VIEWLogic: **Powerview**

An Introduction to Digital Electronic Computer Aided Design/Simulation Software at Duke University version 1.24

J.D. Sterling Babcock & Brent Ward
Mark Perreault & Randy VanBuren & Scott Crutchfield

Dr. Apostolos Dollas
Department of Electrical Engineering
Duke University

September 12, 1994

Abstract

Computer Aided Design (CAD) and Simulation software provides engineers and scientists with a set of tools to develop, design and simulate simple to complex systems without actually constructing physical models. The computer carries out tedious calculations, design verification and analysis of system performance without the time and cost expenditures associated with building and debugging actual implementations. *Powerview* is a highly comprehensive package. The capabilities and usefulness of the software will be revealed in gradual progression, beginning with simple design tools and basic concepts such as developing a circuit drawing, revising the design, and printing out a final schematic. Eventually, the capability to simulate and analyze a circuit design before implementation will be introduced.

Contents

The purpose of this tutorial is to provide the new user with the basic information needed to run and use Powerview at Duke University. For further information, consult the online documentation discussed in section 7.

The most current edition of this tutorial can be found online in the following directory:

On the DEC ACPUB cluster:

~cad/viewlogic/powerview/powerview_tutorial

On the SUN EE cluster:

/opt/digital/share/tutorials/powerview_tutorial

as powerview_tutorial.dvi. Executing the script **pvman** will bring up the tutorial using **xdvi**. The tutorial may be printed with

lpr powerview_tutorial.ps

or with

mpage -2a powerview_tutorial.ps

You may wish to print the simple example with

lpr tutorialfig.ps

Note: To run the software on the SUN EE cluster, you must have a special account requested by your professor. Powerview 5.1.3 for solaris 2.2 runs on dsil1 to dsil12. Powerview 5.1.1 for Sun OS 4.1.2 runs on jefferson, sverige, blackbird, macedonia, and vergina.

Aside: ViewLogic provides us with several more Powerview tutorials in one book. If you are interested in more depth than this tutorial, get one of the books from your powerview administrator. Please note that there are several tutorials in the Powerview Tutorial book. The sequence is analog design tutorial, digital design tutorial, occurrence attribute tutorial, and the VHDL tutorial. The tutorial of digital design begins about 1/4 of the way into the book.

1 Current Known Problems with Powerview at Duke

Please be aware of the following problems in our Powerview installation. This is not all the known bugs, but just the most major ones.

1. The following has been corrected, but advanced users might run into it: If you create new schematic in ViewDraw and save it the very first time, you might get a Fatal error 1310 and ViewDraw will quit. However, the file is saved and you can reload it into ViewDraw. This error means that you have too many libraries defined in your `viewdraw.ini`.

2. ViewPLD 1.3: Model : Timing Model or Model : Schematic Model: ABEL files which do not explicitly use the clock will give errors.

3. ViewPLD 1.3: The filename must match the module name and both must be lower case.

4. A change has been made to licensing to alleviate crowding at the end of the semester. If you have not used your license in the last 30 minutes, someone else may use your license. One word of CAUTION: If you are working in a module and when you try to save your work, if the program tells you "Licenses not available" do not panic. If you have changes that you want to save, just wait 15 to 30 minutes and try again. It is likely that someone with a license has been inactive for 30 minutes and you can grab a license and save at that time.

2 Setting up and Entering Powerview

Running powerview is done using a script which checks to see if you are set up to run powerview and then runs powerview. If you are not setup, the script will set up the directories you need, set the environment variables, and run powerview. If you are already set up, it just sets the environment variables and runs powerview.

NOTE: These instructions assume the use of the C-SHELL. For the script program to work properly, you must use the C-SHELL. If you are using sh or bash or other shell, you must start a c-shell using the csh command. See your lab TA if you have problems.

1. If you have not run powerview before, change directory to your home directory. Create a directory called 'powerview' with the command:

   mkdir ~/powerview (RETURN)

   Note that the directory must be called 'powerview' or the script will not run correctly.

2. The first step to entering the Powerview environment is changing from the home directory to the powerview directory. This is crucial, because all work must be done within the powerview subdirectory. Execute the following command:

   cd ~/powerview (RETURN)

3. You are now in the powerview subdirectory. To invoke the program execute the following command:

   On the DEC ACPUB cluster:

   ~cad/viewlogic/powerview/powerview (RETURN)

   The SUN EE cluster is limited to graduate students and Electrical Engineering students. On the SUN EE cluster:

   /opt/digital/bin/powerview (RETURN)

   There will be a short pause (up to a few minutes if the network is crowded), and a window will appear. Place the window on the screen with the mouse and "open" it by pressing and releasing (clicking) the left mouse button (LMB). This is the Powerview Cockpit. The Cockpit is the central point for all activity in Powerview.

4. You may wish to create an alias by adding the following command to your .cshrc:

   For DEC ACPUB:

   alias powerview ~cad/viewlogic/powerview/powerview

   For SUN EE:

   alias powerview /opt/digital/bin/powerview

Note 1: If you encounter an **Alert** message box indicating that one or more libraries were not opened, you will need to click on **Accept**. To rectify the situation contact your powerview administrator (currently Sterling: jdsb@ee.duke.edu). You will more than likely need to run ~jdsb/public/pv/viewdraw.fix your "powerview" directory. This script replaces all your .ini files in all your subdirectories, so if you customized this file it will be lost.

Note 2: References to clicking on **Accept** assume you are running Powerview under the Open-Look interface. If you are using the Motif interface, **Accept** will appear as **OK** and the interface will be slightly different.

3 Powerview Projects

Powerview manages a design using the concept of a project. To create a project, select **Project** with the left mouse button, and then select **Create**. In the future, this type of command sequence will be indicated in the form:

- Choose **Project** ⇒ **Create**

Place the window that appears by clicking the left mouse button. From the Create Project window, click on the **ViewDraw** button. You may not see a change since it is most likely already selected. Then, click the left mouse button to the right of Project Pathname and type in a full pathname for your project. For example, **~/powerview/tutor**. Why enter the full pathname? Because if you do not, Powerview will create the new project as a subdirectory to the current project! (This is a feature?) (That is, if you are in project A and create a new project B without specifying the path, then project B will be in ~/powerview/A/B.)

Click the **Accept** button with the left mouse button. You will be returned to the cockpit window. Note that the *Current Project* and *Current Library* now point to your project.

4 Drawing a Schematic

The first procedure in developing a circuit design is the process known as "schematic capture", which is using the computer to draw the circuit layout.

To create a blank schematic sheet, **double click** the left mouse button on the **ViewDraw** icon on the right side of the screen. After a few seconds, place the window on the screen with the LMB. You will see the *ViewDraw File Open* dialog box. This dialog lists the schematics available in the current project. In this case there are no schematics available. To create a schematic, point the cursor at the window and type a name such as **schematic1**. Select the **Schematics** option button, and then click the **Accept** button. When the window appears, place the window on the screen with the left mouse button. The software is "menu-driven" which means that there are very few commands to memorize. Each function can be accessed through the "menu bar" at the top of the screen.

The top line of the screen indicates the functions of the mouse buttons at any given time. Currently, if your mouse pointer is on the menu bar the first line will show:

L: MenuDown M: _ R: MenuDown

If your mouse pointer is on the blank sheet, the first line will show:

L: select M: FastDraw R: menu-schematic

The bottom line of the screen will show status messages and information. If you left click on almost any menu button, this lower line will tell you what the highlighted subcommand command will do. One of the command menus is accessed by clicking the Red Button at the top left of the screen. If you have clicked on this button, unselect the menu by clicking anywhere on the drawing. Note that when the button is not specified in this document, it means to use the left mouse button. If you are instructed to click the MIDDLE mouse button, but your mouse only has two buttons, click both buttons simultaneously instead.

4.1 Adding Circuit Components

Hotkey: i ← A hotkey tells you how to invoke the action from the keyboard and to bypass the menus.

1. Select **Add** ⇒ **Comp**. Place the window that appears with the left mouse button. The upper box shows components in the current library (empty right now) and the lower box lists the libraries available.
2. Select "vl74ls" with the left mouse button. After a pause, the upper dialog box will display a list of 74LS components. Note that the 74LS components are the components available to the EE151 students in their lab kits. Each list can be "scrolled" by clicking the arrows at the top and bottom of the list or moving the scroll box with the left mouse button.
3. When you find the component you desire, select it with the left mouse button.
4. Move your mouse pointer into the schematic window. When the component outline appears, place it in the drawing window using the center mouse button.
5. Please add at least two components for this tutorial as the next section will instruct you on how to connect components. An example using a 74LS00 and 74LS04 is shown in the figure `tutorial.ps` in the powerview_tutorial directory where you got this document. It was removed from this document to reduce the postscript size and allow the use of mpage.
6. If you wish to remove the Add Component window, click on the **Cancel** button.

NOTE1: Components taken from the "builtin" directory are pure logical entities; they do not simulate real devices, so no timing information is provided. This can lead to problems if you mix these components with real device models. It has also been noticed (although not completely verified) that some of the "builtin" components are simply pictures of gates, and have no simulation capability at all. To be safe, do not use logic gates from the builtin directory. When in doubt, use the 74LS parts.

NOTE2: EE 151 students have the following 74LS parts available in their lab kit: 00, 02, 04, 08, 10, 20, 27, 30, 32, 85, 86, 109, 138 151, 168, 169, 194, 283

4.2 Moving about the Drawing Window

Panning

To center the window on a specific area of the schematic, point to the area you wish to center and press the **F6** key.

Zooming

The view may be zoomed in or out to the drawing window's center. To zoom in use the **F7** key. Pressing it more than once will zoom in successive steps until the maximum magnification is reached. Likewise, zooming out is accomplished using the **F8** key. To zoom in on an area, move the mouse cursor to a corner of the area of interest, click the **F9** key and then move your mouse to the opposite corner of the area and click the left button to commit the zoom operation.

Full View

You may view the whole schematic sheet by pressing the **F10** key.

4.3 Wiring Components Together

Hotkey: n

1. Zoom in on your components so that you can see their pin numbers.
2. Select **Add** ⇒ **Net**. Note that the labels on the line above the menu change to:

   L: select M: net R: menu-schematic

   This line indicates the functions of the mouse buttons at this time.
3. Point to a pin and click the center mouse button. Drag the mouse pointer to the point you wish to connect and click the middle mouse button. The wire will follow the pointer, and it will automatically move around components, or you may manually anchor the wire's path using the center mouse button.
4. If you wish to end a wire without connecting it to a component, it is necessary to first click the middle mouse button, followed by the right mouse button.

4.4 Adding Text and Labels

Hotkeys: t,l

This function is useful for providing additional information such as your name, the date, and a title. You should get in the habit of labeling all input, output and hanging wires since this information will be necessary to simulate your designs.

1. Select **Add** ⇒ **Graphics** ⇒ **Text**.
2. Type in some text, such as your name, and press **(RETURN)**.
3. Position the text outline with the mouse and place it using the center mouse button.

Wires may be labeled using the **Add** ⇒ **Label** command. Wire labels are generated as follows:

1. Use the left mouse button to select an interconnecting wire, an input, or an output net. The wire should be highlighted in white when selected.
2. Select **Add** ⇒ **Label**.
3. A label box will appear.

4. Click on "Label String" and then type in the desired label name.
5. Clicking on invert will cause the label to appear with a line over it indicating an active low signal.
6. The other options in this dialog box will not be covered here. Just accept the defaults for now.
7. Select **Accept**.
8. Position the text and place it with the middle mouse button.

NOTE: Be very careful not to label inputs and outputs alike; if two wires are labeled the same, powerview will assume that they are the same wire whether or not you have physically connected the two wires.

4.5 Saving your Drawing and Exiting Powerview

We have had a problem in the past where ViewDraw would bomb the first time you save a file with an fatal error 1310. We believe that this problem has been corrected, but if this happens, do not panic. Your file is saved correctly. You just need to relaunch ViewDraw. Later saves of the file should not be a problem.

Hotkeys: w,Alt-F4

1. To save your drawing select **File ⇒ Write**. This will save the file, checking it for problems. If no problems occur, you will be returned to the drawing window and the status line at the bottom of the screen will read: *Check complete. 0 errors and 0 warnings.*
2. If there are errors in the drawing, a dialog box will appear. Place it with the LMB, read the messages and click **Dismiss**.
3. To leave ViewDraw, go to the ViewDraw window and select the **Red button ⇒ Quit view-draw** from the menu that drops down when you click on the red button in the upper left corner with the left mouse button. If you have not saved your file before exiting, you will be asked if you are sure. Respond accordingly. If you have saved your file, the dialog will say "To quit ViewDraw, click on Accept or press Return". Click **Accept**.
4. To leave Powerview, select the **Red button ⇒ Quit powerview** from the red menu of the powerview cockpit. The dialog will say "To quit powerview, click on Accept or press Return". Click **Accept**.
5. If you are following this tutorial, restart both Powerview and ViewDraw before continuing.

NOTE: Please make sure you exit from Powerview when you are done. If you do not, you will lock up a license that someone else could use.

5 Editing a Schematic

Three basic editing concepts will be presented in this section. They are *Selecting, Deleting, and Moving.* The procedures can be used either on the current schematic or on a previously saved drawing. To edit a previously saved drawing you must use the procedure described above to start ViewDraw. Simply select the button in the dialog box for the file you wish to edit instead of entering a new file name.

5.1 Undo - Fixing an error

Hotkey: u

If you run into any problems in editing, you can undo the last change you have made by selecting **Edit** ⇒ **undo**.

5.2 Selecting and Unselecting

To delete or move an item it is first necessary to *Select* the item. This is accomplished by pointing to the item to be selected and pressing the left mouse button. If you are selecting a component or text, a box will appear around the selected item. If a wire is selected the wire will be highlighted. This focuses Powerview's attention on the item of interest. To unselect an item, point to a blank area of the screen and press the left mouse button. This is important because the next two topics to be discussed work on selected items only!

NOTE: To select more than one item, select the first item with the left button. Then hold down the **Ctrl** key and click the left button over each of the items you wish to include in the selected group. Repeat until you have everything you want selected. If you select the wrong thing, you can unselect one thing from the group with **shift** left button.

5.3 Moving

Hotkey: m

Components, wires and text strings can all be moved.

1. Select the item(s) you wish to move.
2. Select **Edit**⇒ **Move**.
3. Move the pointer to position the item in its new location. Note that any wires connected to components will move with them.
4. "Anchor" the item in its new location by clicking the center mouse button.
5. Note that the move command remains active and can be repeated by selecting an object with the left mouse button and then moving it and placing it with the center mouse button.

5.4 Deleting

Hotkey: d

1. Select the item(s) you wish to delete.
2. Select the **Edit** ⇒ **Delete**.

This will delete all items that are currently selected. If you make a mistake and delete the wrong item(s), you may recover them using the **Edit** ⇒ **Undo** command.

6 Printing a Schematic

1. Open the drawing file you wish to print.
2. Select **Red Button** ⇒ **Plot Setup...** and place the window that appears.
3. From the Devices column, select **Postscript**. You may need to use the left mouse button to move the scroll bar down.
4. Select paper size **A** and **US** if not already selected.
5. Select output **FILE** if not already selected.
6. Highlight the output filename by dragging over it with the left mouse button. You may have to press delete several times to remove the default filename. Type in a filename. For example: **schematic.ps**
7. If you want the whole sheet of your schematic printed, select **Full Sheet**. If you want just the portion you have zoomed into on your screen, select **Zoomed View**.
8. If you need the picture rotated, that is available under **Graphics**. Typically you will not need to change the rotation. To save a file for LaTeX, use **Align 0** for portrait sized files and use **Align 270** for landscape sized files. **Best Fit** should do this automatically, but it is unpredictable.
9. Select **Accept**.
10. The plot will be spooled. Place the window that appears. Wait for 'Issuing the plot' disappear BY ITSELF.
11. When spooling is complete, the dialog box will change to 'Plot complete. Please Dismiss this box'. Select **Dismiss**.
12. Note that the plot was spooled from the plot setup screen. There is no need to select **Red Button** ⇒ **Plot Go** unless you are making a plot with no changes to the plot setup.
13. Look at the powerview cockpit. The Current Project path tells you where your postscript file was stored. It is still necessary to send this file to a printer.
    (a) Select an available terminal window (NOT A POWERVIEW WINDOW!) or exit Powerview.
    (b) Change directory to the Current Project path.
    (c) type the following command at the prompt:

        lpr -P*PRINTER-NAME* **FILENAME**

        FILENAME is the name of the printer output file you created.
        PRINTER-NAME is the name of the printer you wish to use. It should come from the following list:
        On the DEC stations: teerlp1, teerlp2, redlp1, soclp1, hudsonlp1, biolp1, carrlp1
        On the SUN EE machines: hp

The teer printers are located in the Teer library cluster, redlp1 is located in the engineering building, soclp1 is located in the Sociology-Psychology cluster, hudsonlp1 is located in the hudson cluster in the Engineering building, biolp1 is located in the Biology cluster, and carrlp1 is located in Carr cluster on East Campus. Note that ACPUB charges for printing.

7 The Help Function and Online Documentation

Hotkey: F1

The system has a menu driven help function and online documentation. If you are experiencing difficulty, you can ask the system for information on the various commands as follows:

1. Select **Red Button** ⇒ **Help** ⇒ **Help on commands**.
2. To get help on a particular command, click on **Contents...**. Select the command that you need information about, using the scroll bar as necessary. For example, left click and hold on the thumb of the scroll bar. Scroll down until you find **Add** ⇒ **Comp**. Left click on this button. Left click **Dismiss**. The window now shows you information about using **Add** ⇒ **Comp** to add components.
3. When you are finished with the command help, select **Red Button** ⇒ **Dismiss Window** from the ViewDraw Commands window to return to your schematic.

NOTE: ViewDoc for the following is not currently available for the Solaris 2 version of Powerview. It is available on the DECs and SunOS 4.1.2.

To examine the online documentation, select **Red Button** ⇒ **Help** ⇒ **ViewDoc**. The Interleaf WorldView window make take some time to appear. When the window comes up, place it. Double click on the box beside the desired topic to view documentation. Select **File** ⇒ **Exit** ⇒ **Accept** to leave WorldView.

8 Preparing a Schematic for Simulation

The Powerview simulator is capable of performing several different types of circuit analysis. In its simplest form the simulator acts as a pure logic simulator for digital devices, performing appropriate operations on various logic values as they propagate through the circuit from input to output. This allows the specification of input logic levels, and the automatic calculation of the corresponding output values as defined by the circuit. A great deal of time can be saved using this feature, especially when testing large, complicated circuits. More advanced applications support timing and voltage level analysis. These topics are very important aspects in "the real world" where logic wave forms are not perfectly rectangular, and the delay in propagating a signal from the input of a gate to the output can appreciably affect the overall circuit performance.

8.1 Adding I/O Connectors

The first step in preparing a schematic for simulation is to add input and output connectors to any "dangling" wires.

1. Input connectors are located in the "builtin" directory. Select **Add** ⇒ **Component**. Use the the scroll bar to locate **(builtin)**. Select "(builtin)".
2. When the upper dialog box appears, search through the entries and select "in.1" or just left click on "Enter Name" and type **in (RETURN)**.
3. Place the connector near the dangling input wire with the middle mouse button. After placing the connector, use **Add** ⇒ **Net** to connect the wire to the connector.
4. Likewise, output connectors can be added in the same fashion; the output connector is called "out.1" (our just "out") and is also located in the "(builtin)" directory.

8.2 Labeling the I/O Connectors

The input and output connectors must be labeled for the simulation to proceed. The connectors are labeled by following the procedure outlined in section 4.4 of of the **VIEW***Logic*: **Powerview** tutorial. Note that only alphanumeric characters should be used as labels for wires and connectors.

8.3 Checking the Design

The next step in preparing the schematic for simulation is to have the powerview software check the design for possible errors such as unused pins, unconnected wires, nodes with more than one label, and other inconsistencies. This is accomplished by selecting **Utils** ⇒ **Check** from the menu bar or more simply by (**w**)riting the file. A window may appear listing any errors or warnings. Click on the button to close the window. If there were any errors, go back to the schematic window and correct them now. If there were no errors, proceed with the simulation.

9 Netlist Extraction

In order to simulate the design, the Powerview software package must have a "map" of the electrical connections in the circuit. The program builds this map through a process known as "netlist extraction". To begin the extraction, do the following:

1. Resize the ViewDraw window to be smaller and move it off to a corner of the screen.
2. Bring the Powerview cockpit window forward on your screen.
3. Double click the left mouse button on the **vsm** tool icon in the Powerview cockpit to invoke the ViewSim netlister.
4. Enter your schematic name in the design name entry if not already present.
5. Click **Accept**. The status of the extraction process will be shown in the Powerview cockpit window. When the extraction has completed successfully, you may begin simulation.
6. Note that EACH time you change your schematic you need to repeat the wirelist extraction.

10 Interactive Simulation

One of the more powerful aspects of the Powerview system is its ability to display the simulation visually. You will want enough room to display both the schematic, simulation, and trace windows.

1. For interactive simulation, make sure that a ViewDraw window still appears somewhere on the screen.
2. Open a simulation window by double clicking the left mouse button over the ViewSim tool icon in the Powerview cockpit.
3. Enter your schematic name in the design name entry if not already present.
4. If you are not using VHDL or Fault Simulation you may wish to left click **Off** for *VHDL Source Window, Fault Simulation*, and *VHDL Debugging* at this time. This will simplify your window and make the window smaller.

5. Click **Accept**. Place the VIEWSIM window with the mouse.
6. The simulation data can be stored in a logfile with the following command:

   logfile *schematic-name*.log (RETURN)

   The use of the **logfile** command is recommended only if it is desired to save the simulation results for many different simulator runs. If the **logfile** command is not used, the information will be stored in a temporary logfile, "vsimtemp.log". By opening a logfile, all commands and simulation results will be stored for later examination. This is especially helpful for long, complicated simulations.
7. Specify the nodes to be saved for generating a graphical wave form trace. Note that this brings up the ViewTrace application, so skip this step if you are low on memory. This is done with the following command:

   wave *schematic-name*.wfm *NODE1 NODE2 NODE3 . . .* (RETURN)

   The NODE names are the names you used to label signals. The values of the signal at each label will be saved to produce a waveform display, similar to the logic analyzer used in the laboratory. Typically, the inputs and outputs will be specified, but intermediate steps can also be labeled and stored to aid in circuit verification.

   NOTE: Any signals that were specified as inverted when the label was placed on the schematic must be referenced with a ~ preceding its name.
8. To specify the nodes to be traced in the text simulation, use the following command:

   trace *NODE1 NODE2 NODE3 . . .* (RETURN)

   Note that the trace command shows you exactly when your signal changed. For example:

   `[time = 19.0ns] C X -> 0`
9. Next, declare the nodes you wish to "watch" or have information displayed during the simulation by typing:

   watch *NODE1 NODE2 NODE3 . . .* (RETURN)

   Note that the watch command only shows you your signals at the end of the simulation time frame. For example:

   `time = 100.0ns A=0 B=0 C=0`
10. Now specify the initial input values for each of the input nodes.
    (a) To set node values to logic "0", use the "l" command as follows:
    l *NODELIST . . .* (RETURN)
    This command causes all of the nodes in *NODELIST* to be set "low".
    (b) To set node values to logic "1", use the "h" command as follows:
    h *NODELIST . . .* (RETURN)
    This command causes all of the nodes in *NODELIST* to be set "high".
11. Simulate the design with the assigned values using the "sim" or "cycle" command as follows:

    sim (RETURN)

    or

cycle (RETURN)

Sim gives information on all the watched nodes. Cycle gives information only on those signals that change value and the time they changed value. The value button gives the value of all watched nodes.

The results of the simulation step will be printed. The information includes transitions and the time at which they occurred during the simulation. The values of all the nodes being "watched" are also displayed. The default simulation step is 100 nS.

12. The ViewDraw schematic window will display the signal values for the circuit, at the input and output of every gate or component. Signal values appearing in a box are those values set by an "h" or "l" command. Those values without a box are values produced by the circuit based on the specified inputs. A value of "X" indicates an as yet unspecified or undefined logic level. These signal values are updated automatically as the simulation proceeds.

13. In the ViewSim window, use the "h" and "l" commands to change the input logic values and resimulate with the "sim" command. The circuit's response to all possible combinations of input values can easily be determined and recorded. Techniques for automating this process (making it even easier) will be explained later.

14. After completing the simulation, save the log file with the following command:

    logfile (RETURN)

15. You can exit ViewSim by selecting **Red Button** ⇒ **Quit ViewSim** ⇒ **Accept** You may be asked if you wish to exit ViewTrace. Click Accept.

11 Printing the Waveform Trace

The Waveform Trace can be printed using a procedure similar to that used to print the schematics. Follow the printing commands for printing schematics from section 6 of this tutorial on the ViewTrace window.

12 Using a Command File to Control Simulation

During the design process, the circuit under construction may go through many changes. Typing in all of the simulation commands used in the above section would be a great inconvenience if it was required for each and every simulation run of a complex schematic. Fortunately, Powerview provides a means of storing a "batch" of simulation commands, so it is a simple matter of calling up the simulation instructions for each run. The command file is created using a text editor such as "vi" or any other editor you may be familiar with. (If you are not familiar with any text editors, tutorials and other information can be obtained from the Academic Computing Consulting Desk in the North building or by typing "man vi" or "man emacs" at a prompt).

To create the command file:

1. The Current Project path n the Cockpit window tells you where your command file should be stored.

2. Obtain an available window (not the same window in which Powerview was started) and change to the Current Project directory.

3. Use the text editor to create a text file with the list of simulation commands. These commands are the same commands used in the ViewSim section; the only difference is that instead of being typed into the simulator interactively, they are being stored in a file. Name the command file *file-name*.cmd as shown in the following example: **vi** ***filename*.cmd**

4. In the ViewSim window, the command file can be executed by selecting **Misc** ⇒ **Execute** ⇒ **CmdFile** from the menu bar. Type the name of the command file (don't forget the ".cmd" !) at the prompt. After (RETURN) is pressed, the simulation will proceed automatically. The results can be seen in the viewdraw and viewtrace windows.

   The command file may be edited between simulation runs without having to re-Export the wirelist (The wirelist must be Exported only after a change has been made to the schematic). To reset the timing information, choose the **Sim** ⇒ **Restart** option from the menu bar. If the simulation is not restarted, the timing information for the current run will begin where the previous run halted.

13 Quitting Powerview

To leave the Powerview environment, you must exit from each application. Note that this is necessary to prevent locking of the license for Powerview. Only a specified number of powerview programs may be running on the Duke network at any given time. By quitting powerview gracefully through the menu bar commands, you ensure that the active count of the number of powerview programs running is reduced, and another powerview session will become available for someone else to use.

1. To save your drawing select **File** ⇒ **Write**. This will save the file, checking it for problems. If no problems occur, you will be returned to the drawing window and the status line at the bottom of the screen will read: Check complete. 0 errors and 0 warnings.

2. To leave ViewDraw, go to the ViewDraw window and select the **Red button** ⇒ **Quit viewdraw** from the menu that drops down when you click on the red button in the upper left corner with the left mouse button. If you have not saved your file before exiting, you will be asked if you are sure. Respond accordingly. If you have saved your file, the dialog will say "To quit ViewDraw, click on Accept or press Return". Click **Accept**.

3. Exit ViewSim or any other PowerView applications similarly.

4. To leave Powerview, select the **Red button** ⇒ **Quit powerview** from the red menu of the powerview cockpit. The dialog will say "To quit powerview, click on Accept or press Return". Click **Accept**.

14 Advanced Features

14.1 Vectors

Powerview has the ability to define, or "Setup" vectors to enable easier control of the simulation. The vector command allows a group of nodes to be associated with a common name, providing a shorthand notation for setting signal levels. The statement **vector** *vectorname signal1 [signal2] [signal3] ...* can be included after the watch command in the command file. The listed signals can then be set to high (1) or low (0) with the assign statement as follows: **assign** *vectorname* [bitpattern] where bitpattern is a list of ones and zeroes specifying the value of signal1 [signal2] [signal3] ..., etc.

This is shown in the following example:

vector OPCODE A0 A1

The previous statement defines the vector OPCODE to be composed of the signals A0 and A1.

assign *OPCODE* 01

The assign statement above assigns the value of 0 to A0 and 1 to A1. This is equivalent to executing the following sequence.

l A0
h A1

From this example it can be seen that as the number of signals to be dealt with increases, it takes considerably less effort to treat groups of signals as vectors.

14.2 CLOCK Waveform Generation

A clock wave form can be defined by using the clock command immediately after the trace command as follows:

clock *NODENAME* 0 0 0 1 1 1

The above command defines a wave that will transition from logic 0 to logic 1 over the course of three simulation steps; i.e. in the example above, the node *NODENAME* will be logic 0 for 3 simulation steps (default : 1 simulation step = 100 ns) and will then become logic 1 for 3 simulation steps. The transition will occur at whatever *NODENAME* is specified. The above sequence can be repeated *n* times with the cycle command as follows:

cycle *n*

14.3 PLDs: Incorporating the use of PLDs in Logic Designs

Powerview has the ability to incorporate Programmable Logic Devices into designs. The first step requires knowledge of the ABEL PLD programming language to create a description of the device.

ABEL

Powerview has the capability to generate it's own JEDEC files from ABEL source code.

NOTE: ViewPLD for the following is not currently available for the Solaris 2 version of Powerview. It is available on the DECs and SunOS 4.1.2.

NOTE: You cannot use ~ in your ABEL file to indicate negation. Powerview will not like it.

WARNING: Back up your original **.abl** file to another filename! When you select **Model ⇒ Schematic** your **.abl** file is written over by Powerview to back annotate.

1. The first step is to use an editor (such as vi or emacs) and enter the ABEL source code into an ascii text file. This file must have a .abl suffix and must be placed in your Current Project directory.

2. To compile the ABEL source code, from the powerview cockpit, double click **ViewPLD**.

3. Place the window that appears.

4. NOTE: We are currently having a problem with ViewPLD on the ACPUB DEC cluster with the **Prepare** ⇒ **Abel** command. Go to another window with **Red Button** ⇒ **System**. cd to your powerview directory. cd to your project directory. execute "ahdl2pla filename.abl". Go back to ViewPLD and do **Prepare** ⇒ **Abel**.

5. Select **Prepare** ⇒ **Abel**. If you placed your .abl file in your Current Project directory, you should see a button with it's filename. If you do not, correct the file location. If you want to type in your filename instead of the button, type the name of your ABEL source code file without the .abl suffix.

6. Click on the *Partitioning Schematic* ⇒ *No* box with the left mouse button. You May also wish to select whether or not to use reduction of your source by clicking the appropriate box with the left mouse button.

7. Click on **Accept** with the left mouse button.

8. When the run is complete you can view the results by using the left mouse button on the scroll bar.

9. Next, use the menu bar to select **Specify** ⇒ **Create JEDEC**. Verify the proper device, and click *Accept* with the left mouse button. This will complete the JEDEC file creation.

10. It is HIGHLY reccommended that you use test vectors to verify that the PLD created is what you had in mind. The test vectors are automatically converted to a .cmd file so that you can test your PLD using ViewSim. However, this requires running ViewSim, setting it up, and testing it. If you are using a SunOS machine (only PV 5.1.1 has the JEDSIM sommand), an easier way at this point in the process to simulate is to do **Red Button** ⇒ **System**, and use the command **jedsim filename**. This will print hash marks for passed(.)/failed(*) test vectors. If one or more vectors failed, open the file **filename.sim** in another window to see what vectors failed. This will give you a quick check to see whether your PLD is correct. We are trying to get this capability for the other systems.

11. Now you must make a timing model for your new device. Select **Model** ⇒ **Timing Model**.

12. Place the window. You should see your filename. Select Manufacturer (usually AMD) and select **Accept**. NOTE: Your timing model will not be activated until you reopen ViewDraw after you do this step. So if ViewDraw is running, you willl need to close it and reopen it to see the changes.

13. Now you must create the schematic symbol for your new device. Select **Model** ⇒ **Schematic**.

14. Place the window. You should see your filename. Select Create Symbol **Yes**. Select Input format **JEDIC**. Select Translate Test Vectors **Yes**. Select **Accept**.

15. When you see *ViewPLD: Task complete*, exit ViewPLD by selecting **Red Button** ⇒ **Exit ViewPLD** ⇒ **Accept**

Placing your PLD into your logic design

To use the PLD in your design, use the following procedure:

1. Open your drawing using ViewDraw. NOTE: Make sure you have stopped and restarted ViewDraw since you ran ViewPLD, or you will not see your changes.

2. Select **Add ⇒ Comp**.
3. The second directory listed is your project directory. Select your project directory (Don't select symbols in memory).
4. Select the component that appears that has the name of your ABEL file.
5. Place that component and connect it to your circuit.

This completes the special instructions for using PLD. To simulate, simply export the wirelist for the schematic and follow the standard procedures for simulation outlined previously.

14.4 BUS Implementation

In Powerview, buses are used instead of separate wires to carry several related signals. They are very useful in schematic capture of datapaths, for example, since they allow quick and easy connection of all the data signals. Rather than connecting the outputs of a 4-bit adder to the inputs of a register with four separate wires extending all over the schematic and taking up a lot of space, you can use a single bus, connect it to the adder and register, and route all four signals together as simply as a single wire. Buses can be any width, which means that they can carry as many related signals as you want. Buses should only carry signals that are logically related.

The following procedure is used to instantiate and utilize busses in Powerview schematics.

1. Using the menu bar, select **Add ⇒ Bus**
2. Treat it like a normal "net" wire: press middle button NEAR the outputs of one chip, move to a point NEAR the inputs of another chip, and press the middle mouse button again. Don't try to connect the bus to any pins yet. The best way to route the bus is so that it runs parallel to the side of the chip that you will connect it to. Right click to finish running the bus.
3. The bus is labeled similarly to other wires or I/O connectors as far as the labeling process is concerned. The difference is in the label itself. Bus labels are of the following form : *DATA[3:0]*. This means that this particular bus us carrying 4 wires that are separately labeled as *DATA3, DATA2, DATA1, and DATA0*. It is important to name all your buses consistently, and the convention is to name them in most significant to least significant order (e.g. *[MSB:LSB]*). Select a bus with the left mouse button and then select **Add ⇒ Label** to label the bus.
4. Use **Add ⇒ Net** from the menu bar to connect the pins from each chip to the bus.
5. Use **Add ⇒ Label** to label these single wires with names corresponding to the bus: *i.e. DATA3 DATA2 DATA1 DATA0*.
6. Note that each wire leading to or from a bus must be labeled to be consistent with the bus label.

It is very important to pay close attention to the naming of buses. You tell Powerview how many signals are on the bus with its label. If a bus is labeled ADDRESS[3:0] and you connect a wire labeled ADDRESS5 to it, there will be an error. Also, the order of the numbers in square brackets makes a difference, especially in simulation. It is best to name them consistently.

It is also important to label all the buses on your schematic.

There is a shortcut to naming the nets that connect to a bus when entering a schematic. This technique assumes that you are connecting one net to each bit of the bus. In some cases it is simpler just to manually label the nets, as you have done before, but the following makes labeling go faster when you are connecting to large buses. After the nets are drawn between the bus and the part they connect to, choose **Add Label** from the menu. Select the most significant net, and click the middle mouse button. When you are prompted for the label, enter the same name that you entered for the bus including the numbers in brackets. Powerview knows that this is a bus name and will "expand" it for you, allowing you to easily name the rest of the nets. The first net will be named automatically as the MSB, and you will be prompted to place the label. Next, select the next net in order and press the middle button once to label it and again to place the label. Notice that Powerview automatically decrements/increments the number that is at the end of the label each time you do this. Continue selecting the nets in order and clicking the middle button to give the next name.

There are a few important simulation commands to use with buses. The vector command lets you give them a convenient name, for example

```
vector data MEMORY_DATA[7:0]
```

allows you to refer to the bus by the name `data` in the command file. To give the bus a value, the assign command is useful:

```
assign data FF
```

or

```
assign MEMORY_DATA[7:0] FF
```

sets all the lines to high (FF_h means the number FF treated in hexadecimal) The tedious alternative to the assign command is to enter separate h and l commands:

```
h MEMORY_DATA0 MEMORY_DATA1 ....  MEMORY_DATA6
```

accomplishes the same thing.

14.5 Design Hierarchy

Powerview has a built in hierarchy that can greatly simplify the design process. This ability allows the user to develop a design (at any hierarchical level - gate, MSI, LSI, VLSI, and even VHDL) and then capture that design into a single symbol. This allows the incorporation of highly complex schematics into higher level designs as a single symbol. This symbol will have the same inputs and outputs as the lower level schematic and will function identically to that lower level schematic during simulation. The general procedure for incorporating the hierarchical features into the design are as follows:

1. Develop and test (simulate) your design using whatever level is appropriate.
2. To create a symbol for your design, select **File** ⇒ **Level** ⇒ **Push** ⇒ **Symbol**.
3. Create a body for your symbol with the **Add** ⇒ **Graphics** ⇒ **Box** command. Put the box about two dots inside the given square with the MIDDLE mouse button.
4. Add text to describe the box's function with **Add** ⇒ **Graphics** ⇒ **Text**.
5. You must also add pins to the symbol. These pins must correspond to the inputs and outputs of your lower level schematic design. The pins also must be labeled with names that are labeled in the schematic. Use **Add** ⇒ **Pin**. Middle click on the side of your symbol body. Then middle click on the outer box. Repeat for each input and output.

6. A useful feature is the command sequence Change Pin Sense. Select a pin with the left mouse button and select **Change Pin Sense**. This places an inversion bubble on the pin (properly sized) for specifying active low logic. Note that this is a cosmetic feature only and does not actually affect the behavior of the module.

7. Add labels for each pin: Select a pin with the left mouse button and select **Add Label**. Type the label and return. Repeat for each pin.

8. Once your symbol design is complete select **File** ⇒ **Write**.

9. To return to your design, select File ⇒ Level ⇒ Pop.

10. You may incorporate this lower level schematic into future designs simply by "Adding" the symbol to your design.

14.6 VHDL Models

VHDL is the VHSIC (Very High Speed Integrated Circuit) Hardware Description Language. VHDL can be used for both behavioral modeling and circuit synthesis. VHDL can be used to model components that you don't have but will be using in your system. This provides a means for simulating the design with the component and checking for any errors in the control signals or wiring before going to fabrication or wire-wrapping (prototyping). VHDL can also be used to develop systems and components that do not yet exist. The versatility of the VHDL behavioral modeling capabilities will make this tool a must in the future. We have included a short tutorial on the tools needed to test a VHDL model, convert it into a symbol, incorporate it into a design and then simulate the system of which the VHDL model is now a component.

VHDL Source File

The user will need to have a file that has been either obtained from the current database of available VHDL modules or have entered VHDL source through one of the available text editors (i.e. vi). The code will then be used to create the usable symbol to be used in system simulation.

For more information on VHDL see the Powerview Tutorials written by VIEWlogic or the VHDL Tutorial written by Lee House and available on the EE network in

/opt/digital/tutorials/vhdl_tutorial

VHDL Analyzer

The VHDL **analyzer** tool (found in the **Powerview Cockpit**) is used to parse the VHDL code. The analyzer alerts the user to any inconsistencies in the code and any misuses or unknown calls. The Analyzer is the first step in creating a usable VHDL module.

1. Before beginning, make sure that the .vhd source file is located in your current project directory.

2. Double click on the **VHDL analyzer** tool from the Powerview Cockpit (you may need to scroll down in the tools window to find it).

3. In the window that appears, in the *VHDL Source Filename* field, enter the name of your VHDL source file (i.e. brentsmachine.vhd).

4. Make sure that all the options are selected as **On**.

5. Click on **Accept**.

6. The current progress and errors from the parsing of the file will be displayed in the Powerview Cockpit window. You can use these error messages to debug your source file.

7. When the source is error free, and the parser is able to completely parse the source, a .vsm file will be created for the model. This .vsm file will be used by the **ViewSim** and **vsm** tools to create simulate the model and the circuit in which it is placed.

VHDL in ViewSim

ViewSim can be used with just the .vsm file created by the **VHDL analyzer** so that you can simulate the model with a minimal number of steps and also to verify the functionality of your model during development.

NOTE: You may want a .cmd file for your model. A discussion of the command file can be found in section **12 - Using a Command File to Control Simulation**.

1. Select the **ViewSim** tool from the **Powerview Cockpit** (you may need to scroll through the tools to find it.)

2. In the *Design Name* field, enter the name of your VHDL model name (i.e. brentsmachine).

3. Optionally, In the *Command File* field, enter the name of your command file (i.e. brentsmachine.cmd).

4. Turn *On* the *Graphical Interface*, and all the other options should be turned **Off**.

5. Click on **Accept**.

You will receive a **ViewSim** window that will display the current progress of the simulation through the model and will allow interactive simulation of the model. If you have a waveform command in your .cmd file, then you will also receive a **ViewTrace** window in which the simulation variables designated will be displayed as the simulation progresses. These outputs can be used to verify the correct operation of your VHDL source.

VHDL → SYM

The VHDL model source can be used to create a symbol that can be used in a schematic (just as the symbols available in the various Powerview libraries). The symbols will simulate just as their Powerview symbol counterparts - based on the information used to create the symbol.

1. Select the **vhdl → sym** tool from the **Powerview Cockpit** window.

2. In the *VHDL Source Filename* field, enter the name of your VHDL source file (i.e. brentsmachine.vhd).

3. Click on **Accept**.

The progress of the creation of the symbol can be viewed in the **Powerview Cockpit** message window. The conversion will display the number of inputs and the number of outputs on the symbol and will then exit. The symbol has now been created.

Use of the VHDL symbol

The VHDL symbol is used exactly like any other symbol/component available through the Powerview libraries.

1. Select the **ViewDraw** tool from the **Powerview Cockpit** window.
2. Enter the name of the schematic that you wish to add the VHDL symbol to and click on *Accept*.
3. Select **Add** ⇒ **Comp**.
4. From the window that appears, click on your project directory, you should find your symbol in the available components.
5. Click on the symbol name and then move the mouse over to the **ViewDraw** window and place the symbol.

From this point forward, the VHDL model will act as any other component and can be used in a simulation of the schematic by going through the normal schematic simulation steps. (i.e. running the **vsm** tool on the schematic file and then running **ViewSim** with a .cmd file).

DISCLAIMER: VHDL symbols simulate with delays reduced by a factor of 10 when placed in a schematic due to a ViewLogic bug. (Actually, it just simulates with a default timing of 1/10 nS.)

A Powerview Setup Script

The following is the setup script for powerview for the SUN workstations:

```
#!/bin/csh -f
# powerview  by J. D. Sterling Babcock  Ver: 4/21/94
# CSH script to set up an account for use with Viewlogic Powerview
# or to actually run Powerview automatically.
# Instructions for use:
#       1. Change to your powerview directory
#          New users change to your home directory.
#       2. Run the program with the command /opt/digital/bin/powerview
#
echo ""
echo ' VIEWLOGIC:  POWERVIEW Startup  Ver 4/21/94'
echo ""

# Check to see if license deamon is running
set vnsd_running =  `/bin/ps axc | /bin/grep vnsd`
if ( $#vnsd_running == 0 ) then
  echo "Viewsim Network Services Deamon (vnsd) is not running on this machine."
 echo "Starting up vnsd"
  setenv WDIR /opt/digital/powerview/standard
  /opt/digital/powerview/vnsd 2> /dev/null
  echo Finished starting up vnsd.
#else
#        /bin/echo "vnsd is up"
endif
unset vnsd_running

echo Welcome to Powerview 5.1.1\!
echo       Please email any problems to jdsb@ee.duke.edu
echo NOTE: Please email Powerview version, platform, and error message\!
echo -------------  Known Problems:  ----------------------------------
echo Abel PLD files:  The filename must match the module name and both
echo                  must be lower case.
echo ------------------------------------------------------------------
echo ""

#
# if the user is not on a sun4 machine, he cannot run powerview
#
if (`arch` != "sun4") then
   echo You TTneed to be on a Sun workstation to run Powerview
   goto end
endif

#
# If the user is in his powerview directory, run powerview.
```

```
# If not and a powerview directory exists, tell the user to change to
# that directory.
#
if (${cwd:t} == "powerview") then
   goto foundpv
endif

if ((-e ./powerview) || (-e ~/powerview)) then
   echo Please cd to your powerview directory and restart.
   echo .
   goto end
endif

# If dir does not exist, ask user if it should be installed.
echo You are not inside a directory named "powerview".
echo Do you wish to install powerview in \~/powerview \(y/n\)\?
set ans = $<
if ($ans == "Y" || $ans == "y" || $ans == "yes" || $ans == "YES") then
   cd ~
   goto mkpv
endif

echo Do you wish to install powerview in the current directory \(y/n\)\?
set ans = $<
if ($ans == "Y" || $ans == "y" || $ans == "yes" || $ans == "YES") then
   goto mkpv
endif

echo Powerview TTcannot execute until it is installed on your account.
echo Change to your powerview directory or where you want it installed
echo and rerun this program.
goto end

mkpv:
mkdir powerview
# cd powerview
echo The powerview directory has been made.  Please cd to that directory
echo and rerun this script.  Each time you wish to run Powerview, you must
echo run this script from inside your powerview directory.
echo
goto end

# Within the powerview directory will be a directory called standard with the
# 'viewdraw.ini' file in it and an environment directory.
foundpv:
if (! -e ./environment) then
   mkdir environment
endif
```

```
if (! -e ./standard) then
   mkdir standard
endif

if (! -e ./standard/viewdraw.ini) then
   cp /opt/digital/powerview/standard/viewdraw.ini standard
endif

setenv PVPATH /opt/digital/powerview
set pvdir="$cwd"
setenv FONT1 6x10
setenv FONT2 8x13
setenv FONT3 9x15
setenv SYSPLT $pvdir/environment
setenv WDIR $pvdir/standard:/opt/digital/powerview/standard
setenv VIEWSTATE_DIR /opt/digital/powerview/standard
setenv ABELPROT viewlogic
# ABEL4DEV must be less than 60 characters to work (PV 5.1) 80 for PV 5.1.1
# setenv ABEL4DEV /opt/digital/powerview.5.1.1/standard/lib4
#                123456789 123456789 123456789 123456789 123456789 1234567890
setenv ABEL4DEV /opt/digital/powerview/standard/lib4
setenv ABEL4DB  /opt/digital/powerview/standard/lib4/devices
# setenv LD_LIBRARY_PATH /depot/X11R5/lib
setenv LD_LIBRARY_PATH /usr/openwin/lib
setenv XAPPLRESDIR /opt/digital/powerview/viewdoc
setenv XFILESEARCHPATH ${XAPPLRESDIR}/Iview
setenv MADSSIM_ANALOG_ARGS madssim
setenv MADSSIM_ANALOG      /opt/digital/share/hspice/hspice_exe
setenv MADSNET_ANALOG      /opt/digital/powerview/standard/wmix.ini
# The pv/bin directory is for ViewSweep
set path= (/opt/digital/powerview/bin $path)
set path= (/opt/digital/powerview $path)

# Is anyone else running powerview on this machine?
set pv_running =  `/bin/ps aucx | /bin/grep powerview`
if ( $#pv_running == 0 ) then
  echo "Checking other users: Powerview is not running on this machine."
  echo "It is OK to use this machine."
else
  echo \!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!
  echo "Checking other users: ALERT\!TT"
  echo "Someone else is running powerview on this machine\!"
echo "If you both try to simulate at once, the machine will be extremely slow."
  echo ""
  echo "USER       PID %CPU %MEM   SZ  RSS TT STAT START  TIME COMMAND"
  /bin/ps aucx | /bin/egrep '(powerview|viewdraw|viewsim|viewtrace)'
  echo \!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!
```

```
endif
unset pv_running
echo ""

exec /opt/digital/powerview/powerview

end:
```

SPARCstation Set Up Instructions

The following setup instructions are for people who wish to set up their EE account for the SUN Powerview permanently and not use the script. Only do this if you are an expert user.

1. Create a directory in which you will work with powerview. This can be any directory, and you can have as many powerview directories as you like. The reason for this is that powerview creates many files and subdirectories of its own. The separate powerview directory helps maintain organization.

2. cd to your new directory. Create a directory called "environment". Create a directory called "standard". cd to this standard directory. Copy the viewdraw and powerview initialization files to your standard directory as follows:

   ```
   cp /opt/digital/powerview/standard/viewdraw.ini . [RETURN]
   ```

3. Add the following environment variables to your .cshrc. Be sure to edit them appropriately. Source your .cshrc to register them (note that if you have the default .cshrc, it has been designed to be run only once per session. If this is the case, you will have to log out and then back in again in order for this to work.)

   ```
   set pvdir='YOUR_POWERVIEW_DIRECTORY'
   setenv FONT1 6x10
   setenv FONT2 8x13
   setenv FONT3 9x15
   setenv SYSPLT $pvdir/environment
   setenv WDIR $pvdir/standard:/opt/digital/powerview/standard
   setenv VIEWSTATE_DIR /opt/digital/powerview/standard
   setenv ABELPROT viewlogic
   # ABEL4DEV must be less than 60 characters to work (PV 5.1) 80 for PV 5.1.1
   setenv ABEL4DEV /opt/digital/powerview/standard/lib4
   setenv ABEL4DB  /opt/digital/powerview/standard/lib4/devices
   setenv LD_LIBRARY_PATH /usr/openwin/lib
   setenv XAPPLRESDIR /opt/digital/powerview/viewdoc
   setenv XFILESEARCHPATH ${XAPPLRESDIR}/Iview
   set path= (/opt/digital/powerview/bin $path) # for ViewSweep
   set path= (/opt/digital/powerview $path)
   ```

4. Add the following directory to your path:

   ```
   /opt/digital/powerview
   ```

5. To run powerview, ensure that you are logged into an EE Dept SPARCstation and cd to your powerview directory. This is very important to prevent cluttering your home directory (or other directories) with powerview files. Ensure your display is set properly. Run powerview by executing the file

   ```
   /opt/digital/powerview/powerview
   ```

   An alias would be very useful here.