SECTION 3
OPERATION GUIDE FOR
MENTOR EWS (210)
Q24000 BICMOS ARRAYS

TABLE OF CONTENTS

1 New this Release 1
1.1 Overview 1
1.2 Special Notes 1
1.3 Warning 2
2 Shell Scripts (run_amcc) 2
3 AMCC EWS-Specific Schematic Rules - Introduction 2
3.1 Off-page Cross-Reference Program For Mentor Graphics Users 5
4 AMCC Logic Array Design Steps On The Mentor Ews 6
5 Use Of amccsimfmt 7
6 Back-Annotation 8

1 New this Release

This release is dated (210) and incorporates all previous software release updates. This software is compatible with the Mentor 8.2 release. *MISL is not supported.* Mentor is supported on 1/4 inch data cartridge.

Version 7.0 is no longer supported by either Mentor or AMCC.,

This release includes an expanded *run_amcc* script, including ***amccpackage*** and ***amccann, amccsimfmt, amccvrc, amccsubmit, amccpas*** and ***amccerc*** , which has been expanded with checks specific to the Q20000 series.

1.1 Overview

The netlister, **AGIF**, is called by ***run_agif*** and produces a standard netlist, *circuit.sdi* which is used as input by all MacroMatrix software.

The MacroMatrix package contains ***amccerc*** , which includes an automated I/O list report and cross reference report; 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. The output files are: *amccio.lst, amccxref.lst and amccerc.lst* located in the **/erc** subdirectory beneath your design. The erc error messages are documented in Volume III, Section 4.

The two programs ***amccpackage*** and ***amccann*** provide package pin capacitance and allow the description of system loading as well as the commentary toggle frequency and unusual terminations. The output files for amccpackage are: *amccpackage.err*, *<product_name>.pkg,* and the data file *output.dly.* The output files for ***amccann*** are: *amccann.lst, amccpkg.lst, output.dly* (with system editing added), and the *fntxxxx.ews* or *bckxxxx.ews* delay files.

The process of running the simulator ends with the production of simulation output files. All simulation output files that are to be submitted to AMCC must be formatted using ***amccsimfmt***. *simfmtxx.xxx* and *pocfmtxx.xxx* are suggested names for the files produced (where the xx.xxx are unique identifiers).

All sampled ***amccsimfmt*** files that are to be submitted must have been processed through the vector rules checking software, ***amccvrc***. The program ***amccvrc*** is documented in Volume III. The vector rules checker program provides expanded screening for some of the common errors committed during vector development. It produces *amccvrc.lst*, located in the calling directory. AMCC requires its use for all submitted, sampled, maximum worst-case functional, AC Test and parametric simulation output files.

Running ***amccvrc*** for a clocked circuit requires a ***signal analysis file*** describing signal-clock relationships. Non-clocked circuits do not perform race checking.

The design submission program **amccsubmit** must be used. The software prompts the user for all required information and performs initial error checking. It produces reports that replace the manually-generated design submission and design validation documents and provides an error file. The ***amccsubmit*** error messages are documented in Volume III. Output files are: *amccsubmit.dat, amccsubmit.err* and *amccsubmit.lst*.

1.2 Special Notes

The **amccsimfmt** simulation format program requires a list output file. This output file must contain all primary inputs, all primary outputs, all bidirectionals and all 3-state enable or bidirectional control enable signals. Documentation on the amccsimfmt program is in Appendix A for this section. The file may be sampled (for functional, at-speed, AC test or parametric simulations) or print on change (for at-speed and AC test simulations).

1.3 Warning

Prior to starting your design, please call AMCC to inquire whether you are in possession of the most current release of libraries, software, and engineering rules checks! The chip macro rev number should match the software release.

2 Shell Scripts (run_amcc)

A script has been provided by AMCC to make use of MacroMatrix easier for you.

Note: the script may create directories and files which are not created by normal manual operation of design steps. The shell script ***run_amcc*** does not include Quicksim but Options 3, 4 prompts the user as to the proper way Quicksim should be called.

1. After completing of your schematic, set your working directory to be top of your design.

2. The script is invoked by typing

/amccsource/amcc/com/run_amcc <design_name>.

where amccsource is the home directory of the AMCC design list.

The <design_name> is optional; if it is not entered, the customer is prompted for it.

3. The MacroMatrix main menu will appear on the screen with nine menu items:

1: Build VIEWPOINT, create AMCCAGIF and run AMCCERC

2: Run AMCCPACKAGE

3: Run AMCCANN

4: Build min and max VIEWPOINTS for SIMULATION

5: Run EXPAND for SIMULATION (BACK ANNOTATE)

Exit and run Simulator before going to next option

6. Run AMCCSIMFMT

7: Run AMCCVRC

8: Run Placement Annotation System (AMCCPAS)

9: Run AMCCSUBMIT

0: Exit

During processing of each menu selection, error checking is done. If errors are found at any time, the script will terminate.

The screen will prompt for the *library name* in steps 5 and 4.

3 AMCC EWS-Specific Schematic Rules - Introduction

This section contains Mentor-specific guidelines and restrictions. Generic information is found in Volume III, Section 2, *EWS Schematic Rules and Conventions.* By following these guidelines, problems can be avoided in creation of 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. $$GROUND, ZTERM, $$PORTIN, $$PORTOUT, VDD, and VSS can be found in:

/amccsource/amcc/amcc_gen_lib

2. To assign a user-defined name to a net or macro use the following syntax (these names must be unique to the entire design):

ADD PROP INST <instancename>

ADD PROP NET <net name>

The INST and NET properties should be used in a hierarchy.

3. **Bus names** must be no more than six (6) alphanumeric characters (not including the subscript) and bus width cannot exceed 99 wires. If you are using buses for primary inputs, outputs, or bidirectionals, the bus names cannot be subscripted. Example: data(0) must be data0.

4. **All wires** attached to a page connector must have a user-defined name.

5. **All pins** of all macros must be connected to a wire. Any unused input pins are connected to $$GROUND and any unused output pins to ZTERM. Refer to Volume III, Section 2 for rules for VDD and VSS usage with BiCMOS designs.

6. **Modification, in any way, of the macros or their primitives**may result in creation of a schematic that cannot be correctly manufactured by AMCC.

7. **Adding properties to the chip macro** (one way): Put the cursor on the text to be changed. Select the text and type:

CHA TEXT VAL <property> <property values>

where <property values> is the instance property. Example:

CHA TEXT VAL POWER_SUPPLY STD5

8. **Adding the FOD property to a net** (one way): Once for each page type:

OWNER FOD -NET

Select the net and type:

ADD PROP FOD <fodvalue>

where <fodvalue> is the net property. Example:

ADD PROP FOD 20

9. **Adding SWGROUP property to a macro** (output or added power or added ground) (one way): Once for each page type:

OWNER SWGROUP -INSTANCE

For each macro to which a swgroup name is to be attached, select the macro and type:

ADD PROP SWGROUP <group_number>

where <group_number> is an instance property.

ADD PROP SWGROUP AAAA

10. **Adding GTO property to a macro** (output): Type:

OWNER GTO -INSTANCE

For the macro which is the gate tree output, select the macro and type:

ADD PROP GTO <name>

where <name> is an instance property.

ADD PROP GTO AAAA

11. **For Designs using Frames, Nesting and Hierarchy**

a. All I/O signals must be captured on the top level, un-nested and named.

b. All hierachical blocks, components and critical Nets must be named uniquely (use INST property for components and use NET property for nets).

12. For notes, refer to the Mentor system manual and see also Section 2, this manual, Application Note 1.MENTOR.

13. Activating Components: AMCC has provided menus of macro names to aid in ACTivating COMPonents. To activate the AMCC menus,

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX **fill in instructions**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

READ MENU /amccsource/amcc/com/amcc.rmenu_easy_7.0

14. Activating Components: *Activated components should be resident only in the /amccsource/amcc/library-name.* If not using AMCC menus, then activate components using:

ACT COMP /amccsource/amcc/(libname)/(comp_name).

15. Q24000 library contains both the Q24091/Q24021 I/O library and the Q24140/Q24060/Q24008 I/O libraries. The macros differ in their names and the ERCs are programmed to detect any mismatches.

16. AMCC Neted[1] Pull-down Menus:: The Neted pull-down menus have been customized to add several new features for AMCC customers. They include:

1) On-line help

On-line help for AMCC Neted commands are available by selecting the HELP > AMCC_HELP menu item within Neted.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX **fill in instructions**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

2) Command to add an AMCC sheet border and title block.

When you choose the SETUP > AMCC_BORDER menu item, the sheet size is set to "B" size (17 by 11 inches) with a 0.125 inch pin and grid spacing. A border and title block are placed on the sheet. Title block information can be changed using the PROPERTIES > MODIFY menu item or the SHIFT-F7 function key.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX **fill in instructions**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3) Activate and place an AMCC component.

This popup menu item is similar to the static parts menus. A macro may be selected and placed on a NETED sheet. Unlike the static menus, the pop-up parts menus are subdivided by function to help in the macro selection process. For example, the EDIT > Q20000_PARTS menu has the following submenus:

- generic
- chip_macros
- interface_macros
- adder
- counter
- comparator
- decoder

[1] Neted – not nested

- exor
- ff
- latch
- or
- and
- and_or
- buffer
- mux
- pll
- power_ground

17. Mentor simulation with bidirectionals The Mentor Simulator currently will not simulate bidirectionals correctly when contention is present in the vectors unless the force statement in the stimulus force file contains the extension -W. Use the **$$portout** connector with bidirectional macros.

When in input mode (example):

force dat2 1 32000 -W (EITHER)

When in output mode (example):

force dat2 0z 48000 -W (TTL)

force dat3 0r 52000 -W (ECL)

No schematic workaround is required.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX **fill in instructions**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3.1 Off-page Cross-Reference Program For Mentor Graphics Users

A program is now available that will add page cross-references to the off-page connectors placed on Mentor Graphics Neted schematics.

There are 2 steps required to add the cross references.

1) Execute the RUN_OFFPAGE program by typing

/amccsource/amcc/com/run_offpage <design>

where <design> is the name of the directory that contains the Neted sheets.

The RUN_OFFPAGE program reads the Neted sheets, determines what cross-references are needed, and then creates a Neted script file called OFFPAGE.DO.

2) Execute the OFFPAGE.DO script file by typing:

OFFPAGE.DO

The script will automatically invoke Neted and add the cross-references to each sheet in the design directory.

Notes:

a) The design must use the off-page connectors ($$offpag.in and $$offpag.out) supplied with the released MacroMatrix macro library. Earlier versions of these connectors do not support this program.

b) You may re-run the RUN_OFFPAGE program more than once on the same design. The new cross-references will over write the old ones.

c) For hierarchical designs you must run the RUN_OFFPAGE program separately on each block of the design. The program will not automatically traverse the hierarchy.

d) The design does NOT need to be expanded to run the OFFPAGE program.

e) For more information refer to the online help file by typing: HELP RUN_OFFPAGE.

19. AMCC requires that I/O macros are placed outside of nested blocks (on a top-level schematic sheet).

4 AMCC Logic Array Design Steps On The Mentor Ews

The following steps are integral in the design of an AMCC logic array:

1. Schematic Capture. Neted is used to capture the design. You must use an AMCC border on each page. The AMCC border must be placed on the page before any other macros are instantiated! The border provided is B size and can be placed on the schematic via the pull-down menus "SETUP" command. Refer to AMCC *EWS Schematic Rules and Conventions in Volume III* on how to use chip macros. A chip macro must be included in the schematic. AMCC software will not function without it. By AMCC convention, the chip macro and any extra power/ground macros must be documented on sheet 1 of your schematic.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX **fill in instructions**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

2. Expand For ERCs. To run the ERCs, select option 1 from the main menu.

The final netlist is ***circuit.sdi.*** this file is an input file to the ***amccerc***, ***amccann, amccsimfmt, amccvrc, amccsubmit*** and AMCC place and route software.

The input files to ***amccerc*** are your ***circuit.sdi*** file and a macro property file provided by AMCC for each library **(/amccsource/amcc/params/?*.fpm).** The output files are: ***amccerc.lst, amccxref.lst,*** and ***amccio.lst.*** Consult Volume III, Section 4 (MacroMatrix Reference Manual) for details on any error messages from ***amccerc*** .

3. Run amccpackage and then run Front-Annotation. Run the Packager and the Annotation program:

run_amcc option 2

run_amcc option 3

These two programs use your ***circuit.sdi*** file as input as well as the appropriate macro property file provided by AMCC (**/amccsource /amcc/params/?*.fpm**). The output file (***amccann.lst***) will contain any error messages. Consult the AMCC MacroMatrix Reference Manual for any error message explanations.

The annotation delay files are named depending upon the library, the *power_supply* and the *product_grade* value (mil or com) you have assigned to the schematic chip macro.

The actual Front-Annotation files generated depends on the array series. Consult Volume I, Section 3 for the files names that correspond to the library. The designators com4, com5, mil4 and mil5 are not used in the annotation file names. ***amccann*** uses the correct data based on the product grade and power supplies specified on the chip macro.

These steps must complete without errors in order to proceed.

4. Expand For Simulation To expand for simulation, AMCC has provided an option for expanding for AMCC primitives. To run expand for simulation:

run_amcc option 7

5. Simulation. Simulate with the annotated design.erel file. When running the shell, the exit to step 3 will prompt the screen for quicksim. Exit the ***run_amcc*** script (option 0) and create the force file. Then type as directed.

Examples:

For q24000 designs, military maximum:

```
quicksim design...name sim_mil 0.001 -max
```

Commercial maximum:

```
quicksim design...name sim_com 0.001 -max
```

Minimum:

```
quicksim design...name sim_min 0.001 -min
```

For q20000 designs, there is a different accuracy:

```
quicksim design...name sim_mil 0.001 -max
```

Use -max for military or commercial simulations to obtain the maximum rise and fall times.

Use -min for minimum simulations for the Q24000 and Q20000 to obtain the minimum rise and fall times.

Simulation output interfaces to the AMCC test department.

For sampled files use period list 100 99.99.

For print on change files. the command to be used is list .... -change.

For bidirectionals, include -w in for force statement:

```
Force signal state time -w
```

Drive TTL with 0z, 1z or xz during output mode and ecl with or.

6. Run amccsimfmt. This program called by:

run_amcc option 5

See Appendix A of this section.

10. Run amccvrc This program is called by:

run_amcc option 6

7. The signal analysis file: Follow the instructions in Section 8, Appendix B for the creation of the signal analysis file. The ***amccvrc*** software requires a user-created signal analysis file, the ***circuit.sdi*** file used in the simulation, and the ***amccsimfmt*** output file to be checked.

Only sampled maximum worst-case functional and AC test ***amccsimfmt*** output files must be checked and the ***amccvrc.lst*** report for each must be submitted along with the corresponding signal analysis file. Errors must resolved or an AMCC waiver obtained.

8. Run amccsubmit. This program is called by:

run_amcc option 7

Follow the instructions in Section 8, Appendix D for the operation of amccsubmit. It is invoked from the same directory containing the AC test simulation output files (formatted). Both ***circuit.sdi*** and ***output.dly*** must be in the same directory.

9. Design Submission - hierarchy.

If you are doing a hierarchical design, document the full path name to the design subtrees (top level down). AMCC needs to update the design on the user-created block once the design has been submitted. The component is not to be located in the /amccsource/amcc subtree.

10. Run amccpas: This program is called by:

run_amcc option 8

5 Use Of amccsimfmt

The ***amccsimfmt*** program converts *sampled* or list -change list files into the required format for submission to the AMCC test program. (List files and waveforms are not part of the design submission.)

The program reads in a Mentor simulation file and produces a standard ***amccsimfmt*** file as output.

The program accepts only one Mentor simulation file at a time as input.. Run ***amccsimfmt*** for each list file to be formatted.

The program prompts for the input and output file names.

The input file to ***amccsimfmt*** is a list file created by using the write list command in QuickSIM see Application Note 1.Mentor.

The output file is the user-defined name for the final formatted file.

Macros with 3-state outputs are modified to produce Z as the high-impedance state instead of 1Z or 0Z.

6 Back-Annotation

After place and route, AMCC will send you files needed to generate back-annotation delay values.

The back annotation files which are in the same syntax as the Front-Annotation files.

delay.dat	Q24000 only
rcdelay.dat	Q20000 only
circuit.pkg	all functions

To generate the BA delay numbers, install the files in your erc subdirectory where the output.dly and circuit.sdi files reside. The file ***output.dly*** must be present in the same directory. Reinvolking ***amccann*** adds the external load delays (system load from ***output.dly*** and package pin capacitive load from ***circuit.pkg***) to the ***delay.dat*** or ***rcdelay.dat*** files and names the new files as follows.

These ***bckxxxx.men*** files are used in the same manner as the ***fntxxxx.men*** files when simulating.

bckm5mx.men

bckm5mn.men

bckc5mx.men

bckc5mn.men

bckm4mx.men

bckm4mn.men

bckc4mx.men

bckc4mn.men