Inter-Office Memorandum

To	Dorado Project	Date	April 18, 1980
From	Ed Taft	Location	PARC/CSL
Subject	Dorado Board Tester	File	[Ivy]<DoradoDocs>BoardTester.press

XEROX

The Dorado board tester is used for static testing of Dorado logic boards, either stuffed or unstuffed, to verify the continuity of all nets and the absence of shorts between nets. It also has a limited ability to detect stuffing errors (e.g., backward SIPs) and leaky gates.

The tester operates by measuring the resistance between each pin of each IC and the power and ground planes (which are shorted together). For ECL nets, this resistance can be predicted on the basis of how many terminating resistors the net has. A discrepancy between measured and predicted resistance is an indication of an error. For example, a net that is broken will show higher resistance than expected on some or all of its pins, while a net that is shorted to some other net will show lower resistance than expected (assuming the other net has at least one terminator).

Physically, the board tester consists of a low chassis containing Dorado edge connectors into which a Dorado board may be slid. The chassis contains a small quantity of electronics and a cable that connects to the Diablo printer port of an Alto I. Any of various probes may be plugged into the chassis, including 16- and 24-pin DIP probes, a QIT package probe, and a single probe for testing other types of packages. Finally, there is a button mounted on the side of the chassis and a foot pedal, either of which may be used to advance from one IC or pin position to the next.

A program called Resist exists (file [Ivy]<Dorado>Resist.run) for driving the board tester. It reads in the wire list for the board being tested (e.g., ProcH-Rev-Ce.wl) and a data file containing information about what resistances to power and ground to expect at each pin of each IC (e.g., ProcH-Rev-Ce.resist). It then interacts with the operator to test each IC position, and produces an output file describing any errors that are detected (e.g., ProcH-Rev-Ce-SN-0123-U.defects, where 0123 is the board serial number and U indicates that it is unstuffed). This file may in turn be used as input to the program, to assist in tracking down and correcting errors.

Usual test procedure

This is a prototype procedure for testing a new Dorado board. Many variations are possible, and undoubtedly this procedure will be improved as experience is gained with it.

1. Mount the board

Slide the board into the top slot of the tester and close the connector jaws. Make sure the contacts line up there is a lot of slop in the connectors.

If the board is not stuffed, plug SIPs into all terminator positions. For bare Multiwire boards, there exist SIPs with special plugs on all their pins. Plug into the chassis the 16-pin DIP probe and the QIT probe with spring-loaded pins.

If the board is already stuffed, just plug in the 16-pin and 24-pin DIP clip probes.

Plug the black cable into the Diablo printer port of an Alto I.

Caution: Make sure the chassis does not contact ground. The chassis is part of the test circuit, so if it is grounded the tester will not work.

2. Obtain data files and start Resist program

Use FTP to load the two required files, extensions .wl'' and .resist'', from the dump file for the board you are testing. Make sure you use the correct technology (stitchweld or Multiwire) and revision. For example, load files ProcH-mwRev-Ce.wl and ProcH-mwRev-Ce.resist from [Ivy]<DoradoLogic>ProcH-mwRev-Ce.dm. (At present, for some boards the .resist'' file is stored separately rather than as part of the dump file.) There is an optional .exceptions'' file (e.g., ProcH-mwRev-Ce-U.exceptions or ProcH-mwRev-Ce-S.exceptions, for unstuffed and stuffed boards, respectively); if one exists you should retrieve it also.

Start the Resist program with the command:

Resist *filename*

where *filename* is the name of the .resist file. (If you are testing a BaseBoard, or any other board that uses 40-pin ICs, you must supply the additional parameter 40/P'' in the command line.)

Resist asks you to supply the board serial number (followed by RETURN), and also asks you whether or not the board is stuffed. It then churns away for about a minute while it reads the input files; most of the display is black during this time. When the main part of the display appears, Resist is ready for you to begin testing.

3. Test the board

Below the system window at the top of the screen, there is a window containing the current board location and instructions about what to do next, another window showing the current reading of the single probe, and a large window showing the current readings of all the pins of the DIP probe, along with the pin numbers, expected resistances, and net names where known. (A net name of ?'' indicates that the pin is not mentioned in the wire list either it is trace-wired or it is not connected to anything.)

Resist tests ICs in the following order:

1. All 16-pin and 14-pin packages
2. All 24-pin DIP packages
3. All QIT packages
4. All other packages

Within each group, ICs are tested in alphabetical order by board location.

When you are at a 14-, 16-, or 24-pin DIP package or a QIT package, Resist asks you to install the appropriate probe on that package. The resistance measurements in the bottom window are continuously updated, and pins whose readings are out of spec are displayed with a black background. When you have installed the probe correctly, all the black bars should go away; if they don't, either the probe is installed incorrectly or there is a real problem with the board. When you are satisfied that the probe is installed correctly, press the button, the foot pedal, or the space bar on the Alto keyboard to advance to the next IC. (The test results saved in the output file are sampled at the instant you press the button or whatever.)

When Resist is testing an IC position with an oddball package, it asks you to put the single probe on each pin in turn and press the button. In this case, the DIP measurements in the bottom window are accumulated one pin at a time. When you have done this for all pins, Resist then waits

for you to press the button once more before advancing to the next IC.

It may happen that Resist asks you to mount a DIP or QIT probe on an IC position for which that is impossible (e.g., because it contains an IC with legs cut or wires soldered on). In this case, type S'' to cause Resist to test this IC using the single probe rather than the DIP or QIT probe.

The board tester contains an audible signal that may optionally be enabled by means of a switch on the side of the chassis. If enabled, the signal will sound whenever one or more of the measured resistances is out of spec. (While you are using one of the DIP or QIT probes, the signal is disabled until you have lifted the probe from the previous IC position and placed it in contact with the new one.)

4. Finishing up

When you have tested all IC positions, Resist writes the .defects file and exits to the Alto Executive. The .defects file name includes the board serial number and unstuffed/stuffed indication; e.g., ProcH-mwRev-Ce-SN-0123-U.defects for an unstuffed board or ProcH-mwRev-Ce-SN-0123-S.defects for a stuffed board. At this point, you may examine the .defects file using Bravo, print it using Empress, or whatever.

At the end of the .defects file is a summary of errors found, organized by net. Each net that exhibited any errors is shown, with the resistance measured at each pin. Within a net, the pins are shown in the order that they are wired on the board; however, only those pins that are mentioned in the .resist file are shown edge pins, terminator pins, and the like are not shown. A measured resistance of --'' means that the pin was not tested; this occurs only if you skipped over board locations or Quit'' before testing the entire board.

You can also re-run Resist using the .defects file as input (instead of the .resist file). When you do this, Resist tests only those IC positions for which errors were reported. This gives you an opportunity to eliminate any errors that might have been caused by operator error, bad contact with the probe, etc. When Resist finishes this time, it writes the updated results on top of the input file, after saving the input file by appending $'' to its name (just like Bravo).

When you are satisfied that all the errors reflect real board problems, then, for an unstuffed Multiwire board, you should print the .defects file and put it in the board log as a record of what modifications must be made to the board after it is stuffed. For a stuffed board, you should proceed to find and correct the problems.

5. Other details

If a test is interrupted (either by the Alto failing or by your issuing the Quit'' command), you can restart it simply by issuing the command Resist/R'', with no other parameters on the command line. Resist will resume the test at the point of interruption. You must issue the Resist/R'' command *immediately* after the interruption; if you start up Resist without the /R'' switch, you will lose the information required to restart the previous session.

All pin numbers are counted on the package, not on the board. In particular:

> 14-pin DIPs have pin numbers 1 through 7 up one side and 8 through 14 down the other. The Resist program tests 14-pin DIPs using the 16-pin DIP probe, which you must position so that pin 1 of the probe contacts pin 1 of the 14-pin DIP.
>
> QIT packages have pins numbered 1 through 24, starting in the usual corner and proceeding counterclockwise. However, Route (and therefore Resist) believes that a QIT package has 26 pins, because that is the number of pins on the QIT platform (the last two pins are extra grounds). If you test a QIT package position using the single probe, you must hold the probe on pin 24 while advancing the Resist program through pins 25 and 26.

The board tester's resistance range is limited to about 150 ohms; resistances greater than that are reported as open''.

The Resist program will optionally read an additional file, with extension .exceptions'', which contains exceptions to the .resist file. The .resist file may be incorrect due to the presence of analog components, configuration jumpers, and other aberrations that Route was never told about. To avoid having these show up as errors every time a particular board is tested, you can prepare an .exceptions file containing the corrected information for each affected IC position.

The .exceptions file has the same format as the .resist file and should be created by editing either the .resist file or a derived .defects file. (This procedure should be used in preference to editing the .resist file, since such edits would disappear whenever a rework was performed.) The name of the .exceptions file must be in the form ProcL-mwRev-Ce-U.exceptions or ProcL-mwRev-Ce-S.exceptions, where the board name and revision level match those of the .resist file, and the U or S indicates that the file is applicable to unstuffed or stuffed boards. If a .exceptions file with the appropriate name exists on your disk, Resist will read it automatically when it starts up.

Resist commands

The following is a complete description of all commands and facilities available in the Resist program.

Command line

The general form of the command line for starting Resist is:

Resist/*global-switches parameter*/*switch parameter*/*switch* ...

The *global-switches* may include any one of the following:

/R	Restart an interrupted session. (If you specify this switch, you should not supply any *parameter*s on the command line.)
/S	The board being tested is stuffed.
/U	The board being tested is unstuffed. If you don't specify either /S'' or /U'', Resist will ask you for this information right after it starts up. (If you are reading a .defects file rather than a .resist file, Resist extracts this switch from the name of the file.)

A *parameter* without a *switch* specifies the input file name; there must be at most one of these. (If you don't specify one, Resist will ask you for it right after it starts up.) Other parameters are as follows:

number/S	The board serial number. (If you don't specify this, and Resist needs it, it will ask you for it right after it starts up. If you are reading a .defects file rather than a .resist file, Resist extracts the serial number from the name of the file.)
number/P	The maximum number of pins permitted on any IC. The default is 26; you must specify 40 for a BaseBoard. (Increasing this number increases the size of the display, which reduces the amount of memory available for storing the information in the .resist and .wl files.)
number/T	The error tolerance, as a percentage of the expected resistance. The default is 12.
location/L	The location of the first IC to test.

An example of a fairly complex command line is:

Resist/U ProcH-mwRev-Ce.resist 40/P 123/S 10/T

Interactive operation

At any time when Resist is expecting you to press the button to advance to the next IC or pin, you may type any of the following single-character commands:

Single	Test the current IC using the single probe rather than the DIP probe. If you issue this command when you are already using the single probe, you will reset Resist to pin 1 of the current IC.
DIP	Test the current IC using the appropriate DIP or QIT probe rather than the single probe.
Back up	Back up one pin. This command is meaningful only when you are using the single probe.
Ignore	Ignore this IC: that is, advance to the next IC, but do not write anything into the .defects file even if there are errors. Use this command when you know that an error is benign (e.g., due to a component with legs cut for configuration purposes).
Location	Go to a new board location, which you must then type in (precisely e03'', not E03'' or e3''). Resist proceeds to test ICs beginning at that location, without remembering about any locations you may have skipped over. You can only specify locations that are mentioned in the input .resist (or .defects) file.
Quit	Exit to the Executive. Resist will quit without putting anything into the .defects file; however, you can later restart using Resist/R'.

Resist, Defects, and Exceptions file format

The .resist, .defects, and .exceptions files are text files containing information about each IC. Text between a semicolon and a CR is a comment and is ignored by the Resist program. A .resist file contains information about all ICs on the board (as well as some other component positions such as terminators and bypass capacitors, which are ignored by Resist). A .defects file contains the same information, but only for those ICs for which errors were detected; following each IC description are one or more comments describing the error. An .exceptions file contains manually-prepared corrections for specific ICs in the .resist file.

An IC description is:

location (*package-type*, *number-of-pins*, *IC-type*) *resistance*, *resistance*, ...

The *location* is a board location in standard format (e.g., g13'' or #-1f13''). The *package-type* is one of DIP3 (.3-inch wide DIP), SIP, or OddPkg (anything besides a DIP3 or a SIP). The *IC-type* is the full IC designation (e.g., MC10176''); in a .resist file (but not in a .defects file), some additional information appears after a /'' it is ignored by Resist.

The remainder of the description consists of the expected resistance at each pin, from pin 1 to pin *number-of-pins*. A short is indicated by 0'' and an open circuit by $''. If the resistance at a pin cannot be predicated, this is indicated by ?''.