Chapter 8.2

Electroglas Autoprobe in DCL

(autoprobe - 373 Cory)

1.0 Equipment Purpose

1.1 (autoprobe) Electroglas Autoprobe in DCL (Device Characterization Laboratory, 490 Cory) is capable of electrical testing of 4" to 8” wafers.

1.2 This is a users' manual for the Electroglas autoprobe and the Metrics ICS program. The station can be used to test baseline chips and other test structures. Large amounts of data can be collected to monitor the baseline process. This automated testing system was developed in conjunction with the BCAM test chip now used for baseline processing, and currently also by other members for their specific application.

2.0 Material Controls & Compatibility

2.1 To be determined

3.0 Training Procedure & Applicable Documents

3.1 Member Qualify Tool

3.1.1 This tool may or may not require enabling for use.

3.1.2 Timeline (estimated time to completion: 1 day)

3.1.2.1 Train and qualify with any qualified member. All qualified members are superusers on these tools.

3.2 Hardware

This document describes how to turn on the instrument, put in the probe card and align the wafer. See Section 9.1.

3.3 Software

This section describes how to use Metrics ICS System Tools and Testing Tools to setup measurements for the test structure. See Section 9.3, the software setup and on line help files, documentation at: http://www.metricstech.com/icv/icv.shtml

4.0 Definitions, Process Terminology, Equipment Configuration

4.1 Probe Card

A PC board with specific conductive path and probe tips.

5.0 Safety

The surgical/poly gloves should be worn when handling the probe card. Special care should be taken when dealing with the probe tips as they are easily damaged.

6.0 Statistical/Process Data

6.1 To be added as available

7.0 Available Processes, Process Notes, & Hardware Modules

7.1 Machine Components

The probe station is comprised of different components.

7.1.1 Electronic Rack: Tower is located to the left side of the probe station, 4084B switching matrix controller (SMC), and the 4142B DC source/monitor, and the 4280 1 MHZ Capacitance meter/C-V plotter.

7.1.2 Electroglas 2001 probe station: Microscope and 4085A switching Matrix module on top, joystick box (key pads on it), control key pad and **monitor**, in front of the station and **camera lighting** electronic module/power supply units, just below the station.

7.1.3 PC and Unix stations to invoke the test programs and their associated key pads located to the right of the EG 2001 probe station.

8.0 Operation

The automatic probe station (autoprobe) operation involves hardware and software set up to successfully perform electrical parametric (I-V, C-V and resistance) measurement on device/IC under test (4"-8" wafers). The hardware part needs to be setup first, which includes probe card and wafer set up. Once the wafer is ready (aligned and set on the chuck), the software program can then be invoked to drive the stage, control metric switch, perform the test routine/s and extract parametric data in a PC or a Unix based workstation environment. The PC windows based program (recommended) called Metrics I/CV by Metrics Technology is described in this section, as a primary mode of operation. The second method Sunbase3 program is a Unix based package older software), also offers various measurement routines, but it is somewhat more difficult to manipulate or add new test routines in it. For more information on setup files, test routines and how to run the sunbase3 program, refer to Appendix 2 and Appendix 3 (specific to baseline runs) at the end of this chapter. See Figure 13 for a picture of the entire system.

9.1 Hardware Set Up

Probe Card Set Up (Mounting/Replacing A Probe Card)

9.1.1 Swing out the microscope assembly on the probe station (the orange arm and the attached microscope) to the right and out of the way. See Figure 14 in Section 10.

9.1.2 Unlock the two metal locks, lift the buckle and slide down to free the locks at the lower right side of the probe station. See Figure 15.

9.1.3 Slowly lift the cover by the black handle. Be careful, it is very heavy.

9.1.4 Loosen the three screws locking in the probe card plate, so that the pins held down by the screws can rotate, Figure 16.

9.1.5 Rotate the pins to free the plate.

9.1.6 Take the plate out and put it back upside down, Figure 17.

9.1.7 If a probe card is already in there, carefully take the probe card out. It might be a little tight, but be careful not to break it.

9.1.8 Put the probe card into an empty probe card container with the contact tips (not the measuring pins) facing up. Close the lid of the container and put it away.

9.1.9 Take the probe card that you want to use back in.

9.1.10 Line up the two very small markers (two small dots) on the probe card with the red arrows on the machine. Make sure the probe card is securely put in. On most probecards pin 26 is marked with a hole. This hole should be exactly above a little screw on the mounting unit (there is only one smallest screw).

9.1.11 Take out the plate and put it back right side up. That is, flip the plate and put it back the way you first saw it. Pin number 12 on the mounting plate has to be in the downmost position (towards the user).

9.1.12 Line up the plate as before and rotate the pins to lock the plate in place.

9.1.13 Tighten the screws so that the pins can no longer move. Now the plate is securely locked again.

9.1.14 Slowly put the cover back down by the black handle. Again, be careful, because it is very heavy.

9.1.15 Lock the two metal locks on the right of the cover. If the locks are locked correctly, you should see a red light go on directly to the left of the cover's black handle. The red light is labeled as **OUTPUT ENABLED**.

9.1.16 Put the orange arm with the microscope attached back in.

Wafer Set Up

9.1.17 Use different menus on the small monitor in front of the probe station to setup your test parameters. The control key pad just to the front of this monitor (Figure 18), can be used to enter parameters on various screens shown in Appendix 1. Normally the die size (x and y) and wafer diameter are the only parameters that are needed to get changed, unless someone else has changed other parameter for special application or mode of operation. It is suggested to doublecheck all the options in the main menu and submenus for changes before each start.

Note: If the equipment is turned off, then turn on the 4084B switching matrix controller (SMC), and the 4142B DC source/monitor on the electronic rack. Turn on the probe station, next. There are two power switches on the front panel of the station (**camera lighting** electronic module on top, and power supply below it). Turn the top switch first then the lower switch (do the reverse when powering down). The eglas should power up with the message **XY motor blank** on the monitor screen.

9.1.18 With the main menu shown on the screen, press **SET PRMTR** (set parameter) key on the control pad to go to the parameter screen. Set the die size and the diameter of the wafer with option 1 and 4 (type 01 and 04 followed by enter key). Input the new value/s followed by return key. (Enter key is a black button on the bottom, where the broken white key should have been.) Press the **enter** key again to go back to the initial screen, shown in Appendix 1.

9.1.19 Press **SET OPTION** and make sure that option 02 (Auto Align), option 03 (Auto Profile) and option 08 (Auto Temperature) are all enabled.

9.1.20 Press **SET MODE** key on the control panel to check the status of **theta compensation** on the screen. For automatic mode of operation, testing many dies automatically, you will need to have **theta compensation** disabled. You can enable theta compensation (option 11) in the **SET MODE** menu, if you need to perform manual wafer alignment/testing.

Note: when inside a menu you can type in the option number followed by the enter key to select, and then change the value of that particular option. Otherwise you can get out of the menu by pressing the designated key on the control pad. Check Appendix 1 for more detail on the content of different menus and submenus.

9.1.21 Press the **force release button**, which is a small button inside the circular hole on the left side of the joystick box to release the vacuum on the wafer chuck assembly. See Figure 19.

9.1.22 Now with the vacuum released, the chuck assembly should be free to slide around on the platform. Slide the chuck assembly to the front-right corner of the work surface (probe station platform) closest to you, so that you can load your wafer. Check that the orientation of the dice is roughly parallel to the axes of the probe workspace. The wafer can only be rotated by ~15 degrees after it is loaded, and it's much easier to do it right from the start.

9.1.23 Press the small button (force release button) on the left side of the joystick box again to enable the vacuum on the chuck assembly.

9.1.24 Setting a reference point - press [FIND TARGET] the wafer should move under the camera resulting in an image appearing on the monitor. Looking at the monitor and image of the die, use the joystick to find a reference. This will be a unique non-repeating pattern on your die that can be moved into the white square box on the screen. Once set, go to the next step. This works best with simple horizontal/vertical structures; corners of dies, intersections are suggested to try first. If recognition fails, retry in an other spot.

Note 1: Press **CAMR** the camera button on the joystick, If no image is shown on the screen post **FIND TARGET** step. If still a problem, proceed to instruction given in the Troubleshooting Guidelines, Section 9.2.

Note 2: The joystick has three modes of speed, which can be selected by turning it clockwise or counter clockwise. These speed settings can give you the fine movement, die size stepping and continuous motion, depending on what mode you choose (jog/index/scan)

9.1.25 Press [PAUSE/CONT] on the joystick box to start scan. A number should show up on the screen, while the wafer gets mapped (scanned). The bigger this number the better the fit and referencing process. Refer to Section 9.2, If any error or warning comes up, i.e. **Ref Not Stored**. If all goes well, then you should end up in the initial screen with a message asking you for a second reference (2nd REF). You do not need to scan the second target, just press [ENTER] twice on the next two messages (**second ref** and **find edge** questions). This will take you back to the initial screen.

Note: If you only have a few dies on your wafer and do not need to perform automatic probing, you will still need to perform the **find target** procedure. This time however, you will end up with **Ref Not Stored** warning, which can be bypassed (Do not trouble shoot it, proceed to the next step, 9.1.26). Please also note, in that situation you would have enabled the theta compensation option earlier (SET MODE screen and line 11, enabled). which means the auto align operation are to be skipped while testing the die at the later steps.

9.1.26 Press **PROG** (program) on the monitor panel.

9.1.27 Select option 04, to profile. This will adjust the tilt of the wafer automatically. The chuck will go to a pre-specified location, and then will move around a bit as the wafer is profiled. After the profiling is done, the chuck should end up under the microscope.

9.1.27.1 Occasionally, the chuck will not initiate from a correct position on the platform. If this happens, press **load** to move the chuck back home, and then press

force release button on the same joystick box, and move the chuck to front right corner of the platform, essentially need to start over at step 9.1.21.

9.1.28 Press 3 in the G-PROFILER menu (same menu) to set the Z - height, which is the vertical height of the wafer.

9.1.29 Press **Z** on the joystick box; adjust the height with respect to the probe tips as per next step.

Note: You may need to move your wafer into a proper position for the Z-height adjustment procedure. You can do this by pressing the **X** and/or **Y** keys on the control panel (Figure 18).

9.1.30 While looking into the microscope, use the joystick to move the wafer up and down to adjust the Z-height. The probe tips should come in focus under the microscope, but you need to move the wafer into focus manually. This means moving the stage up, while looking into the microscope (always). You can press and hold the red button on the joystick to speed up the z-movement, but be careful not to crash the tips into your wafer (avoid tip damage/deformation). Release this button when tips get close to surface of your wafer, and watch carefully for any sign of tips getting deflected (bend outward). Probe tips' shape change is an indication of wafer pressing against the probe tips, therefore you should not move the wafer up any further. Slow down Z movement when wafer image is getting sharper. Adjust the z height, so that the probes just touch the wafer, but not pressed against it too much. The z height is numerically displayed on the display monitor, during the Z-height adjustment. The correct value (height) should be somewhere around 310 mils for a typical 6-inch wafer.

Important note! Always look into the microscope instead of the monitor, while adjusting the z height. This will prevent you from overdriving the tips into the substrate, hence damaging the probe card.

9.1.31 Look at the final z height on the monitor after you are done adjusting the height; make a mental note of the value.

9.1.32 Press **ENTER** to go back to the original menu and follow with the next step. This is an additional step required to store the z-height value by essentially repeating the z- height set up twice, as per follows.

9.1.33 You need to check to make sure that the z height is now set at the same value as in the previous step. This can be done by pressing **PROG** key, and selecting option 3 in the next menu, G-PROFILER menu for the Z-height set- up. Once you confirmed that the Z-height has a correct value, press **ENTER** to get back to the original menu. This second round is always needed, as it will store your previously set Z- height value. The saved value for Z height will be decreased with 7 mils for safety. If this breaks contact with wafer and pins, repeat procedure with an elevated Z value.

9.1.34 The wafer setup is now complete, however, if you need to make any final X& Y adjustment, then press the **Z** button on joystick box to lift the probe tips off the wafer, before stepping the wafer in such direction/s. Press **Z** again to put the probe tips back on the wafer, after your X and/or Y movement has placed the probe tips exactly where you want them to land (device pad, so on).

9.1.35 Press **LAMP** button on joystick box to turn off the lamp.

9.1.36 You can start your measurements as per instructions defined in Section 9.3 for a PC based program or in a UNIX environment (sunbase3 program), as per described in Appendix 2.

Note: Baseline runs (test chips), which are regularly processed by staff can also be tested through sunbase3 program, as described in Appendix 3.

Optional Stage / Wafer Rotation Adjustment

Please note, the Theta compensation should be enabled for this section. This can be done by changing its status in the **SET MODE** sub- menu (press **SET MODE** key on the control panel to get to this sub-menu).

9.1.37 The rotation is automatically adjusted, but if more accurate theta setting is required (probe tips are not stepping in the exact position on all dies, across your wafer), then follow the next step.

9.1.38 Press **ALIGN SCAN** on the joystick box, see Figure 19.

9.1.39 Pressing **PAUSE/CONT** on the joystick box once, will scan the probe over the wafer once. Correct the theta rotation as per follows.

9.1.40 While the wafer is getting scanned (moving back and forth horizontally follow the scribe lanes or an edge of a unique feature under microscope. You need to have no movement along the Y-axis looking at these features. You can correct for any misalignment observed by rotating/holding the joystick in a proper direction to correct for rotational error. You can also hold the red button on top of the joystick to speed up the process. (9.1.37 and 9.1.38).

Note: Holding the joystick button for a second or two will activate a buzzing sound.

9.1.41 Once there is no movement in the Y direction and pattern cruise by nicely aligned with the probe tips, then your Theta is perfectly aligned. You can stop the scan, go to next step.

Press **PAUSE/CONT** to stop the scan. You should be ready to start your measurement as per Section 9.3 or Appendices 2 and 3, as was described earlier.

Changing or Unload A Wafer

9.1.42 One can change or unload the wafer at anytime by the following steps.

9.1.43 Pressing [LOAD] button on the joystick box (Figure 19) will bring the chuck back to the lower right corner of the platform. You can push the force release button on the left side of the joystick box to lift up/release the wafer. Now you can remove or exchange wafer at this point.

HELPFUL KEYS

[x] enables the joystick to move in the x and y direction

[z] enables the joystick to move in the z direction

[lamp] turns on the lamp for the microscope

[enter] exits out of current menu if an exit option is not given.

Note: See Figures 18 and 19.

9.2 Troubleshooting Guidelines (Hardware Set Up Part)

No Image After Find Target Step (9.1.24)

9.2.1 If after pressing **FIND TARGET** button, no image shows up on the screen, follow the next steps to resolve this issue.

9.2.2 Press **CAMR** button on the joystick box, and if there is still no image on the screen, then go to next step.

9.2.3 Press load on the joystick pad, stage comes back to the right corner of the station platform.

9.2.3.1 Press the **force release** button on side of the joystick box to release the wafer.

9.2.3.2 Remove the wafer [just like in step 9.1.22, above].

9.2.4 Reboot the upper electronic rack just below the probe station called **camera lighting** by pressing the ON/OFF switch and wait for 20 seconds before turning it back on again.

9.2.5 Follow steps 9.1.22 through 9.1.24 to load the wafer back on stage and go through **FIND Target** step again.

XY Motor Blank Warning

9.2.6 You may get the **XY motor blank** warning/error, on the monitor, which will stop you from getting into the menu and various sub-menus.

9.2.7 Press **force release** button on left side of the joystick box, see Figure 19.

9.2.8 Once the stage is vacuumed down the warning should clear.

Ref Not Stored Warning

9.2.9 Try a new non-repeating target to try.

9.2.10 Press **FIND TARGET** on the control panel, Figure 18. this time find a more recognizable non-repeating target, then press **PAUSE/CONT** key on the joystick box to resolve the issue.

9.2.11 If still a problem, check your die size and remember you need to account for the scribe lane when calculating for this. Check/verify correct die size on the setup menu to remedy the problem.

Error "Clean position Z (Z-height) not adjusted"

9.2.12 [step 9.1.33] if the messages **clean position z not adjusted** or/and **continuity test z not adjusted** error/warning comes up, z height can still be set with these messages, however, it is better to perform the set ups again, as a precautionary measure to keep the probe card from getting damaged.

9.3 Software Set Up

Using the Metrics I/CV Tools

We will be using the Metrics I/CV system tools and testing tools. Shortcuts to these two programs can be found on the desktop of the computer by the probe station. Figure 1 shows the icon for these two shortcuts. **I-CV System Tools** is a package that allows you to setup the tests that you want to run. **I-CV Testing Tools** is a package you can use to run the test/s. We will not need many of the features in these two packages for simple tests. If more advanced features are needed, please refer to the help files or the online documentation:
http://www.metricstech.com/icv/icv.shtml.

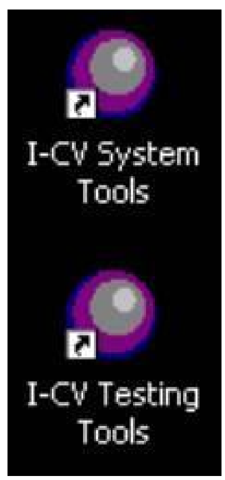


Figure 1 - Metrics I/CV Software Shortcuts

The metrics software is a modular package that is configurable from the device to wafer levels.

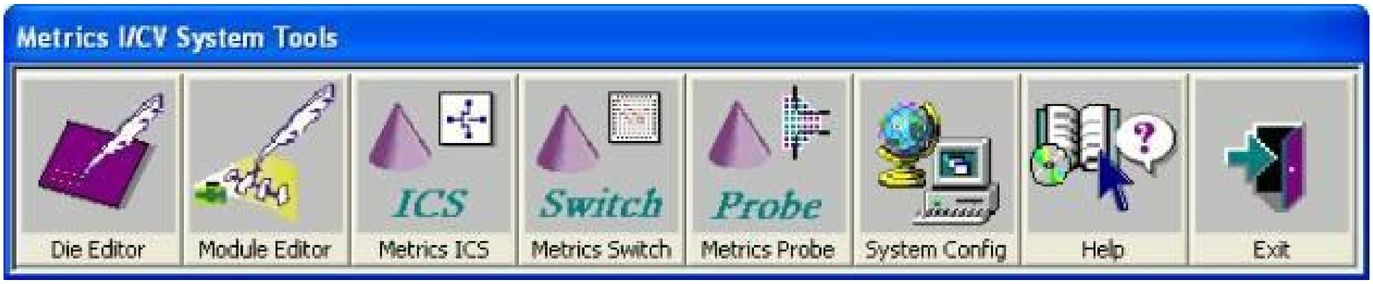


Figure 2 - Metrics I/CV System Tools Menu

How To Build Up A Test Using Metrics I/CV Tools

The following steps are necessary to build up a test plan for a particular test (Figure 3).

i. Pin assignment:connection between the pins on the probe card and measurement instruments (Metrics Switch)

ii. Measurement definition for a single device (Metrics ICS)

iii. Module definition: measurements at a single placement of the pins (Module Editor)

iv. Die module: multiple measurements within a single die (Die Editor)

v. Wafer Plan: which dice on the wafer are to be evaluated (Wafer Map Editor)

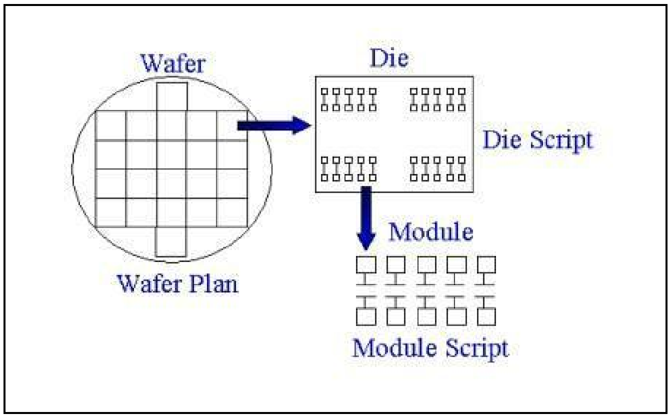


Figure 3

9.3.1 Pin Assignment

Connections between the pins on the probe card and measurement instruments are defined in the Metrics Switch software. This is where the physical connections between the device under test (DUT) and measurement channels are defined. A subset of the probe card pins is associated with the inputs and outputs of the HP-4142B.

9.3.1.1 Open the Metrics Switch program from the Metrics I/CV System Tools program bar (Figure 2).

9.3.1.2 Select New from the menu to configure a new switch setting or open an existing template to modify. If opening a template, use the **Save As…** command to save the new switch setting before modifying the file.

9.3.1.3 Verify that the file name is correct in the upper portion of the window.

9.3.1.4 Set the **Autoprobe** switch as active (Figure 4).

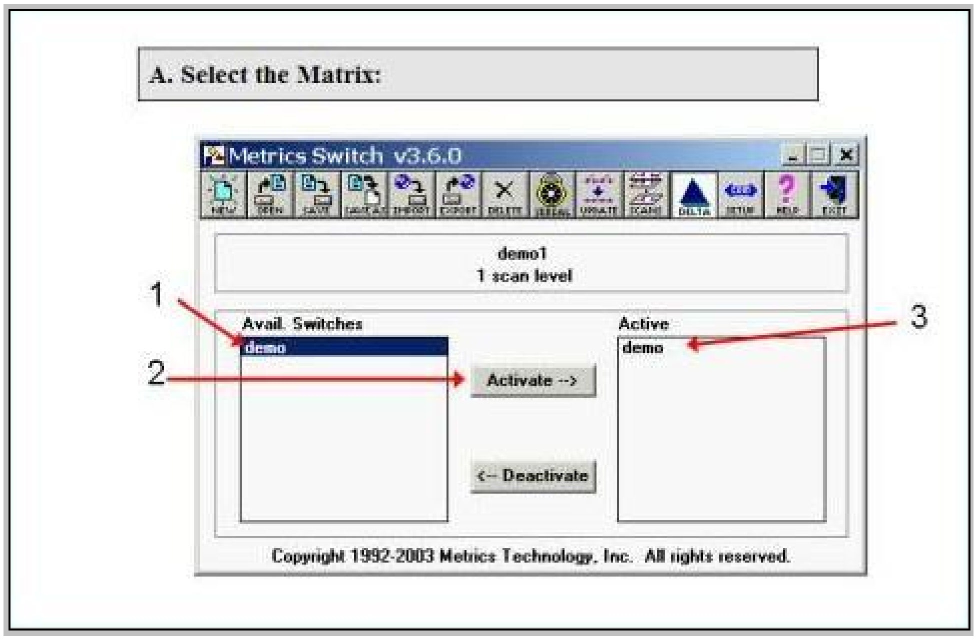
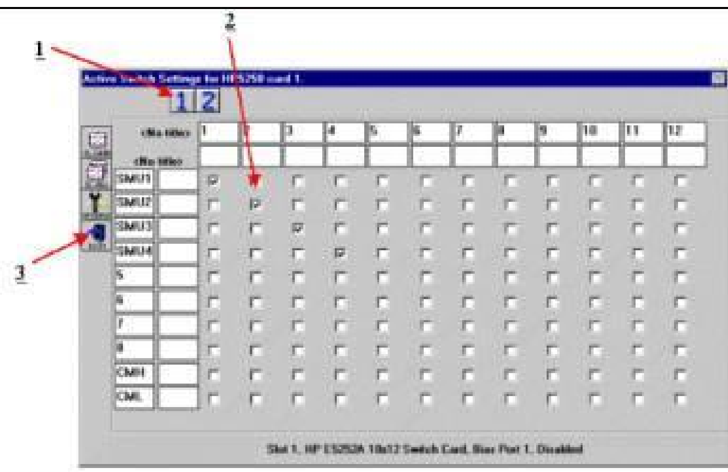


Figure 4

9.3.1.5 Double click on the active switch to configure the switch settings.

9.3.1.6 Select the check boxes that correspond to the pin to connect to the desired source/measurement channel. The upper labels on the columns correspond to the connections to the pin board that interfaces with the switching matrix. The lower column label shows the pin number on the probe card. The correspondence between the switching matrix channels and the probe card pins is determined by how the probe card is installed in the mount. The labels serve as a reference only and maybe edited for measurement clarity (Figure 5).



1. Click the button corresponding to the matrix card to be used.
2. Click each of the crosspoints to be set.
3. Click the "Exit" button.

Figure 5

9.3.1.7 Press the Exit button to close the active switch dialogue.

9.3.1.8 Save the new switch configuration.

9.3.2 Measurement Definition

Measurements for a single device are configured in the Metrics ICS software. Measurement parameters are defined and associated with a switch setting for a single pin placement.

9.3.2.1 Open the Metrics Switch program from the Metrics I/CV System

9.3.2.2 Select **New** from the from the file menu to create a new measurement.

9.3.2.3 Measurements are saved in project files with other measurements. From the File menu select Set project to define a new project or open an existing project. When the measurement is saved, it will be a unique measurement in the project file.

9.3.2.4 Select the instruments icon to select the measurement instruments. The available instruments are the HP4142B and HP4280.

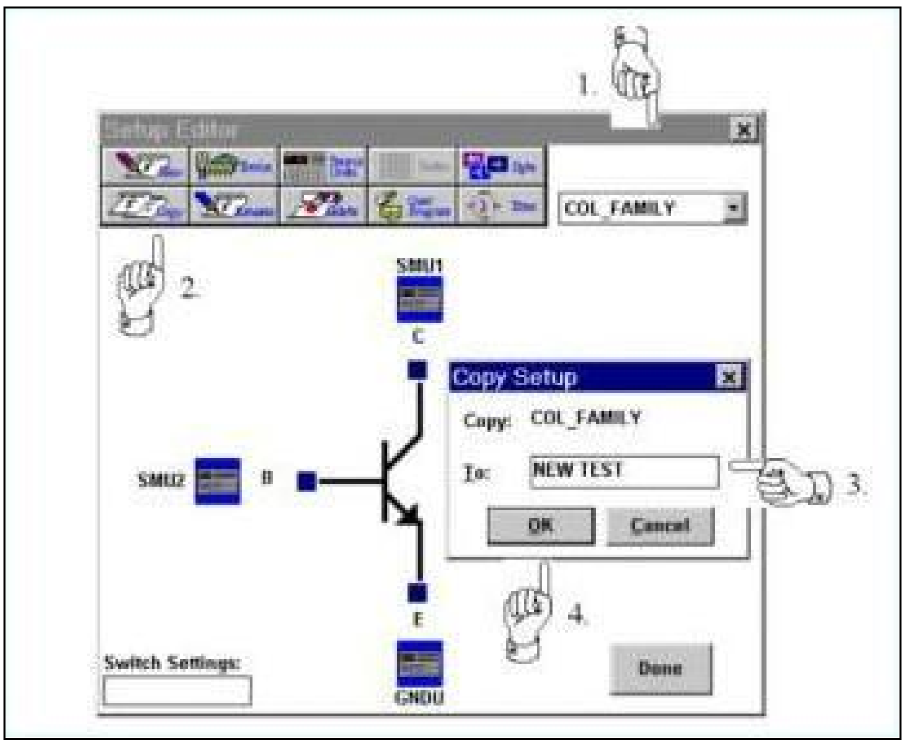


Figure 6

9.3.2.5 The Metrics Switch interface maybe controlled through the ICS software. Be sure that Enable Switch Control is selected so switch settings maybe defined in the measurement definition.

9.3.2.6 Select the **Edit Test Setup** command to configure a new measurement.

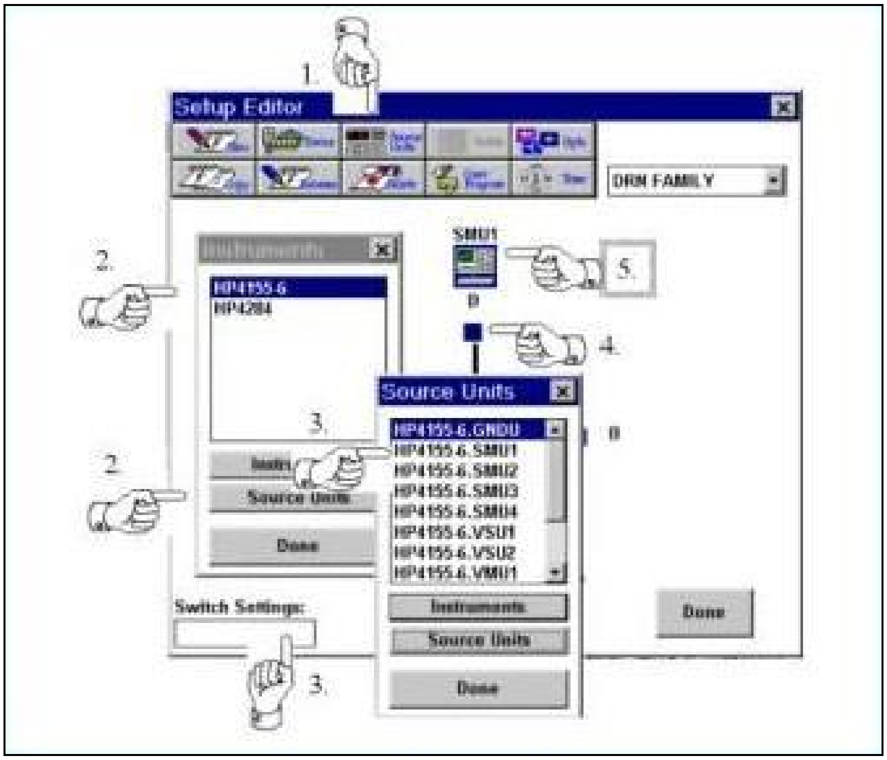


Figure 7

i. Select the New button to configure a new measurement.

ii. Select the Device button to define a new measurement with the proper polarity (Figure 6).

iii. Select Source Units (Figure 7)

1. Use the Instruments button to choose the desired source/ measurement instrument.
   a. You cannot use source units from more than one instrument.
2. Use the Source Units button to pick the desired measurement channel.
   a. Highlight the desired channel in the Source units window and select the appropriate terminal on the device. Repeat for each terminal on the device.
   b. Parameters for the source units can be configured by clicking on the source unit icon at the device terminal.

iv. Select the Switch button to choose the switch configuration to use with the measurement.

v. The Time button will open the time based measurement configuration dialogue.

9.3.2.7 To test a measurement or make a spot measurement, position the probe card at the appropriate device and open the Measurement Remote Control (Figure 8).

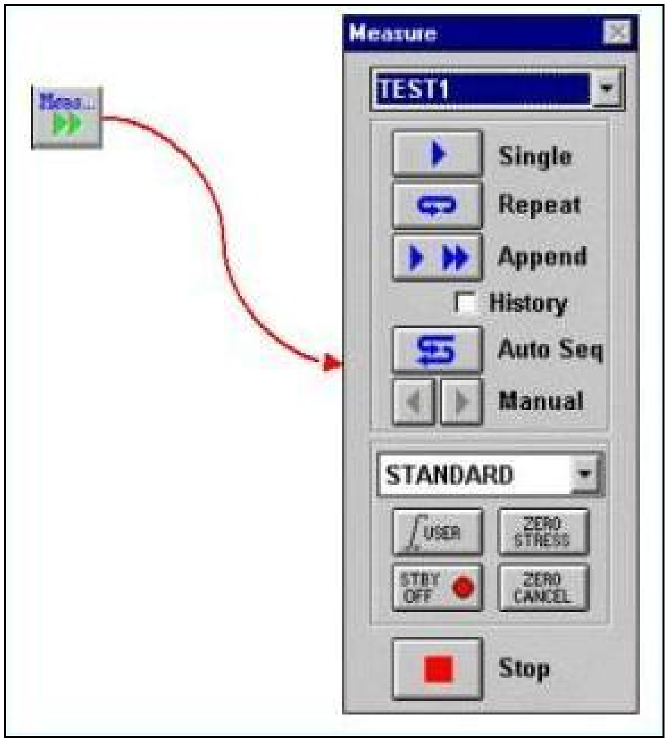


Figure 8

9.3.2.8 The data plotter is used to display data taken by the current measurement (Figure 9).

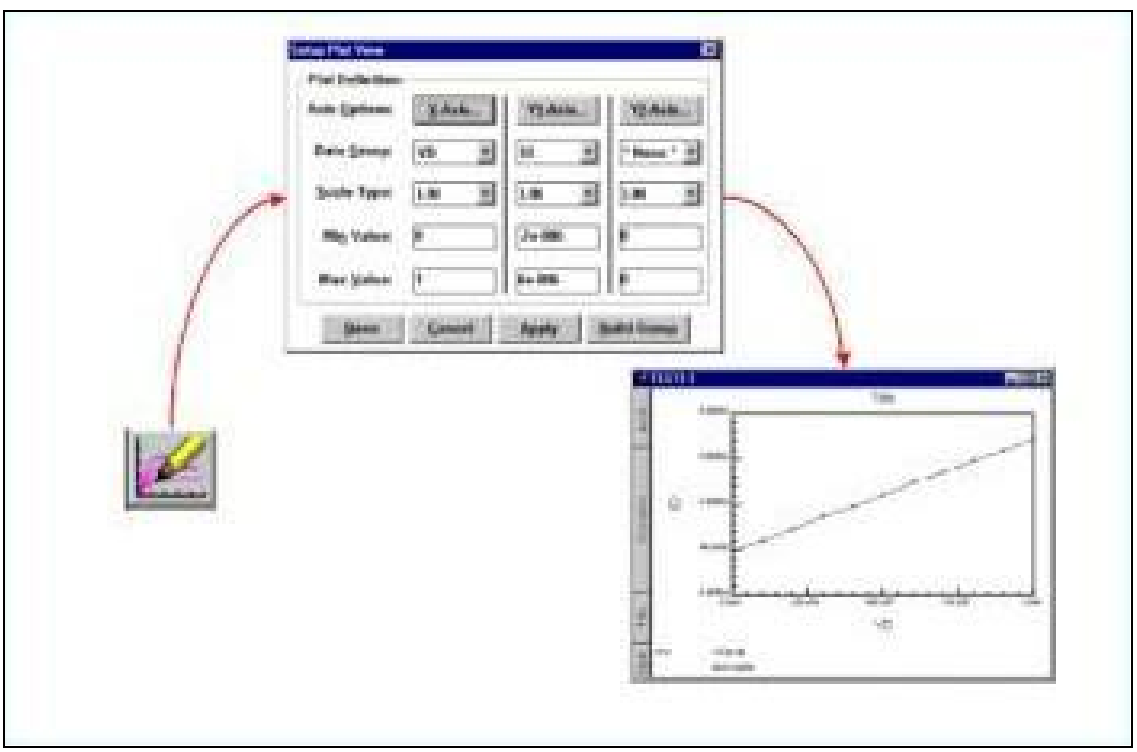


Figure 9

9.3.2.9 The transform editor is used to automatically extract parameters from the current measurement. The extracted parameter will be saved in the raw data file (Figure 10).

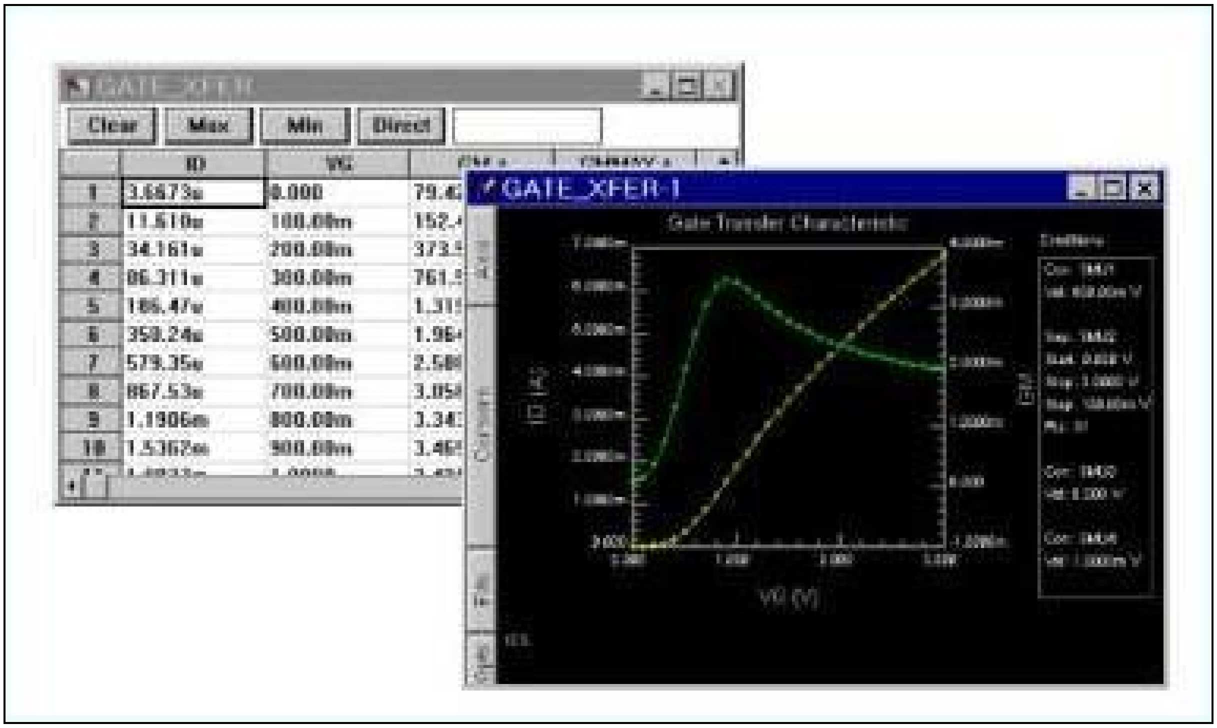


Figure 10

9.3.2.10 Save the measurement to the appropriate project file.

9.3.3 Test Module

Defines the switch settings and measurements to be executed at a single placement of the pins. Test modules are defined in the Module Editor. Parameters and variables may be passed to the measurement in the module script.

9.3.3.1 0pen the Module Editor from the Metrics I/CV System Tools program bar.

9.3.3.2 Select a new or existing script file (*.scr) from your directory.

9.3.3.3 Use the **Add Line** and **Insert Line** commands to add new measurements.

9.3.3.4 A basic script file will consist of connecting a switch setting and running an ICS Test.

9.3.3.5 More advanced scripting can be executed through the module editor. Refer to the help file for definitions and examples of the available commands.

9.3.4 Die Module

Multiple measurements within a single die are defined in the Die Editor. The die script defines the movements of the pins within a single die and which module script is to be executed with each pin placement. All movements are defined with respect to the starting location that is aligned by the user when the wafer is initialized.

9.3.4.1 Open the Die Editor from the Metrics I/CV System Tools program bar.

9.3.4.2 Select a new or existing die file (*.die) from your directory.

9.3.4.3 Use the **Add Line** and **Insert Line** commands to add new pin placements with in a die and configure the module script to run.

9.3.4.4 A basic die script will consist of Module Move step and an execute module script step.

9.3.4.5 All movements are defined with respect to the initial position that is manually defined when the wafer is loaded.

9.3.5 Wafer Plan

The die size is specified in the wafer plan along with which dice on the wafer are to be evaluated.

9.3.5.1 Open up the Metrics Probe program, an option available on the Metrics I/CV System Tools program bar. This will open the Wafer Probing Tools program bar.

9.3.5.2 Open the Wafer Map Editor from the Wafer Probing Tools program bar.

9.3.5.3 Define a new file or select an existing file from your directory to edit. Do not change any of the default wafer maps.

9.3.5.4 Enter the appropriate sizes for the wafer and die size to be tested.

9.3.5.5 Select the die locations to be visited on the wafer map by clicking in the appropriate cell.

9.3.5.6 Define the origin location on the wafer map. The origin of the coordinate system may be specified in the text fields on the left (Figure 11).

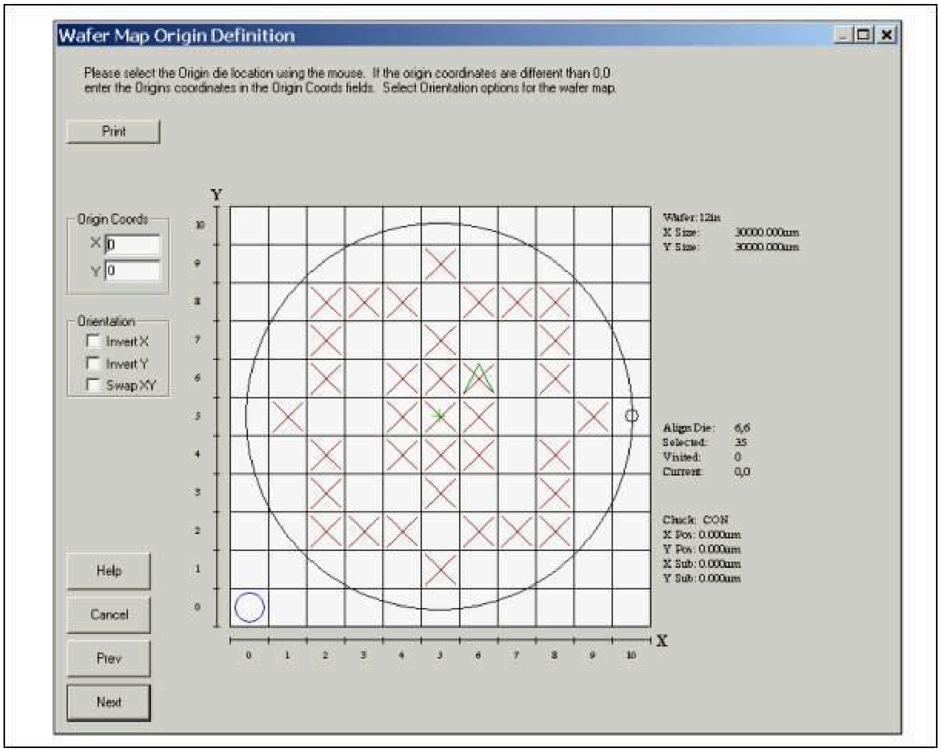


Figure 11

9.3.5.7 Define the location of the alignment die. This is the location where the initial alignment will be made prior to beginning a measurement (Figure 12).

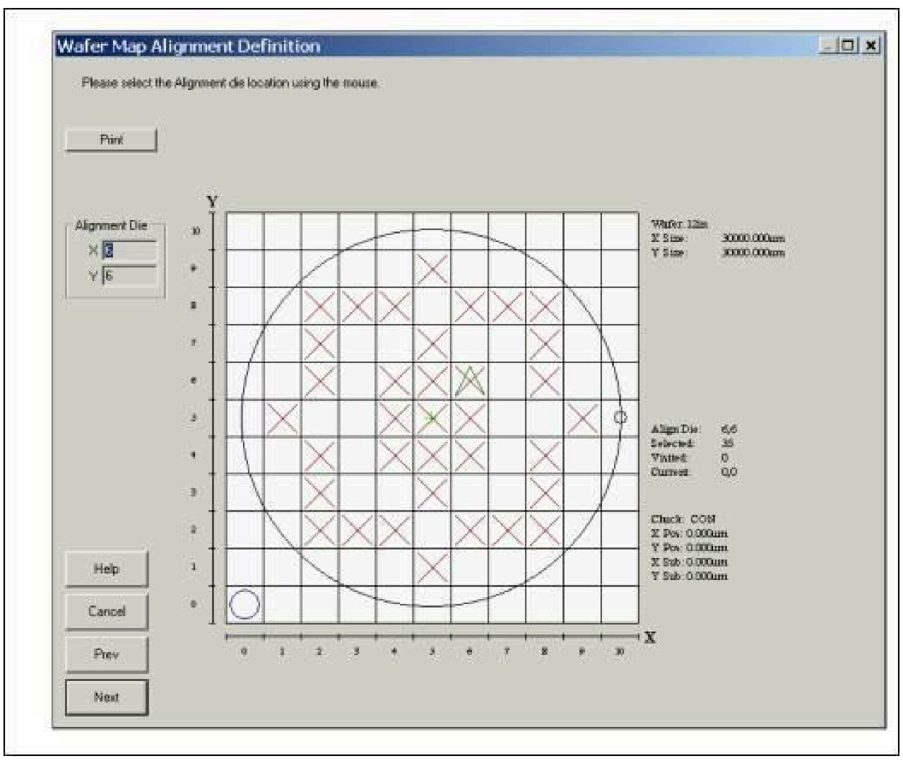


Figure 12

9.3.5.8 Select Next to finish and save the file.

9.3.5.9 The movement through the wafer map may be verified using the prober controller program.

i. Open the prober controller program from the Wafer Probing Tools program bar.

ii. Select the appropriate file when prompted.

iii. Check that the probe card is at the alignment die coordinates.

iv. The entire wafer map may be stepped through using the Play command.

9.3.6 Data collection

First check to make sure the switch box located under the PC terminal is switched onto PC, see Figure 13.

9.3.6.1 Open the Auto Test program from the Metrics I/CV Testing Tools program bar.

9.3.6.2 Select a new data file.

9.3.6.3 Select the appropriate wafer map.

9.3.6.4 Align the probe card to the initialization position on the alignment die.

9.3.6.5 Select the die module to collect data.

9.3.6.6 Start the data collection.

9.3.7 Extract Data

Data may be extracted using the Metrics Miner program. This program has the capability to return statistics and display wafer maps of measured parameters. Data can be exported to a tab delimited text file through the Metrics Miner interface. The Data Browser may be used to plot the measurement results for data in vector format

10.0 <u>Figures & Schematics</u>

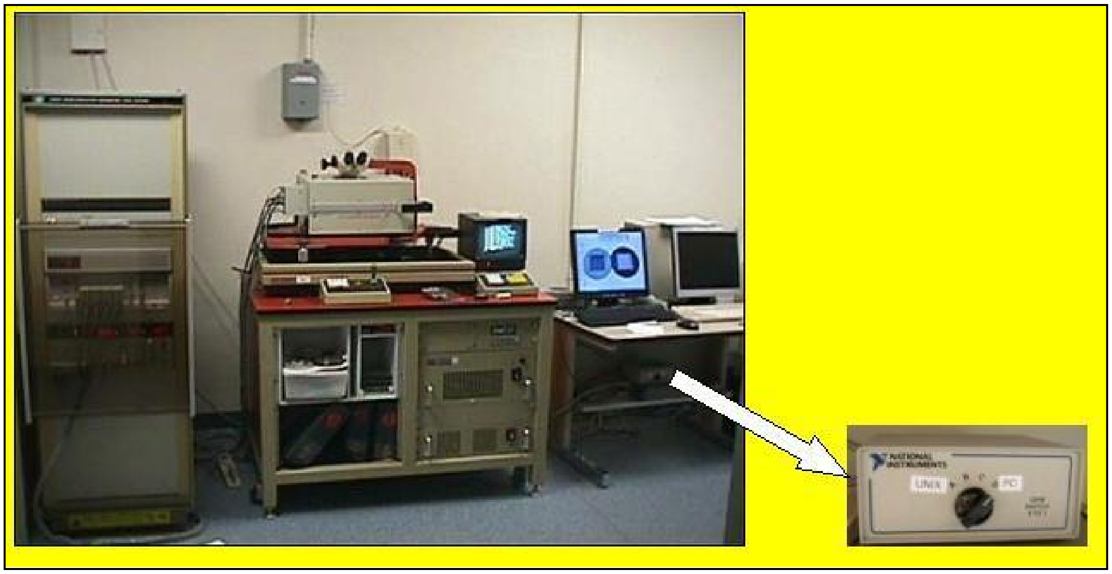


Figure 13 - Auto Probe Station

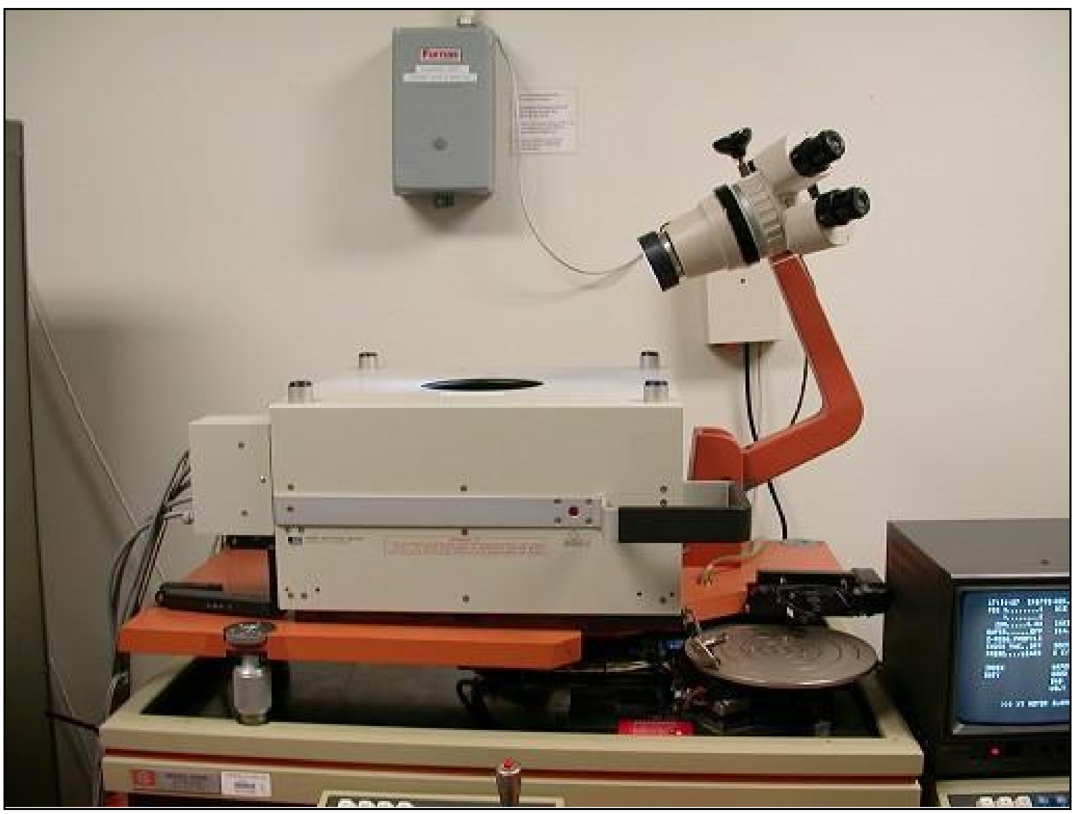


Figure 14 - Microscope and 4085A Switching Matrix

Figure 15 - Side Latches on Microscope and 4085A Switching Matrix

Figure 16 - Probe Card in the Cover With the Tips Down

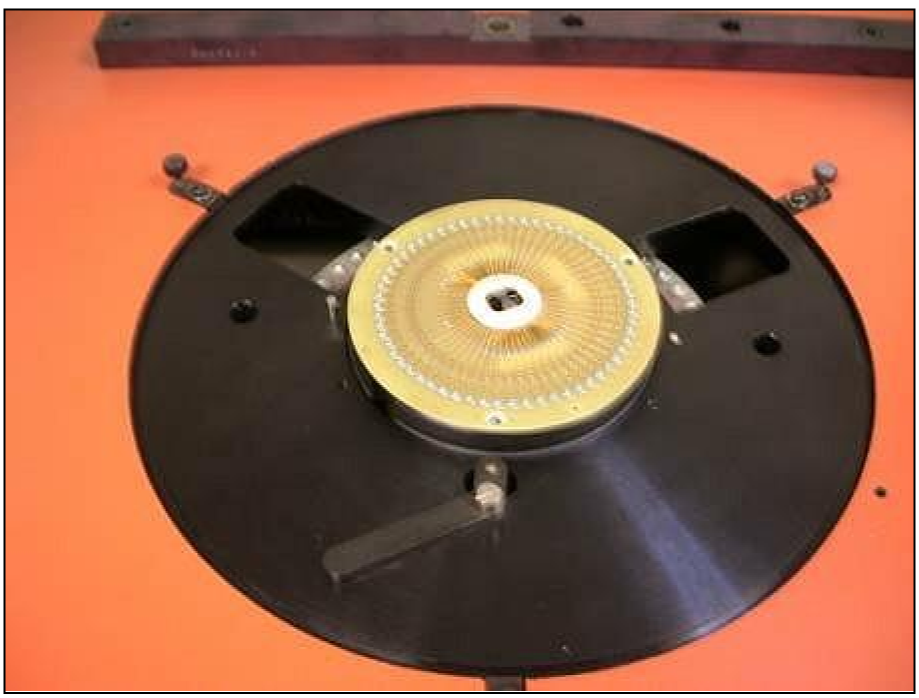


Figure 17 - Probe Card in the Cover With the Tips Up

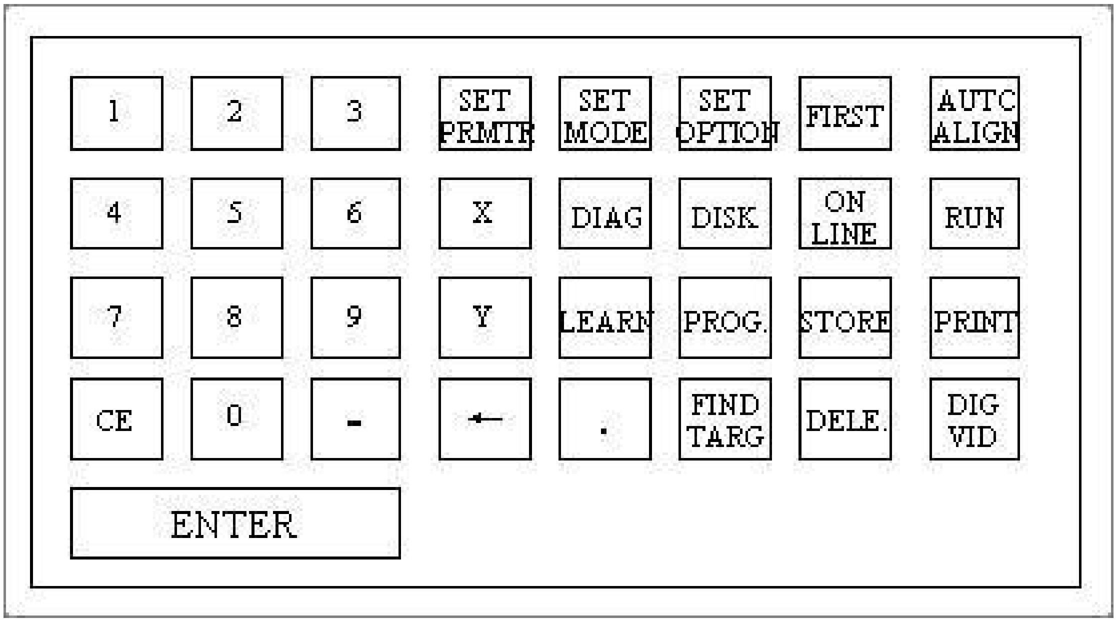


Figure 18 - Auto Probe Station Control Key Pad

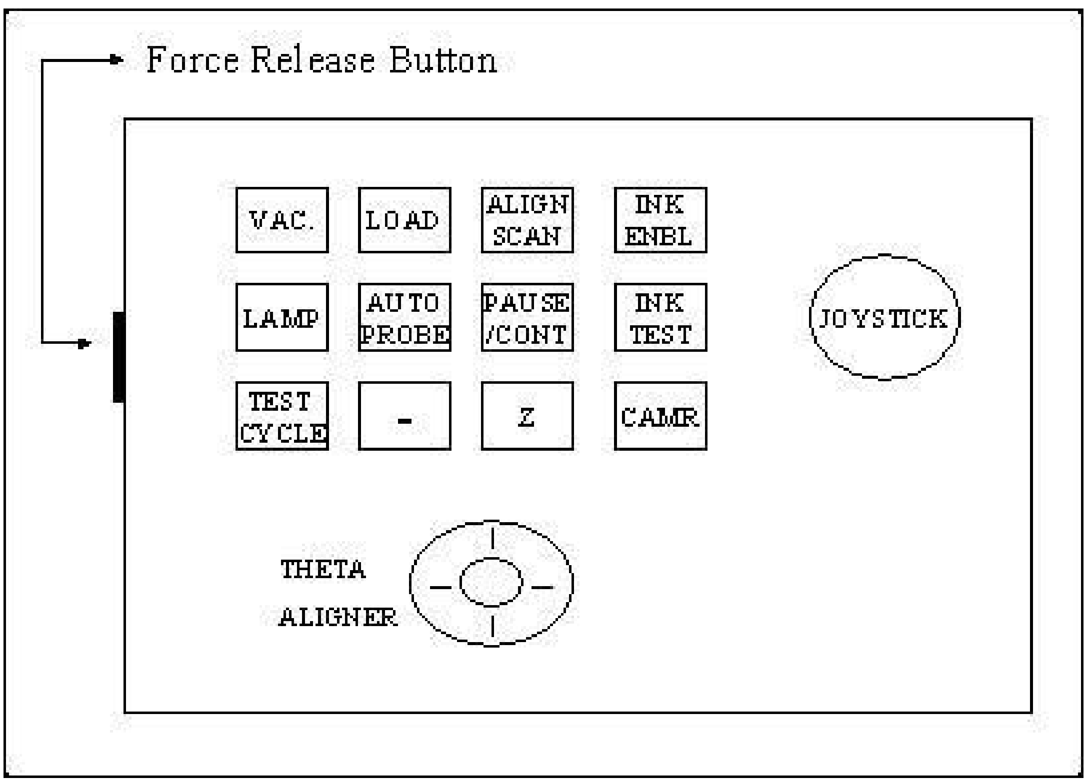


Figure 19 - Joystick Box

Appendix 1

Auto Probe Station Parameter Screens

MENU A - Initial screen # One can always press enter to get back to this screen. If you are in submenus, just keep pressing enter or quit menu if given the option.

POS X	0	DIE X	8.46667
Y	0	Y	8.46667
ZDN	0.00	INKER	DIS
WAFER	OFF	DIA	100 MM
Z-MODE	PROFILE		
CHUCK VAC	OFF	SECS	DIS
PROBE	-> MATRX	X I/O	OFFLINE
FF		WAFER #	1
		GOOD DIE	0
		BAD DIE	0
		UGLY DIE	0

>> X Y MOTOR BLANK <<

MENU B- SET MODE # To get to this menu, press [SET MODE]

01	METRIC/ENGLISH	METRIC	
02	AUTO-PROBE PATTERN	MATRX	
03	PROBING MODE	MENU	# This option brings up MENU H
04	Z-TRAVELING MODE	PROFILE	
05	MICRO PROBLING	ENB	
07	I/O CONTROL	MENU	
08	DATA LOGGING MODE	MENU	
09	MISCELLANEOUS OPTIONS	MENU	
10	ASSIGN LOGICAL INK-CODE	MENU	
11	THETA COMPENSATION	DIS	

MENU C- SET PARAMETER # To get to this menu, press [SET PARAMTER]

01	DIE	X	8.46667	MM
		Y	8.46667	MM
03	PRESET	X 4	Y 3	DIE
04	WAFER DIAMETER		100	MM
05	ALIGN SCAN VEL		2000	MPS
06	Z OVERTRAVEL		7.00	MILS
07	Z CLEARANCE		7.00	MILS
08	Z UP LIMIT		330.00	MILS
09	Z DOWN LIMIT		200.00	MILS
10	Z ALIGN		200.00	MILS
1	Z UNDERTRAVEL		0.00	MILS
12	NEXT PAGE			

[PAGE 2]

01	Z SCALE UNITS/MIL			8
02	SET RUNTIME DISPLAY CLOCK			
03	AIR-SENSOR	X 47300	Y 46600	
04	TEMP. COMPENSATION			AUTO

05	PROBE COUNT	184
06	INK DOT COUNTER & TIMER MENU	
07	PROBING TIME (HR & MIN)	0:00
08	% YIELD TO PASS WAFER	0
09	MAX X-Y VELOCITY	10000 MPS

MENU D-SET OPTION # To get to this menu, press [SET OPTION]

01	AUTO-LOAD SWITCH	DIS	
02	AUTO-ALIGN SWITCH	ENB	
03	AUTO-PROFILE SWITCH	ENB	# After ENB this switch, MENU E pops up
04	WAFER ID READER	DIS	
05	SECS PROTOCOL OPTION	DIS	
06	WAFER MAPPING OPTION	DIS	
07	HOT CHUCK OPTION	DIS	
08	AUTO TEMP COMPENSATION	ENB	
09	INK DOT INSPECTION	DIS	
10	PROBE MARK INSPECTION	DIS	

MENU E-AUTO PROFILER OPTION MENU # Submenu of MENU D

01	PROFILE WITH FIND CENTER	ENB
02	ENHANCED PROFILE	DIS
03	PROFILER RETRIES	4
04	USE AUTO DIAMETER	ENB

MENU F-AUTO ALIGN OPTION MENU# Submenu of MENU D

01	STOP IF AUTO ALIGN FAILS	DIS
02	AUTO ALIGN MODE	FINE
03	Q THRESHOLD	NORMAL
04	TEACH DIE CORNER	DIS
05	LIGHT CALIBRATION	MENU
06	PERFORM AUTO LIGHT TRAIN	
07	PERFORM COARSE THETA ALIGN	
08	PERFORM CAMERA CALIBRATION	
09	SELF-TEACH SWITCH	MANUAL
10	USE CENTER OF REFERENCE	DIS

MENU G-PROFILER MENU# To get this menu, press [PROG]

PRESS	Z	TO ENABLE Z JOG
PRESS	X	TO ENABLE X-Y JOG
PRESS	1	TO SET PROBE CLEAN XYZ
PRESS	2	TO SET PROBE TIP CENTER
PRESS	3	TO SET Z HEIGHT
PRESS	4	TO PROFILE (STANDARD)
PRESS	5	TO CLEAN PROBE TIPS
PRESS	6	TO SET PROBE TEST XYZ
PRESS	7	TO TEST PROBE TIPS
PRESS	8	TO SET CAMERA CENTER

PRESS 9 TO GET NEXT PAGE MENU
PRESS ENTER

PAGE 2

01	AUTO ADJUST CLN/CT Z HT	ENB
02	MAUAL SET CLN/CT Z HT	ENB

MENU H-PROBE MODE MENU # Submenu of MENU B

01	PROBE QUADRANT	3
02	COORDINATE QUADRANT	3
03	REPROBE LIMIT	0 DIE
04	PROBE CLEAN LIMIT	0 DIE
05	MATRIX	X 5 Y 5 DIE
08	USE AUTO DIAMETER	ENB
09	CONT. AT LAST DIE	DIS
10	PROBE TIP SCRUB	DIS
11	INKING MODE	MENU
12	UGLY DIE MENU	

MENU I - I/O CONTROL MENU # Submenu of MENU B

01	I/O PROTOCOL	ENHANCED	
02	EXTERNAL I/O MODE	MENU	# This option brings up MENU J
03	I/O PORT	GRIB-SP	
05	GPIB ADDRESS	14	
06	TERMINATOR	LF	
07	GPIB SRQ	ENB	
09	TIMEOUT TIMER 1 IN MS	5000	
10	TIMEOUT TIMER 2 IN MS	5000	

MENU J - ENHANCED EXTERNAL I/O MODE MENU (PAGE 1) # Submenu of MENU I

01	MF/MC ON X-Y MOTION	ENB
02	MF/MC ON Z MOTION	ENB
03	MF/MC ON OPT DEVICES	ENB
04	MF/MC ON REST OF CMDS	ENB
05	TEST START MESSAGE	ENB
06	TEST COMPLETE MESSAGE	ENB
07	PATTERN COMPLETE MESSAGE	ENB
08	PAUSE/CONTINUE MESSAGE	DIS
09	ALARM MESSAGES	DIS
10	WAFER COMPLETE MESSAGE	ENB
11	ENHANCED PC MESSAGE	ENB
12	NEXT PAGE	

[NEXT PAGE]

01	ENHANCED TS MESSAGE	ENB

02	UGLY DIE REPORT	DIS
03	MAP TRANSFER RETRIES	6
04	SEND MAP COORDS WITH TS	DIS
05	SEND EC/BC MESSAGE	ENB

Appendix 2

Sunbase Program

The Microlab's Auto-Probe station can also be run by a Unix based program called sunbase, which has gone through some revisions currently on its third revision called **sunbase3**. The sunbase program was initially capable of measuring layouts/test structures with a 10-pin arrangement. This capability was later enhanced to provide access to 30-pin devices. A new testing interface program (~eglas/src/sunbase2) was then developed to support the new HP 4142B DCS measurement subsystem. This version of the sunbase program is however, no longer valid, as we have moved to sunbase3 program.

Another new enhancement came in, due to the requirement of testing an 8-bit adder device. This measuring layout will require a 32-pin arrangement. A whole new set of software interface was developed under (~eglas/src/sunbase3), which supports 32-pin measurement of adder. This new interface is also fully supports 30-pin and 10-pin existing modules in sunbase and sunbase2 directories. Therefore, the ~eglas/src/sunbase3 should now be the **only** interface to use to test any device.

In conjunction with the new interface program that was developed, and the new adder module, adder.c, a new parsing function, adder_parse.c, was also created to take the user's additional input configuration. The detail format of this new input file is stated below.

A. Testing Through Sunbase3 Program.

Sunbase3 codes/program is currently available on the tellurium Unix work station (running Solaris) just to the right of the probe station. The sunbase3 program requires proper setup files (routines) to perform specific test/s, as well as a wafer map and a die map to properly step the probe tips over the wafer and within the die/s, respectively. Measurement results are automatically written into **output. text** and are processed to give **final.out** data. Members often copy desire routines in their own directory, invoke and collect test results in the same directories. Here is how you can start your electrical measurements/test:

A.1 Log into the Unix station (tellurium), just to the right of the probe station by entering your user name and password (silicon account).

A.2 Right click on Solaris screen and select **tool** then in the drop down menu, **terminal**.

A.3 You should now be at the **%tellurium** prompt, where you can change directory or start a test routine. Remember you need to be in the directory that contains pertinent die.map, and prober.text files. The sunbase3 program looks into the prober.text to perform the required test, which is specific to a particular routine.

A.3.1 Start the test by typing **~eglas/src/sunbase3/sunbase3.**

A.3.2 Again, the sunbase3 program looks into the prober.text to tests different subroutines called upon.

Note: Please note that the output file/s will get written to the current directory.

A.4 After the test is completed (sunbase3) exists, then press [LOAD] button to unload the wafer. You can now load the next wafer.

Note: Be sure to bring chuck back into the lower right corner before leaving the probe station, by pressing the [LOAD] key.

B. Setup Files/Sub-Routines

These are default set up files available in the eglas/src/sunbase3. These files can be copied into your directory, and also be modified to serve your specific measurement requirement/s.

B.1 adder.text

This is a configuration file specific designed for adder module. If you are not using adder module, please skip this file. The following is a sample format of the adder.text. Everything after a '*' on the same line is a comment.

```
* This is a input format for a 8-bit adder module
*
*         A7 A6 A5 A4 A3 A2 A1 A0
*         B7 B6 B5 B4 B3 B2 BI B0
*
* Sample     0 0 0 0 1 0 0 0
*         0 0 0 0 0 0 0 0
* 0 -> low-input pin
* 1 -> high-input pin
01000000
00000000
```

B.2 output.text

The output is written into **output.text** in the order it was measured. These files are overwritten by new test results. Make sure to save your results.

B.3 Available Sub Routines

The user can utilize a set of existing measurement subroutines by configuring two text files, or may choose to add extra subroutines to the current library (~eglas/src/sunbase3/ on the Microlab's main file server (silicon). Available subroutines are listed in Table 1, below.

Measurement	Routine Name	Author	Output
I_d-V_{ds} curve	IdVds	V. Gutnik	I_d-V_{ds} characteristic
I_d-V_g curve	IdVg	V. Gutnik	I_d-V_g characteristic
Four Point Probe	4ptprb	V. Gutnik	Resistance
Van der Pauw	VDP	V. Gutnik	Sheet resistance (R_s)
Split-Cross-Bridge	SCBR	D. Rodriguez	R_s, ΔW, Spacing, Pitch
Fallon Ladder	Fallon	D. Rodriguez	Ladder resistance, min. line width resolved
Contact Resistance	Conr	D. Rodriguez	Contact resistances (left, right, avg.)
Comb Defect	Comb	D. Rodriguez	5 Binary results showing shorts (defects)
Serpentine Resistance	Serp	D. Rodriguez	Resistances for 5 serpentines in pad set
Setpentine/Comb Defect	SerpComb	D. Rodriguez	2 Binary results showing opens/shorts (defects)
Individual MOSFETs	VTWDLD	V. Gutnik, S. Fang, and D. Rodriguez	threshold voltage(vt), delta width(wd), delta length(ld), body *effect (bodye),
	Kelvin/Kelmod	Tim Duncan	Support 30-pin testing measurement
	Adder	T.K. Chen	Support 8-bit (32 pins) adder device

Table 1 - Available Subroutines

B.4 prober.text

prober.text must be in the directory from which sunbase3 is run.

This file specifies the tests that the user wants to run. It starts with an array of characters, either 0 (zero), 1 (one), or x, that describe the map of dies on the wafer:

```
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 x 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

This is a crude model of the wafer to be probed: x marks the location of the initial die, 1 marks a die that will be probed, 0 marks a die that will not be probed. Sunbase3 will step from the die marked x through all dice marked 1 in a non-obvious, but complete pattern. It is suggested that the dies marked with an x be near the center of the wafer, to alleviate misalignment problems.

This is followed by names of modules to be run on specific devices, and the names of the devices (from die.map) to be probed. The list of devices passed to any routine is terminated with a period. Lines beginning with * are ignored (comments) and **+** should be used to pass parameters to the routines.

The module name must match one of the names defined below (case is important) and the device name must match one of the devices from die.map exactly.

So, a few lines may look like this:

```
IDVG    m2      mosfet31
IDVDS   + VDSstop = 2   + VDSstep = .05
mosfet31
```

Sunbase3 will run the IDVG routine on m2 and mosfet31, then run IDVDS (with the given changes) on mosfet 31, print out the data, step on to the next die, and repeat the process.

B.5 die.map

This file describes the location of the devices on the die, as well as switching matrix connections to the probe tips (voltage/currents values). In principle, it should contain all the necessary information about the die that can be known before testing. The general format is:

DeviceName X,Y Terminal1, Terminal2, ..., Other Info.

The first letter of the device name determines the type of the device. X and Y are measured in microns, and increase to the right and up, respectively.

Formats for the devices already created follow:

```
mosfet x,y drain gate source bulk
4ptprb x,y iin gnd v1 v2
conr x,y iin gnd
fallon x,y iin gnd v1 v2 lw
scbr x,y i1 i2 i3 v1 v2 v3 v4 v5 v6 v7 layer
serp x,y
serp_comb x,y
cchain x,y
```

A typical mosfet might look like this:

```
m2 0,0 7 8 4 5
```

The **m** shows it is a mosfet. The only way to refer this device is by name. The two integers separated by a comma give the location of this device in microns relative to the starting location. Thus, the **0,0** indicates that sunbase3 will assume the probes start out touching this device every time the die is probed, specifically with probe #7 on the drain, #8 on the gate, #4 on the source, and #5 on the bulk. (Actually, unless the device is called in prober.text, sunbase3 will ignore it completely, so there can be several different devices at any location as long as prober.text doesn't call more than one per run. Probes are numbered clockwise from the upper left probe as they appear in the microscope:

```
probe #'s:
1 2 3 4 5
10 9 8 7 6
```

Another example: mosfet31 320,640 2 3 9 10

Another mosfet, 320 microns to the right and 640 microns up from m2.

die.map can contain any number of entries, and sunbase3 ignores blank lines and lines starting with * which can be used as comments. The line **@home 0,0** must appear in die.map. Sunbase3 goes to this device after the wafer is finished to put the probes on a device similar to the ones it started on.

Sunbase3 does no error checking of die.map. If necessary fields are not specified, it will crash.

C. Sunbase3 Programming

1. Adding a Routine (i.e., IC_VCE)

- ► Write it. Probably best to find a routine similar to what you're writing and copy it - (i.e. that's why IdVdg looks like IdVds).
- ► Ad it to modules.c and modules.h in the obvious way.
- ► Add it to make the file.
- ► Recompile the program sunbase3. (Please contact the system administrator to compile the code for you)

2. Adding a Structure

- ► Add the name to the DevType enumeration in hash.h.
- ► Add the structure declaration to hash.h.
- ► Add a line to identify the device in FindDev (hash.c).
- ► Write a routine to **fill** the structure, in hash.c, and a declaration for the fill routine into hash.h.

3. Usage Comments/Troubleshooting

- ► Be sure to run the most current version of sunbase3 (the binary is located under /usr/users/micro/bin/); be sure you have the previous path define in your login profile.
- ► Sunbase3 reads the prober.text in the directory from which it is run if you're not sure, just before you type **sunbase3**, type **more prober.text**; you should see the prober.text you expect to run.
- ► Check spelling. Sunbase3 will ignore routines names that do not match the spelling (and capitalization) in modules.c, and will abort if it looks for a non-exist device.
- ► Don't run more than one version of sunbase3 at a time; the second incarnation will not be able to use the gpib and will stall. This tends to happen during debugging.
- ► Start with the probes down, (more correctly, the chuck up). There's no error checking for this.
- ► Check the position on the wafer- sunbase3 has no way of checking if it has moved off the wafer; this is important when probing several dice.
- ► When running long jobs, redirect screen output to /dev/null; you won't be around to see it and it slows things down.

4. Description of the Code

Sunbase3 consists of a core that handles communication to the instruments, reading configuration files and finding devices, and a set of **modules** that do the measurement. For the sake of consistency, I will call the actual user routines (i.e. FPP_meas()), **mroutines** and anything passed to them, be it device names or options, **mparameters**.

The core files are: main.c initial.c hash.c instruments.c modules.c modtools.c. Each routine is described in turn, in order of appearance in the source file.

(a) main.c ::: The primary probing loop

- ► main: initialize, probe, clean up. No actual work done here, just function calls.
- ► probe: This is the loop that does all the probing. It calls preprocess to extract the wafer description (i.e. which dice to probe) and sets puts bookmark at the place where probe information (what routines and what devices to probe) starts. Then the repeated loop starts: As long as there's another die to probe, go back to the bookmark, parse, call the mroutine, continue to the end of the file. At the end of prober.text, move the wafer to the

next location and loop. At the end of the wafer, move the probes to the location of the device they started on.

- ► parse: This collects the name of the mroutine (I.E. 4ptprb) as written in prober.text and the mparameters into a list, and figures out what the function name of mroutine is (in this case, FPP_meas). Basically, read a line, if it isn't empty and isn't a comment, add it to the list until the line start with a '.' which signifies the end of the list. Then call another routine to figure out which mroutine the first string in the list corresponds to.
- ► preprocess: Count the number of dice to probe and set the bookmark. This should be a little smarter, but isn't.
- ► Wafermove: Finds the next wafer to probe in the wafer, updates the current location (Xcurrent, Ycurrent) and calls **move**.
- ► Pdie: Prints the current die coordinates separated by tabs into a string.
- ► cleanup: Should close files, check the gpib, etc, but it just prints a message to the eglas.

(b) initial.c ::: Preliminary stuff- initialization, etc.

- ► opening_message: Hello to terminal and eglas.
- ► startup: Open the probefile (typically prober.text) the outputfile (typically output.text) and device file (typically die.map) for reading and writing as necessary. Checks the gpib (sort of), converts the device file into an array.

(c) hash.c ::: Routines that relate to devices (i.e. mosfets)

► FindDev: Converts a string like **m123 0,0 1 2 3 4** into a device structure for, in this case, a mosfet, by calling the appropriate routine.

► *Fill: Actual routines to accomplish the above. There should be one for every type of device.

► hash: This should put the devices into some sort of hash table for quick searches. In fact it just puts everything into a sequential array.

► locate: Given a device name, looks up the entry (originally from device.map, put into an array by hash) corresponding to the name.

(d) instruments.c ::: Routines that relate to test equipment (i.e. eglas)

- ► devwrt: Send characters over gpib to an instrument. Adds the requisite newline.
- ► init_devs: Opens the device files for the various instruments, sends some initialization codes. This should detect errors, but doesn't.
- ► check_gpib: Placeholder for any real testing of communication.
- ► screenwrite: Writes a string to the screen of the eglas terminal.
- ► move: Lowers the chuck, moves it (in die steps), catches acknowledgement from eglas (though it doesn't check errors) and raises the chuck back. Also resets the **microdie**, or intradie position.
- ► align wafer: Prompts the user through alignment. Written out during testing.
- ► prober_self_test: This will someday check all the instruments and the gpib before operation. Does absolutely nothing now.

- ► need_move: keeps track of current location within a die to avoid moving by 0 (which takes time and chuck lowering/raising).
- ► MoveTo: Lowers the chuck, moves it to a new location *within the die* and raises the chuck back. Updates current location.
- ► connect: **port** refers to the signal input/output channel on the 4141A. **pin** refers to one of the ten probes that touch pads on the wafer. If both port and pin are nonzero, this connects the given port to that pin through the SMC. (NOTE: The SMC will not allow two ports to connect to the same pin. For example, if port 1 is connected to pin 10, and you send a command to connect port 2 to pin 10, the SMC will break the connection between port 1 and pin 10 before making the new connection. It is best to make this explicit in the code, though.)
- ► ReadBuf: Returns, as a string, the contents of the DCS buffer.
- ► DCShold: Directs the DCS to source a current or voltage at a given setting and with a specified compliance. **source** is the port number, **mode** is either
- ► DCSMeasure: Single point measurement of a given channel and mode. Note that the DCS will not measure the voltage of a voltage source or the current from a current source directly, so if you really need to confirm the source output, you have to connect another channel to it. See connect for caveat. DCSmeasure returns the contents of the DCS output buffer.
- ► DCSsweep: Sets up sweep parameters for a channel; no measurement is actually done. Some of the options (i.e. linlog, SECONDARY vs. PRIMARY sweep) have not been fully tested.
- ► DCStrack: is the parallel to DCSMeasure – given a list of sources, it directs the DCS to keep track of the measurements for those channels, then triggers a sweep measurement. DCStrack returns a string; see the DCS manual for the specific format.
- ► dmake: This routine parses the output string into a dtype, keeping all the information (i.e. flags, which channel it was, etc.) his should be used to parse the output of DCSMeasure.
- ► DatFormat: Parses an array, as returned by DCS track. Check DCS documentation for the format of the output, or look at idvds.c for an example.
- ► numpoints: Queries the DCS for the number of measurement points in the last sweep; this is mostly a sanity check, because DatFormat can simply count the points as they're read.

(e) modules.c ::: Command-> routine parser

- ► ident: Matches a string from prober.text to the appropriate mroutine.
- ► Ignore: Throws away arguments passed to a command that doesn't match an mroutine, and handles comments, etc.

(f) modtools.c ::: Declarations and utilities for the modules

- ► V_diff: Finds voltage difference between two given pins and cleans up.
- ► Discon: Disconnect a list of pads.

5. List of Current test modules

The test modules (idvds.c, FPP.c, Fallon.c, etc) are located in ~eglas/src/sunbase. The existing modules are:

Module	Writer	What it does
~eglas/src/sunbase3/scbr.2.c	D. Rodriguez	R_s, ΔW, Spacing
~eglas/src/sunbase3/FPP.c	V. Gutnik	Four-point-probe (any resist)
~eglas/src/sunbase3/vdp.c	V. Gutnik	Van-der-Pauw resistance
~eglas/src/sunbase3/idvds.c	V. Gutnik	Data for (nmos) Id-Vds curves
~eglas/src/sunbase3/idvdg.c	V. Gutnik	Data for (nmos) Id-Vdg curves
~eglas/src/sunbase3/Fallon.c	D. Rodriguez	Ladder resistance
~eglas/src/sunbase3/conr.c	D. Rodriguez	Contact resistances
~eglas/src/sunbase3/serp.c	D. Rodriguez	Resistances for 5 serpentines
~eglas/src/sunbase3/comb.c	D. Rodriguez	Showing shorts (defects)
~eglas/src/sunbase3/serpcomb.c	D. Rodriguez	2 Binary results showing opens/shorts
~eglas/src/sunbase3/kelvin.c ~eglas/src/sunbase3/kelmod.c	T. Duncan	Support 30-pin testing measurement
~eglas/src/sunbase3/adder.c	T.K. Chen	Support 8-bit (32 pins) adder device

Note: The software describes how sunbase is written, and how to add measurement routines for a new devices, as well as what future improvements could be added.

D. Sunbase3 Programming Notes

General Notes

- Remember - all compiling must be done on tellurium - sun3 with GPIB-SCSI library.
- The device created in /dev are spa (address 7), eglas (address x), dcs (address 23), smc (add 22), cv (address 17). Addresses in decimal..To add device, root must run ibconf and then reboot the machine.
- The pad numbers are in sunbase3.has comments. If there is a mistake in the numbers in die.map, no error will be flagged.
- You don't need to recompile when changing die.map.
- Numbers below are in mils - electroglas powers up to move in microns!!!!

To Identify Coordinates of Second Transistor

- Best method - find correct VEM/KIC/MAGIC layout and measure.
- Otherwise -

```
start ibic (/usr/tools/gpib/bin)
 ibfind eglas - talking to eglas.  use name from /dev file
 ibwrt "?H0  - use capitals, and  to terminate.
 should echo cmpl (without err)
 ibrsp
 ibrd 20  - reads 20 bytes
 eglas: ibrd 20
 [2100]   ( end cmpl )
 count: 15
 48  58  38  38  34  38  39  59        H X 8 8 4 8 9 Y
 36  33  32  39  35  0d  0a          6 3 2 9 5 . .
 x is x position in absolute machine position, in mils,
             y is y position
```

move to **next** transistor
eglas: ibrsp
[0100] (cmpl)
Poll: 0x00
eglas: ibwrt "?H0
[0100] (cmpl)
count: 3
eglas: ibrsp
[0100] (cmpl)
Poll: 0x40
eglas: ibrd 20
[2100] (end cmpl)
count: 15
48 58 38 38 34 38 39 59 H X 8 8 4 8 9 Y
36 33 31 37 30 0d 0a 6 3 1 7 0 . .
we have moved 125 mils up!
quit exits ibic. must exit or you'll tie up hpib.

In case of problems:

ibrsp - Clears serial poll in case of mistake.
Hit online to clear eglas if it looks confused.
Try again.
Changed air sensor x & y from 23768, 47022 respectively
to 23000, 46000 to avoid profiling errors.
In SET PARAMETER, 12-3 menu.

Appendix 3

Sunbase Testing of the Baseline Run (Staff)

Baseline test chips (runs) can be tested through sunbase3 program in a directory that contains the die.map, and prober.text files defined for each of the subroutines (NVt, PVt and resistivity). A simple script such as the one noted below as **run** can to automatically switch in and out of the defined directories to perform above noted tests (NVt, PNt and contact resistivity).

A. Testing The Baseline Runs Through Sunbase3 Program

A.1 Log into the Unix station (tellurium), just to the right of the probe station by entering your user name and password (silicon account).

A.2 Right click on Solaris screen and select **tool** and in the drop down menu, **terminal**.

A.3 You should now be at the %tellurium prompt, where you can change directory or start a test routine. Remember you need to be in the directory that contains pertinent die.map, and prober.text files. The sunbase3 program looks into the prober.text to perform the required test, which is specific to a particular routine. Our main directory called **etest**, contains Nvt, PVt and Resistivity directories.

A.4 Start the test by typing one of the following commands in the desired directory or from the main etest directory, as per follows.

A.4.1 From any of the three subdirectories; NVt, PVt and resistivity, type in the command **~eglas/src/sunbase3/sunbase3**. Programs follows the test routines defined in the prober.text file to perform the test/s utilizing the die.map stepping instructions .

A.4.1 type **source run&**, if you like to perform all three test routines (baseline) as per instructions in the script called **run**, noted below.

Note: The output file will get written to the current directory, as output.text.

A.5 After the test is completed (sunbase3 exists), press [LOAD] button to unload the wafer. You can now load the next wafer. You can abort the program at any time by pressing **Ctrl-c** on the Unix station.

Note: Be sure to bring chuck back into the lower right corner before leaving the probe station, by pressing the [LOAD] key.

B. Test Routines and Supporting Files for Baseline Runs

<u>Content of the **run** script.</u>

```
#!/bin/csh
set sunbase=~eglas/src/sunbase3/sunbase3
set thisdir=`pwd`
echo Sunbase program is $sunbase
echo Start directory is $thisdir
cd nvt
echo `pwd`
$sunbase
cd $thisdir
cd pvt
echo `pwd`
$sunbase
cd $thisdir
cd resist
echo `pwd`
$sunbase
```

<u>NVt prober.text</u>

prober.text 50 lines, 645 characters

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 1 0 1 0 0 0 0
0 1 0 0 1 0 0 0 0 0
0 0 1 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 1 0
0 1 0 1 x 1 0 0 0 0
0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0

VTWDLD
VT
+ VBSstop = -0.1
```

Mn10x0.8
Mn10x0.9
Mn10x1.0
Mn10x1.1
Mn10x1.2
Mn10x2
Mn10x3
Mn10x5
Mn5x0.8
Mn5x0.9
Mn5x1.0
Mn5x1.1
Mn5x1.2
Mn5x2
Mn5x3
Mn5x5
Mn15x0.8
Mn15x0.9
Mn15x1.0
Mn15x1.1
Mn15x1.2
Mn15x2
Mn15x3
Mn15x5

* 9x9 default. Good programs would probably determine this on the fly.
*
* sample call to parameter extract function. Use '+' to pass
* parameters, 'M' for transistors, and '.' to end functions, * for comments

*
<u>PVt prober.text</u>

prober.text 49 lines, 643 characters
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 1 0 1 0 0 0 0
0 1 0 0 1 0 0 0 0 0
0 0 1 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 1 0
0 1 0 1 x 1 0 0 0 0
0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
VTWDLD
VT
+ VBSstop = 0.1
Mp10x0.8
Mp10x0.9
Mp10x1.0
Mp10x1.1
Mp10x1.2
Mp10x2
Mp10x3

Mp10x5
Mp5x0.8
Mp5x0.9
Mp5x1.0
Mp5x1.1
Mp5x1.2
Mp5x2
Mp5x3
Mp5x5
Mp15x0.8
Mp15x0.9
Mp15x1.0
Mp15x1.1
Mp15x1.2
Mp15x2
Mp15x3
Mp15x5

```
* 9x9 default.  Good programs would probably determine this on the fly.
*
* sample call to parameter extract function.  Use '+' to pass
* parameters, 'M' for transistors, and '.' to end functions, * for comments
*
```

<u>resistivity (resist) prober.text</u>

prober.text 32 lines, 477 characters

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 x 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 x 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
* 9x9 default.  Good programs would probably determine this on the fly.
*
* sample call to parameter extract function.  Use '+' to pass
* parameters, 'M' for transistors, and '.' to end functions, * for comments
*
SCBR
scbrp+
scbrn+
```

scbrPO
.Conr
conrp+
conrn+
conrpo

NVt_die.map
die.map 42 lines, 1300 characters

```
@home 0,0
* All dimensions are in microns
*  Coordinate system: x increases to the right, y increases up.

* probe #'s      1 2  3  4  5
*              10 9  8  7  6
*mosfet x,y drain gate source bulk
*4ptprb x,y iin gnd v1 v2
*conr   x,y iin gnd
*fallon x,y iin gnd v1 v2 lw
*scbr   x,y i1 i2 i3 v1 v2 v3 v4 v5 v6 v7 LB WBD LStop LSbot WSDrawn layer
*xposition,yposition relative to starting probe position.
*mosPW_L
Mn10x0.8 0,0     8 9 3 7  W=10  L=0.8
Mn10x0.9 0,320   8 9 3 7  W=10  L=0.9
Mn10x1.0 0,640   8 9 3 7  W=10  L=1.0
Mn10x1.1 0,960   8 9 3 7  W=10  L=1.1
Mn10x1.2 0,1280  8 9 3 7  W=10  L=1.2
Mn10x2   0,1600  8 9 3 7  W=10  L=2
Mn10x3   0,1920  8 9 3 7  W=10  L=3
Mn10x5   0,2240  8 9 3 7  W=10  L=5
Mn5x0.8 0,0     10 2 1 7  W=5  L=0.8
Mn5x0.9 0,320   10 2 1 7  W=5  L=0.9
Mn5x1.0 0,640   10 2 1 7  W=5  L=1.0
Mn5x1.1 0,960   10 2 1 7  W=5  L=1.1
Mn5x1.2 0,1280  10 2 1 7  W=5  L=1.2
Mn5x2   0,1600  10 2 1 7  W=5  L=2
Mn5x3   0,1920  10 2 1 7  W=5  L=3
Mn5x5   0,2240  10 2 1 7  W=5  L=5
Mn15x0.8 0,0     6 4 5 7  W=15  L=0.8
Mn15x0.9 0,320   6 4 5 7  W=15  L=0.9
Mn15x1.0 0,640   6 4 5 7  W=15  L=1.0
Mn15x1.1 0,960   6 4 5 7  W=15  L=1.1
Mn15x1.2 0,1280  6 4 5 7  W=15  L=1.2
Mn15x2   0,1600  6 4 5 7  W=15  L=2
Mn15x3   0,1920  6 4 5 7  W=15  L=3
Mn15x5   0,2240  6 4 5 7  W=15  L=5
```

PVt_die.map
die.map 44 lines, 1392 characters

```
********************************************************************************
*
@home 0,0
* All dimensions are in microns
*  Coordinate system: x increases to the right, y increases up.
* probe #'s      1 2  3  4  5
```

```
*             10 9  8  7  6
*mosfet x,y drain gate source bulk
*4ptprb x,y iin gnd v1 v2
*conr   x,y iin gnd
*fallon x,y iin gnd v1 v2 lw
*scbr   x,y i1 i2 i3 v1 v2 v3 v4 v5 v6 v7 LB WBD LStop LSbot WSDrawn layer
*xposition,yposition relative to starting probe position.
*mosNW_L
Mp10x0.8 0,2560  8 9 3 7  W=10  L=0.8
Mp10x0.9 0,2880  8 9 3 7  W=10  L=0.9
Mp10x1.0 0,3200  8 9 3 7  W=10  L=1.0
Mp10x1.1 0,3520  8 9 3 7  W=10  L=1.1
Mp10x1.2 0,3840  8 9 3 7  W=10  L=1.2
Mp10x2   0,4160  8 9 3 7  W=10  L=2
Mp10x3   0,4480  8 9 3 7  W=10  L=3
Mp10x5   0,4800  8 9 3 7  W=10  L=5
Mp5x0.8 0,2560  10 2 1 7  W=5  L=0.8
Mp5x0.9 0,2880  10 2 1 7  W=5  L=0.9
Mp5x1.0 0,3200  10 2 1 7  W=5  L=1.0
Mp5x1.1 0,3520  10 2 1 7  W=5  L=1.1
Mp5x1.2 0,3840  10 2 1 7  W=5  L=1.2
Mp5x2   0,4160  10 2 1 7  W=5  L=2
Mp5x3   0,4480  10 2 1 7  W=5  L=3
Mp5x5   0,4800  10 2 1 7  W=5  L=5
```

<u>Resistivity (resist) die.map</u>

die.map 26 lines, 780 characters

```
@home 0,0
* All dimensions are in microns
*  Coordinate system: x increases to the right, y increases up.
@home 0,0
* All dimensions are in microns
*  Coordinate system: x increases to the right, y increases up.
* probe #'s      1 2  3  4  5
*             10 9  8  7  6
*mosfet x,y drain gate source bulk
*4ptprb x,y iin gnd v1 v2
*conr   x,y iin gnd
*fallon x,y iin gnd v1 v2 lw
*scbr   x,y i1 i2 i3 v1 v2 v3 v4 v5 v6 v7 LB WBD LStop LSbot WSDrawn layer
*xposition,yposition relative to starting probe position.
scbrp+         0,6400 10 9 5 1 2 6 8 6 8 6 647.5 4.5 429.0 429.0 2.25 p+
scbrPO          0,5760 10 9 5 1 2 8 7 6 3 4 219.0 6.0 204.5 247.0 2.0 poly
scbrn+         0,6080 10 9 5 1 2 6 8 6 8 6 647.5 4.5 429.0 429.0 2.25 n+
conrpo         0,6720 1 6
conrp+         0,7040 1 6
conrn+         0,7360 1 6
cchainPO         920,7040
```