Section 7:

Installation and Operations Guide

Table of Contents

7-1 INTRODUCTION

This section describes AMCC design methodology for macro conversion and simulation on a Daisy. AMCC assumes that the customer has:

1. A good understanding of how to use the Daisy. AMCC offers a seminar for new users which covers basic operations on the DAISY EWS and the use of AMCC MacroMatrix support software, including AGIF, AMCCERC, AMCCANN, AMCCVRC, AMCCSIMFMT and the AMCCFILUTL. Consult your local AMCC sales office.

2. Read the following:

   AMCC Application Note 1.DNIX (1.ACE pending)
   Volume II, Section 2.
   AMCC Application Note 2.DNIX (hierarchy)
   Volume II, Section 2.
   AMCC Application Note 3.DNIX (nesting)
   Volume II, Section 2.
   AMCC EWS Schematic Rules and Conventions,
   Volume II, Section 3.
   AMCC Vector Submission Rules and Conventions,
   Volume II, Section 4.
   AMCC Design Submission: DAISY
   Volume II, Section 6.
   AMCC MacroMatrix User's Guide
   Volume II, Section 8.

3. Created a schematic that follows AMCC schematic rules. Refer to EWS Schematics Rules and Conventions in Section 3 for generic rules and refer to this document, Section 7-6, for DAISY-specific rules.

4. Fully simulated the circuit on the Daisy following the procedures outlined in Section 4, Vector Submission Rules and Guidelines.

7-2 INSTALLATION OF RELEASE FLOPPY DISK

7-2-1 NEW LIBRARY; NEW RELEASE

Use the following syntax to install the release floppy:

```
MOUNT /F
COPY /F TO / -DE -B
DIS /F
```

After the release floppy disks are installed, type one of the following files to get current information on changes or additions to this release.

```
/AMCC/Q3500_LIBS/TYPE_ME.INFO
/AMCC/Q5000_LIBS/TYPE_ME.INFO
/AMCC/Q14000_LIBS/TYPE_ME.INFO
/AMCC/Q20000_LIBS/TYPE_ME.INFO
```

All previously installed macro patches (custom macros, etc.) must be re-installed after the new library has been installed. (See Section 7-2-3.)

7-2-2 NEW PATCHES WITH AN EXISTING LIBRARY; EXISTING PATCHES

Use the following to install a patch floppy:

The script "INSTALL" is included in every patch release. The script will append any patched macro(s) to an existing PATCH.CLIB, PATCH.VIF and PATCH.SLIB so previously patched macros are not disturbed or lost. All of these files are added to the system under //AMCC/Qxxxx_LIBS. An updated Qxxxx.FPM is also installed under /AMCC/PARAMS.

PATCH.VIF is the file that is used in DED2 and PATCHxxxx.LIB is the file that is used in ACE. If there is no PATCH.VIF file available under /AMCC/Qxxxx_LIBS, the script will automatically create one.

The PATCHxxxx.LIB files are sent with a date extension (i.e., PATCH0927.LIB). These .LIB files will keep building as additional patches are added.

ACE: The .LIB files will be automatically added to the ACE menu when ACE is involved.

At the end of the INSTALL script, LIB_INV of the PATCH.CLIB and PATCH.VIF files are produced. These inventories show the macros added to the existing library.
The file "TYPE_ME.INFO" contains instructions on loading the patch files:

1. Type in: MOUNT /F
2. Type in: /F/INSTALL

7-2-3 NEW LIBRARY, OLD PATCHES

On receiving a new library, when custom macro patches are in use:

1. Install the library
2. Re-install the patches

You need to retain the Qxxx.FPM file located in /AMCC/PARAMS. Once the new library is installed, this Qxxx.FPM file must be re-installed after the patch floppies are installed.

7-3 THIS RELEASE

This release is dated (809) and incorporates all previous release updates, plus an (809) bipolar or (810) BiCMOS macro library update. AMCC supports the Daisy 5.02 DNIX using SPARC files. AMCC also supports DEDII and ACE on the 5.02 operating system. DEDII does not yet support the "bold box" AMCC normally uses around latches, flip/flops and MSI macros.

DAISY is supported on 8 and on 5 1/4 inch floppy disks. Design submission should be via 8 or 5 1/4 inch floppy disks. A DAISY LOGICIAN format 1/2 inch magnetic tape can be accepted.

Macros have been deleted in this release as per MacroMatrix News # 9. Refer to Volume I, Section 6-index for a list of the macros included in this release. New macros are marked by an * in the index.

A new MacroMatrix package is being released. It includes an expanded AMCCERC, including an automated I/O list report and cross reference report. The ERCs include the ability to check for excessive loading against a derated fan-out load limit, and the ability to count and check the power/ground requirements for simultaneously switching outputs. Added in this release is the user-interface AMCCANN that allows the description of package pin capacitance and system loading as well as the commentary toggle frequency and unusual ECL resistance terminations.

The output report files are: AMCCIO.LST, AMCCPKG.LST, AMCCXREF.LST and AMCCERC.LST located in the /ERC subdirectory beneath the top of the design. The ERC and VRC error messages are documented in Volume II, Section 8.

AMCCSIMFMT simulation format program requires an $OUTPUTS section output file. This $OUTPUTS section must request an output file which contains all primary input, all primary output, all bidirectional I/O, and all 3-state enable or bidirectional control enable signals. Excluded are the thermal diodes, AC monitors and VB generator signals. Documentation on the AMCCSIMFMT program is in Appendix A for this section. The file may be sampled (for functional, at-speed and AC test simulations) or PRINT_ON_CHANGE (for at-speed and AC test simulations).

AMCCFILUTL file utilities program provides a comment stripper as one of its options. The customer creating a data input file for simulation can add comments to that file to document the tests being performed. The utility program will strip out these comments and a second pass will strip out blank lines. Comments should be bracketed within /* and */. Other functions are also performed and are documented in Appendix B for this section.

AMCCVRC vector rules checker program provides a screening for some of the common errors committed during vector development.
All sampled, maximum worst-case functional, AC test and parametric AMCCSIMFMT output files that are to be submitted must have been processed through the vector rules checking software, AMCCVRC. AMCCVRC is documented in Appendix B of Section 8. AMCCVRC produces AMCCVRC.LST, located in the calling directory. AMCCVRC requires a signal analysis file describing signal-clock relationships (clocked circuits only).

7-4 WARNING

PRIOR TO START OF DESIGN, PLEASE CALL AMCC TO INQUIRE WHETHER YOU ARE IN POSSESSION OF THE MOST CURRENT RELEASE OF LIBRARIES AND MACROMATRIX SOFTWARE, INCLUDING THE ENGINEERING RULES CHECKS!

7-5 SHELL SCRIPTS

If you want to use AMCC-provided shell scripts, they are included in the release floppy disk. The DAISY-DNIX-ACEII and ACE AMCC shells are documented in Design Methodology in Section 2. To be properly set up to use the scripts, the user needs a loginfile as documented in Section 2.

7-6 AMCC EWS-SPECIFIC SCHEMATIC RULES - INTRODUCTION

This section contains some of the general guidelines and restrictions which you must follow. Generic information is found in Volume II, Section 3, EWS Schematic Rules and Conventions. By following AMCC schematic guidelines, you may be able to avoid problems in creating the circuit schematic or in later operation of the various software support programs.

The guidelines are as follows:

1. Only AMCC-created components can be used on the schematic, except for standard Daisy connectors and terminators.
   - o To update a drawing page:
     UPDATE ERASE REPLACE COPY ALL{DEF}/*
     {EXTRACT}{EXECUTE}
     - erases parameters so be careful
     UPDATE C{MACRO}{EXECUTE}
     UPDATE S{MACRO}EXECUTE
     - ignore error messages until all three are complete

2. If you are designing using HIERARCHICAL OR NESTED blocks, there are additional AMCC guidelines which must be followed. Refer to Section 2, APNOTE 2.DNIX for hierarchy-only and to Section 2, APNOTE 3.DNIX for nested-hierarchy rules and guidelines and an example circuit.

   These are in addition to the rules listed in Section 3.

3. The user is warned against modification in any way of the libraries or the Back-Annotation file. This could result in the creation of a circuit that cannot be manufactured by AMCC.

4. The RAM10 and RAM20 macros are considered to be nested macros by the DAISY software. There is a nested file, /AMCC/Q5000_LIBS/Q5000_RAM.NEST, that already exists to handle this nested structure. If RAM macros are used in a nested design, copy this file to your design area and add to it.

   If you edit it where it is, the next boot will erase your edits. Copy the edited file back prior to running DAISY software (PREFERRED METHOD) or adjust the PROFILE file PROQxxx to point to the new location. If edited, the xxx.NEST file MUST be submitted at design submission.

5. The character limit for the CONTS parameter (bundles) is 64 characters. This is a DAISY limit. When more than 64 characters are needed, use the "CONTS" file and indirect mode for the parameter CONTS command:

   SET CTL PARAMETER TO "B"
   SET CONTS PARAMETER to (TAG)

   where (TAG) is a unique label (NOT the name) for the bundle. (TAG) and signal names are entered into the /CONTS file located at /PROJECT/CONTS. Copy this file into your work area for submission. If edited, the /PROJECT/CONTS file MUST be submitted at design submission.

6. The directory name for a design is limited to 10 characters. This is a DAISY-TCAL limit. For deeply nested structures, keep block names short. AMCC is investigating this problem.

7-7 STEPS IN AMCC LOGIC ARRAY DESIGN ON THE DAISY EWS

All of the following steps are required to ensure that a schematic does not contain a circuit which cannot be manufactured by AMCC. The following gives a brief summary of the procedures discussed in detail in Section 2, EWS Design Methodology.

0. Log on, check the current context, check space, clean directory prior to starting a design. If you are going to make changes in the PROFILE files, edit the PROQ<lib> file and not the PROFILE file itself. (PROQ5000, PROQ1400, PROQ2000)

1. Select library by programming the loginfile. (See Section 2.) Make certain that you have set-up for the correct library.

2. Enter DED 1, DED2 or ACE. Follow AMCC EWS Schematic Rules and Conventions or refer to Application Note 1.DNIX in Section 2 of this volume. (Application Note 1 uses DED, can be used as a guide for DED2.)

3. A special border must be used on AMCC designs. The name of the border is /AMCCPAGEB. (If you are using ACE, the border is automatic.) After this border is placed on the page, add notes to identify the design and the page itself.

4. Every circuit requires a chip macro. Parameterize it and wire it as defined in the AMCC EWS Schematic Rules and Conventions and in Section 6-5, special macro section of the macro library documentation.

5. DANCE each drawing page one at a time or all at once.

```
DANCE -T -N -ERR -M3   <----all at once
DANCE x -M3            <----one page x.DRAW
ADANCE x -M3           <---- ACE users
```

   All pages of the circuit schematic MUST pass through DANCE without errors. The following errors may be considered warnings and as such may be safely ignored:

   (a) Error 71 - Caused by border component
   (b) Error 11 - Caused by border component

   DANCE error files are n.DFR.

6. DRINK the entire design from the top of the design tree. A partial circuit can be processed with DRINK. Make sure that nested designs have a properly edited /AMCC/CONFIG/NEST file. Refer to APNOTE 3.DNIX.

```
DRINK -T -M3
```

DRINK error file is TREE.DFR.

The AMCC DANCE-DRINK shell can be used to run both DANCE and DRINK:

```
RUN_DD
```

THE AMCC super-shell can also be used to run DANCE and DRINK, along with AGIF, ERCs, and Front-Annotation:

```
RUN_AMCC
1
```

In both cases, the error files are DANCE.ERR and DRINK.ERR. Remove circuit errors detected by these files BEFORE proceeding. (Unless you are compiling a partial circuit - then remove appropriate errors and ignore page interconnects not yet captured.) The shell scripts will now halt on any DANCE or DRINK errors. If you are running a partial circuit, use the individual DANCE and DRINK commands.

7. Run SING-TO-AGIF netlist reformat to change the DAISY netlist into the AMCC interface format required for the ERC and Front-Annotation software. The resulting CIRCUIT.SDI file must be submitted in the design submission package (on floppy; do not print it out). Use the AMCC shell:

```
RUN_AGIF
```

8. Run the AMCC MacroMatrix ERC software by changing to the /ERC subdirectory and then typing AMCCERC.

```
CD ERC
AMCCERC
```

By running the AMCC shell script RUN_ERC (or the super-shell above) there is no need to change to the /ERC subdirectory.

```
RUN_ERC
```

Files created are AMCCERC.LST, ERC reports and error messages, AMCCIO.LST, the automated I/O list, and AMCCXREF.LST, the cross-reference list for any renamed items.

9. Run Front-Annotation. This is run from the same subdirectory by typing:

   AMCCANN

   The program will ask you if you need to enter or change the package type, package pin capacitance or system load values via:

   "NEED TO EDIT PACKAGE PIN DATA? (Y or N):"

   response "N" will cause it to use default values or those you have previously defined. AMCCANN can be re-run as many times as desired.

   By running the AMCC shell script RUN_ANN there is no need to change to the /ERC subdirectory.

   RUN_ANN

   The files created are: FNTMIN.DSY, FNTNOM.DSY, and either FNTMIL.DSY or FNTCOM.DSY, located at the calling node path. It also creates OUTPUT.DLY and AMCCPKG.LST, located in the /ERC subdirectory.

10. Run SIFT. This is run from the top of the design tree. The designer can use the AMCC shell script, RUN_SIFT xxx, where xxx = MIN, NOM, COM (Bipolar), COM4 (BiCMOS, -4.5V) or COM5 (BiCMOS, -5.2V or +5V) or MIL.

    RUN_SIFT MIL RUN_SIFT COM RUN_SIFT COM4
    RUN_SIFT MIN RUN_SIFT NOM RUN_SIFT COM5

TECHNOLOGY	TIMING LIBRARY	DELAY ANNOTATION	PRODUCT GRADE	ECL POWER SUPPLY
BIPOLAR	MIN	FNTMIN.ews	any	any
	NOM	FNTNOM.ews	any	any
	COM	FNTCOM.ews	COM	any
	MIL	FNTMIL.ews	MIL	any
BICMOS	MIN	FNTMIN.ews	any	any
	NOM	FNTNOM.ews	any	any
	COM4	FNTCOM.ews	COM	-4.5V
	COM5	FNTCOM.ews	COM	-5.2V; +5V
	MIL	FNTMIL.ews	MIL	any

PRODUCT_GRADE is only COM or MIL

11. Run SOM_MAKER. This will build a default FMT and an initial SOM_MCF.SING file, defined for T=0, and then call up the TEC text editor so the designer can complete the simulation control file definition. An $OUTPUTS section will be required. Again, there are so many steps (see Section 2 on Design Methodology for listings of the shell scripts) that the user should use the shell:

```
RUN_SMAKER
```

SIFT and SOM_MAKER can be run under the super shell by typing a "3" on the reappearance of the menu:

```
RUN_AMCC
1       <---- DANCE, DRINK, AGIF, ERC, FA
2       <----- AMCCANN
3       <---- RUN_SIFT, RUN_SMAKER
MIL          <---- it will prompt you
```

12. Using TEC, edit the SOM_MCF.SING file until all input data are entered. An $OUTPUTS section must be created which defines a sampled output file composed of all primary inputs, all primary outputs and all 3-state enable and bidirectional enable signals in that order.

```
TEC <filename>
```

NOTE: DO NOT USE DATA OR INITIALIZE STATEMENTS WITH A FUNCTIONAL, AC-TEST OR PARAMETRIC SIMULATION.

The super shell step 3 ends with the SOM_MCF.SING file open under the TEC editor and the editor screen active. When you exit the editor you are still in the super-shell.

BIDIRECTIONAL MACROS:

To correctly simulate a design with bidirectional using an $INPUTS file (as opposed to $SIGNAL_GENERATORS, each bidirectional signal must have a type field of CON. An example $INPUTS section would look like:

```
$INPUTS
FILE /USER/CLASS/ALPHA1/INPUTS ->
@ALPHA1/1:INP1, INP2, INP3,
@ALPHA1/2:BIDI1::CON, BIDI2::CON, BIDI3::CON,
@ALPHA1/2:BIDI4::CON;
```

The data field would contain "#" for each bidirectional that was behaving as an output and a "1" or "0" when it was behaving as an input. The designer must be careful to properly switch the direction.

SIMULTANEOUSLY SWITCHING OUTPUTS:

AMCCVRC rules require the only 8 ECL and 8 TTL or 16 of either bidirectional and 3-state outputs change directional mode or state in any one vector. Enable busses must be degated for test. The 8 ECL, 8 TTL limit applies to mixed ECL/TTL circuits regardless of power supply. For 100% TTL or 100% ECL, the limit is 16 per vector.

13. Optional: Outside of the super-shell, use TEC to create an input data file (useful when test vectors already exist for a circuit).

14. Run SOM to link the information together:

```
SOM som-ctl-file -M3 -L SOM.ERR
```

where SOM.ERR will contain a copy of the SOM control file with any errors flagged. Data errors will be flagged during DLS/DTV execution.

15. Run TCAL to incorporate the Front-Annotation data into the simulation model. TCAL stands for timing calculation.

```
TCAL -MCF FNTxxx.DSY -M3
```

where xxx = NOM, MIN, COM or MIL. The FNTxxx.DSY files are created by the Front-Annotation program.

The shell RUN_SOM can also be used:

```
RUN_SOM som-ctl-file  tcal-file
```

The super shell can also be used to run SOM:

```
RUN_AMCC
 1                    <----- DANCE, DRINK, AGIF,
ERC
 2                    <----- AMCCANN
 N                    <----- no changes
 3                    <----- sift
 product_grade
 4                    <-----  SOM
 som-ctl-file         <----- you will be prompted
 tcal-file            <----- you will be prompted
 0                    <----- exit the shell
```

16. Run the Daisy Logic Simulator by typing in:

```
DLS
GET FMT
```

to load the previously created format file.

```
FO
```

to edit for whatever you want to see in WAVE, LIST.

```
{ENTER}
```

to return to DLS

```
MODE
```

to edit for the proper mode (MIN, NOM, MAX) worst-case multiplier within the timing library.

When using a MIL or COM library, set mode to MAX. When using a MIN library, set the mode to MIN. If you wish to run a nominal simulation, set the mode to NOM. (Nominal simulations are not submitted.) When you want the timing checks run, turn them on. All submitted simulations must be run with the timing checks on.

{ENTER}
to return to DLS

Now run the simulation (VIEW, RUN, START, LIST, WAVE, etc.) as desired. AMCC would prefer a submit shell script be written and submitted with the design submission. Refer to the Design Methodology section in this volume for a sample RUN_DLS shell for functional sampled and a RUN_ASDLS shell for at-speed sampled. These are user-created and tailored to your submission.

For Functional simulation use:

```
VIEW 9999 10000
```

17. Run AMCCSIMFMT. Follow the directions in Section 7-8 and in Appendix A of Section 7. AMCCSIMFMT requires the file created under control of the som-ctl-file $OUTPUTS section. All simulation $OUTPUTS file, PRINT_ON_CHANGE or sampled, may be processed through AMCCSIMFMT. Only AMCCSIMFMT output files are submitted.

    AMCCSIMFMT may also be run under RUN_AMCC as option 5. The version number will appear followed by the AMCCSIMFMT prompts.

18. Run AMCCVRC. Follow the instructions in Section 8, Appendix B, AMCC Vector Rules Checker. The AMCCVRC software requires a user-created signal analysis file, the CIRCUIT.SDI file, and the AMCCSIMFMT output file to be checked. Only sampled files for the maximum worst-case simulations for AC test and Functional simulations must be checked and the AMCCVRC.LST reports for those files must be submitted.

    AMCCVRC may also be run under RUN_AMCC as option 6. The system will respond with the AMCCVRC version number and the AMCCVRC prompts.

7-8 USE OF AMCCSIMFMT

AMCCSIMFMT is an AMCC tool to aid in the use of DLS. The purpose of AMCCSIMFMT is to reformat the sampled output created by the DLS $OUTPUTS section and add the option of merging as many input files as the user wishes.

This program generates an output file that looks like a TEGAS file. The output from AMCCSIMFMT will be fed to the TESTFMTR program at AMCC. DO NOT EDIT THE OUTPUT OF THE AMCCSIMFMT PROGRAM. TO COMMENT THE VECTORS, COPY TO A .TXT FILE. AMCCSIMFMT allows the user to have 50 signals on a page with spacing and over 120 without spacing before column wrap on a 132 character printer.

NOTE: if the simulation time runs into the millions, the time count inside the list function will truncate the time after 6 digits. For example, one hundred thousand will appear the same as one million on the printout.

NOTE: when an input data file is being used, the file may have imbedded comments enclosed in /* and */ (PASCAL-like notation). AMCCFILUTL is used to strip the comments to create an execution file. This removes the need to comment a version of the AMCCSIMFMT output file. The prompts are:

1. Run DLS using the correct timing step for the array from the functional (sampled) simulation.
   VIEW 9999 10000 or as required

2. The simulation control file must have an $OUTPUTS section which lists all primary inputs, all primary outputs, and all 3-state enables and bidirectional enables.

3. QUIT DLS.

4. Run AMCCSIMFMT.

```
DLS                     simple run
GET DLS_FMT             pre-edited format/mode
VIEW 9999 10000         function step
RUN 1000000             limit for vector set
QUIT                    quit DLS - output file
N                             already exists
AMCCSIMFMT              call AMCCSIMFMT
```

It will prompt you for output file name, for spacing between signals, and for input file name(s).

7-9 BACK-ANNOTATION

AMCC will provide Back-Annotation files called BCKxxx.DSY where xxx = MIN, NOM, COM, or MIL. Substitute these in the SOM step (TCAL files):

The shell RUN_SOM:

```
RUN_SOM som-ctl-file  BCKxxx.DSY
```

If the control file is FUNCTION.SING and Back-Annotation is run at MIL, then this becomes:

```
RUN_SOM FUNCTION.SING BCKMIL.DSY
```

The super shell can also be used to run SOM:

```
RUN_AMCC
 1
 2
 3
 som-ctl-file     <----- you will be prompted
 BCKxxx.DSY       <----- you will be prompted
 0                <----- exit the shell
```

Proceed as before. The Back-Annotation file will contain the output load delays.

AMCCSIMFMT

The AMCCSIMFMT VLAIF conversion program accepts a DAISY VLA Intermediate Formatted input file (VLAIF) (a file produced by an $OUTPUTS section specification in the simulation control program), and produces an AMCC-formatted output file. This output file is used as an input file to other software programs.

During the processing of the VLAIF input file, the data field will be translated from the DAISY output strength level (refer to the DAISY User's Manuals) into the SIM formatted strength level.

The program asks the user for the following:

Enter VLAIF (input) file name:

This is the file produced as a result of the $OUTPUTS section in the SOM simulation control file.

Enter SIM format (output) file name:

This requests the file name that you want used for the formatted output. Use the AMCC standard file naming convention, i.e.: SIMFMTFN.nnr, SIMFMTAS.nnr, SIMFMTAC.nnr, SIMFMTPR.nnr, POCFMTAS.nnr or POCFMTAC.nnr. (See VOlume II, page 4-1-4.)

Choose from menu -
1: VLAIF to SIM format (no parenthesis)
2: VLAIF to SIM format (with parenthesis)
Enter choice [1 or 2]:

Choose 2

Do you want data separated in columns [Y/N]?

This is requesting spaces between columns. Choose N.

Enter number of DATA per column.

This refers to the above. DATA can range from 1 to n; 0 will get no columns.

The VLAIF input file is processed by extracting the net names from the header section, then processing the address and data until the EOF is reached.

The following table lists the translation from VLAIF characters to SIM format characters in the DATA field:

VLAIF	SIM FORMAT
0	0
1	1
Z	Z
H	1
L	0
+	Z
-	Z
*	X
#	X
X	X
U	X
!	X
@	X

ERROR MESSAGES

Error: Token (:) not found, line xxx.
Error: END parameter not found in VLAIF input file.
Error: Data length out of range on line of VLAIF input file.
Data'a length (#), Number of nets (#).

AMCC FILE UTILITY - AMCCFILUTL

OUTLINE:

1 ABSTRACT

The file utility program provides several generally useful features:

- Comment Stripper - Strips all text between comment markers (ie. /* ... */) and the markers themselves.
- Set Characters Case - Convert text's case (upper to lower and lower to upper).
- Column Stripper - Strips out columns of text between the start and ending column entered by the user.
- Column Maker - Breaks text into columns.
- Blank line remover - Removes blank lines.
- Trailing blank remover - Removes trailing blanks.

To run the FILE UTILITY program you simply enter:

For DAISY MAESTRO and DNIX
AMCCFILUTL <cr>

2 OVERVIEW

The AMCCFILUTL program is used to manipulate TEXT files. The program provides six (6) utility functions. The program is menu driven and very simple to use.
The program prompts for the Input and Output file names, then displays a menu of operations. The format of the menu is:

```
Choose from MAIN menu -
   (A)  Comment Stripper.
   (B)  Strip columns of TEXT.
   (C)  Delete trailing blanks.
   (D)  Delete blank lines.
   (E)  Break TEXT into colums.
   (F)  Change characters case.
   (Q)  Quit (break to OS).
Enter selection [A, B, C, D, E, F or Q]:
```

Notes:

- The Input file must exist or a DAISY run time error will occur.
- The Output file will erase an existing file of the same name (DAISY only).
- A carrage return entered as the Output file name will open a higher version of the Input file (VAX/VMS only).

3 DESCRIPTION

The following sections describe each of the file utilities individually.

3.1 COMMENT STRIPPER

The COMMENT STRIPPER will delete all DAISY and "C" comments (i.e., "/* ... */") from the text file entered. All text between comment markers and the comment markers themselves will be removed from the input file when copied to the output file. Select option (A) in the menu for the comment stripper.

3.2 SET CHARACTER CASE

The SETCASE option sets the case of all non-commented, and non-quoted characters when coping text from the input file to the output file. The user can select either UPPER to LOWER or LOWER to UPPER case.

The user is prompt with a sub menu for selection:

```
Enter selection from the SETCASE menu -
   (L) Convert to Lowercase.
   (U) Convert to Uppercase.
   (Q) Quit.
Enter SETCASE choice [L, U or Q]:
```

Select option (F) in the menu for setcase.

3.3 COLUMN STRIPPER

The COLUMN STRIPPER selection will strip columns of text between START and END indexes when coping from the input file to the output file. The user will enter both the START and END indexes and both indexes will be checked to insure they are correct (i.e., [(both>0), (end>start), (end<maxstr) and (start<maxstr)]). MAXSTR is currently set to 256.

Example with (Start index = 3) and (End index = 6):

```
1234567890        - Input file
1234567890
127890            - Output file
127890
```

Select option (B) in the menu for column stripper.

3.4 COLUMN MAKER

This option allows the user to break a text file into columns of text separated by a blank column when coping text from the input file to the output file. The user will enter the amount of text per column.

Example:

```
xxxxxxxxxxxx          ->  input file
xxx xxx xxx xxx x     ->  output file with text
                          per column = 3
```

Select option (E) in the menu for column maker.

3.5 TRAILING BLANK REMOVER

The Trailing Blank Remover utility will remove all trailing blanks from all lines in the input file when coping to the output file.

Select option (C) in the menu for removing trailing blanks.

3.6 BLANK LINE REMOVER

The Blank Line Remover will remove all blank lines from the input file when coping to the output file.

Select option (D) in the menu for deleting blank lines.