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.

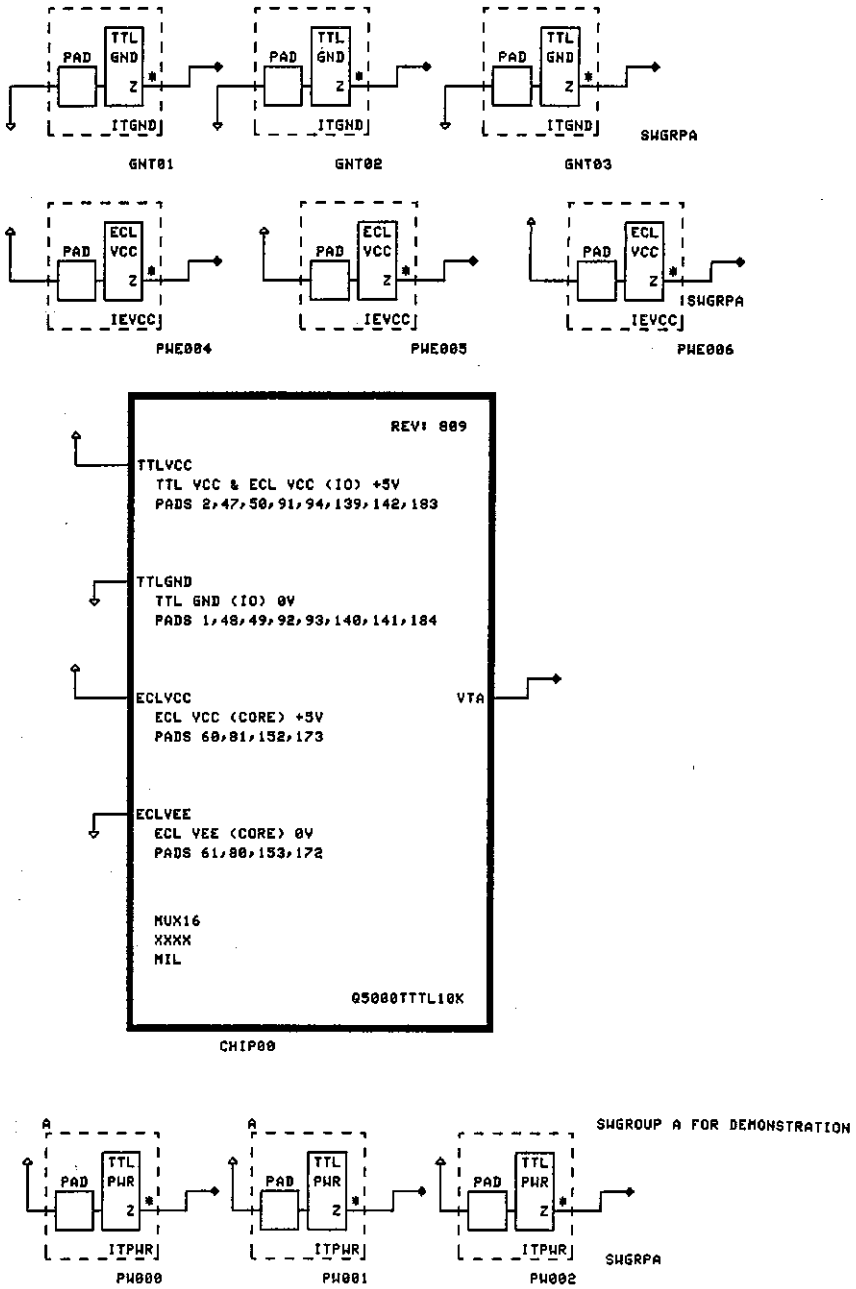

PAD
TTL GND
Z
ITGND
GNT01
GNT02
GNT03
SWGRPA
ECL VCC
IEVCC
PWE004
PWE005
PWE006
REV: 009
TTLVCC
TTL VCC & ECL VCC (IO) +5V
PADS 2,47,50,91,94,139,142,183
TTLGND
TTL GND (IO) 0V
PADS 1,48,49,92,93,140,141,184
ECLVCC
ECL VCC (CORE) +5V
PADS 60,81,152,173
VTA
ECLVEE
ECL VEE (CORE) 0V
PADS 61,80,153,172
MUX16
XXXX
MIL
Q5000TTL10K
CHIP00
SWGROUP A FOR DEMONSTRATION
TTL PWR
ITPWR
PW000
PW001
PW002

FIGURE 1

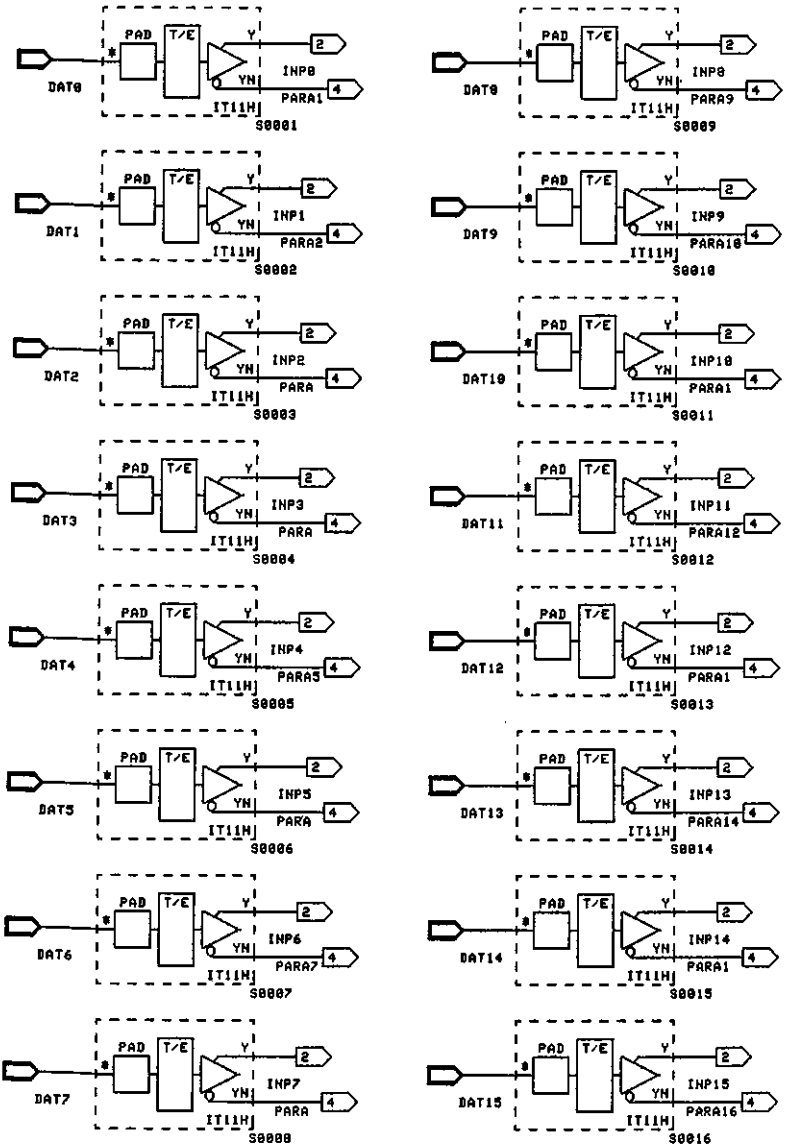

DAT0
PAD
T/E
Y
YN
2
INP0
PARA1
4
IT11H
S0001
DAT1
INP1
PARA2
S0002
DAT2
INP2
PARA
S0003
DAT3
INP3
S0004
DAT4
INP4
PARA5
S0005
DAT5
INP5
S0006
DAT6
INP6
PARA7
S0007
DAT7
INP7
S0008
DAT8
INP8
PARA9
S0009
DAT9
INP9
PARA10
S0010
DAT10
INP10
PARA1
S0011
DAT11
INP11
PARA12
S0012
DAT12
INP12
S0013
DAT13
INP13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

FIGURE 2

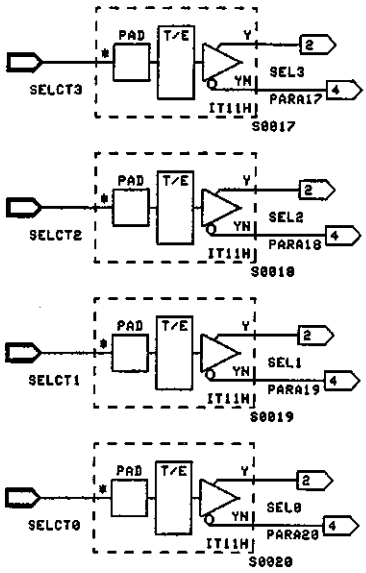


FIGURE 2 CONTINUED

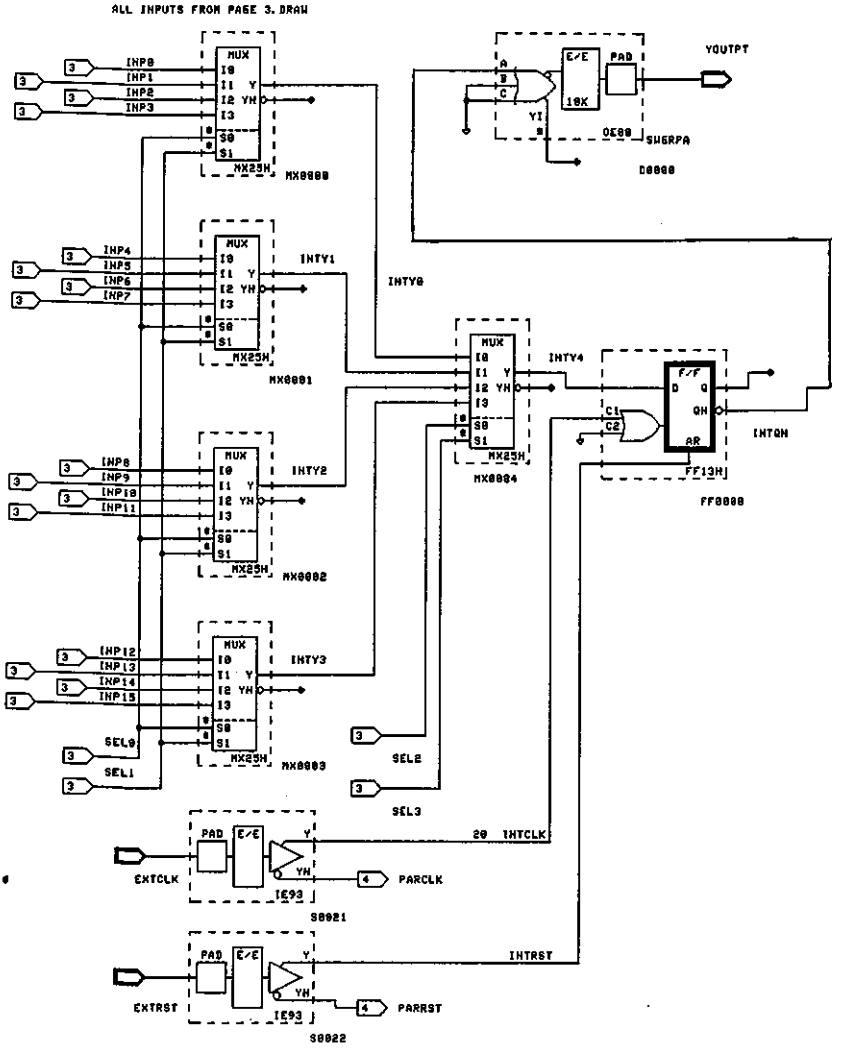

ALL INPUTS FROM PAGE 3.DRAW
INP0
INP1
INP2
INP3
MUX
I0
I1
I2
I3
S0
S1
Y
YH
MX25H
MX0000
INP4
INP5
INP6
INP7
INTY1
MX0001
INP8
INP9
INP10
INP11
INTY2
MX0002
INP12
INP13
INP14
INP15
INTY3
MX0003
SEL0
SEL1
SEL2
SEL3
INTY0
INTY4
MX0004
E/E
PAD
YOUTPT
A
B
C
YI
10K
OE00
SWERPA
D0000
F/F
D
Q
QH
AR
C1
C2
INTQH
FF13H
FF0000
EXTCLK
E93
S0021
PARCLK
INTCLK
EXTRST
S0022
PARRST
INTRST

FIGURE 3

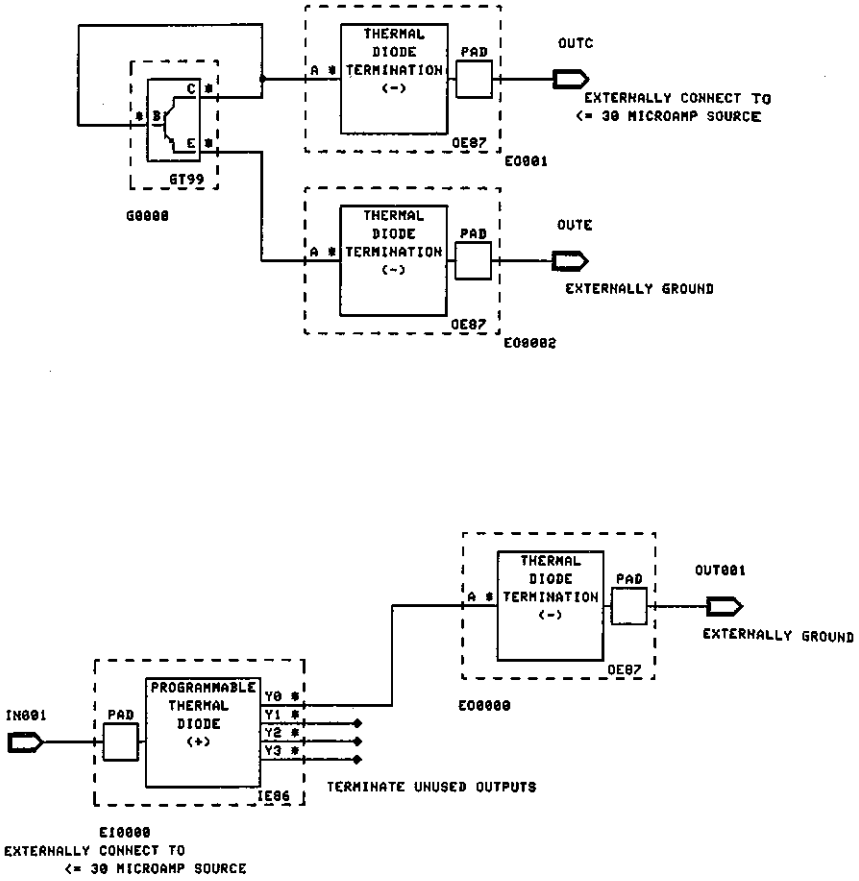

THERMAL
DIODE
TERMINATION
(-)
PAD
OUTC
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE
OE8Z
E0001
C
B
E
GT99
G0000
A
THERMAL
DIODE
TERMINATION
(-)
PAD
OUTE
EXTERNALLY GROUND
OE8Z
E0002
THERMAL
DIODE
TERMINATION
(-)
PAD
OUT001
EXTERNALLY GROUND
OE8Z
E0000
PROGRAMMABLE
THERMAL
DIODE
(+)
PAD
IN001
Y0
Y1
Y2
Y3
IE86
TERMINATE UNUSED OUTPUTS
EI0000
EXTERNALLY CONNECT TO
<= 30 MICROAMP SOURCE

FIGURE 4

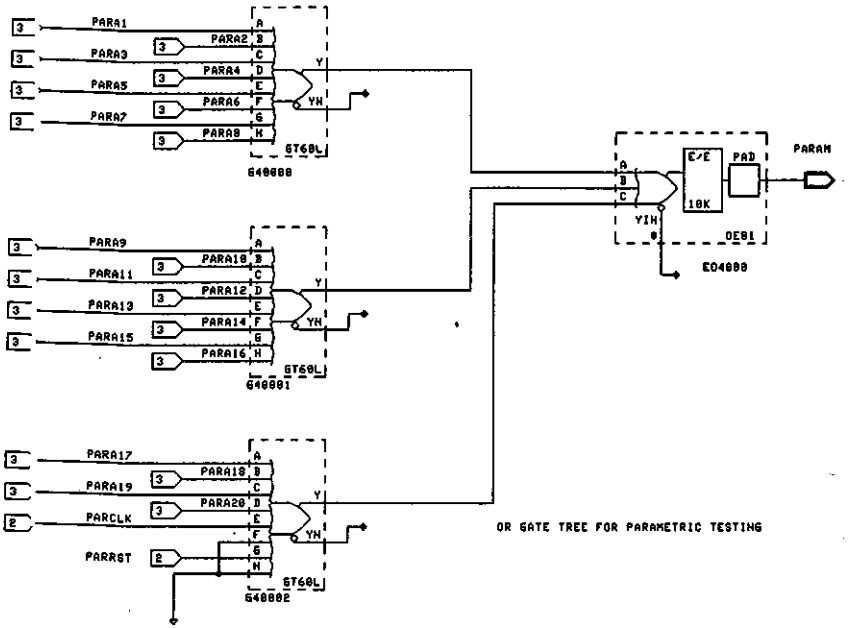

PARA1
PARA2
PARA3
PARA4
PARA5
PARA6
PARA7
PARA8
GT60L
G40000
PARA9
PARA10
PARA11
PARA12
PARA13
PARA14
PARA15
PARA16
GT60L
G40001
PARA17
PARA18
PARA19
PARA20
PARCLK
PARRST
GT60L
G40002
YH
C/E
PAD
10K
YIH
OEBI
E04000
PARAM
OR GATE TREE FOR PARAMETRIC TESTING

FIGURE 5

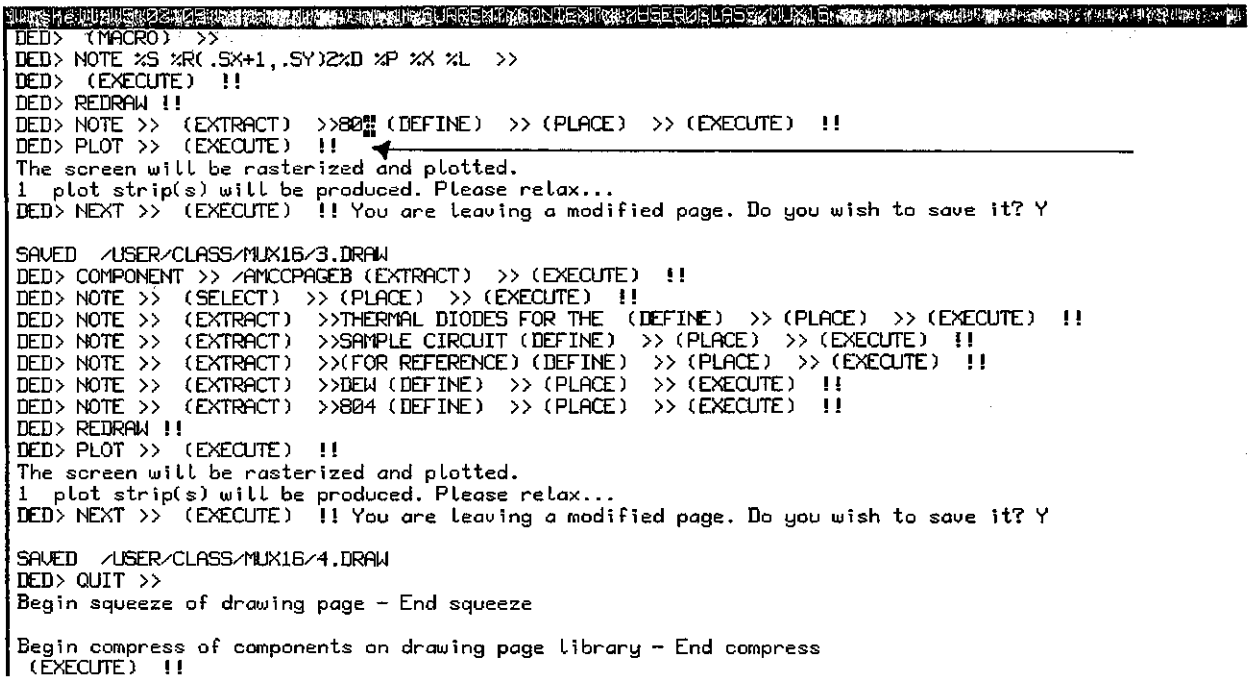


FIGURE 6

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
```

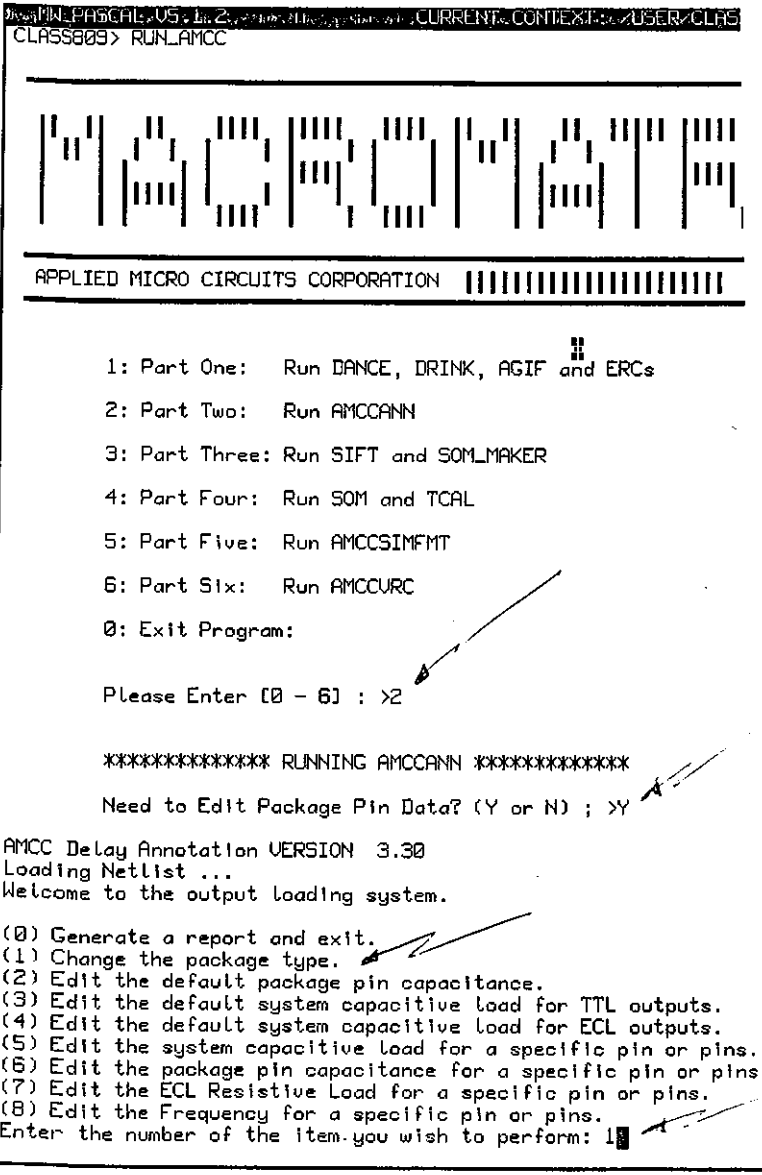
Or by using the Super-Shell

```
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.



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 V5.1.2 ... CURRENT CONTEXT /USER/CLASS/NW>
(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  1.00                                    *
*                                                           *
*************************************************************
   Netlister version number =                  700
   SDI version number =                        2.10
   Netlist generation date =                   20 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 =                               Q5000
   EWS Library Rev =                           809
   Macro Parameter family =                    Q5000
   The ARRAY type =                            Q5000T
   Chip Macro Name =                           Q5000TTTL10K
   Circuit Family =                            Q5000
   Circuit Technology =                        P
   ECL Level =                                 10K

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	Typ pf	Max pf
1	!DAT0	!S0001	!IT11H							
2	!DAT1	!S0002	!IT11H							
3	!DAT10	!S0011	!IT11H							
15	!DAT8	!S0009	!IT11H							
16	!DAT9	!S0010	!IT11H							
17	!EXTCLK	!S0021	!IE93		100.0					
18	!EXTRST	!S0022	!IE93							
19	!IN001	!EI0000	!IE86							
20	!OUT001	!EO0000	!OE87				6.0	3.80	4.40	4.90
21	!OUTC	!EO0001	!OE87			50.0	6.0	3.80	4.40	4.90
22	!OUTE	!EO0002	!OE87				6.0	3.80	4.40	4.90
23	!PARAM	!EO4000	!OE81		100.0		6.0	8.00	8.00	8.00
24	!SELCT0	!S0020	!IT11H							
25	!SELCT1	!S0019	!IT11H							
26	!SELCT2	!S0018	!IT11H							
27	!SELCT3	!S0017	!IT11H							
28	!YOUTPT	!D3000	!OE80		100.0	55.0	12.0	12.00	12.00	12.00
29	!ITGND	!GNT03	!ITGND							

this edit to default

default

previous edit

this edit

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
0AT0@MUX16/3:DAT0
3DAT1@MUX16/3:DAT1
3DAT10@MUX16/3:DAT10
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
3IN00!0@MUX16/4:IN00!1
3SELCT0@MUX16/3:SELCT0
3SELCT1@MUX16/3:SELCT1
3SELCT2@MUX16/3:SELCT2
3SELCT3@MUX16/3:SELCT3
3OUT00!0@MUX16/4:OUT00!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.

```
DLS 05.02.02 [illegible] CURRENT CONTEXT: /USER/CLASS/MUX16
** REPLACE **

NAME            BASE  POLARITY STRN TRC? SIGNAL_LIST

DAT0            BIN      +       OFF  ON  @MUX16/3:DAT0
DAT1            BIN      +       OFF  ON  @MUX16/3:DAT1
DAT10           BIN      +       OFF  ON  @MUX16/3:DAT10
DAT11           BIN      +       OFF  ON  @MUX16/3:DAT11
DAT12           BIN      +       OFF  ON  @MUX16/3:DAT12
DAT13           BIN      +       OFF  ON  @MUX16/3:DAT13
DAT14           BIN      +       OFF  ON  @MUX16/3:DAT14
DAT15           BIN      +       OFF  ON  @MUX16/3:DAT15
DAT2            BIN      +       OFF  ON  @MUX16/3:DAT2
DAT3            BIN      +       OFF  ON  @MUX16/3:DAT3
DAT4            BIN      +       OFF  ON  @MUX16/3:DAT4
DAT5            BIN      +       OFF  ON  @MUX16/3:DAT5
DAT6            BIN      +       OFF  ON  @MUX16/3:DAT6
DAT7            BIN      +       OFF  ON  @MUX16/3:DAT7
DAT8            BIN      +       OFF  ON  @MUX16/3:DAT8
DAT9            BIN      +       OFF  ON  @MUX16/3:DAT9
EXTCLK          BIN      +       OFF  ON  @MUX16/2:EXTCLK
EXTRST          BIN      +       OFF  ON  @MUX16/2:EXTRST
IN001           BIN      +       OFF  ON  @MUX16/4:IN001
SELCT0          BIN      +       OFF  ON  @MUX16/3:SELCT0
SELCT1          BIN      +       OFF  ON  @MUX16/3:SELCT1
SELCT2          BIN      +       OFF  ON  @MUX16/3:SELCT2
SELCT3          BIN      +       OFF  ON  @MUX16/3:SELCT3
OUT001          BIN      +       OFF  ON  @MUX16/4:OUT001
OUTC            BIN      +       OFF  ON  @MUX16/4:OUTC
OUTE            BIN      +       OFF  ON  @MUX16/4:OUTE
YOUTPT          BIN      +       OFF  ON  @MUX16/2:YOUTPT

TOTAL NUMBER OF PRIMITIVES =        671

DLS > FORMAT >>
FORMAT_SPEC>
```

FIGURE 14: DLS FORMAT WINDOW

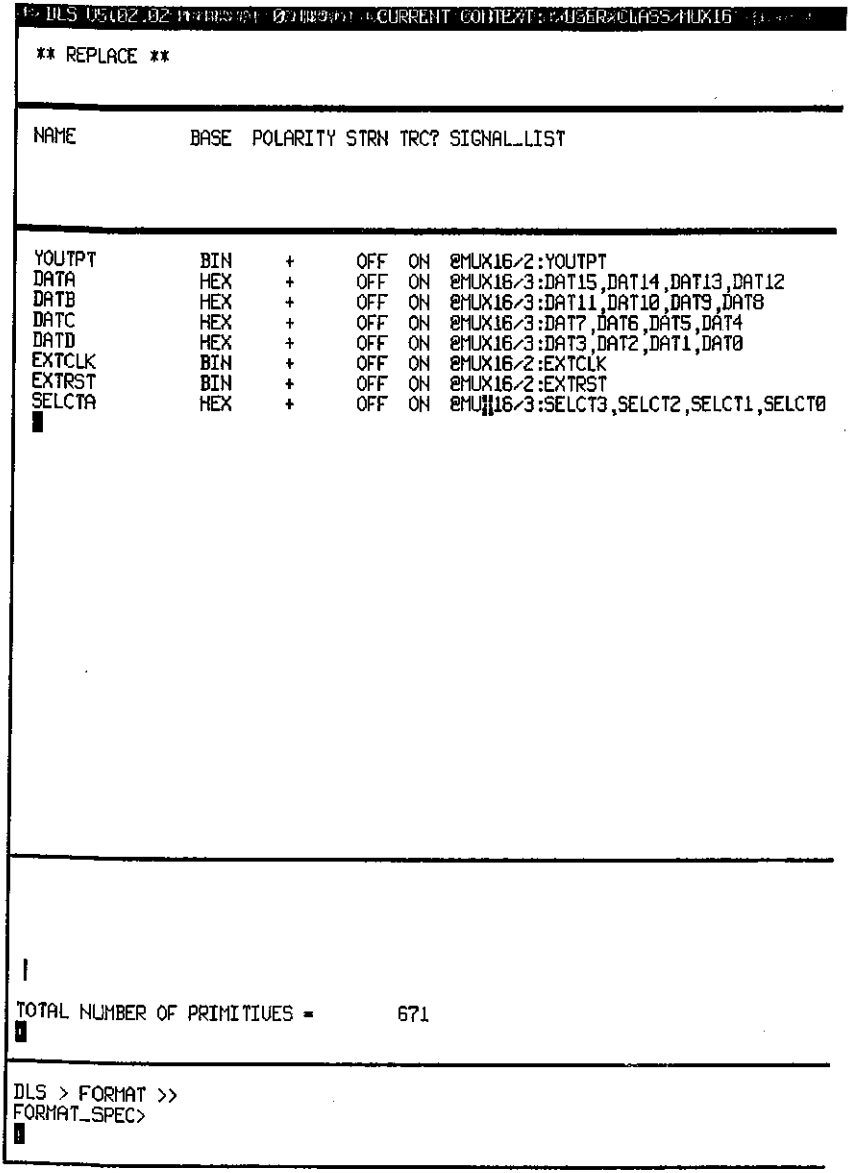


FIGURE 15: Edited DLS FORMAT WINDOW – HEX

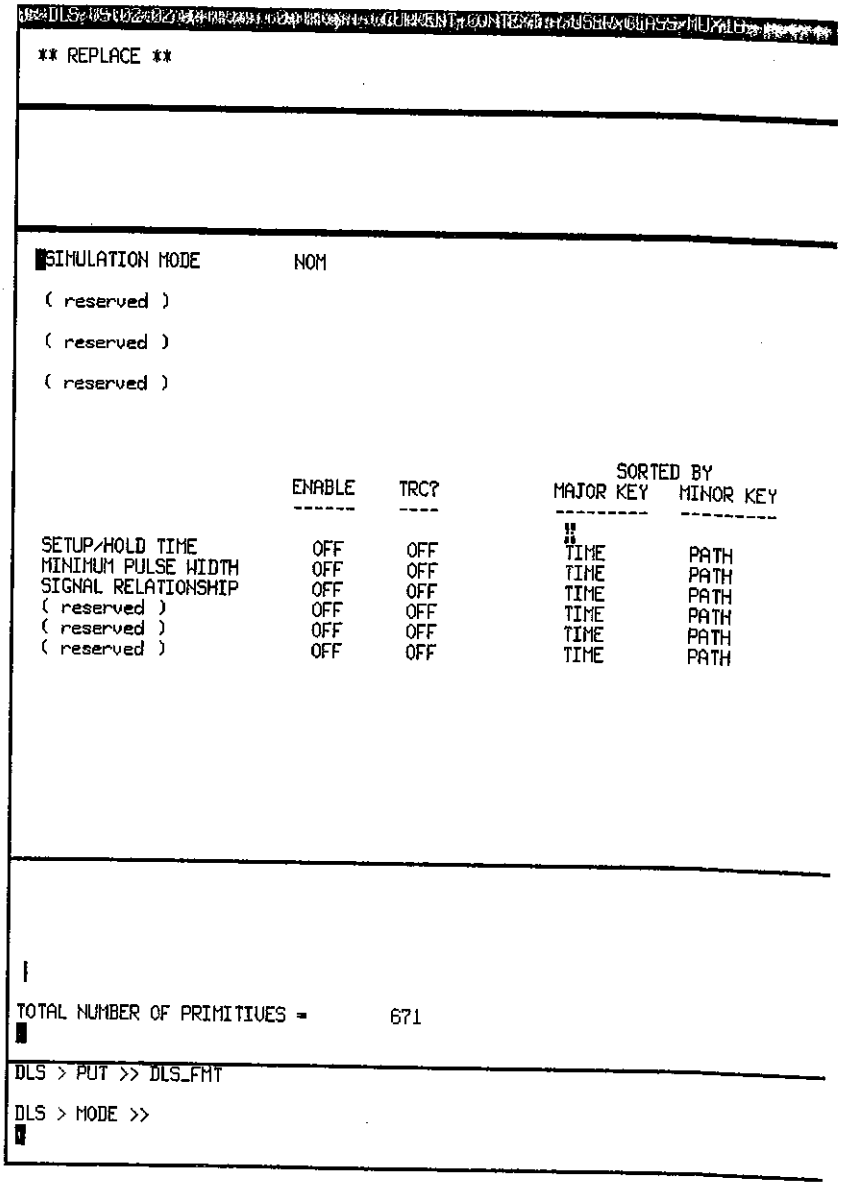


FIGURE 16: DLS MODE WINDOW – NOMINAL

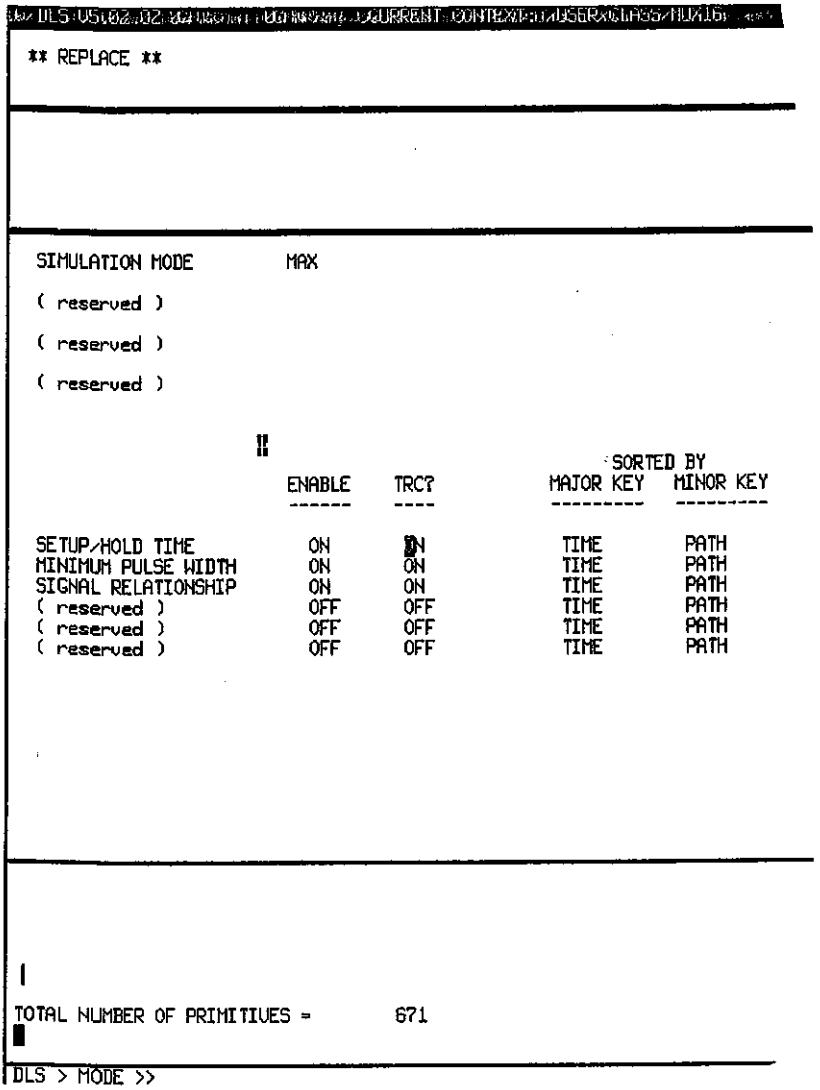


FIGURE 17: DLS MODE WINDOW – Edited for MAXIMUM

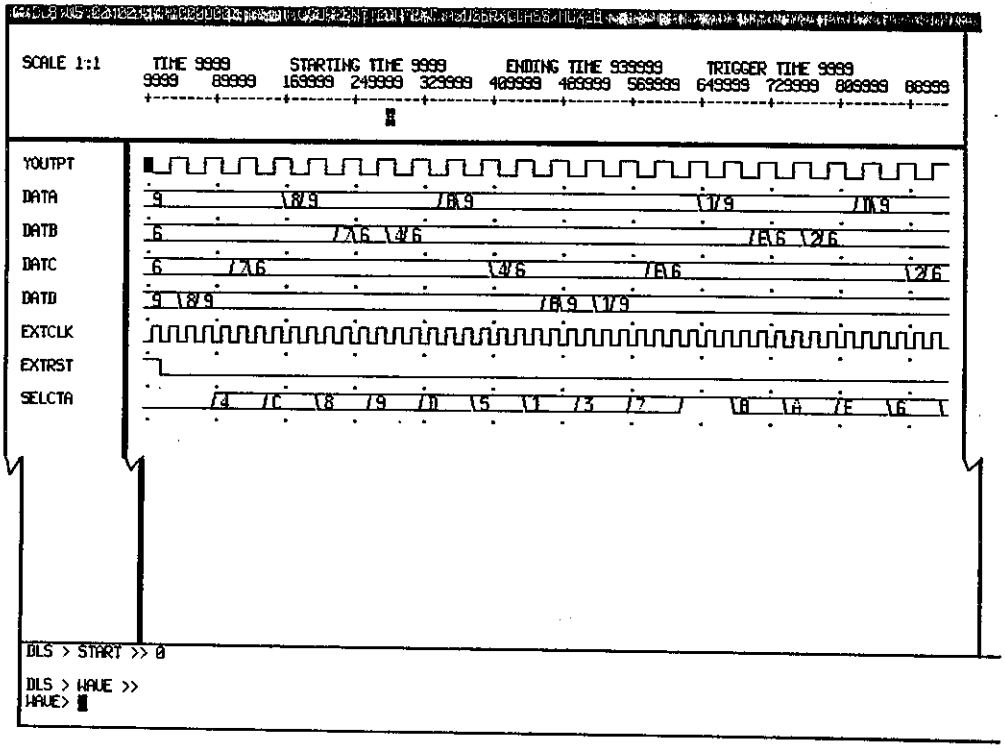
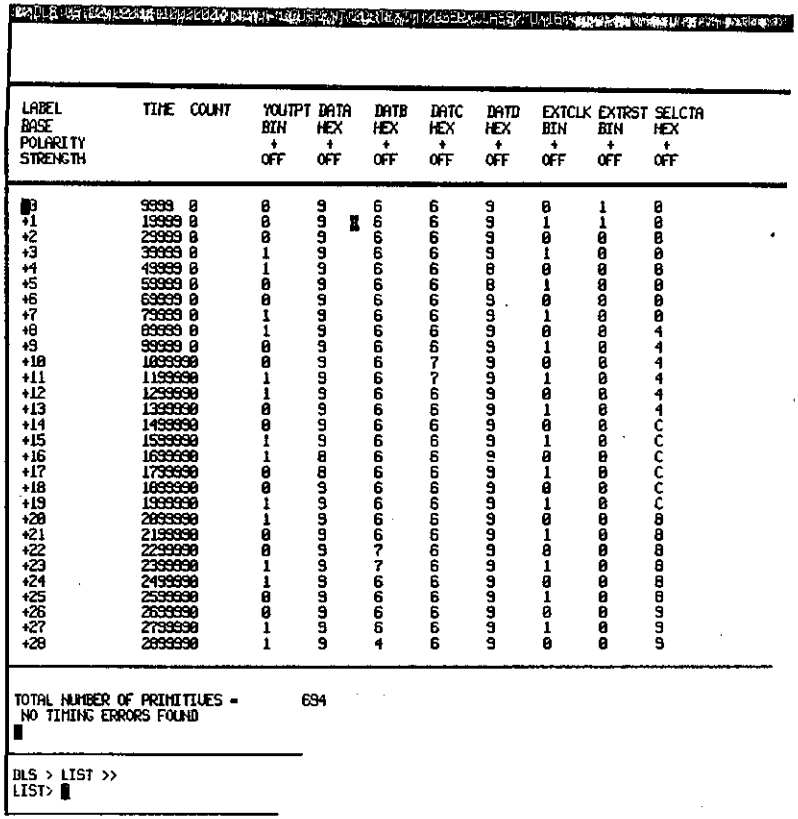


FIGURE 18: DLS WAVE - HEX FORMAT



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	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	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	VOUTPT BIN + OFF	DATA HEX + OFF	DAT8 HEX + OFF	DATC HEX + OFF	DAT0 HEX + OFF	EXTCLK BIN + OFF	EXTRST BIN + OFF	SELCTA HEX + OFF
Ø	9999	Ø	Ø	9	6	6	9	Ø	1	Ø
+1	19999	Ø	Ø	9	6	6	9	1	1	Ø
+2	29999	Ø	Ø	9	6	6	9	Ø	Ø	Ø
+3	39999	Ø	1	9	6	6	9	1	Ø	Ø
+4	49999	Ø	1	9	6	6	8	Ø	Ø	Ø
+5	59999	Ø	Ø	9	6	6	8	1	Ø	Ø
+6	69999	Ø	Ø	9	6	6	9	Ø	Ø	Ø
+7	79999	Ø	1	9	6	6	9	1	Ø	Ø
+8	89999	Ø	1	9	6	6	9	Ø	Ø	1
+9	99999	Ø	Ø	9	6	6	9	1	Ø	1
+1Ø	1Ø9999Ø		Ø	9	6	7	9	Ø	Ø	1
+11	1199999Ø		1	9	6	7	9	1	Ø	1
+12	1299999Ø		1	9	6	6	9	Ø	Ø	1
+13	1399999Ø		Ø	9	6	6	9	1	Ø	1
+14	1499999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+15	1599999Ø		1	9	6	6	9	1	Ø	Ø
+16	1699999Ø		1	8	6	6	9	Ø	Ø	Ø
+17	1799999Ø		Ø	8	6	6	9	1	Ø	Ø
+18	1899999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+19	1999999Ø		1	9	6	6	9	1	Ø	Ø
+2Ø	2Ø99999Ø		1	9	6	6	9	Ø	Ø	Ø
+21	2199999Ø		Ø	9	6	6	9	1	Ø	Ø
+22	2299999Ø		Ø	9	7	6	9	Ø	Ø	Ø
+23	2399999Ø		1	9	7	6	9	1	Ø	Ø
+24	2499999Ø		1	9	6	6	9	Ø	Ø	Ø
+25	2599999Ø		Ø	9	6	6	9	1	Ø	Ø
+26	2699999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+27	2799999Ø		1	9	6	6	9	1	Ø	Ø
+28	2899999Ø		1	9	4	6	9	Ø	Ø	Ø
+29	2999999Ø		Ø	9	4	6	9	1	Ø	Ø
+3Ø	3Ø99999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+31	3199999Ø		1	9	6	6	9	1	Ø	Ø
+32	3299999Ø		1	9	6	6	9	Ø	Ø	Ø
+33	3399999Ø		Ø	9	6	6	9	1	Ø	Ø
+34	3499999Ø		Ø	8	6	6	9	Ø	Ø	Ø
+35	3599999Ø		1	8	6	6	9	1	Ø	Ø
+36	3699999Ø		1	9	6	6	9	Ø	Ø	Ø
+37	3799999Ø		Ø	9	6	6	9	1	Ø	Ø
+38	3899999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+39	3999999Ø		1	9	6	6	9	1	Ø	Ø
+4Ø	4Ø99999Ø		1	9	6	4	9	Ø	Ø	Ø
+41	4199999Ø		Ø	9	6	4	9	1	Ø	Ø
+42	4299999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+43	4399999Ø		1	9	6	6	9	1	Ø	Ø
+44	4499999Ø		1	9	6	6	9	Ø	Ø	1
+45	4599999Ø		Ø	9	6	6	9	1	Ø	1
+46	4699999Ø		Ø	9	6	6	8	Ø	Ø	1
+47	4799999Ø		1	9	6	6	8	1	Ø	1
+48	4899999Ø		1	9	6	6	9	Ø	Ø	1
+49	4999999Ø		2	9	6	6	9	1	Ø	1
+5Ø	5Ø99999Ø		Ø	9	6	6	9	Ø	Ø	1
+51	5199999Ø		1	9	6	6	9	1	Ø	1
+52	5299999Ø		1	9	6	6	1	Ø	Ø	1
+53	5399999Ø		Ø	9	6	6	1	1	Ø	1
+54	5499999Ø		Ø	9	6	6	9	Ø	Ø	Ø
+55	5599999Ø		1	9	6	6	9	1	Ø	Ø
+56	5699999Ø		1	9	6	6	9	Ø	Ø	Ø
+57	5799999Ø		Ø	9	6	6	9	1	Ø	Ø

FIGURE 20: PARTIAL "SAVED" LIST FILE

USE:

DLS>LIST S SAM

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

$DATA_HEADERS
$TYPES
I/O
$FORMATS
TIME_VALUE
$TOTAL_COLUMNS$
10 23
$BASES$
0 0
$FIELDS 1
@MUX16/2:EXTCLK 1
$FIELDS 2
@MUX16/2:EXTRST 2
$FIELDS 3
@MUX16/3:SELCT3 3
$FIELDS 4
@MUX16/3:SELCT2 4
$FIELDS 5
@MUX16/3:SELCT1 5
$FIELDS 6
@MUX16/3:SELCT0 6
$FIELDS 7
@MUX16/3:DAT0 7
$FIELDS 8
@MUX16/3:DAT1 8
$FIELDS 9
@MUX16/3:DAT2 9
$FIELDS 10
@MUX16/3:DAT3 10
$FIELDS 11
@MUX16/3:DAT4 11
$FIELDS 12
@MUX16/3:DAT5 12
$FIELDS 13
@MUX16/3:DAT6 13
$FIELDS 14
@MUX16/3:DAT7 14
$FIELDS 15
@MUX16/3:DAT8 15
$FIELDS 16
@MUX16/3:DAT9 16
$FIELDS 17
@MUX16/3:DAT10 17
$FIELDS 18
@MUX16/3:DAT11 18
$FIELDS 19
@MUX16/3:DAT12 19
$FIELDS 20
@MUX16/3:DAT13 20
$FIELDS 21
@MUX16/3:DAT14 21
$FIELDS 22
@MUX16/3:DAT16 22
$FIELDS 23
@MUX16/2:YOUTPT 23
$ENDS
0          0100001001011001101001L
9999       0100001001011001101001L
19999      1100001001011001101001L
29999      0000001001011001101001L
39999      10000010010110011010011
49999      00000000010110011010011
59999      1000000001011001101001L
69999      0000001001011001101001L
79999      10000010010110011010011
89999      00010010010110011010011
99999      1001001001011001101001L
109999     0001001001111001101001L
119999     10010010011110011010011
129999     00010010010110011010011
139999     1001001001011001101001L
149999     0011001001011001101001L
159999     10110010010110011010011
169999     00110010010110011000011
179999     1011001001011001100001L
189999     0011001001011001101001L
199999     10110010010110011010011
209999     00100010010110011010011
219999     1010001001011001101001L
229999     0010001001011011101001L
239999     10100010010110111010011
249999     00100010010110011010011
259999     1010001001011001101001L
269999     0010011001011001101001L
279999     10100110010110011010011
289999     00100110010110001010011
299999     1010011001011000101001L
309999     0010011001011001101001L
319999     10100110010110011010011
329999     00110110010110011010011
339999     1011011001011001101001L
349999     0011011001011001101101L
359999     10110110010110011011011
369999     00110110010110011010011
379999     1011011001011001101001L
389999     0001011001011001101001L
399999     10010110010110011010011
409999     00010110010010011010011
419999     1001011001001001101001L
429999     0001011001011001101001L
439999     10010110010110011010011
449999     00000110010110011010011
459999     1000011001011001101001L
469999     0000011101011001101001L
479999     10000111010110011010011
489999     00000110010110011010011
499999     1000011001011001101001L
509999     0000111001011001101001L
519999     10001110010110011010011
529999     00001110000110011010011
539999     1000111000011001101001L
549999     0000111001011001101001L
559999     10001110010110011010011
569999     00011110010110011010011
579999     1001111001011001101001L
589999     0001111001011101101001L
599999     10011110010111011010011
609999     00011110010110011010011
619999     1001111001011001101001L
629999     0011111001011001101001L
639999     10111110010110011010011
649999     00111110010110011010001
659999     1011111001011001101000L
669999     0011111001011001101001L
679999     10111110010110011010011
689999     00101110010110011010011
699999     1010111001011001101001L
709999     0010111001011001111001L
719999     10101110010110011110011
729999     00101110010110011010011
739999     1010111001011001101001L
749999     0010101001011001101001L
759999     10101010010110011010011
769999     00101010010110010010011
779999     1010101001011001001001L
789999     0010101001011001101001L
799999     10101010010110011010011
809999     00111010010110011010011
819999     1011101001011001101001L
829999     0011101001011001101011L
839999     10111010010110011010111
849999     00111010010110011010011
859999     1011101001011001101001L
869999     0001101001011001101001L
879999     10011010010110011010011
889999     00011010010100011010011
899999     1001101001010001101001L
909999     0001101001011001101001L
919999     10011010010110011010011
929999     00001010010110011010011
939999     1000101001011001101001L
949999     0000101011011001101001L
959999     10001010110110011010011
969999     00001010010110011010011
979999     1000101001011001101001L
989999     0000001001011001101001L
999999     10000010010110011010011
```

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 =

                EESSSSDDDDDDDDDDDDDDDY
                XXEEEEEAAAAAAAAAAAAAAO
                TTLLLLTTTTTTTTTTTTTTTU
                CRCCCC#12345678911111T
                LSTTTT          #12345P
                KT321#                T

          TIME

9999            #1#####1##1#1##11##11#1##1#
19999           11#####1##1#1##11##11#1##1#
29999           #######1##1#1##11##11#1##1#
39999           1######1##1#1##11##11#1##11
49999           ##########1#1##11##11#1##11
59999           1#########1#1##11##11#1##1#
69999           #######1##1#1##11##11#1##1#
79999           1######1##1#1##11##11#1##11
89999           ##1##1##1##1#1##11##11#1##11
99999           1##1##1##1#1##1##11##11#1##1#
1#9999          ##1##1##1###1##11##11#1##1#
119999          1##1##1##1###1##11##11#1##11
129999          ##1##1##1#1#1##11##11#1##11
139999          1##1##1##1#1#1##11##11#1##1#
149999          ##1##1##1#1#1##11##11#1##1#
159999          1#1##1##1#1#1##11##11#1##11
169999          ##1##1##1#1#1##11##1#####11
179999          1#1##1##1#1#1##11##11#####1#
189999          ##1##1##1#1#1##11##11#1##1#
199999          1#1##1##1#1#1##11##11#1##11
2#9999          ##1####1##1#1##11##1#1##11
219999          1#1####1##1#1##11##11#1##1#
229999          ##1##1##1#1#1#111##11#1##1#
239999          1#1##1##1#1#1#111##11#1##11
249999          ##1##1##1#1#1##11##11#1##11
259999          1#1####1##1#1##11##11#1##1#
269999          ##1##1##1#1#1##11##11#1##1#
279999          1#1##1##1#1#1##11##11#1##11
289999          ##1##1##1#1#1####1#1##11##1
299999          1#1##1##1#1#1####1#1##11##1#
3#9999          ##1##1##1#1#1##11##11#1##1#
319999          1#1##1##1#1#1##111#11#1##11
329999          ##1#1#1##1#1#1##11#1#1##11#
339999          1#1#1#1##1#1#1##11#1#1##1#1
349999          ##1#1#1##1#1#1##11#1#1##1#1
359999          1#1#1#1##1#1#1##11#1#1##11#
369999          ##1#1#1##1#1#1##11#1#1##11#
379999          1#1#1#1##1#1#1##11#1#1##1#1
389999          ##1#1#1##1#1#1##11#1#1##1#1
399999          1##1#1##1#1#1##11#1#1##11#
4#9999          ##1#1#1##1##1##11#1#1##11#
419999          1##1#1##1##1##11#1#1##1#1#
429999          ##1#1#1##1#1#1##11#1#1##1#
439999          1##1#1##1#1#1##11#1#1##11#
449999          ######1##1#1#1##11#1#1##11#
459999          1#####1##1#1#1##11#1#1##1#1
469999          ######111#1#1##11#1#1##1#1
479999          1#####111#1#1##11#1#1##11#
489999          ######1##1#1#1##11#1#1##11#
499999          1#####1##1#1#1##11#1#1##1#1
5#9999          ######111#1#1##11#1#1##1#1
519999          1####111#1#1##11#1#1##11#

529999          ####111#####11##11#1##11#
539999          1####11#####11##11#1##1#1
549999          #####111##1#1##11#1#1##1#1
559999          1####111##1#1##11#1#1##11#
569999          ###111#1##1#1##11#1#1##11#
579999          1##111#1##1#1##11#1#1##1#1
589999          ###111#1##1#111#1#1#1##1#1
599999          1##111#1##1#111#1#1##11#1#
6#9999          ###111#1##1#1##11#1#1#11#1
619999          1##111#1##1#1##11#1#1##1#1
629999          ###111#1##1#1##11#1#1#1#1#
639999          1#1111#1##1#1##11#1#1##11#
649999          ###111#1##1#1##11#1#1#1##1
659999          1#1111#1##1#1##11#1#1#####
669999          ###1111#1##1#1##11#1#1##1#
679999          1#1111#1##1#1##11#1#1#1##11
689999          ##1#111#1##1#1#1##11#1##11
699999          1#1#111#1##1#1##11#1#1##1#
7#9999          ##1#111#1##1#1##1111#1##1#
719999          1#1#111#1##1#1##1111#11##11
729999          ##1#111#1##1#1##11#1#1##11
739999          1#1#111#1##1#1##11#1#1#1#1#
749999          ##1#1#1#1##1#1##11#1#1#1#1#
759999          1#1#1#1#1##1#1#1##11#1#1##11
769999          ##1#1#1#1##1#1#1##11#1#1#11
779999          1#1#1#1#1##1#1#1##1#1#1##1#
789999          ##1#1#1#1##1#1#1##11#1#1#1#
799999          1#1#1#1#1##1#1#1##11#1#1##11
8#9999          ##111#1#1##1#1#1##11#1#1##1
819999          1#111#1#1##1#1#1##11#1#1#1#
829999          ##111#1#1##1#1#1##11#1#111#
839999          1#111#1#1##1#1#1##11#1#1111
849999          ##111#1#1##1#1#1##11#1#1##1
859999          1#111#1#1##1#1#1##11#1#1#1#
869999          ##1#1#1#1##1#1#1##11#1#1#1#
879999          1##1#1#1#1##1#1#1##11#1#11#
889999          ##1#1#1#1##1#1#1####1#1##11
899999          1##1#1#1#1##1#1#1####1#1#1#
9#9999          ##1#1#1#1##1#1#1##11#1#1#1#
919999          1##1#1#1#1##1#1#1##11#1#11#
929999          ####1#1#1##1#1#1##11#1#11#
939999          1###1#1#1##1#1#1##11#1#1#1#
949999          ####1#1#1#1#1#1##11#1#1#1#
959999          1###1#1#1#1#1#1##11#1#1##1
969999          ####1#1#1##1#1#1##11#1#11#
979999          1###1#1#1##1#1#1##11#1#1#1#
989999          #####1#1##1#1#1##11#1#1#1#
999999          1#####1#1##1#1#1##11#1#11#
```

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