	27.00	0.00
ITGND	3	0.00	0.00	0.00	0.00
ITPWR	3	0.00	0.00	0.00	0.00
MX25H	5	2.07	0.00	10.35	0.00
OE80	1	5.58	0.00	5.58	0.00
OE81	1	5.58	0.00	5.58	0.00
OE87	3	0.00	0.00	0.00	0.00

	ICC mA	IEE mA
TOTAL TYP MACRO CURRENT mA	57.74	0.00
TOTAL TYPICAL POWERED DOWN CURRENT mA	4.68	0.00
TOTAL TYP OVERHEAD CURRENT mA	309.00	0.00
TOTAL TYP CURRENT mA	362.06	0.00
TOTAL MAX CURRENT mA (TYP CURRENT TIMES 1.40) =	506.88	0.00
WORST CASE POWER DISSIPATION VCC (5.5)V X (506.884)mA/1000 VEE (0)V X (0)mA/1000	2.79 WATTS 0.00 WATTS	
ECL OUTPUT POWER DISSIPATION (14.0)mA X 1.3V X (2)outputs/1000	0.04 WATTS	
TOTAL POWER DISSIPATION	2.82 WATTS	

FIGURE A2 CONTINUED

```
**************************************************************
*          CIRCUIT TECHNOLOGY ERC                            *
*          VERSION  2.8Ø                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.


**************************************************************
*          ERC report summary                                *
*          VERSION  3.4Ø                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


ERC NAME                              ! ERRORS ! WARNINGS
--------------------------------------------------------
Simultanously Switching Output          Ø          Ø
Population                              Ø          Ø
Valid Name Check                        Ø          Ø
Pin Class                               Ø          Ø
Fanout                                  Ø          Ø
Internal Pin Count                      Ø          Ø
Pin hookup and Unused pins              Ø          Ø
Bipolar Macro Occurrence and ...        Ø          Ø
Circuit Technolgy                       Ø          Ø
```

FIGURE A2 CONTINUED

FIGURE A3

AMCCIO.LST

```
*****************************************************
*          SIGNAL I/O LIST                          *
*          REVISION  3.10                           *
*****************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5000
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


*********************************
* SIGNAL I/O LIST               *
*********************************
```

ITEM #	SIGNAL NAME	INSTANCE NAME	MACRO NAME	LOGIC LEVEL	I/O TYPE	SWITCH GROUP	DIFF PAIRS	FAN-IN IIL(uA)	FAN-IN IIH(uA)
1	!DAT0	!S0001	!IT11H	TTL	I			0.00	0.00
2	!DAT1	!S0002	!IT11H	TTL	I			0.00	0.00
3	!DAT10	!S0011	!IT11H	TTL	I			0.00	0.00
4	!DAT11	!S0012	!IT11H	TTL	I			0.00	0.00
5	!DAT12	!S0013	!IT11H	TTL	I			0.00	0.00
6	!DAT13	!S0014	!IT11H	TTL	I			0.00	0.00
7	!DAT14	!S0015	!IT11H	TTL	I			0.00	0.00
8	!DAT15	!S0016	!IT11H	TTL	I			0.00	0.00
9	!DAT2	!S0003	!IT11H	TTL	I			0.00	0.00
10	!DAT3	!S0004	!IT11H	TTL	I			0.00	0.00
11	!DAT4	!S0005	!IT11H	TTL	I			0.00	0.00
12	!DAT5	!S0006	!IT11H	TTL	I			0.00	0.00
13	!DAT6	!S0007	!IT11H	TTL	I			0.00	0.00
14	!DAT7	!S0008	!IT11H	TTL	I			0.00	0.00
15	!DAT8	!S0009	!IT11H	TTL	I			0.00	0.00
16	!DAT9	!S0010	!IT11H	TTL	I			0.00	0.00
17	!EXTCLK	!S0021	!IE93	ECL	I			0.00	0.00
18	!EXTRST	!S0022	!IE93	ECL	I			0.00	0.00
19	!IN001	!E10000	!IE86	ECL	I			0.00	0.00
20	!OUT001	!E00000	!OE87	ECL10K	O				
21	!OUTC	!E0001	!OE87	ECL10K	O				
22	!OUTE	!E00002	!OE87	ECL10K	O				
23	!PARAM	!E04000	!OE81	ECL10K	O				
24	!SELCT0	!S0020	!IT11H	TTL	I			0.00	0.00
25	!SELCT1	!S0019	!IT11H	TTL	I			0.00	0.00

26	!SELCT2	!S#018	! TTL	! I				
27	!SELCT3	!S#017	! TTL	! I				
28	!YOUTPT	!OE8#	! ECL10K	! O	!SWGRPA		#.##	#.##
29	!ITGND	!GNT#1	! TTL	! G				
30	!ITGND	!GNT#2	! TTL	! G	!SWGRPA			
31	!ITGND	!GNT#3	! TTL	! G	!SWGRPA			
32	!ITPWR	!PW###	! TTL	! P	!A			
33	!ITPWR	!PW#01	! TTL	! P	!A			
34	!ITPWR	!PW#02	! TTL	! P	!SWGRPA			
35	!ITPWR	!PWE#04	! ECL	! P				
36	!ITPWR	!PWE#05	! ECL	! P	!SWGRPA			
37	!ITPWR	!PWE#06	! ECL	! P				
38	!ECLIOVCC!		! 5V	! P				
39	!ECLIOVCC!		! 5V	! P				
40	!ECLIOVCC!		! 5V	! P				
41	!ECLIOVCC!							
42	!ECLVCC		! 5V	! P				
43	!ECLVCC		! 5V	! P				
44	!ECLVCC		! 5V	! P				
45	!ECLVCC		! 5V	! P				
46	!ECLVEE		! #V	! G				
47	!ECLVEE		! #V	! G				
48	!ECLVEE		! #V	! G				
49	!ECLVEE		! #V	! G				
50	!TTLGND		! #V	! G				
51	!TTLGND		! #V	! G				
52	!TTLGND		! #V	! G				
53	!TTLGND		! #V	! G				
54	!TTLGND		! #V	! G				
55	!TTLGND		! #V	! G				
56	!TTLGND		! #V	! G				
57	!TTLGND		! #V	! G				
58	!TTLVCC		! 5V	! P				
59	!TTLVCC		! 5V	! P				
60	!TTLVCC		! 5V	! P				
61	!TTLVCC		! 5V	! P				

FIGURE A3 CONTINUED

FIGURE A3 CONTINUED

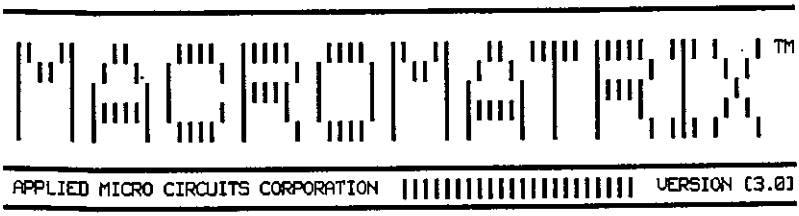
```
**********************************
* I/O LIST SUMMARY               *
**********************************


Total fixed power pins:                12
Total fixed ground pins:               12
Total added power pins:                 6
Total added ground pins:                3

Total input signals:                   23
Total output signals:                   5
Total bidirectional signals:            Ø


**********************************
* SWITCHING GROUPS               *
**********************************

                    !      !            ! ADDED POWER PINS     ! ADDED GROUND PINS
SWITCHING GROUP     !      !   TYPE     ! REQIRED              ! REQUIRED
NAME                ! SIZE ! (ECL/TTL) ! MIN--MAX / SUPPLIED  ! MIN--MAX / SUPPLIED
--------------------------------------------------------------------------------
A                     Ø       TTL          Ø -  Ø /    2          Ø -  Ø /    Ø
SWGRPA                Ø       TTL          Ø -  Ø /    1          Ø -  Ø /    1
SWGRPA                1       ECL          Ø -  Ø /    1
--------------------------------------------------------------------------------
TOTAL # SIMULTANEOUSLY SWITCHING TTL OUTPUTS:  Ø
TOTAL # SIMULTANEOUSLY SWITCHING ECL OUTPUTS:  1
-------------------------------------------------------
```

```
Date:  16 SEP 88 16:11   File:  ERC/AMCCXREF.LST


 Signal Cross-Reference

NETLIST   ! ORIGINAL                        !                      !
SIGNAL    ! SIGNAL                          ! SIGNAL               ! PAGE
NAME      ! NAME                            ! PATHNAME             ! NUMBER
--------------------------------------------------------------------------
3S28      !XSIG1                            !@MUX16                !4
3S29      !XSIG2                            !@MUX16                !4
3S3Ø      !XSIG27                           !@MUX16                !4
3S34      !XSIG42                           !@MUX16                !4
3S36      !XSIG5                            !@MUX16                !4
3S37      !XSIG54                           !@MUX16                !4


 PrePlacement Component Cross-Reference

                  !                ! ORIGINAL          ! NETLIST
COMPONENT         ! PAGE           ! COMPONENT         ! COMPONENT
PATHNAME          ! NUMBER         ! NAME              ! NAME
--------------------------------------------------------------------------


 Component Cross-Reference

NETLIST  ! ORIGINAL                 !                      !
COMPONENT! COMPONENT                ! COMPONENT            ! PAGE
NAME     ! NAME                     ! PATHNAME             ! NUMBER
--------------------------------------------------------------------------
```

AMCCXREF.LST

FIGURE A4

```
CLASS809> RUN_AMCC
```




```
        1: Part One:   Run DANCE, DRINK, AGIF and ERCs

        2: Part Two:   Run AMCCANN

        3: Part Three: Run SIFT and SOM_MAKER

        4: Part Four:  Run SOM and TCAL

        5: Part Five:  Run AMCCSIMFMT

        6: Part Six:   Run AMCCVRC

        0: Exit Program:

        Please Enter [0 - 6] : >2

        ********** RUNNING AMCCANN **********

        Need to Edit Package Pin Data? (Y or N) ; >Y

AMCC Delay Annotation VERSION  3.30
Loading Netlist ...
Welcome to the output loading system.

(0) Generate a report and exit.
(1) Change the package type.
(2) Edit the default package pin capacitance.
(3) Edit the default system capacitive load for TTL outputs.
(4) Edit the default system capacitive load for ECL outputs.
(5) Edit the system capacitive load for a specific pin or pins.
(6) Edit the package pin capacitance for a specific pin or pins.
(7) Edit the ECL Resistive Load for a specific pin or pins.
(8) Edit the Frequency for a specific pin or pins.
Enter the number of the item you wish to perform:
```

Running AMCCANN

FIGURE A5

```
Date:  16 SEP 88 16:11    File:  ERC/AMCCANN.LST


*************************************************************
*          AMCC Schematic Data Interface                    *
*          Revision [1,Ø]                                   *
*************************************************************

   Netlister version number =              7Ø8
   SDi version number =                    2.1Ø
   Netlist generation date =               19 SEP 1988
   Netlist generation time =               14:25
   Engineering workstation type =          DAISY/DNIX
   Engineering workstation path name =     /USER/CLASS/MUX16
   Product Name =                          MUX16
   Product Number =                        XXXX
   Product Grade =                         MIL
   EWS Library =                           Q5ØØØ
   EWS Library Rev =                       8Ø4
   Macro Parameter family =                Q5ØØØ
   The ARRAY type =                        Q5ØØØT
   Chip Macro Name =                       Q5ØØØTTTL1ØK
   Circuit Family =                        Q5ØØØ
   Circuit Technology =                    P
   ECL Level =                             1ØK



*************************************************************
*          AMCC Delay Annotation                            *
*          VERSION  3.3Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name PRODUCT_NAME
   Circuit family Q5ØØØ
   Circuit technology P
   Date 16 SEP 1988
   Time 12:43
   Product Grade PRODUCT_GRADE


   MIN front annotation file created for DAISY/DNIX
   NOM front annotation file created for DAISY/DNIX
   MIL front annotation file created for DAISY/DNIX

   No errors found.
```

Discard This File

FIGURE A6

```
Date:  14 SEP 88 14:09   File:  ERC/OUTPUT.DLY

pkgkey     1 systtl_cap      15.00 sysecl_cap      10.00 pkgmin_cap      4.00 pkgtyp_cap      5.30 pkgmax_cap      6.20
EXTCLK          0.00
OUT001         10.00          0.00          0.00          0.00          100.00
OUTC           10.00          4.00          5.30          6.20
OUTE           10.00          4.00          5.30          6.20
PARAM          10.00          4.00          5.30          6.20
YOUTPT         10.00          4.00          5.30          6.20
                              12.00         12.00         12.00          0.00          55.00

Date:  16 SEP 88 16:14   File:  ERC/OUTPUT.DLY  * Used in execution

pkgkey     3 systtl_cap      15.00 sysecl_cap       5.00 pkgmin_cap      3.80 pkgtyp_cap      4.40 pkgmax_cap      4.90
EXTCLK          0.00          0.00          0.00          0.00          100.00
OUT001          5.00          3.80          4.40          4.90
OUTC            5.00          3.80          4.40          4.90
OUTE            5.00          3.80          4.40          4.90
PARAM           5.00          3.80          4.40          4.90
YOUTPT          5.00         12.00         12.00         12.00          0.00          55.00

Date:  27 AUG 86 13:20   File:  ERC/OUTPUT.DLY

pkgkey    11 systtl_cap      15.00 sysecl_cap       5.00 pkgmin_cap      2.40 pkgtyp_cap      3.00 pkgmax_cap      3.60
DAT0            0.00          0.00          0.00          0.00           80.00
EXTCLK          0.00          0.00          0.00          0.00          100.00
EXTRST          0.00          0.00          0.00          0.00           30.00
OUT001          5.00          2.40          3.00          3.60
OUTC            5.00          2.40          3.00          3.60
OUTE            5.00          2.40          3.00          3.60
PARAM          50.00          2.40          3.00          3.60
YOUTPT         10.00          2.40          3.00          3.60
```

Sample OUTPUT.DLY (data file)

FIGURE A7

AMCCPKG.LST

FIGURE A8

Date: 19 SEP 88 14:32 File: ERC/AMCCPKG.LST

```
*************************************************************
*          AMCC Output Loading System                       *
*          Version  1.00                                    *
*************************************************************
```

Netlister version number =	700
SDI version number =	2.10
Netlist generation date =	19 SEP 1988
Netlist generation time =	14:25
Engineering workstation type =	DAISY/DNIX
Engineering workstation path name =	/USER/CLASS/MUX16
Product Name =	MUX16
Product Number =	XXXX
Product Grade =	NIL
EWS Library =	Q5000
EWS Library Rev =	804
Macro Parameter family =	Q5000
The ARRAY type =	Q5000T
Chip Macro Name =	Q5000TTTL10K
Circuit Family =	Q5000
Circuit Technology =	P
ECL Level =	10K

Item #	Signal Name	Instance Name	Macro Name	PAD	Frq MHz	ECL R	System pf	Package Capacitance Min pf	Typ pf	Max pf
1	DAT0	S0001	IT11H							
2	DAT1	S0002	IT11H							
3	DAT10	S0011	IT11H							
4	DAT11	S0012	IT11H							
5	DAT12	S0013	IT11H							
6	DAT13	S0014	IT11H							
7	DAT14	S0015	IT11H							
8	DAT15	S0016	IT11H							
9	DAT2	S0003	IT11H							
10	DAT3	S0004	IT11H							
11	DAT4	S0005	IT11H							
12	DAT5	S0006	IT11H							
13	DAT6	S0007	IT11H							
14	DAT7	S0008	IT11H							
15	DAT8	S0009	IT11H							
16	DAT9	S0010	IT11H							
17	EXTCLK	S0021	IE93		100.0					
18	EXTRST	S0022	IE93							
19	INA01	EI0000	IE86							
20	OUT001	EO0000	OE87				5.0	3.60	4.40	4.90
21	OUTC	EO001	OE87				5.0	3.60	4.40	4.90
22	OUTE	EO0002	OE87				5.0	3.60	4.40	4.90
23	PARAM	EO4000	OE81				5.0	3.60	4.40	4.90
24	SELCT0	S0020	IT11H							
25	SELCT1	S0019	IT11H							

26	SELCT2	SØØ18	IT11H						
27	SELCT3	SØØ17	IT11H						
28	YOUTPT	DØØØØ	OE8Ø		55.Ø	5.Ø	12.ØØ	12.ØØ	12.ØØ
29	ITGND	GNTØ3	ITGND						
3Ø	ITGND	GNTØ2	ITGND						
31	ITGND	GNTØ1	ITGND						
32	ITPWR	PWØØ2	ITPWR						
33	ITPWR	PWØØ1	ITPWR						
34	ITPWR	PWØØØ	ITPWR						
35	IOECLVCC	PWEØØ6	IEVCC						
36	IOECLVCC	PWEØØ5	IEVCC						
37	IOECLVCC	PWEØØ4	IEVCC						
38	ECLIOVCC			139					
39	ECLIOVCC			2					
4Ø	ECLIOVCC			47					
41	ECLIOVCC			94					
42	ECLVCC			152					
43	ECLVCC			173					
44	ECLVCC			6Ø					
45	ECLVCC			81					
46	ECLVEE			153					
47	ECLVEE			172					
48	ECLVEE			61					
49	ECLVEE			8Ø					
5Ø	TTLGND			1					
51	TTLGND			14Ø					
52	TTLGND			141					
53	TTLGND			184					
54	TTLGND			48					
55	TTLGND			49					
56	TTLGND			92					
57	TTLGND			93					
58	TTLVCC			142					
59	TTLVCC			183					
6Ø	TTLVCC			5Ø					
61	TTLVCC			91					

FIGURE A8 CONTINUED

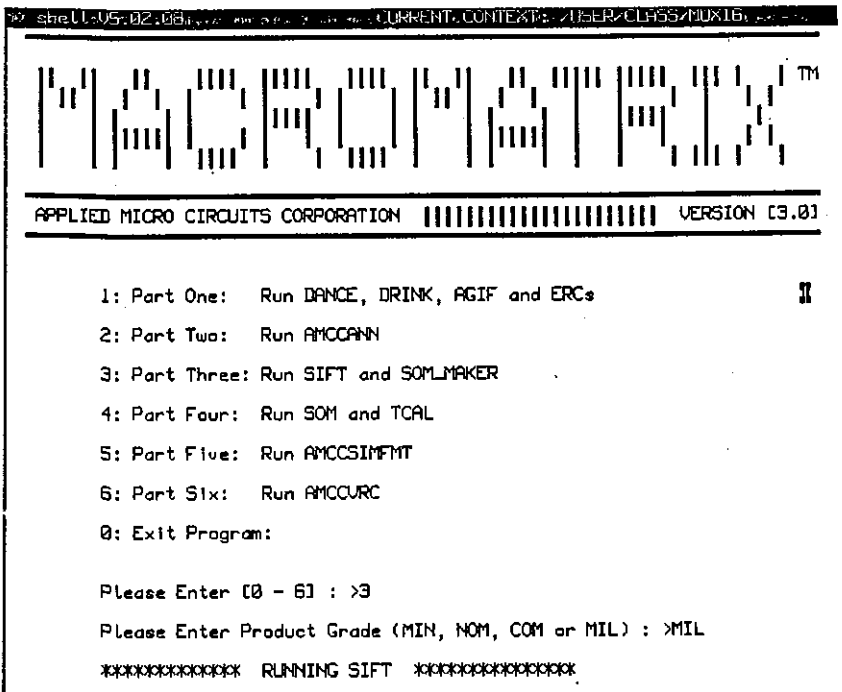
```
Date:  19 SEP 88 14:34    File:  FNTMIL.DSY

/* Daisy design pathname:  /USER/CLASS/MUX16  */
$CONTROL
MODE ADD
SECTION DELAY:N:6
$DELAY
/*                                     R I S E           F A L L         */
/*                                 Typ   Min   Max   Typ   Min   Max     */
@MUX16/4:XSIG1          =    Ø     Ø     Ø     Ø     Ø     Ø   ;
@MUX16/4:XSIG2          =    Ø     Ø     Ø     Ø     Ø     Ø   ;
@MUX16/4:XSIG27         =   26    23    28    51    46    56   ;
@MUX16/4:XSIG42         =   26    23    28    51    46    56   ;
@MUX16/4:XSIG5          =    Ø     Ø     Ø     Ø     Ø     Ø   ;
@MUX16/4:XSIG54         =   26    23    28    51    46    56   ;
@MUX16/2:INPØ           =   13    12    14    26    23    28   ;
@MUX16/2:INP1           =   13    12    14    26    23    28   ;
@MUX16/2:INP1Ø          =   13    12    14    26    23    28   ;
@MUX16/2:INP11          =   13    12    14    26    23    28   ;
@MUX16/2:INP12          =   13    12    14    26    23    28   ;
@MUX16/2:INP13          =   13    12    14    26    23    28   ;
@MUX16/2:INP14          =   13    12    14    26    23    28   ;
@MUX16/2:INP15          =   13    12    14    26    23    28   ;
@MUX16/2:INP2           =   13    12    14    26    23    28   ;
@MUX16/2:INP3           =   13    12    14    26    23    28   ;
@MUX16/2:INP4           =   13    12    14    26    23    28   ;
@MUX16/2:INP5           =   13    12    14    26    23    28   ;
@MUX16/2:INP6           =   13    12    14    26    23    28   ;
@MUX16/2:INP7           =   13    12    14    26    23    28   ;
@MUX16/2:INP8           =   13    12    14    26    23    28   ;
@MUX16/2:INP9           =   13    12    14    26    23    28   ;
@MUX16/2:INTCLK         =   26    23    28    26    23    28   ;
@MUX16/2:INTQN          =   13    12    14    26    23    28   ;
@MUX16/2:INTRST         =   26    23    28    26    23    28   ;
@MUX16/2:INTYØ          =   13    12    14    26    23    28   ;
@MUX16/2:INTY1          =   13    12    14    26    23    28   ;
@MUX16/2:INTY2          =   13    12    14    26    23    28   ;
@MUX16/2:INTY3          =   13    12    14    26    23    28   ;
@MUX16/2:INTY4          =   13    12    14    26    23    26   ;
@MUX16/3:PARA1          =   13    12    14    26    23    28   ;
@MUX16/3:PARA1Ø         =   13    12    14    26    23    28   ;
@MUX16/3:PARA11         =   13    12    14    26    23    28   ;
@MUX16/3:PARA12         =   13    12    14    26    23    28   ;
@MUX16/3:PARA13         =   13    12    14    26    23    28   ;
@MUX16/3:PARA14         =   13    12    14    26    23    28   ;
@MUX16/3:PARA15         =   13    12    14    26    23    28   ;
@MUX16/3:PARA16         =   13    12    14    26    23    28   ;
@MUX16/3:PARA17         =   13    12    14    26    23    28   ;
@MUX16/3:PARA18         =   13    12    14    26    23    28   ;
@MUX16/3:PARA19         =   13    12    14    26    23    28   ;
@MUX16/3:PARA2          =   13    12    14    26    23    28   ;
@MUX16/3:PARA2Ø         =   13    12    14    26    23    28   ;
@MUX16/3:PARA3          =   13    12    14    26    23    28   ;
@MUX16/3:PARA4          =   13    12    14    26    23    28   ;
@MUX16/3:PARA5          =   13    12    14    26    23    28   ;
@MUX16/3:PARA6          =   13    12    14    26    23    28   ;
@MUX16/3:PARA7          =   13    12    14    26    23    28   ;
@MUX16/3:PARA8          =   13    12    14    26    23    28   ;
@MUX16/3:PARA9          =   13    12    14    26    23    28   ;
@MUX16/2:PARCLK         =   26    23    28    26    23    28   ;
@MUX16/2:PARRST         =   26    23    28    26    23    28   ;
@MUX16/2:SELØ           =   36    33    4Ø    72    65    8Ø   ;
@MUX16/2:SEL1           =   36    33    4Ø    72    65    8Ø   ;
@MUX16/2:SEL2           =   13    12    14    26    23    28   ;
@MUX16/2:SEL3           =   13    12    14    26    23    28   ;
```

OUTPUT LOAD DELAYS

```
@MUX16/4:OUTØØ1         =   56    47    65    46    39    53   ;
@MUX16/4:OUTC           =   56    47    65    46    39    53   ;
@MUX16/4:OUTE           =   56    47    65    46    39    53   ;
@MUX16/4:PARAM          =   56    47    65    46    39    53   ;
@MUX16/2:YOUTPT         =  1Ø1    91   111    83    75    91   ;
$END
```

Front-Annotation

FIGURE A9



SIFT and SOM_MAKER

FIGURE A10

APNOTE 1.DNIX A25

```
CLASS809> RUN_AMCC
```


```
APPLIED MICRO CIRCUITS CORPORATION                    VERSION [3.0]

      1: Part One:   Run DANCE, DRINK, AGIF and ERCs
      2: Part Two:   Run AMCCANN
      3: Part Three: Run SIFT and SOM_MAKER
      4: Part Four:  Run SOM and TCAL
      5: Part Five:  Run AMCCSIMFMT
      6: Part Six:   Run AMCCVRC
      0: Exit Program:

      Please Enter [0 - 6] : >4
      Please Enter SOM Master Control File : >FUNCTION.SING
      Please Enter TCAL Delay File : >FNTMIL.DSY
```

SOM and TCAL
Link For Simulation

FIGURE A11

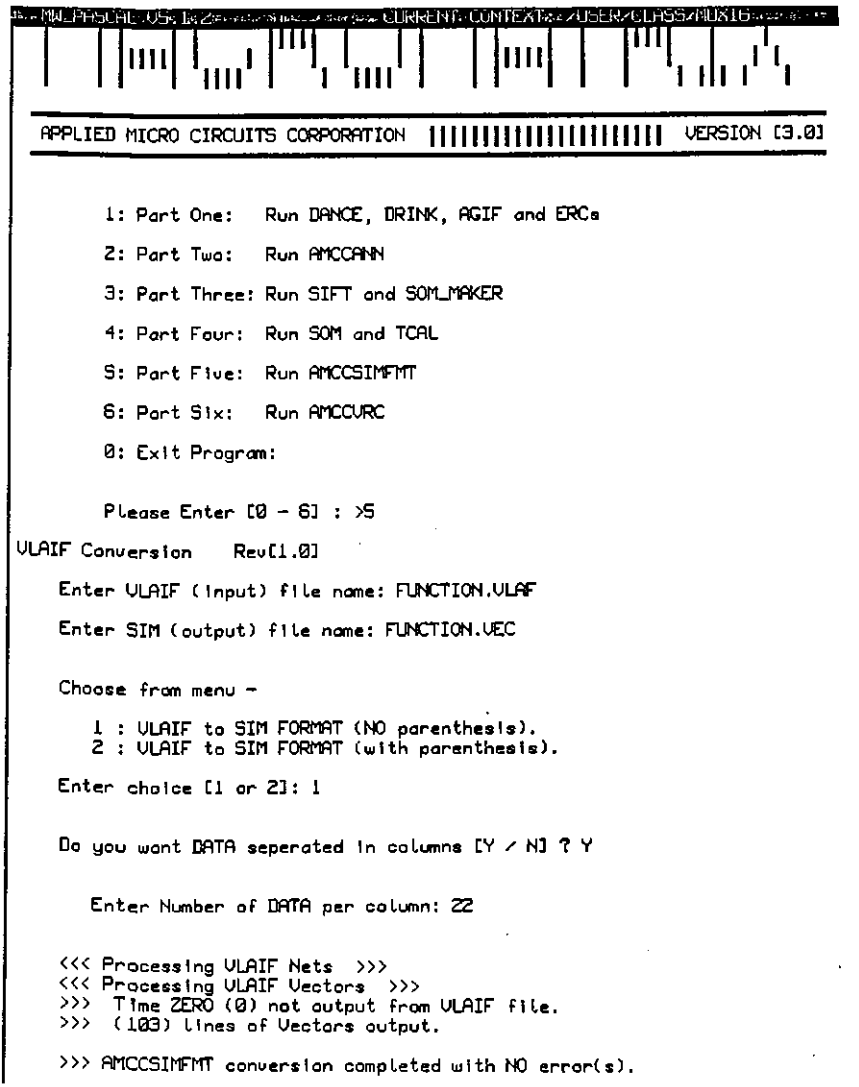
```
Date:  19 SEP 88 16:46   File:  FUNCTION.SING

/****
*
* DESIGN PATH /USER/CLASS/MUX16   DATE 24 MAY 1988 13:30
*
* COMPANY  ___AMCC__________   CIRCUIT_NAME  __16:1 MUX_
*
* ARRAY    _Q5000 SERIES PO#  ___-_______   REV  ______
*
* DESIGNER _____DEW_________________
*
* What tests does this control file support:  _________
*
* __FUNCTIONAL TESTS - ALL_____________________________
*
* __100 % FAULT GRADE COVERAGE VIA MINIMAL TEST SEQUENCE
*
****/

/***   Configuration section    ****/

$CONFIGURATION

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

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

$SIGNAL_GENERATORS

@MUX16/3:DAT0  :=  00:F1,   @40000:F0,   @60000:F1;
@MUX16/3:DAT1  :=  00:F0,  @460000:F1,  @480000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT2  :=  00:F0,  @940000:F1,  @960000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT3  :=  00:F1,  @520000:F0,  @540000:F1;
@MUX16/3:DAT4  :=  00:F0,  @100000:F1,  @120000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT5  :=  00:F1,  @400000:F0,  @420000:F1;
@MUX16/3:DAT6  :=  00:F1,  @880000:F0,  @900000:F1;
@MUX16/3:DAT7  :=  00:F0,  @580000:F1,  @600000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT8  :=  00:F0,  @220000:F1,  @240000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT9  :=  00:F1,  @280000:F0,  @300000:F1;
@MUX16/3:DAT10 := 00:F1,  @760000:F0,  @780000:F1;
@MUX16/3:DAT11 := 00:F0,  @700000:F1,  @720000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT12 := 00:F1,  @160000:F0,  @180000:F1;
@MUX16/3:DAT13 := 00:F0,  @340000:F1,  @360000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT14 := 00:F0,  @820000:F1,  @840000:F0, @1000000:F1, @1020000:F0;
@MUX16/3:DAT15 := 00:F1,  @640000:F0,  @660000:F1;

@MUX16/2:EXTCLK := 00:F0, [10000:F0, 10000:F1]**;

@MUX16/2:EXTRST := 00:F1 , @20000:F0, @1000000:F1, @1020000:F0;

@MUX16/3:SELCT0 := 00:F0, @260000:F1,  @740000:F0, @1000000:F1, @1020000:F0;

@MUX16/3:SELCT1 := 00:F0, @500000:F1,  @980000:F0, @1000000:F1, @1020000:F0;

@MUX16/3:SELCT2 := 00:F0,  @80000:F1,  @200000:F0,
                           @320000:F1,  @440000:F0,
                           @560000:F1,  @680000:F0,
                           @800000:F1,  @920000:F0, @1000000:F1, @1020000:F0;

@MUX16/3:SELCT3 := 00:F0, @140000:F1,  @380000:F0,
                           @620000:F1,  @860000:F0, @1000000:F1, @1020000:F0;
```

FUNCTION.SING: SOM_MCF.SING edited for functional simulation

FIGURE A12

```
/*  SET THERMAL DIODE INPUT STEADY */
@MUX16/4:IN@@1 := @@:F1;

/* ========================================================== */

$OUTPUTS

FILE /USER/CLASS/MUX16/FUNCTION.VLAF <-

/* ---------------------------------------------------- */
/* LIST ALL PRIMARY INPUTS; EXCLUDE THERMAL DIODES,     */
/* MONITOR POINTS AND V8xx MACRO SIGNALS                */
/* ---------------------------------------------------- */

@MUX16/2:EXTCLK, EXTRST,
@MUX16/3:SELCT3, SELCT2, SELCT1, SELCT@,
@MUX16/3:DAT@, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7,
@MUX16/3:DAT8, DAT9, DAT1@, DAT11, DAT12, DAT13, DAT14, DAT15,

/* ---------------------------------------------------- */
/* LIST ALL PRIMARY OUTPUTS; EXCLUDE THERMAL DIODES,    */
/* MONITOR POINTS AND V8xx MACRO SIGNALS                */
/* ---------------------------------------------------- */

@MUX16/2:YOUTPT,
@MUX16/4:PARAM;

/* ---------------------------------------------------- */
/* LIST INTERNAL 3-STATE ENABLES, BIDIRECTIONAL         */
/*      ENABLES HERE                                    */
/*   IF ANY                                             */
/* ---------------------------------------------------- */

$END
```

FIGURE A12 CONTINUED



AMCCSIMFMT

FIGURE A13

APNOTE 1.DNIX A29

```
Date:  27 AUG 86 12:45    File:  SIMFMTFN.Ø1

1***CIRCUIT IDENTIFICATION =

              EESSSSDDDDDDDDDDDDDDD YP
              XXEEEEAAAAAAAAAAAAAAA OA
              TTLLLLTTTTTTTTTTTTTTT UR
              CRCCCCØ123456789111111 TA
              LSTTTT          Ø12345 PM
              KT321Ø                 T

        TIME

9999          ØlØØØØlØØlØlllØØllØlØØl Ø1
19999         llØØØØlØØlØlllØØllØlØØl Ø1
29999         ØØØØØØlØØlØlllØØllØlØØl Ø1
39999         lØØØØØlØØlØlllØØllØlØØl 11
49999         ØØØØØØØØØlØlllØØllØlØØl 11
59999         lØØØØØØØØlØlllØØllØlØØl Ø1
69999         ØØØØØØlØØlØlllØØllØlØØl Ø1
79999         lØØØØØlØØlØlllØØllØlØØl 11
89999         ØØØlØØlØØlØlllØØllØlØØl 11
99999         lØØlØØlØØlØlllØØllØlØØl Ø1
lØ9999        ØØØlØØlØØlllllØØllØlØØl Ø1
119999        lØØlØØlØØllllØØllØlØØl 11
129999        ØØØlØØlØØlØlllØØllØlØØl 11
139999        lØØlØØlØØlØlllØØllØlØØl Ø1
149999        ØØllØØlØØlØlllØØllØlØØl Ø1
159999        lØllØØlØØlØlllØØllØlØØl 11
169999        ØØllØØlØØlØlllØØllØØØØl 11
179999        lØllØØlØØlØlllØØllØØØØl Ø1
189999        ØØllØØlØØlØlllØØllØlØØl Ø1
199999        lØllØØlØØlØlllØØllØlØØl 11
2Ø9999        ØØlØØØlØØlØlllØØllØlØØl 11
219999        lØlØØØlØØlØlllØØllØlØØl Ø1
229999        ØØlØØØlØØlØllØlllØlØØl Ø1
239999        lØlØØØlØØlØllØlllØlØØl 11
249999        ØØlØØØlØØlØlllØØllØlØØl 11
259999        lØlØØØlØØlØlllØØllØlØØl Ø1
269999        ØØlØØllØØlØlllØØllØlØØl Ø1
279999        lØlØØllØØlØlllØØllØlØØl 11
289999        ØØlØØllØØlØllØØØlØlØØl 11
299999        lØlØØllØØlØllØØØlØlØØl Ø1
3Ø9999        ØØlØØllØØlØlllØØllØlØØl Ø1
319999        lØlØØllØØlØlllØØllØlØØl 11
329999        ØØllØllØØlØlllØØllØlØØl 11
339999        lØllØllØØlØlllØØllØlØØl Ø1
349999        ØØllØllØØlØlllØØllØllØl Ø1
359999        lØllØllØØlØlllØØllØllØl 11
369999        ØØllØllØØlØlllØØllØlØØl 11
379999        lØllØllØØlØlllØØllØlØØl Ø1
389999        ØØØlØllØØlØlllØØllØlØØl Ø1
399999        lØØlØllØØlØlllØØllØlØØl 11
4Ø9999        ØØØlØllØØlØØlØØllØlØØl 11
419999        lØØlØllØØlØØlØØllØlØØl Ø1
429999        ØØØlØllØØlØlllØØllØlØØl Ø1
439999        lØØlØllØØlØlllØØllØlØØl 11
449999        ØØØØØllØØlØlllØØllØlØØl 11
459999        lØØØØllØØlØlllØØllØlØØl Ø1
469999        ØØØØØlllØlØlllØØllØlØØl Ø1
479999        lØØØØlllØlØllØØllØlØØl 11
489999        ØØØØØllØØlØlllØØllØlØØl 11
499999        lØØØØllØØlØlllØØllØlØØl Ø1
5Ø9999        ØØØØllØØlØllØØllØlØØl Ø1
519999        lØØØllØØlØllØØllØlØØl 11
```

FUNCTION.VEC

FIGURE A14

```
529999      0000111000011001101001 11
539999      1000111000011001101001 01
549999      0000111001011001101001 01
559999      1000111001011001101001 11
569999      0001111001011001101001 11
579999      1001111001011001101001 01
589999      0001111001011101101001 01
599999      1001111001011101101001 11
609999      0001111001011001101001 11
619999      1001111001011001101001 01
629999      0011111001011001101001 01
639999      1011111001011001101001 11
649999      0011111001011001101000 11
659999      1011111001011001101000 01
669999      0011111001011001101001 01
679999      1011111001011001101001 11
689999      0010111001011001101001 11
699999      1010111001011001101001 01
709999      0010111001011001111001 01
719999      1010111001011001111001 11
729999      0010111001011001101001 11
739999      1010111001011001101001 01
749999      0010101001011001101001 01
759999      1010101001011001101001 11
769999      0010101001011001001001 11
779999      1010101001011001001001 01
789999      0010101001011001101001 01
799999      1010101001011001101001 11
809999      0011101001011001101001 11
819999      1011101001011001101001 01
829999      0011101001011001101011 01
839999      1011101001011001101011 11
849999      0011101001011001101001 11
859999      1011101001011001101001 01
869999      0001101001011001101001 01
879999      1001101001011001101001 11
889999      0001101001010001101001 11
899999      1001101001010001101001 01
909999      0001101001011001101001 01
919999      1001101001011001101001 11
929999      0000101001011001101001 11
939999      1000101001011001101001 01
949999      0000101011011001101001 01
959999      1000101011011001101001 11
969999      0000101001011001101001 11
979999      1000101001011001101001 01
989999      0000000100101100110100 01
999999      1000000100101100110100 11
1009999     01111111111111111111111 01
1019999     11111111111111111111111 00
1029999     0000000100101100110100 01
```

FIGURE A14 CONTINUED

```
CLASS809> RUN_AMCC

MACROMATRIX TM

APPLIED MICRO CIRCUITS CORPORATION                     VERSION [3.0]

        1: Part One:    Run DANCE, DRINK, AGIF and ERCs

        2: Part Two:    Run AMCCANN

        3: Part Three: Run SIFT and SOM_MAKER

        4: Part Four:   Run SOM and TCAL

        5: Part Five:   Run AMCCSIMFMT

        6: Part Six:    Run AMCCVRC

        0: Exit Program:

        Please Enter [0 - 6] : >6
AMCCVRC VERSION  3.00

 Enter the SDI file name (input) >  ERC/CIRCUIT.SDI

 Enter the AMCCSIM file name (input) > FUNCTION.VEC



   What type of check do you want to do?
     0) Quit
     1) Race conditions
     2) Simultaneously Switching Outputs
     3) Unknown signals
     4) Signal Toggle check
     5) Required Signal check
     6) Differential Input check
     7) Vector Length check
     9) All but Race (Tests 2 to 7)
    10) All the above (Tests 1 to 7)
    11) Change error limit
   Enter choice (0..11) >10

 Enter the signal analysis file (input)>SIGNAL.DAT


 >>> Creating working files


 ----------- Race CHECK -----------

 >>> phase 1 - phase 2 - phase 3 - phase 4 - phase 5 - phase 6
```

FIGURE A15 **AMCCVRC**

```
Enter choice (0..11) >10

Enter the signal analysis file (Input)>SIGNAL.DAT

>>> Creating working files

---------- Race CHECK ----------

>>> phase 1 - phase 2 - phase 3 - phase 4 - phase 5 - phase 6

---------- SSO CHECK ----------

>>> phase 1 - phase 2 - phase 3 - phase 4

---------- Unknown signal CHECK ----------

>>> phase 1 - phase 2 - phase 3 - phase 4

---------- Signal Toggle Check ----------

>>> phase 1 - phase 2 - phase 3 - phase 4

---------- Required Signal Check ----------

>>> phase 1 - phase 2 - phase 3 - phase 4

---------- DIFFIP CHECK ----------

>>> phase 1 - phase 2 - phase 3 - phase 4

---------- Vector Length Check ----------

>>> phase 1

   What type of check do you want to do?
     0) Quit
     1) Race conditions
     2) Simultaneously Switching Outputs
     3) Unknown signals
     4) Signal Toggle check
     5) Required Signal check
     6) Differential Input check
     7) Vector Length check
     9) All but Race (Tests 2 to 7)
    10) All the above (Tests 1 to 7)
    11) Change error limit
   Enter choice (0..11) >0
```

FIGURE A15 CONTINUED

SIGNAL ANALYSIS FILE

```
        Date:  19 SEP 88 14:31    File:  SIGNAL.DAT
        EXTCLK + EXCEPT!
```

```
Date:  19 SEP 88 10:13    File:  AMCCVRC.LST


*****************************************************************
*          Race Condition Checker                               *
*          VERSION  2.90                                        *
*****************************************************************


   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5000
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.



*****************************************************************
*          Simultaneously Switching Output Checker              *
*          VERSION  2.90                                        *
*****************************************************************


   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5000
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.



*****************************************************************
*          Unknown Signal State Checker                         *
*          VERSION  2.90                                        *
*****************************************************************


   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5000
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

AMCCVRC.LST

FIGURE A16

APNOTE 1.DNIX A34

```
*************************************************************
*          Differential Input Checker                       *
*          VERSION  2.9Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


 The following signals are differential:


   No errors found.


*************************************************************
*          REQUIRED SIGNAL CHECK                            *
*          VERSION  2.9Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL




   No errors found.


*************************************************************
*          Signal Toggle Check                              *
*          VERSION  2.9Ø                                    *
*************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

FIGURE A16 CONTINUED

```
**************************************************************
*          Vector Length Check                               *
*          VERSION  3.ØØ                                     *
**************************************************************

   Path Name /USER/CLASS/MUX16
   Product Name MUX16
   Circuit family Q5ØØØ
   Circuit technology P
   Date 19 SEP 1988
   Time 14:25
   Product Grade MIL


   No errors found.
```

```
**************************************************************
*          VRC report summary                                *
*          VERSION  3.ØØ                                     *
**************************************************************

    Path Name /USER/CLASS/MUX16
    Product Name MUX16
    Circuit family Q5ØØØ
    Circuit technology P
    Date 19 SEP 1988
    Time 14:25
    Product Grade MIL


Test number  1  Had   Ø  Errors
Test number  2  Had   Ø  Errors
Test number  3  Had   Ø  Errors
Test number  4  Had   Ø  Errors
Test number  5  Had   Ø  Errors
Test number  6  Had   Ø  Errors
Test number  7  Had   Ø  Errors
```

FIGURE A16 CONTINUED

```
Date:  30 AUG 86 12:49   File:  SIMFMTAS.01

1***CIRCUIT IDENTIFICATION =

                EESSSSDDDDDDDDDDDDDDDD YP
                XXEEEEAAAAAAAAAAAAAAAA OA
                TTLLLLTTTTTTTTTTTTTTTT UR
                CRCCCC0123456789111111 TA
                LSTTTT           012345 PM
                KT3210                  T

        TIME

499             0100001001011001101001 01
999             1100001001011001101001 01
1499            0000001001011001101001 01
1999            1000001001011001101001 01
2499            0000000001011001101001 11
2999            1000000001011001101001 01
3499            0000001001011001101001 01
3999            1000001001011001101001 01
4499            0001001001011001101001 11
4999            1001001001011001101001 01
5499            0001001001111001101001 01
5999            1001001001111001101001 01
6499            0001001001011001101001 11
6999            1001001001011001101001 01
7499            0011001001011001101001 01
7999            1011001001011001101001 01
8499            0011001001011001100001 11
8999            1011001001011001100001 01
9499            0011001001011001101001 01
9999            1011001001011001101001 01
10499           0010001001011001101001 11
10999           1010001001011001101001 01
11499           0010001001011011101001 01
11999           1010001001011011101001 01
12499           0010001001011001101001 11
12999           1010001001011001101001 01
13499           0010011001011001101001 01
13999           1010011001011001101001 01
14499           0010011001011000101001 11
14999           1010011001011000101001 01
15499           0010011001011001101001 01
15999           1010011001011001101001 01
16499           0011011001011001101001 11
16999           1011011001011001101001 01
17499           0011011001011001101101 01
17999           1011011001011001101101 01
18499           0011011001011001101001 11
18999           1011011001011001101001 01
19499           0001011001011001101001 01
19999           1001011001011001101001 01
20499           0001011001001001101001 11
20999           1001011001001001101001 01
21499           0001011001011001101001 01
21999           1001011001011001101001 01
22499           0000011001011001101001 11
22999           1000011001011001101001 01
23499           0000011101011001101001 01
23999           1000011101011001101001 01
24499           0000011001011001101001 11
24999           1000011001011001101001 01
25499           0000111001011001101001 01
25999           1000111001011001101001 01
```

SAMPLED AT-SPEED

FIGURE A17

```
25499       0000111100001100110100l  11
25999       1000111100001100110100l  01
27499       0000111001101100110100l  01
27999       1000111001101100110100l  01
28499       0001111001101100110100l  11
28999       1001111001101100110100l  01
29499       0001111001101110110100l  01
29999       1001111001101110110100l  01
30499       0001111001101100110100l  11
30999       1001111001101100110100l  01
31499       0011111001101100110100l  01
31999       1011111001101100110100l  01
32499       0011111001101100110100   11
32999       1011111001101100110100   01
33499       0011111001101100110100l  01
33999       1011111001101100110100l  01
34499       0010111001101100110100l  11
34999       1010111001101100110100l  01
35499       0010111001101100111100l  01
35999       1010111001101100111100l  01
36499       0010111001101100110100l  11
36999       1010111001101100110100l  01
37499       0010101001101100110100l  01
37999       1010101001101100110100l  01
38499       0010101001101100100100l  11
38999       1010101001101100100100l  01
39499       0010101001101100110100l  01
39999       1010101001101100110100l  01
40499       0011101001101100110100l  11
40999       1011101001101100110100l  01
41499       0011101001101100110101   01
41999       1011101001101100110101   01
42499       0011101001101100110100l  11
42999       1011101001101100110100l  01
43499       0001101001101100110100l  01
43999       1001101001101100110100l  01
44499       0001101001010001101001   11
44999       1001101001010001101001   01
45499       0001101001101100110100l  01
45999       1001101001101100110100l  01
46499       0000101001101100110100l  11
46999       1000101001101100110100l  01
47499       0000101011011100110100l  01
47999       1000101011011100110100l  01
48499       0000101001101100110100l  11
48999       1000101001101100110100l  01
49499       0000001001101100110100l  01
49999       1000001001101100110100l  01
```

FIGURE A17 CONTINUED

```
Date:  19 SEP 88 15:48    File:  ATSPEED_POC.SING

/****
*
* DESIGN PATH /USER/CLASS/MUX16   DATE 12 FEB 1986 13:30
*
* COMPANY  ___AMCC__________    CIRCUIT_NAME  __16:1 MUX_
*
* ARRAY    _Q5000 SERIES PO#  ___-_______    REV  ______
*
* DESIGNER _____DEV_____________________
*
* What tests does this control file support:  _________
*
* __ATSPEED    TESTS - ALL PATHS _____________________
*      70/30 CLOCK
****/

/***   Configuration section    ****/

$CONFIGURATION

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

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

$SIGNAL_GENERATORS

@MUX16/3:DAT0 :=  @0:F1,   @2000:F0,   @3000:F1;
@MUX16/3:DAT1 :=  @0:F0,  @23000:F1,  @24000:F0;
@MUX16/3:DAT2 :=  @0:F0,  @47000:F1,  @48000:F0;
@MUX16/3:DAT3 :=  @0:F1,  @26000:F0,  @27000:F1;
@MUX16/3:DAT4 :=  @0:F0,   @5000:F1,   @6000:F0;
@MUX16/3:DAT5 :=  @0:F1,  @20000:F0,  @21000:F1;
@MUX16/3:DAT6 :=  @0:F1,  @44000:F0,  @45000:F1;
@MUX16/3:DAT7 :=  @0:F0,  @29000:F1,  @30000:F0;
@MUX16/3:DAT8 :=  @0:F0,  @11000:F1,  @12000:F0;
@MUX16/3:DAT9 :=  @0:F1,  @14000:F0,  @15000:F1;
@MUX16/3:DAT10 := @0:F1,  @38000:F0,  @39000:F1;
@MUX16/3:DAT11 := @0:F0,  @35000:F1,  @36000:F0;
@MUX16/3:DAT12 := @0:F1,   @8000:F0,   @9000:F1;
@MUX16/3:DAT13 := @0:F0,  @17000:F1,  @18000:F0;
@MUX16/3:DAT14 := @0:F0,  @41000:F1,  @42000:F0;
@MUX16/3:DAT15 := @0:F1,  @32000:F0,  @33000:F1;

@MUX16/2:EXTCLK := @0:F0, [700:F0, 300:F1]**;

@MUX16/2:EXTRST := @0:F1 , @1000:F0;

@MUX16/3:SELCT0 := @0:F0, @13000:F1,  @37000:F0;

@MUX16/3:SELCT1 := @0:F0, @25000:F1,  @49000:F0;

@MUX16/3:SELCT2 := @0:F0,  @4000:F1,  @10000:F0,
                          @16000:F1,  @22000:F0,
                          @28000:F1,  @34000:F0,
                          @40000:F1,  @46000:F0;

@MUX16/3:SELCT3 := @0:F0,  @7000:F1,  @19000:F0,
                          @31000:F1,  @43000:F0;

@MUX16/4:IN001 := @0:F1;

/* ================================================== */

$OUTPUTS
PRINT_ON_CHANGE

FILE /USER/CLASS/MUX16/ATSPEED.VLAF <-
@MUX16/2:EXTCLK, EXTRST,

@MUX16/3:SELCT3, SELCT2, SELCT1, SELCT0,

@MUX16/3:DAT0, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7,
@MUX16/3:DAT8, DAT9, DAT10, DAT11, DAT12, DAT13, DAT14, DAT15,

@MUX16/2:YOUTPT,
@MUX16/4:PARAM;

$END
```

FIGURE A18

SOM__MCF.SING edited for AT-SPEED PRINT-ON-CHANGE

```
Date:  3Ø AUG 86 13:Ø6    File:  POCFMTAS.Ø1

1***CIRCUIT IDENTIFICATION =

                EESSSSDDDDDDDDDDDDDDDD YP
                XXEEEEAAAAAAAAAAAAAAAA OA
                TTLLLLTTTTTTTTTTTTTTTT UR
                CRCCCCØ123456789111111 TA
                LSTTTT           Ø12345 PM
                KT321Ø                  T

        TIME

7ØØ             11ØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
1ØØØ            ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
17ØØ            1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
2ØØØ            ØØØØØØØØØ1Ø11ØØ11Ø1ØØ1 Ø1
2Ø63            ØØØØØØØØØ1Ø11ØØ11Ø1ØØ1 11
27ØØ            1ØØØØØØØØ1Ø11ØØ11Ø1ØØ1 11
3ØØØ            ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
3Ø18            ØØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
37ØØ            1ØØØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
4ØØØ            ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
4Ø63            ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
47ØØ            1ØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
5ØØØ            ØØØ1ØØ1ØØ1111ØØ11Ø1ØØ1 11
5Ø18            ØØØ1ØØ1ØØ1111ØØ11Ø1ØØ1 Ø1
57ØØ            1ØØ1ØØ1ØØ1111ØØ11Ø1ØØ1 Ø1
6ØØØ            ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
6Ø63            ØØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
67ØØ            1ØØ1ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
7ØØØ            ØØ11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
7Ø18            ØØ11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
77ØØ            1Ø11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
8ØØØ            ØØ11ØØ1ØØ1Ø11ØØ11ØØØØ1 Ø1
8Ø63            ØØ11ØØ1ØØ1Ø11ØØ11ØØØØ1 11
87ØØ            1Ø11ØØ1ØØ1Ø11ØØ11ØØØØ1 11
9ØØØ            ØØ11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
9Ø18            ØØ11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
97ØØ            1Ø11ØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
1ØØØØ           ØØ1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
1ØØ63           ØØ1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
1Ø7ØØ           1Ø1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
11ØØØ           ØØ1ØØØ1ØØ1Ø11Ø111Ø1ØØ1 11
1101Ø           ØØ1ØØØ1ØØ1Ø111Ø111Ø1ØØ1 Ø1
117ØØ           1Ø1ØØØ1ØØ!Ø11Ø111Ø1ØØ1 Ø1
12ØØØ           ØØ1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 Ø1
12Ø63           ØØ1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
127ØØ           1Ø1ØØØ1ØØ1Ø11ØØ11Ø1ØØ1 11
13ØØØ           ØØ1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 11
1301Ø           ØØ1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
137ØØ           1Ø1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
14ØØØ           ØØ1ØØ11ØØ1Ø11ØØØ1Ø1ØØ1 Ø1
14Ø63           ØØ1ØØ11ØØ1Ø11ØØØ1Ø1ØØ1 11
147ØØ           1Ø1ØØ11ØØ1Ø11ØØØ1Ø1ØØ1 11
15ØØØ           ØØ1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 11
1501Ø           ØØ1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
157ØØ           1Ø1ØØ11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
16ØØØ           ØØ11Ø11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
16Ø63           ØØ11Ø11ØØ1Ø11ØØ11Ø1ØØ1 11
167ØØ           1Ø11Ø11ØØ1Ø11ØØ11Ø1ØØ1 11
17ØØØ           ØØ11Ø11ØØ1Ø11ØØ11Ø11Ø1 11
1701Ø           ØØ11Ø11ØØ1Ø11ØØ11Ø11Ø1 Ø1
177ØØ           1Ø11Ø11ØØ1Ø11ØØ11Ø11Ø1 Ø1
18ØØØ           ØØ11Ø11ØØ1Ø11ØØ11Ø1ØØ1 Ø1
```

PRINT-ON-CHANGE

AT-SPEED

FIGURE A19

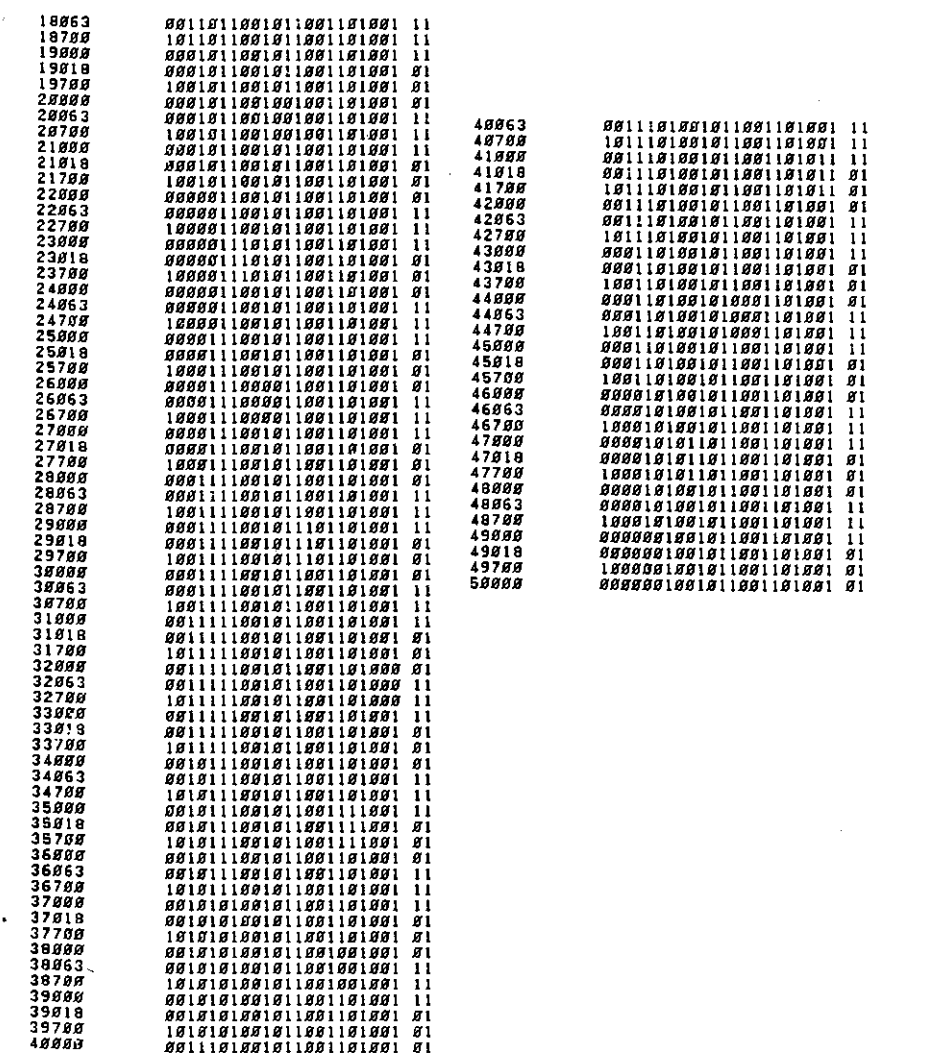


FIGURE A19 CONTINUED

```
Date:  19 SEP 88 14:48   File:  PARAMETER.SING

/****
*
* DESIGN PATH /USER/CLASS/MUX16   DATE 24 MAY 1988 13:30
*
* COMPANY  ___AMCC__________   CIRCUIT_NAME  __16:1 MUX_
*
* ARRAY    _Q5000 SERIES PO#  ___-_______   REV  ______
*
* DESIGNER _____DEW_____________________
*
* What tests does this control file support:  _________
*
* __PARAMETRIC TESTS - ALL_____________________________
*
****/

/***   Configuration section    ****/

$CONFIGURATION

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

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

$SIGNAL_GENERATORS

@MUX16/3:DAT0  :=  00:F1,   @10000:F0,   @20000:F1;
@MUX16/3:DAT1  :=  00:F1,   @30000:F0,   @40000:F1;
@MUX16/3:DAT2  :=  00:F1,   @50000:F0,   @60000:F1;
@MUX16/3:DAT3  :=  00:F1,   @70000:F0,   @80000:F1;
@MUX16/3:DAT4  :=  00:F1,   @90000:F0,  @100000:F1;
@MUX16/3:DAT5  :=  00:F1,  @110000:F0,  @120000:F1;
@MUX16/3:DAT6  :=  00:F1,  @130000:F0,  @140000:F1;
@MUX16/3:DAT7  :=  00:F1,  @150000:F0,  @160000:F1;
@MUX16/3:DAT8  :=  00:F1,  @170000:F0,  @180000:F1;
@MUX16/3:DAT9  :=  00:F1,  @190000:F0,  @200000:F1;
@MUX16/3:DAT10 :=  00:F1,  @210000:F0,  @220000:F1;
@MUX16/3:DAT11 :=  00:F1,  @230000:F0,  @240000:F1;
@MUX16/3:DAT12 :=  00:F1,  @250000:F0,  @260000:F1;
@MUX16/3:DAT13 :=  00:F1,  @270000:F0,  @280000:F1;
@MUX16/3:DAT14 :=  00:F1,  @290000:F0,  @300000:F1;
@MUX16/3:DAT15 :=  00:F1,  @310000:F0,  @320000:F1;

@MUX16/2:EXTCLK :=  00:F1, @330000:F0,  @340000:F1;

@MUX16/2:EXTRST :=  00:F1, @350000:F0,  @360000:F1;

@MUX16/3:SELCT0 :=  00:F1, @370000:F0,  @380000:F1;

@MUX16/3:SELCT1 :=  00:F1, @390000:F0,  @400000:F1;

@MUX16/3:SELCT2 :=  00:F1, @410000:F0,  @420000:F1;

@MUX16/3:SELCT3 :=  00:F1, @430000:F0,  @440000:F1;

@MUX16/4:IN001 :=  00:F1;

/* ============================================================ */
```

SOM_MCF.SING edited for Parametric Vectors

FIGURE A20

```
$OUTPUTS

FILE /USER/CLASS/MUX16/FUNCTION.VLAF <-

/* ------------------------------------------------- */
/* LIST ALL PRIMARY INPUTS; EXCLUDE THERMAL DIODES,  */
/* MONITOR POINTS AND VBxx MACRO SIGNALS             */
/* ------------------------------------------------- */

@MUX16/2:EXTCLK, EXTRST,
@MUX16/3:SELCT3, SELCT2, SELCT1, SELCTØ,
@MUX16/3:DATØ, DAT1, DAT2, DAT3, DAT4, DAT5, DAT6, DAT7,
@MUX16/3:DAT8, DAT9, DAT1Ø, DAT11, DAT12, DAT13, DAT14, DAT15,

/* ------------------------------------------------- */
/* LIST ALL PRIMARY OUTPUTS; EXCLUDE THERMAL DIODES, */
/* MONITOR POINTS AND VBxx MACRO SIGNALS             */
/* ------------------------------------------------- */

@MUX16/2:YOUTPT,
@MUX16/4:PARAM;

/* ------------------------------------------------- */
/* LIST INTERNAL 3-STATE ENABLES AND BIDIRECTIONAL   */
/*   ENABLES HERE                                    */
/*   IF ANY                                          */
/* ------------------------------------------------- */

$END
```

FIGURE A20 CONTINUED

```
Date:  26 AUG 86 11:54    File:  SIMFMTPR.Ø1

1***CIRCUIT IDENTIFICATION =

                EESSSSDDDDDDDDDDDDDDD YP
                XXEEEEAAAAAAAAAAAAAAA OA
                TTLLLLTTTTTTTTTTTTTTT UR
                CRCCCCØ123456789111111 TA
                LSTTTT           Ø12345 PM
                KT321Ø                 T

        TIME

9999            111111111111111111111 ØØ
19999           111111Ø1111111111111111 Ø1
29999           111111111111111111111 ØØ
39999           111111Ø1111111111111111 Ø1
49999           111111111111111111111 ØØ
59999           111111110111111111111 Ø1
69999           111111111111111111111 ØØ
79999           111111111Ø1111111111 Ø1
89999           111111111111111111111 ØØ
99999           111111111Ø1111111111 Ø1
1Ø9999          111111111111111111111 ØØ
119999          111111111Ø1111111111 Ø1
129999          111111111111111111111 ØØ
139999          111111111111Ø1111111 Ø1
149999          111111111111111111111 ØØ
159999          111111111111Ø1111111 Ø1
169999          111111111111111111111 ØØ
179999          111111111111Ø1111111 Ø1
189999          111111111111111111111 ØØ
199999          111111111111111Ø111111 Ø1
2Ø9999          111111111111111111111 ØØ
219999          111111111111111Ø11111 Ø1
229999          111111111111111111111 ØØ
239999          111111111111111Ø1111 Ø1
249999          111111111111111111111 ØØ
259999          111111111111111111Ø111 Ø1
269999          111111111111111111111 ØØ
279999          111111111111111111Ø11 Ø1
289999          111111111111111111111 ØØ
299999          111111111111111111Ø1 Ø1
3Ø9999          111111111111111111111 ØØ
319999          111111111111111111111Ø Ø1
329999          111111111111111111111 ØØ
339999          Ø111111111111111111111 Ø1
349999          111111111111111111111 ØØ
359999          1Ø111111111111111111111 Ø1
369999          111111111111111111111 ØØ
379999          111111Ø1111111111111111 Ø1
389999          111111111111111111111 ØØ
399999          111110111111111111111 Ø1
4Ø9999          111111111111111111111 ØØ
419999          111Ø1111111111111111 Ø1
429999          111111111111111111111 ØØ
439999          11Ø1111111111111111111 Ø1
449999          111111111111111111111 ØØ
```

Parametric Vectors

(Sampled only)

FIGURE A21

APNOTE 2.DNIX

(809)

INTRODUCTION

The (708) release includes a new DAISY netlister for use with 5.01 and 5.02 versions of DAISY-DNIX II. Under the new netlister, nested and hierarchical designs are more fully supported.

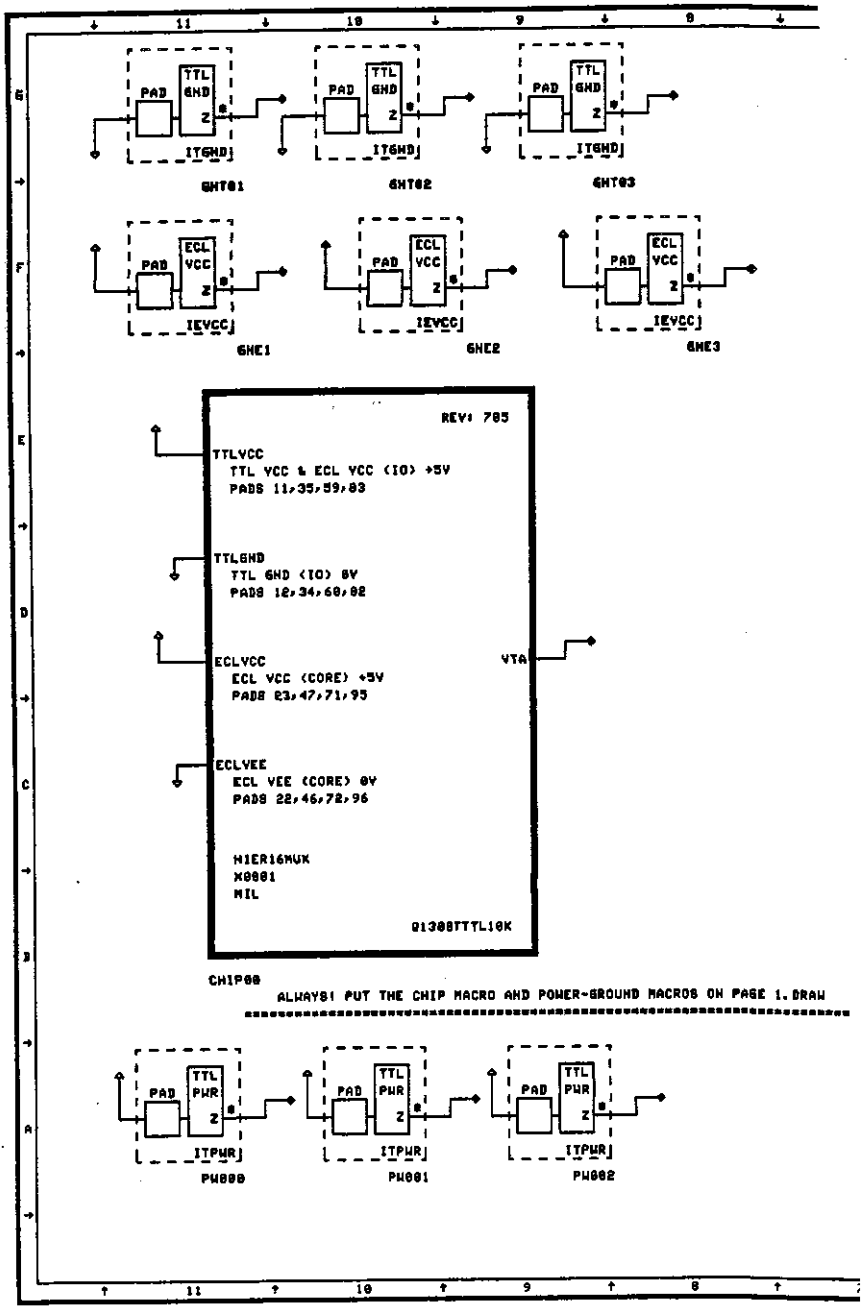
Nested applications are covered in APNOTE 3.DNIX.

The following are the design methodologies that apply to hierarchical designs.

First, DAISY defines such a design to be centered at a top-level directory and to be spread downward in a directory tree. All circuit operations such as DRINK, AGIF, ERCs, Front-Annotation, SIFT, SOM, DLS, etc., are to be performed with the current context set to the top of the design tree.

- Place the chip macro and extra power and ground macros on page 1.DRAW and begin the hierarchy block diagram on page 2.DRAW at the top level.

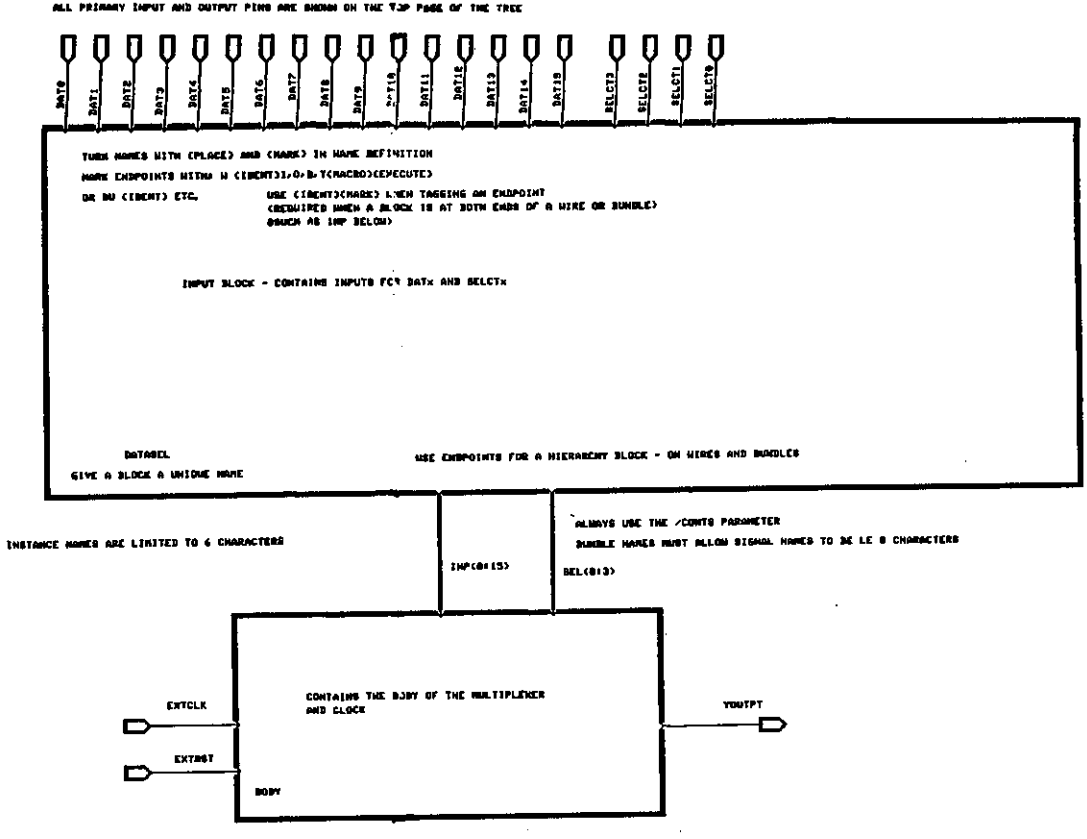
All extra power and ground macros and the chip macro should be on one page. Extra power and ground macros are named for placement. REGARDLESS OF THE TECHNOLOGY OF THE CHIP, all extra power and ground macro input pins are tied to global ground (NOT VDD) and all extra power and ground macro output pins are tied to a terminator. On DAISY, that is usually /LWTERM. Wire the terminator to the macro pin. Touching pins will sometimes cause a reboot.



PAGE 1.DRAW

APNOTE 2.DNIX 2

PAGE 2.DRAW


ALL PRIMARY INPUT AND OUTPUT PINS ARE SHOWN ON THE TOP PAGE OF THE TREE
DAT0
DAT1
DAT2
DAT3
DAT4
DAT5
DAT6
DAT7
DAT8
DAT9
DAT10
DAT11
DAT12
DAT13
DAT14
DAT15
SELCT3
SELCT2
SELCT1
SELCT0
TUBE NAMES WITH (PLACE) AND (MARK) IN NAME DEFINITION
MARK ENDPOINTS WITH# W (IBCNT)3)-0-B-T(MACRO)(EXECUTE)
OR BU (IBCNT) ETC.
USE (IBCNT)(MARK) WHEN TAGGING AN ENDPOINT
(REQUIRED WHEN A BLOCK IS AT BOTH ENDS OF A WIRE OR BUNDLE)
@SUCH AS IMP BELOW)
INPUT BLOCK - CONTAINS INPUTS FOR DATx AND SELCTx
DATASEL
GIVE A BLOCK A UNIQUE NAME
USE ENDPOINTS FOR A HIERARCHY BLOCK - ON WIRES AND BUNDLES
INSTANCE NAMES ARE LIMITED TO 6 CHARACTERS
IMP(0#(5)
ALWAYS USE THE /CONTS PARAMETER
BUNDLE NAMES MUST ALLOW SIGNAL NAMES TO BE LE 8 CHARACTERS
BEL(0:3)
EXTCLK
EXTRST
CONTAINS THE BODY OF THE MULTIPLEXER
AND CLOCK
BODY
YOUTPT

- Create the blocks top down so the system will create the directory for each as it goes.

As a block is created on the drawing page, the system creates a directory for it. If you copy hierarchy pages, you must create the directory for the system or it will become confused. If you rename a block, the system renames the directory at the same time. Try to plan your names in advance and minimize the interference with the system's book-keeping.

- The top sheet or all sheets at the top directory level show all primary inputs and outputs.

Although the input and output macros themselves are probably not on that sheet, all primary inputs and outputs must be placed on a top directory level drawing page. A block diagram may run to more than one page. The primary I/Os must be on one of those pages.

- Use wires and bundles between blocks

There is no restriction on the use of bundles and wires. AMCC prefers that you be consistent across a boundary, i.e., if you leave a block as a bundle and enter the next as a bundle, that the definition pages for those blocks also show bundles.

- If a block is defined as a page which references an additional level of blocks, that is still hierarchy. In DAISY, if the blocks referenced are not unique (are the same), that is nested. DAISY also has a cell - an on-page nested block. Refer the APNOTE 3.DNIX for nested rules.

- Name wires.

Always name wires according to the AMCC EWS Schematic Rules, Volume II, Section 3. All wires crossing a block boundary are named.

- Place a contents parameter on the bundles, not a name.

```
PARA {SELECT}/CONTS{EXTRACT}value
                    {DEF}{PLACE}...{EXECUTE}
```

You may name bundles. AMCC software uses the /CONTS parameter definition.

ALways use the /CONTS parameter for bundles. Bundle nets do not have to be given /CONTS parameters when they are branches from the main bundle on the page. If they are, use the "local" and not the "global" parameter.

Wires feeding into/out of bundles are named with the contents parameter name and the appropriate digits. For example, a bundle with a /CONTS parameter of DTD(0:12) would have signals such as DTD0 and DTD12.

- Place endpoints on wires and bundles:

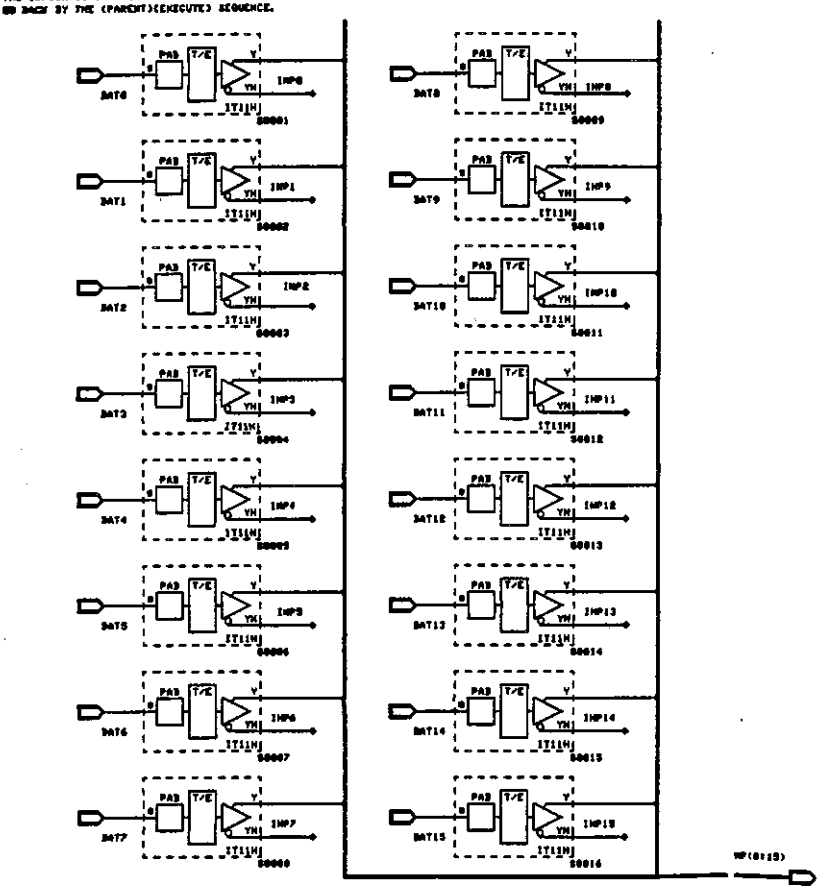
  W {IDENT}x{MACRO}[{MARK}]{EXECUTE} where x = I|O|T|B

  for input, output, tristate, or bidirectional.

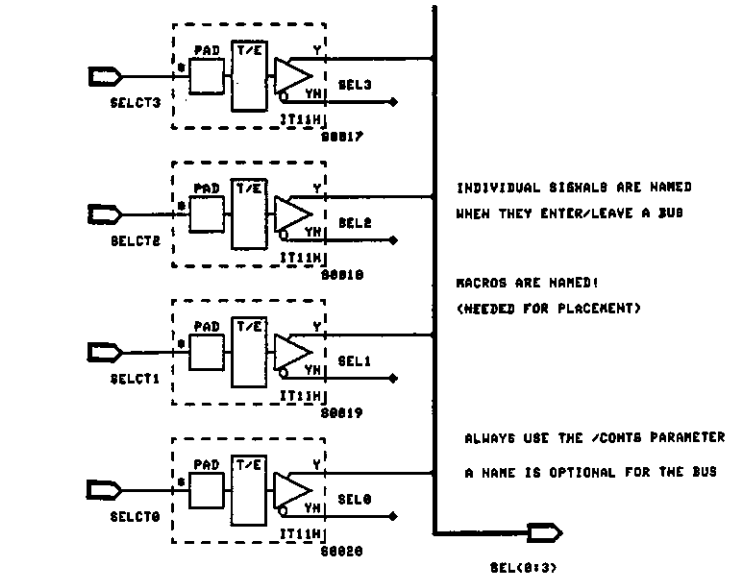
  {MARK} is used to position the endpoint and is required when a bus runs between two blocks. The endpoints tend to dissappear when notes are added. Use the {REDRAW} key to get them back.

  - Place both endpoints on by:

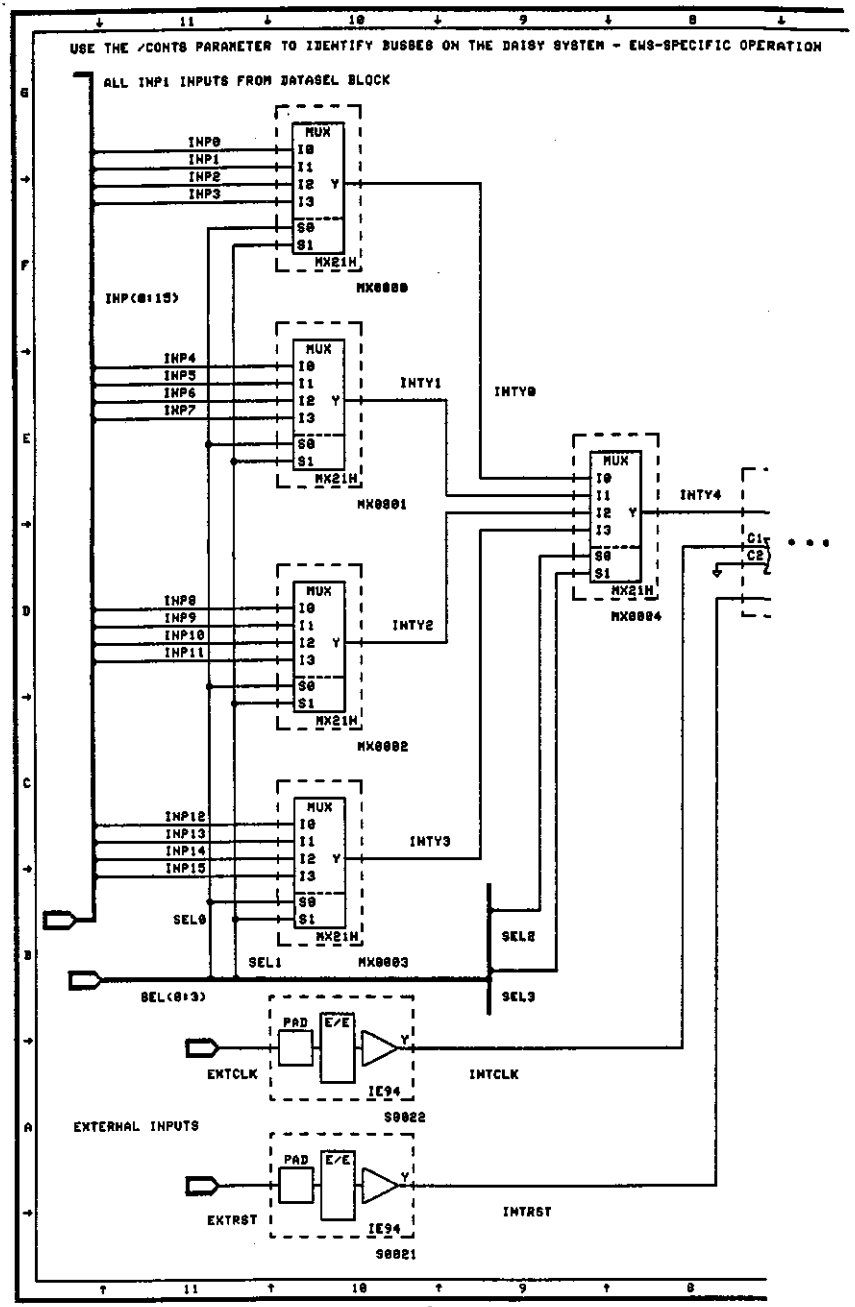
    W {IDENT}I{MACRO}O{MACRO}{EXECUTE}


CHANGE TO THIS PAGE BY THE (CHANGE)(SELECT)(EXECUTE) COMMAND SEQUENCE WHILE
THE CURSOR IS ON THE TOP-LEVEL BLOCK. EACH TOP LEVEL BLOCK HAS ITS OWN DIRECTORY.
GO BACK BY THE (PARENT)(EXECUTE) SEQUENCE.
DAT0
DAT1
DAT2
DAT3
DAT4
DAT5
DAT6
DAT7
DAT8
DAT9
DAT10
DAT11
DAT12
DAT13
DAT14
DAT15
INP0
INP1
INP2
INP3
INP4
INP5
INP6
INP7
INP8
INP9
INP10
INP11
INP12
INP13
INP14
INP15
WP(0:15)

INPUTS

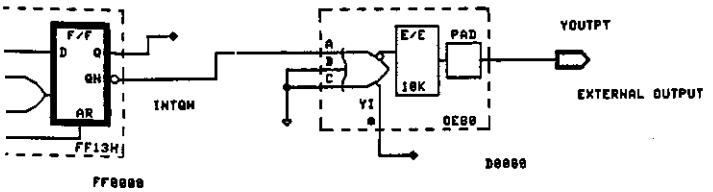


INPUTS CONTINUED



BODY

APNOTE 2.DNIX 8



ALL I/O TO/FROM A BLOCK (SIGNALS CROSSING BLOCK BOUNDARIES)
ARE CONNECTED TO HIERARCHY CONNECTORS

USE /RWHICON/ /RWHOCON/ /RBHICON/ RBHOCON/ ETC.

BODY CONTINUED

- Endpoints dissappear when a pin attribute table is deleted. Pin attribute tables are not used for hierarchy blocks; Pin attribute tables are mandatory for nested blocks. Refer to APNOTE 3.DNIX.

- When on the top level, place a cursor on the block whose directory you wish to go to. Use the {CHNGE} key or type CHANGE THEN {SELECT}{EXECUTE}. The system moves down the hierarchy to the lower directory, page 1.draw.

  {CHANGE}{SELECT}{EXECUTE}

- To go back up, use

  {PARENT}{EXECUTE}

Moving up and down the design tree is via the change key and the select key or via the parent key. Moving around within the tree level, as in multiple page definitions, is done via the next and previous keys. The system will prompt for saving a page if it was edited. Always execute a SAVE command once every 10-20 minutes, or at the end of a difficult edit step, if that step took less than 10 minutes. Protect yourself.

Always save to floppy and to a hard disk at the end of a session.

- Name macros if the design is not nested (if a block is not called more than once in a design).

If the hierarchy structure is deep (more than two levels), the user defined name and the block reference may be too long in which case the AMCC netlister will rename the macro (or signal). Refer to AMCCXREF.LST in the /ERC subdirectory. ALL MACROS MUST BE NAMED. ALL BLOCKS MUST BE NAMED.

```
/* Daisy design pathname:  /USER/CLASS/MUXHIER   */
$CONTROL
MODE ADD
SECTION DELAY:N:6
$DELAY
/*                                R I S E        F A L L       */
/*                             Typ   Min   Max   Typ   Min   Max   */
@MUXHIER 2:INPØ          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP1          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP1Ø         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP11         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP12         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP13         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP14         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP15         =   23    21    25    39    35    42 ;
@MUXHIER 2:INP2          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP3          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP4          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP5          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP6          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP7          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP8          =   23    21    25    39    35    42 ;
@MUXHIER 2:INP9          =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTCLK       =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTON        =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTRST       =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTYØ        =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTY1        =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTY2        =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTY3        =   23    21    25    39    35    42 ;
@MUXHIER/BODY 1:INTY4        =   23    21    25    39    35    42 ;
@MUXHIER 2:SELØ          =   7Ø    63    77   117   1Ø6   129 ;
@MUXHIER 2:SEL1          =   7Ø    63    77   117   1Ø6   129 ;
@MUXHIER 2:SEL2          =   23    21    25    39    35    42 ;
@MUXHIER 2:SEL3          =   23    21    25    39    35    42 ;
$END
```

FNTMIL.DSY FILE

```
/****
*
* DESIGN PATH /USER/CLASS/MUXHIER   DATE 30 MAR 1986 11:59
*
* 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
@MUXHIER/DATASEL/1:DAT0 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT1 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT10 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT11 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT12 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT13 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT14 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT15 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT2 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT3 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT4 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT5 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT6 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT7 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT8 := 00:F0 ;
@MUXHIER/DATASEL/1:DAT9 := 00:F0 ;
@MUXHIER/BODY/1:EXTCLK := 00:F0 ;
@MUXHIER/BODY/1:EXTRST := 00:F0 ;
@MUXHIER/DATASEL/1:SELCT0 := 00:F0 ;
@MUXHIER/DATASEL/1:SELCT1 := 00:F0 ;
@MUXHIER/DATASEL/1:SELCT2 := 00:F0 ;
@MUXHIER/DATASEL/1:SELCT3 := 00:F0 ;
```

UNEDITED SOM_MCF.SING FILE

APNOTE 3.DNIX

(809)

INTRODUCTION

The (708) release includes a new DAISY netlister for use with 5.01 and 5.02 versions of DAISY-DNIX II. Under the new netlister, nested and hierarchical designs are more fully supported.

If a block is defined as a page which references an additional level of blocks, that is still hierarchy. In DAISY, if the blocks referenced are not unique (are the same), that is nested. Design structures may be combinations of these.

Cells are an on-page nest (where a block definition does not require an entire schematic page).

Non-nested hierarchy applications are covered in APNOTE 2.DNIX.

The new AMCC AGIF netlister handles a deeply structured drawing. It can handle a top-level hierarchy block calling a nested block or blocks which in turn call other hierarchy structures.

The following are the design methodologies that apply to DAISY nested hierarchical designs.

First, DAISY defines such a design to be centered at a top-level directory and to be spread downward in a directory tree. All circuit operations such as DRINK, AGIF, ERCs, Front-Annotation, SIFT, SOM, DLS, etc., are to be performed with the current context set to the top of the design tree.

- Place the chip macro and extra power and ground macros on page 1.DRAW and begin the hierarchy block diagram on page 2.DRAW at the top level.

All extra power and ground macros and the chip macro should be on one page. Extra power and ground macros are named for placement. REGARDLESS OF THE TECHNOLOGY OF THE CHIP, all extra power and ground macro input pins are tied to global ground (NOT VDD) and all extra power and ground macro output pins are tied to a terminator. On DAISY, that is usually /LWTERM. Wire the terminator to the macro pin. Touching pins will sometimes cause a reboot.

- Create the blocks top down so the system will create the directory for each as it goes.

As a block is created on the drawing page (the box drawn and given a unique instance name), the system creates a directory for it. If you copy hierarchy pages, you must create the directory for the system or it will become confused. If you rename a block, the system renames the directory at the same time. Try to plan your names in advance and minimize the interference with the system's book-keeping.

- Cells are on-page nesteds, can be off-page nesteds and use the CELL command. Put the cell name inside of the cell.

- Nested pages (definition thereof) can not be at the top level. They require their own individual directory.

Create a directory that is one down from the tree top and name it the same as the LREF parameter you will use to reference it. This makes it easier to find and use by a reviewer. The pathname to this directory is what goes into the /AMCC/CONFIG/NESTED file. A nested block may require more than one definition page.

- For nested blocks, they must have:
  instance name (6 character limit)
  /CTL parameter defined as N
  /LREF parameter with the name the system will use to find the nested page
  pins must be named by the PIN attribute command

The AGIF netlister <u>does</u> recognize and use the /LREF parameter. Make the instance name of the block unique. Use the /LREF parameter to link the structure. Place the /LREF parameter and the path location for the block/cell definition in the /AMCC/CONFIG/NESTED file.

- The top sheet or all sheets at the top directory level show all primary inputs and outputs.

Although the input and output macros themselves are probably not on that sheet, all primary inputs and outputs must be placed on a top directory level drawing page. A block diagram may run to more than one page. The primary I/Os must be on one of those pages.

- Use wires and bundles between blocks

There is no restriction on the use of bundles and wires. AMCC prefers that you be consistent across a boundary, i.e., if you leave a block as a bundle and enter the next as a bundle, that the definition pages for those blocks also show bundles.

- Name wires.

Always name wires according to the AMCC EWS Schematic Rules, Volume II, Section 3. All wires crossing a block boundary are named.

- Name bundles when they are inside a nested block definition.

Bundles may be named at any time but they MUST be named when inside a nested block definition.

- Place a contents parameter on the bundles, not a name.

  PARA {SELECT}/CONTS{EXTRACT}value
  {DEF}{PLACE}...{EXECUTE}

  You may name bundles. AMCC software uses the /CONTS parameter definition.

ALways use the /CONTS parameter for bundles. Bundle nets do not have to be given /CONTS parameters when they are branches from the main bundle on the page. If they are, use the "local" and not the "global" parameter.

Wires feeding into/out of bundles are named with the contents parameter name and the appropriate digits. For example, a bundle with a /CONTS parameter of DTD(0:12) would have signals such as DTD0 and DTD12.

- Place endpoints on wires and bundles going into or out of non-nested blocks.

  W {IDENT}x{MACRO}[{MARK}]{EXECUTE}

  where x = I|O|T|B

  for input, output, tristate, or bidirectional.

  {MARK} is used to position the endpoint and is required when a bus runs between two blocks. The endpoints tend to dissappear when notes are added. Use the {REDRAW} key to get them back.

  - Place both endpoints on by:

    W {IDENT}I{MACRO}O{MACRO}{EXECUTE}

- Endpoints dissappear when a pin attribute table is deleted.

Pin attribute tables are not used for hierarchy blocks; Pin attribute tables are mandatory for nested blocks. Endpoints are required for non-nested blocks.

- When on the top level, place a cursor on the block whose directory you wish to go to. Use the {CHNGE} key or type CHANGE THEN {SELECT}{EXECUTE}. The system moves down the hierarchy to the lower directory, page 1.draw.

  {CHANGE}{SELECT}{EXECUTE}

- To go back up, use

  {PARENT}{EXECUTE}

Moving up and down the design tree is via the change key and the select key or via the parent key. Moving around within the tree level, as in multiple page definitions, is done via the next and previous keys. The system will prompt for saving a page if it was edited. Always execute a SAVE command once every 10-20 minutes, or at the end of a difficult edit step, if that step took less than 10 minutes. Protect yourself.

Always save to floppy and to a hard disk at the end of a session.

- Name macros.

If the hierarchy structure is deep (more than two levels) or is nested, the user defined name and the block reference may be too long in which case the AMCC netlister will rename the macro (or signal). Refer to AMCCXREF.LST in the /ERC subdirectory. ALL MACROS MUST BE NAMED. ALL BLOCKS MUST BE NAMED.

- Identify pins by the PIN command when blocks are nested. For DED2, refer to its manual for the equivalent operation.

PIN {PLACE}
    at this point the block and the pin definition block will flash.
    - the word "name" will be flashing
name{DEF}
    type in the pin name and hit the define key
number{DEF}
    physical identification, default is one
BU{DEF} or just {CONF}
    type of connection, wire is default
OUTPUT{DEF} or INPUT{DEF} or BIDIRECTIONAL{DEF}
    the attribute default is INPUT.
n{DEF}
    n is the width, default is one.

- A pin name must match with the definition page(s) name
  - if the pin is a bundle then a bundle connector must be on the lower level page
  - the pin name at the upper level; the bundle name (if used), the contents parameter, and the individual signal names on the lower page should all match in a nested definition.

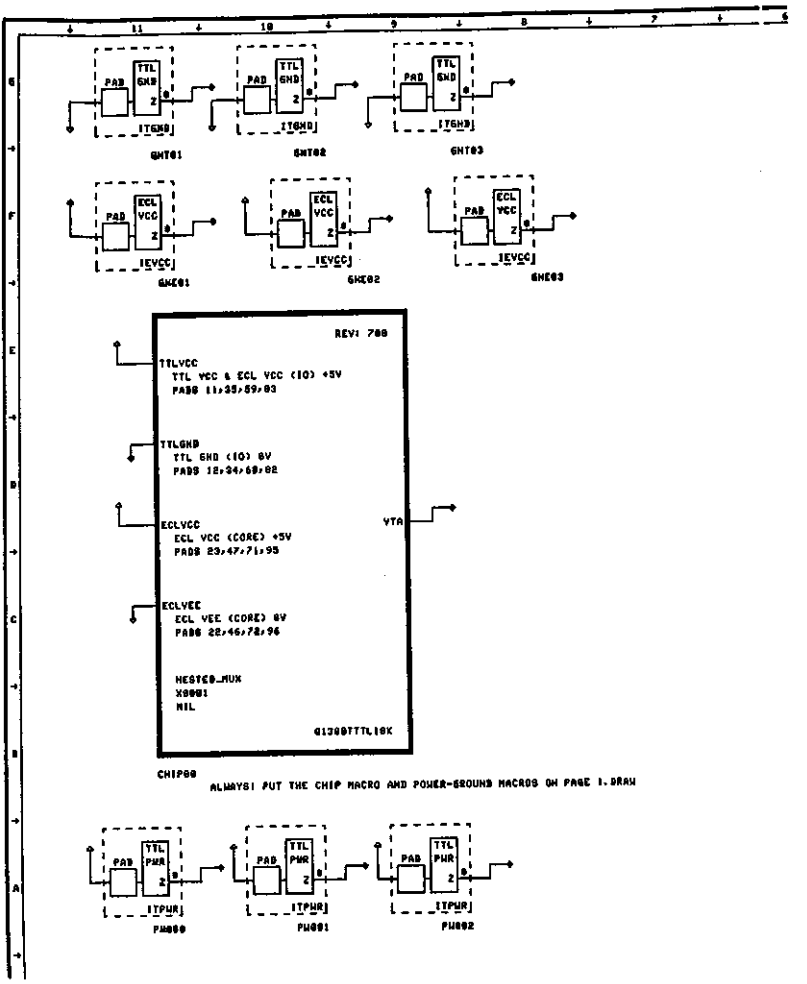
- View a pin definition by PIN {SELECT}
- Confirm an item's definition by {CONF}
- exit a pin edit by {EXECUTE}
- Edit the /AMCC/CONFIG/NESTED file
  LREF_name path_name

path_name is the path where the definition page can be found.

LREF_name is the value you assigned to LREF for each block that will reference the same DEFINITION. The definition can be a page or several pages. It must be uniquely identifiable in the nested file.

If you are using RAM or other predefined nested macros, copy over the library nested file first, such as: /AMCC/Q3500_LIBS/Q3500_RAM.NEST. Copy this file into the /AMCC/CONFIG/NESTED file before adding your structures. Be sure to save the edited file and to submit it on the floppy(s) sent for design submission. The /AMCC/CONFIG/NESTED file will not be changed when you log on or off.

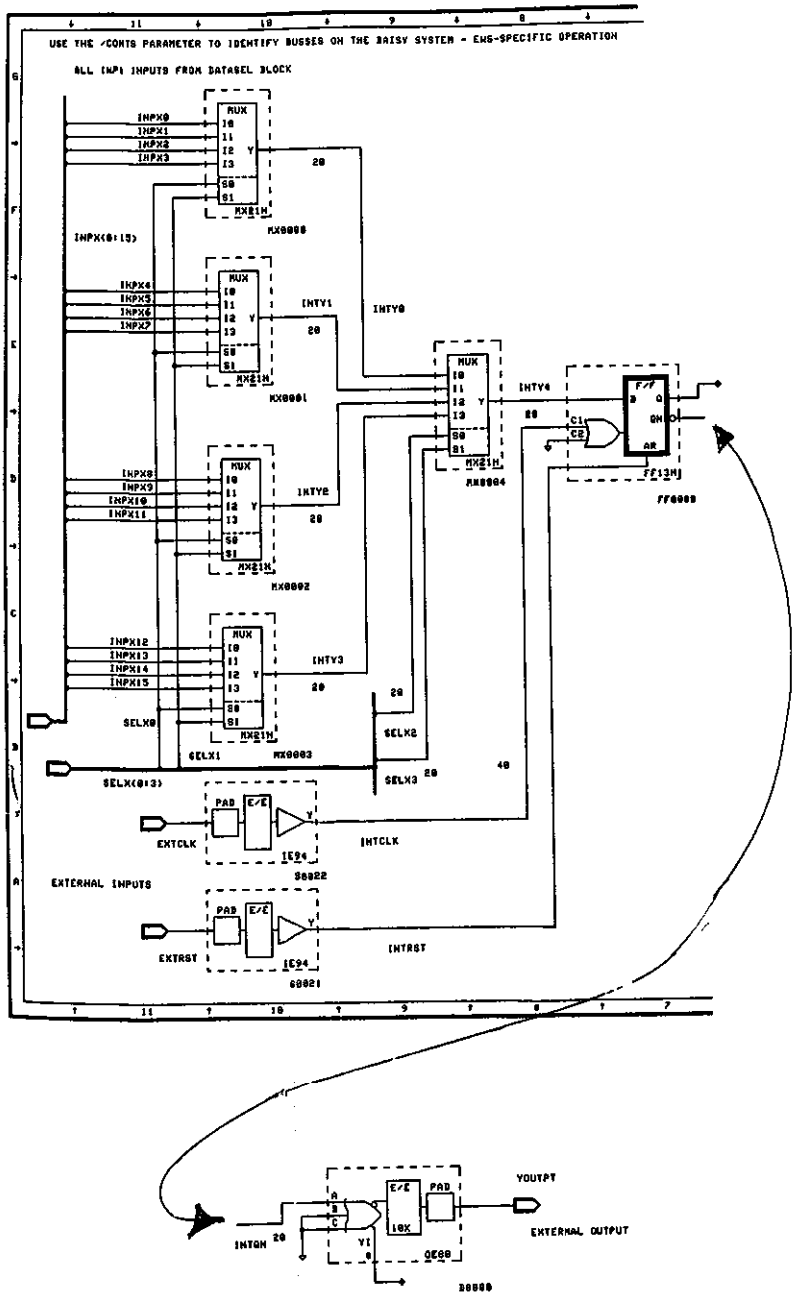
```
Date:  14 SEP 88 14:37   File:  /AMCC/Q5000_LIBS/Q5000_RAM.NEST

$FORMAT_CARD$
16A 60C
$END$

RAM10  $(NETPATH)/Q5000_LIBS/NESTED/MM10
RAM10H $(NETPATH)/Q5000_LIBS/NESTED/MM10H
RAM20  $(NETPATH)/Q5000_LIBS/NESTED/MM20
RAM20H $(NETPATH)/Q5000_LIBS/NESTED/MM20H
```


PAD
TTL GND
ITGND
GNT01
GNT02
GNT03
ECL VCC
IEVCC
GNE01
GNE02
GNE03
REV: 7B8
TTLVCC
TTL VCC & ECL VCC (IO) +5V
PADS 11,35,59,83
TTLGND
TTL GND (IO) 0V
PADS 12,34,60,82
ECLVCC
ECL VCC (CORE) +5V
PADS 23,47,71,95
ECLVEE
ECL VEE (CORE) 0V
PADS 22,46,72,96
NESTED_MUX
X0001
NIL
VTA
G1200TTL10K
CHIP00
ALWAYS! PUT THE CHIP MACRO AND POWER-GROUND MACROS ON PAGE 1.DRAW
TTL PWR
ITPWR
PW000
PW001
PW002

PAGE 1.DRAW

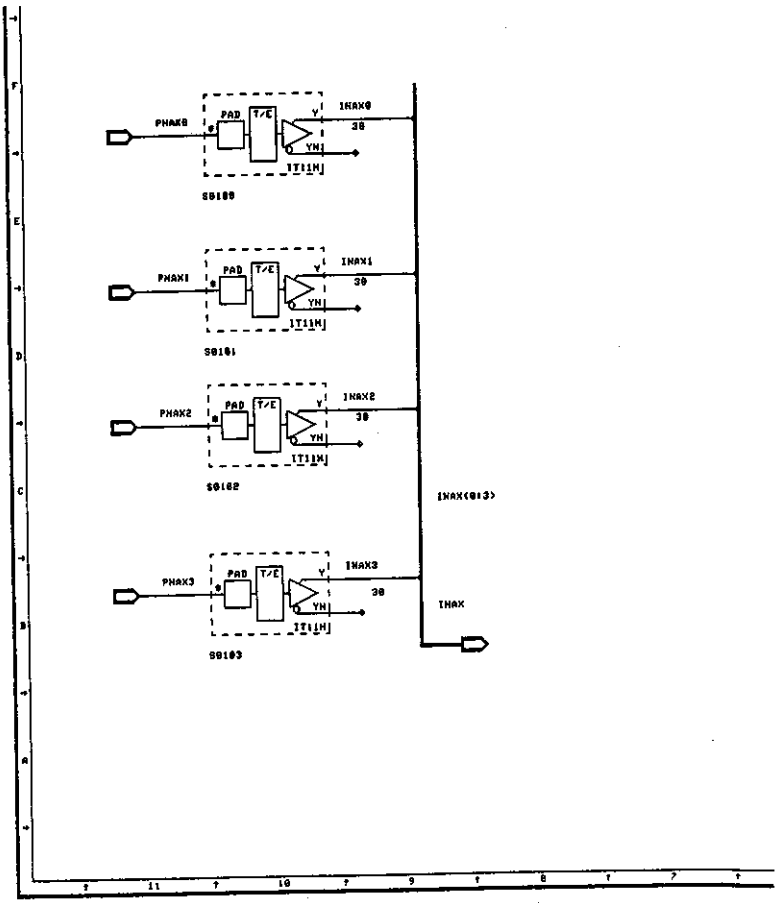
PAGE 2.DRAW

APPNOTE 3.DNIX10



DATBLK PAGE

APPNOTE 3.DNIX11



NESTED BLOCK DEFINITION

APPNOTE 3.DNIX12

```
Date:  Ø3 JUN 86 13:26   File:  FNTMIL.DSY

/* Daisy design pathname:  /USER/CLASS/NESTED1  */
$CONTROL
MODE ADD
SECTION DELAY:N:6
$DELAY
/*                              R I S E          F A L L       */
/*                          Typ  Min   Max   Typ  Min   Max    */
@NESTED1 2:INPXØ          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX1          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX1Ø         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX11         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX12         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX13         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX14         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX15         =  23   21    25    39   35    42 ;
@NESTED1 2:INPX2          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX3          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX4          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX5          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX6          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX7          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX8          =  23   21    25    39   35    42 ;
@NESTED1 2:INPX9          =  23   21    25    39   35    42 ;
@NESTED1/DATBLK 1:INTCLK      =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTQN       =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTRST      =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTYØ       =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTY1       =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTY2       =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTY3       =  23   21    25    39   35     42 ;
@NESTED1/DATBLK 1:INTY4       =  23   21    25    39   35     42 ;
@NESTED1 2:SELXØ          =  7Ø   63    77   117  1Ø6   129 ;
@NESTED1 2:SELX1          =  7Ø   63    77   117  1Ø6   129 ;
@NESTED1 2:SELX2          =  23   21    25    39   35    42 ;
@NESTED1 2:SELX3          =  23   21    25    39   35    42 ;
$END
```

FNTMIL.DSY

APPNOTE 3.DNIX13

```
Date:  03 JUN 86 13:32   File:  SOM_MCF.SING

/****
*
* DESIGN PATH /USER/CLASS/NESTED1   DATE 29-MAY-1986 13:17
*
* 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
@NESTED1/2:DATX0 := 00:F0 ;
@NESTED1/2:DATX1 := 00:F0 ;
@NESTED1/2:DATX10 := 00:F0 ;
@NESTED1/2:DATX11 := 00:F0 ;
@NESTED1/2:DATX12 := 00:F0 ;
@NESTED1/2:DATX13 := 00:F0 ;
@NESTED1/2:DATX14 := 00:F0 ;
@NESTED1/2:DATX15 := 00:F0 ;
@NESTED1/2:DATX2 := 00:F0 ;
@NESTED1/2:DATX3 := 00:F0 ;
@NESTED1/2:DATX4 := 00:F0 ;
@NESTED1/2:DATX5 := 00:F0 ;
@NESTED1/2:DATX6 := 00:F0 ;
@NESTED1/2:DATX7 := 00:F0 ;
@NESTED1/2:DATX8 := 00:F0 ;
@NESTED1/2:DATX9 := 00:F0 ;
@NESTED1/DATBLK/1:EXTCLK := 00:F0 ;
@NESTED1/DATBLK/1:EXTRST := 00:F0 ;
@NESTED1/2:SELC0 := 00:F0 ;
@NESTED1/2:SELC1 := 00:F0 ;
@NESTED1/2:SELC2 := 00:F0 ;
@NESTED1/2:SELC3 := 00:F0 ;
```

UNEDITED

SOM_MCF.SING

APPNOTE 3.DNIX14