HSVIEW 2.2
A Simulation Output Viewer

Alireza Moini

June 1998

Contents

1 Introduction

HSVIEW is an X11 interface for viewing the output from various simulation programs. Its easy interface and intuitive design makes it suitable for circuit design engineers who need to perform many simulations and data analysis, and at the same time be able to manipulate the data easily.

2 Windows in HSVIEW

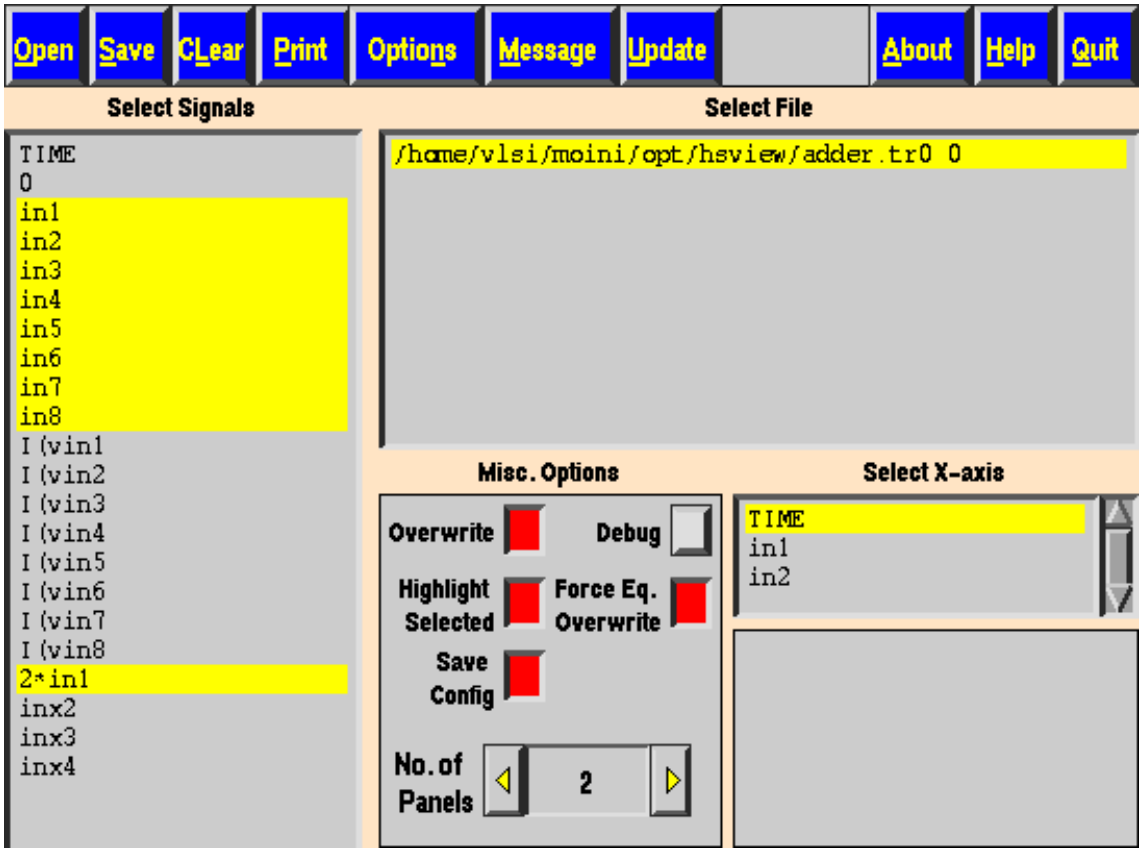
There are several windows in HSVIEW through which different operations of the program can be controlled.

2.1 Main Window

The main window consists of control buttons and scrolling browsers for selecting the files, signals, etc. . The main buttons are:

- **Open:** This brings up the file selector to select an input file. In the files selector the ".tr*", ".sw*", and ".ac*" buttons can be used to filter only one type of inputs.
- **Save:** This brings up the file selector to select an output file. Currently, the only output format provided is the matlab format.
- **CLear:** This erases the data read from the selected file. (The selected file is the one which is highlighted in the "Select File" browser). All the data is erased from the memory and also all the signals which belonged to this file will be deleted from the panels.
- **Print:** This brings up the print window. Functions of this window are described later.
- **Options:** This brings up the Options window. Functions of this window are described later.
- **Message:** This brings up the Message window.
- **Update:** This button updates the contents of all the files that have been read in. The update only occurs only when the modification time of each file has been changed from the last time that it has been read in.
- **About:** This brings up the About window, containing information about the software.
- **Quit:** This closes the program and all the windows. If the "Save Config" option is turned on, then the current configuration of HSVIEW will be saved into a file.
- **Select Signal:** This is a browser in which the signals belonging to the selected file are listed. The signals which are plotted in each panel will be highlighted. Adding or deleting a signal is simply done by clicking any of the mouse butting on the name of the signal.
- **Select File:** This is a browser in which the files which have successfully been read in are listed. In order to select a file simply click on it. Consequently the "Select Signal" browser and other relevant information will be updated.
- **Mis Options:** This set of buttons are used for changing some of the options in HSVIEW .
- **Select X-axis:** This browser is used to select the signal for using as the X-axis. By default the first signal, which is normally the signal used for sweeping, is used. Using other signals instead of the first signal may produce results when performing measurements.
- **Sweep parameters:** If the selected file contains a sweep operation, the information related to the sweep, i.e. the name of the swept parameters, the value and the number of sweeps, will be shown in the lower right corner of the window.

  If the selected fiel does not have a sweep operation, nothing will be shown here.


Open
Save
CLear
Print
Options
Message
Update
About
Help
Quit
Select Signals
Select File
TIME
0
in1
in2
in3
in4
in5
in6
in7
in8
I (vin1
I (vin2
I (vin3
I (vin4
I (vin5
I (vin6
I (vin7
I (vin8
2*in1
inx2
inx3
inx4
/home/vlsi/moini/opt/hsview/adder.tr0 0
Misc. Options
Select X-axis
Overwrite
Debug
Highlight Selected
Force Eq. Overwrite
Save Config
No. of Panels
2
TIME
in1
in2

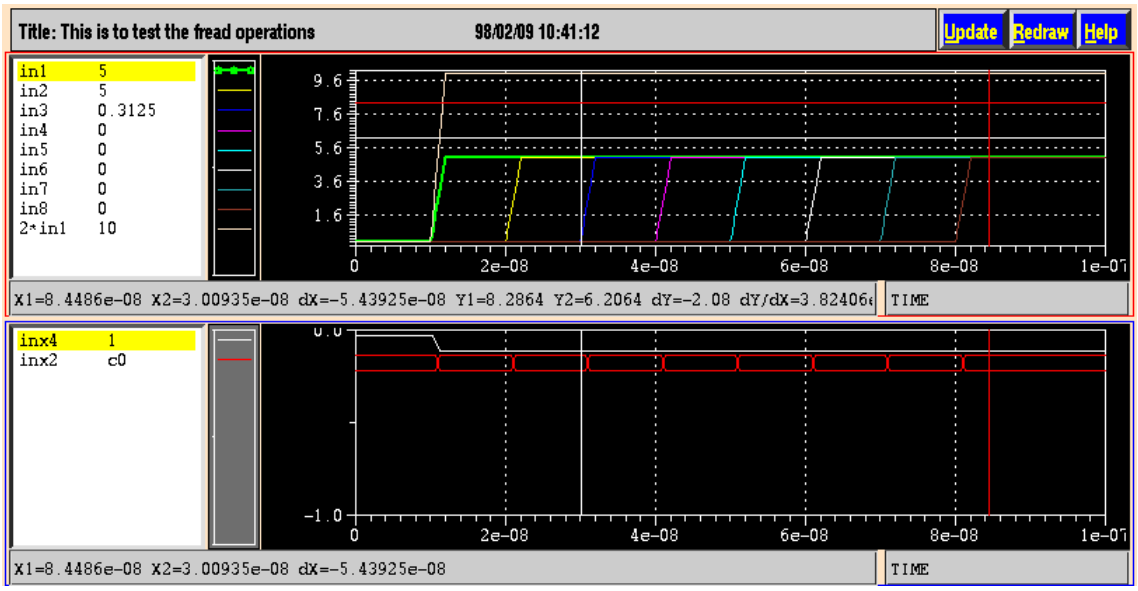
2.2 Panel Window

This window is in effect the main window in HSVIEW . Almost all functions can be activated from within this window. In fact the design has been made such that the user does not need to refer to other windows too often. Also the usage of shortkeys has been maximized. This makes interacting with the tool very easy and quick.

The panel window consists of a header and up to 8 panels. The header contains the title of the simulation file and several buttons for updating and redrawing the window.

Each panel has a plotting area, where the curves are shown, a browser where the signals in the panel are shown, a legend area where the legends for each signal are shown, and an X-axis label.

Note that the title and all the X-axis labels can be changed. This functionality is only provided for the purpose of printing.



2.3 Select Individual Options Window

This window is used for changing the characteristics of each panel and each signal.

The Xlog and Ylog buttons change the axis between a linear and logarithmic axis. The Xtick and Ytick buttons change the number of tick marks on each axis. The Xgrid and Ygrid buttons change the appearance of the grid lines across the X and Y axes.

The X1, X2, Y1, and Y2 boxes show the boundaries of the current panel. The boundaries can be changed via these input boxes. Remember to press the Return key after changing the values.

The "BG color" changes the background color of the current panel. It brings up a color selection window, through which a color can be selected.

The "Font color" changes the color of the font used in the panel. The Signal color changes the color of the signal (currently this does not work).

The "Line Width" changes the linewidth of the selected signal.

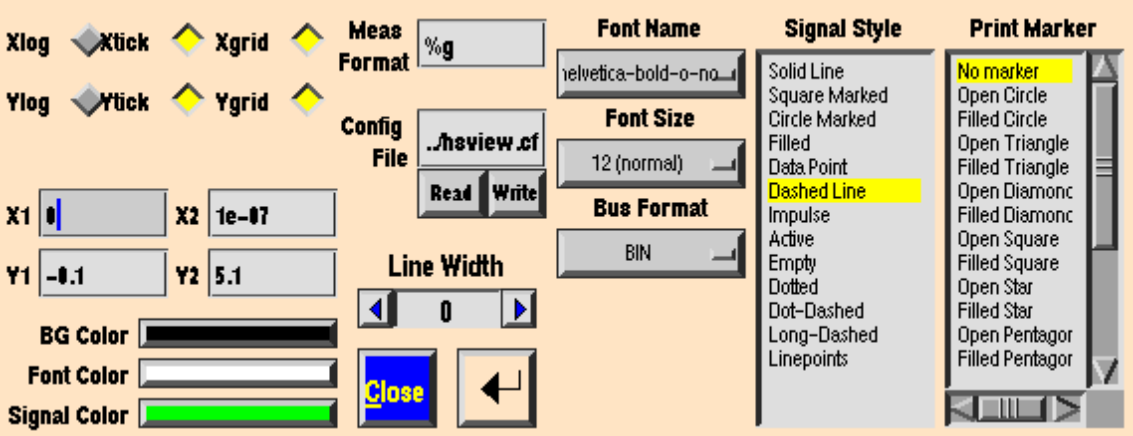
The "Signal Style" changes the style of the selected signal.

The "Font Name" changes the font being used in the panel, and the "Font Size" changes the font size used in the panel.

The "Meas Format" is used for changing the format of the signal values being printed during the measurement operation. This only acts during the operation. The format should follow the guidelines for printing "floating point" values in C language. Its default value is "%g". Any other

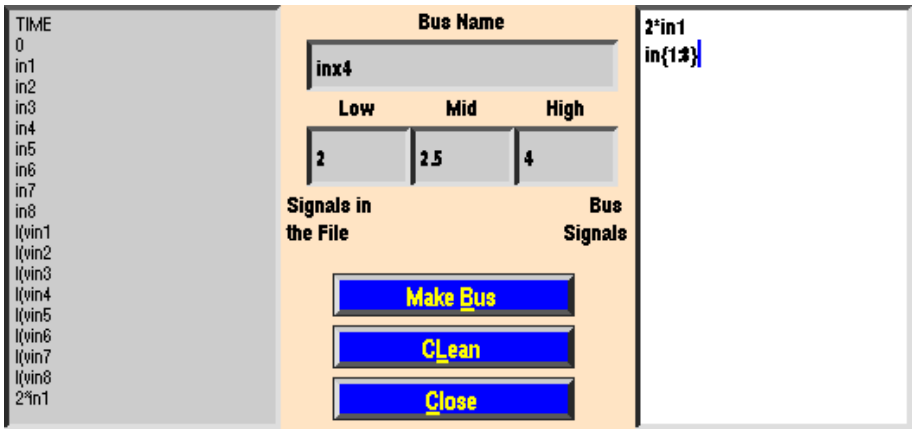
acceptable format in C language can be used. However, the user should be warned against using unacceptable formats.

The "Print Marker" is used for marking the data points using one of the specified styles. This is only used for printing, and it doesn't have any effect on the display of the data.



2.4 Bus Editor Window

This window is used when creating a digital bus from a series of analog signals.



2.5 Panel Help Window

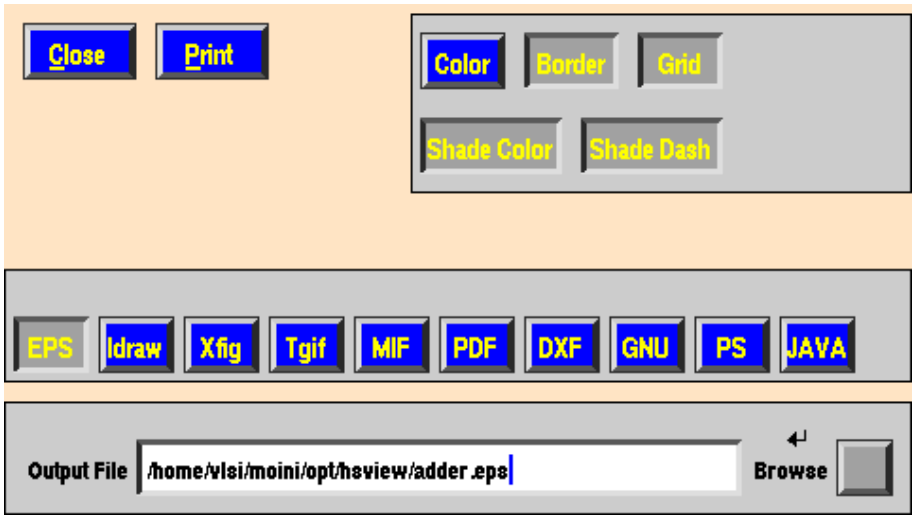
This window contains the shortkeys for HSVIEW , which can be used within the Panel Window.

2.6 Message Window

All the messages printed during the operation of HSVIEW are sent to this window. It only stores 1024 lines at a time.

2.7 Print Window

Through this window the printing options and functions can be controlled. HSVIEW can generate black&white or color plots. The main output format is postscript, which is default format ("EPS"). However through the use of "pstoedit"[1] other formats can also be generated.



The output file can be selected using the file selector, or be written manually.

Note that the DXF, GNU, and JAVA formats require other programs for a complete translation of the files.

Also in the Xfig, Tgif, MIF, and PDF formats, the translation of dashed lines is not complete and they are all translated to solid lines. The reason is that these format only support a limited number of "fixed dash styles". The best format for a complete translation is Idraw.

The options currently provided are: Color, Border, Grid, Shade Color, Shade Dash, and Auto Marker. The function of each option is obvious. The Shade Color and Shade Dash options are used when a signal with several sweep values is printed. In this case the color or the dash style of the lines are gradually changed. The Auto Marker option will force each signal to be printed with a marker style.

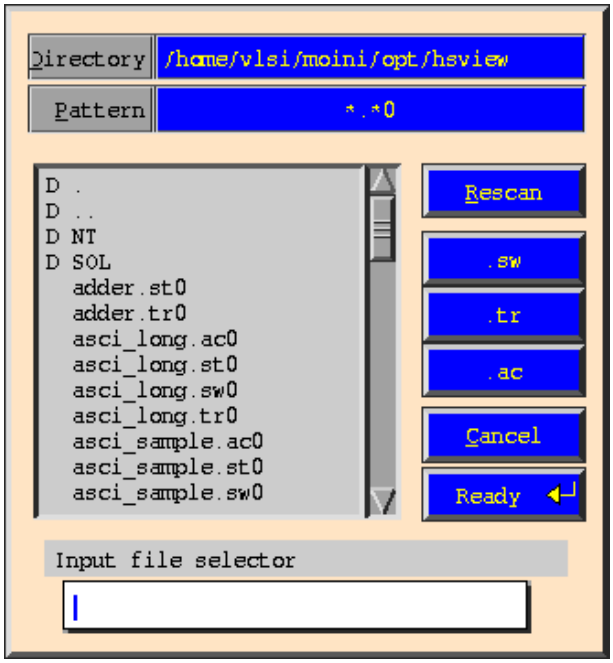
2.8 File Selector Window

The File Selector window is used for selecting a file. It has several fixed buttons, and input areas, such as Directory, Pattern, Rescan, Cancel, and Ready. Other buttons change according to the place where the file selector is invoked. For example, when selecting an input file, three buttons with .tr* .sw* and .ac* appear, or when selecting a file for printing buttons with .ps and .eps appear.

2.9 Select Signal Window

This window only contains a browser, which is an exact copy of the "Select Signals" browser in the Main Window, and has the same functionality. It is mainly used as another method for browsing through the signals, specially when the number of signals is too large.

[1] pstoedit is a program for transforming PS files into many other drawing formats. This is very convenient for changing the contents of the output file. This program can be obtained from ftp://ftp.x.org/contrib/applications/pstoedit/pstoedit.html


Directory
/home/vlsi/moini/opt/hsview
Pattern
*.*0
D .
D ..
D NT
D SOL
adder.st0
adder.tr0
asci_long.ac0
asci_long.st0
asci_long.sw0
asci_long.tr0
asci_sample.ac0
asci_sample.st0
asci_sample.sw0
Rescan
.sw
.tr
.ac
Cancel
Ready
Input file selector

3 Reading files

HSVIEW currently accepts four formats, the **binary** and **ascii** output from Hspice and HspiceStar, the **binary** output from Spice3f4 and later version, and a raw data format (RDF).

There are three methods for reading in the simulation files. The first method is through command line option, the second method is through the configuration file, and the third method is using the graphical interface and the file selector.

During the reading of each file, first the format of the file is detected, and then the file is read according to each format. Warning messages resulted from faulty files, or other problems are printed in the Message Window.

Note that all the data from the file is read into the working memory. Therefore, for very large files this may cause some problems. In this case it is recommended to print out only those signals that are required. However, for all other cases this gives the advantage of a single and very fast data access.

The RDF format is used to add the capability of HSVIEW to render data created manually, or translated from different formats. The RDF syntax is given in the Appendix.

4 Saving the data

The data from a file can be saved into an ascii file in a format readable by Matlab. This is particularly useful for special types of simulations, such as simulating a two-dimensional network, or a large number of statistical analysis, or for post-processing and pretty-plotting the data.

Currently, only the contents of one simulation file can be saved into a file.

For simulations which consist of sweeps as well, the signals are written as two-dimensional data.

At the beginning of the output file, a section is used which clears the signals if they already existed during a Matlab session. This section can be avoided by using the "-C" command line option.

Complex numbers are written as "a+ib" where "a" is real and "b" is the imaginary part. This happens mainly in the AC analysis. By default, however, the magnitude and phase of the signals are written to the output. In order to write the signals in the complex format the "-I" command line option should be used.

As the output file is typically up to ten times bigger than the original simulation files, it is recommended to use the filter files to only print the required signals.

A filter file simply contains the name of the signals to be written into the output file. The filter file can be used as a command line option. For example:

```
hsview -s filter-file infile.tr0
```

As during the translation of the simulation file into a matlab, often the graphical interface is not required, several options are provided to facilitate this. A typical command line option would be:

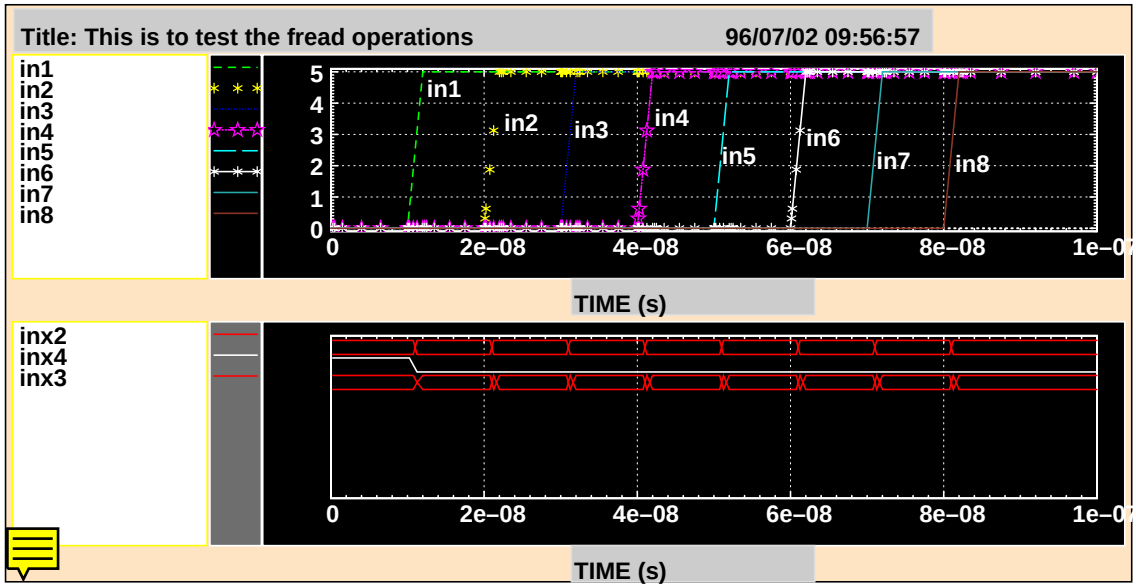
```
hsview -s filter-file -X -m outfile.m infile.tr0
```

The "-X" option disables the graphical interface. However, the program should still be able to access an X-Server. In other words you cannot use the program in a pure text console. The "-m" option specifies the name of the output file and its format (which is Matlab in this case).

As the name of the signals may not comply with the naming strategies of Matlab, some of them are changed during this procedure. This should be given special notice. This includes `-+(),:[]`. For signals which are represented by a number or start with a number, HSVIEW prefixes the signal name by an "X". In this case a warning message will also be printed.

5 Printing

A great effort has been put on creating high quality print out for documentation and presentation purposes. A sample output is shown below.



HSVIEW can print the graphical plots into several format. The main format is an encapsulated postscript format, which can be used in many documentation and word-processing programs. However, whenever other formats are required, through a filter HSVIEW can generate other formats. The filter program is external to HSVIEW and is invoked by HSVIEW . HSVIEW uses pstoedit, which a public domain program for converting postscript files into other vector drawing formats.

Note that if you want to compile pstoedit yourself, you might need to change some parameters[2].

In order to print simply type "P" in the "Hsview Panel". A new window, called "Hsview Print" will appear. You will find several options in this window.

- The "Color/B&W" button sets the output file to be either color or black&white.
- The "Border" button turns on or off the printing of the border of each object in the panel window.
- The "Grid" button turns on or off the printing of the grids.
- The "Shade Color" button sets the shading of the color when printing a swept parameter. This button and the next one are useful for easily recognizing the trend in the signal change when a sweep parameter is used.
- The "Shade Dash" button sets the printing of the swept parameter to follow a shading in the dash style of the line.
- The buttons from "EPS" to "JAVA" set the printing format.
- The output file name can be selected using the browser or typed in. Note that the file name is automatically set to the name of the input file, and its extension is set according to the selected output format.

[2]In the file pstoedit.ph the deviceinfo should be replaced by currentpagedevice. This is a problem with more recent versions of ghostscript.

Note that if you type "p" in the panel window, the graph will be printed without bringing up the print window. This is useful as a quick way for printing the results and looking at them.

6 Measurement

The measurement mode provides a quick way of looking at the values of the data points. In order to start the measurement mode simply type "m" in the panel window. You will notice that a new field will appear. This field will display several parameters related to the position of the measurement cursors. There are two measurement cursors. In order to set the position of the first cursor press the "right" button of the mouse. In order to set the position of the second cursor press the "left" button of the mouse. In order to get continuous measurement while moving the mouse, press the "left" button and drag it on the panel window. You will notice that the output filed will display $X1$, $X2$, $Y1$, $Y2$, ΔX, ΔY, and $\Delta Y/\Delta X$. Also in front of each signal name the value of that signal is displayed. The shown value is the value of the next closest data point to the position of the cursor in the X direction. So there is no interpolation performed between the data points (This may, however, be included in the future versions if necessary).

Note that for a "digital panel", only $X1$, $X2$, and ΔX are shown. In this case the value shown in front of the signal will also be digital. The format of the digital buses can be either binary, hexadecimal, octal, or decimal. By default when a bus with a width more than 8 bits is created, its value will be shown in hexadecimal. For less than 8 bits the value is shown in binary format. You can change the format of a bus by 1- plotting it in a panel, 2- selecting it in the panel, 3- bring up the "Select Individual Options" window (press "s"), 4- choose the desired format through the "Bus Format" menu.

7 Equation evaluator

The equation evaluator is a powerful tool for performing postprocessing mathematical operations without the need to a simulator which provides this functionality, and also without the need to resimulating the design. The mathematical expressions are very general. Here several examples are provided:

$$\begin{array}{l} new_signal = 2 * in1 + 3 * out \\ new_signal = sin(in1 + log10(out) * (out2^3.3)) \\ sin(in1) \end{array} \quad (1)$$

In the first example a new signal names $signalnew$ is created from the existing signals $in1$ and out, based on the equation provided. In the second example a complex equation is defined. In the third example an equation is defined. In the first and second examples the name of the new signal is determined by the string before the "=" sign. However, in the third example the name of the new signal will be assigned to the expression used, i.e. "$sin(in1)$". This is useful when comparing various expressions.

If the name of an equation is similar to an existing and *original*, i.e. it is not the result of an equation or bus creation, then the signal will not be generated and an error message will be printed.

If the name of the equation is similar to an existing signal which has been resulted from another equation, its value will be replaced by the new values. A warning message will be printed to inform about this overwrite.

7.1 Mathematical expressions

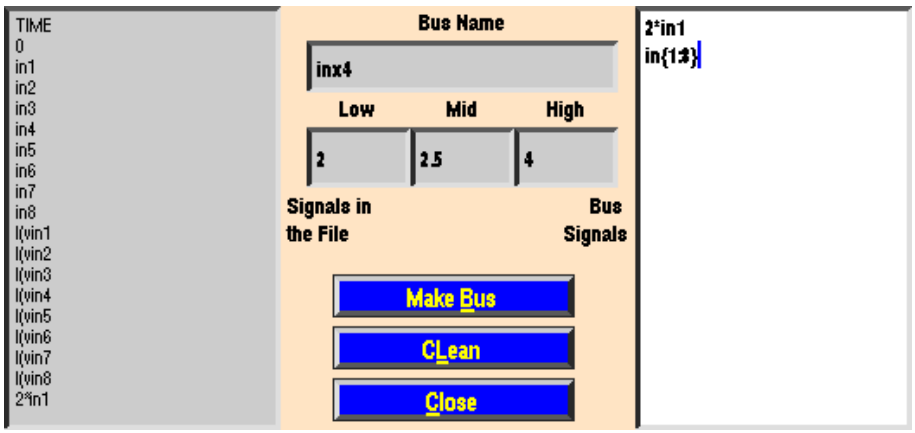
Many functional expressions are implemented in HSVIEW . These include the addition (a+b+c), subtraction(a-b-c), unary negation (-a), multiplication (a*b*c), division (a/b*c), modulus or remainder (a%b), and power ($a\hat{b}$). In addition, the following mathematical functions are allowed.

expression	function
sin(a)	$sin(a)$
cos(a)	$cos(a)$
tan(a)	$tan(a)$
asin(a)	$asin(a)$ (the inverse function of sin)
acos(a)	$acos(a)$ (the inverse function of cos)
atan(a)	$atan(a)$ (the inverse function of tan)
sinh(a)	$sinh(a)$
cosh(a)	$cosh(a)$
tanh(a)	$tanh(a)$
exp(a)	e^a
log(a)	ln(a)
log10(a)	$log(a)$
sqrt(a)	$\sqrt{a}$
floor(a)	$floor(a)$
ceil(a)	$ceil(a)$
abs(a)	$abs(a)$
hypot(a,b)	$\sqrt{a^2+b^2}$
deg(a)	$\frac{a \times 180}{\pi}$
rad(a)	$\frac{a \times \pi}{180}$
adc(a,threshold)	=0 if $a < threshold$, =1 if $a > threshold$
threshold(a,low,mid,high), or thresh(a,low,mid,high)	=0 if $a < low$, =x if $low < a < mid$, =X if $mid < a < high$, =1 if $high < a$

8 Bus Editor

The bus editor is used to create a digital signal from an analog signal. In order to invoke the bus editor press "b" in the Panel window. There are several fields within the bus editor. The browser in the left (*Signals in the File*) has a list of all the signals in the current selected file. The "Bus Name" input is used to enter the name of the bus to be created. This name should be unique. The "Low", "Mid", and "High" are used to specify the low, mid and high threshold levels used in converting the analog signal to the digital signal. The function that is used for this purpose is "threshold", which is specified in table 7.1.

In order to add a signal into the list of the signals the user can either press on its name in the left browser, or enter the name manually. When the name of the signal consists of numbers an easier syntax can be used. For example if signals *in1 in2 in3 in4 in5 in6 in8* all exist, they can all be added by typing *in{1:8}*. This is specially useful for very long buses. An example is shown in the figure.



9 Placing Text

In order to simplify the task of adding text onto the plots, for example for annotating the curves, a text paste facility is implemented. By pressing "t" the "Hsview Text Paste" is invoked. In order to add a text simply type the text, and then press the "left mouse button". In order to relocate text press the mouse button at a new location. In order to delete the text press the "right mouse button". In order to edit an existing text press the "middle mouse button" on top of the text in the panel window. The text will be placed in the edit window and can be deleted and replaced by a new text.

If you want to repeat the same text in several places, you should change the name. This can be done by adding an extra blank space after the name. This should be remembered when trying to delete a text from the plot.

During the time that the text paste mode is on, the left and right mouse buttons have a different function. In order to get back to the normal mode, turn off the text mode by pressing "t" again.

10 Configuration

An important feature of HSVIEW is its ability to save and read the configuration in a simple scripting language. This is specially useful for professional designer going through the same measurements and simulations many times.

By default the configuration is automatically saved when you quit the program into a file named "hsview.cfg". In order to read the configuration file one of the following methods can be used.

- During the start of the program and in the command line type
  `hsview -c hsview.cfg`
  Of course you can specify another name if a configuration fiel with that name exists.
- In the "Select Individual Options" window there is an input field called "Config File". You can set the name to the desired one and then press the "Read" or "Write" buttons in order to read or write the configuration into the file.

You can edit the configuration file manually, as long as the added code is consistent with the scripting language.

Currently the following commands are supported in the configuration file. Note that the program is case sensitive and the words should be exactly typed as follows.

- `"*"`
  A line starting with a * is considered to be a comment.
- `load_file file_name`
  This reads the file named "file_name".
- `equation name expression`
  This creates an equation with the name "name" based on the "expression". For example:
  `equation new_in1 in1+sin(in2)^3.1`
- `bus name format low mid high bus_signal1 bus_singal2 ...`
  This creates a bus with the name "name".

  The "format" indicates the type of the signal, which can be binary=1, hexadecimal=2, octal=3, and decimal=4.

  The "low mid high" indicate the threshold levels used to translate the analog signal into a digital signal. These are floating point numbers.

  The "bus_signal1 ..." are the name of the signals used to create the bus. The ordering of the signals is important. Refer to the section 2.4 for more information on creating a bus.
- `hsview_window xul yul xlr ylr`
  This specifies the location and geometry of the Main window.
- `panel_window xul yul xlr ylr`
  This specifies the location and geometry of the Panel window.
- `digital_panel panel_number`
  This sets the panel "panel_number" to be a digital panel. The digital panel can only be used for drawing digital signals. Analog signals can be plotted in non-digital panels. Note that $0 \leq panel\_number \leq 8$. Any value outside this range will be ignored.
- `panel_numbers N`
  This set the number of panels to "N".

- `add_signal panel name sweep id1 id2 color style file`
  This command will add a signal named "name" into panel "panel".

  "sweep" is the sweep number of the signal.

  "id1" and "id2" are the identification numbers of the signal. These should be unique. "id2" can be -1, and it accepts other values only for "bus signals" with more than one bit.

  "color" is a number which identifies the color, and "style" is the style of the signal.

  "file" indicates the file to which this signal belongs.

- `add_text panel x y text`
  This adds the text to a panel at location (x,y).

- `panel panel_num xmin xmax ymin ymax xlog ylog xgrid ygrid f_size f_sty f_col p_col xtick ytick`

  This command sets all the parameters of a panel. This includes, the axes values, linear/log, grid, font size and color, panel color, and tick values.

- **Various other parameters:**
  There are several other options, which can be specified in the configuration file. These include "print_color", "print_border", "print_grid", "print_sweep_color_shade", "print_sweep_dash_shade".

It is highly advised that for the first time let HSVIEW save the configuration file, and then the user can easily change its content.

NOTE: Whenever a file has already been loaded, and then it is loaded again by reading a configuration file, the file will be erased from memory and all the signals belonging to that file will be deleted from the panels. After this the operations in the configuration file are performed.

The following file shows an example of a configuration file generated by HSVIEW .

```
****************************************
* Config file for hsview
* Date: Thu Jun 11 17:08:29 1998
****************************************

load_file /home/vlsi/moini/opt/hsview/adder.tr0
******
  equation 2*in1 2*in1
  bus inx2 2 2 2.5 3 in{1:8}
  bus inx3 2 2 2.5 4 in{1:8}
  bus inx4 1 2 2.5 4 2*in1
******
load_file /home/vlsi/moini/opt/hsview/sample.sw0
******
******
hsview_window     27 32 545 405
panel_window     232 450 810 415
digital_panel     1
****************************************
* various variables
panel_numbers            2
print_color              0
print_border             1
print_grid               1
print_sweep_color_shade  1
```

```
print_sweep_dash_shade   1

****************************************
* Write the signals in each panel
* add_signal panel signal sweep id1 id2 marker col sty file
add_signal 0    in1     0 1 -1 -1 2 5 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in2     0 2 -1 16 3 4 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in3     0 3 -1 -1 4 9 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in4     0 4 -1 8 5 10 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in5     0 5 -1 -1 6 11 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in6     0 6 -1 16 7 12 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in7     0 7 -1 -1 21 0 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 0    in8     0 8 -1 -1 22 0 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 1    inx2    0 1 2 -1 1 0 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 1    inx4    0 3 -1 -1 7 0 /home/vlsi/moini/opt/hsview/adder.tr0
add_signal 1    inx3    0 4 5 -1 1 0 /home/vlsi/moini/opt/hsview/adder.tr0
****************************************
* text in the panels
add_text 0 1.19626e-08 3.3944 "here"
add_text 0 3.19626e-08 3.8936 "there"
add_text 0 4.13084e-08 2.3544 "there  "
add_text 0 6.16822e-08 3.1448 "there is a life for me"
add_text 0 5.15888e-08 0.9816 "there "
****************************************
* panel panel_num xmin xmax ymin ymax xlog ylog xgrid ygrid f_size f_sty f_col p_col xtick ytic
panel 0 0 4.55 -0.2 10.2 0 0 1 1 12 4 7 0
panel 1 0 1e-07 -1 0 0 0 1 1 12 4 7 0
```

11 Messages

The message window is used to feedback on the operation of HSVIEW . In order to see the message window press "K" in the Panel window. In fact the shortkey "K" toggles the message window, so you can close the window by pressing "K" again.

12 Command line options

Currently the following command line options are accepted.

```
--------------------------------

 USAGE :
 hsview  [-X] [-d] [-m] [-c config_file] [-C] [-o output_file]
         [-s signal_file] input_file1 input_file2 ...

The options are:
 X                => do not display the hsview windows.
                     Just parse the data and write the matlab file.
 c                => read (and write) the configuration into file
 C                => do NOT clear the matlab variables
 d                => debug info. prints out a lot of garbage
 h                => prints out this help message
 m                => prints out the data in matlab format
 s                => only write those signals named in the file
 o                => to specify the output file for matlab
 input_file       => input file name
 config_file      => configuration file name

--------------------------------
Examples
        hsview  sweep.tr0
        hsview -I ac_test.ac0
        hsview -X -m -o output.m sweep.tr0
        hsview *.tr?
        hsview -c hsview.cfg
--------------------------------
```

13 Shortkeys

In order to accelerate the functions, most operations are performed using shortkeys. These can be found in the help window (press "h"). They are listed here. As these may change with the new revisions, it is suggested that the user reads through this help window.

```
Shortkeys for signal
   navigation:

^ = CTRL
# = ALT
^[= ESC
SH= Shift key
S = Space Key
AL= Arrow Left
AR= Arrow Right
AU= Arrow Up
AD= Arrow Down
------------------------
+ : increase the width
 of the name browser
 (useful during measure
  mode)

- : decrease the width
 of the name browser

a : draw a signal(s)
A : toggle autoscale
    default=on
b : toggle the bus editor
    window
d : delete a panel
D : add a panel
e : equation evaluator
ff: full scale
fx: full scale only in X
fy: full scale only in Y
gx: toggle the grid for X
gy: toggle the grid for Y
h : toggle the display
    of the help window
H : toggle the display
    of the main help window
k : kill the selected
    signal
lx: Toggle the log scale
    for X
ly: Toggle the log scale
    for Y
m : Toggle the measure
    mode
M : toggle the display of
    the main window
```

```
p : print
P : toggle the display of
    the print window
Q : quit the program
    (after a query!)
s : toggle the display of
    the panel option
S : toggle the display of
    the signal window
t : toggle the text mode
    and window
U : Update file
x : pan in the X-axis
    relative to mouse
y : pan in the X-axis
    relative to mouse
zx: zoom-in in X-axis
    relative to mouse
zy: zoom-in in X-axis
    relative to mouse
zz: disable zooming acti-
    -vated by the 'z'
Zx: zoom-out in X-axis
    relative to mouse
Zy: zoom-out in Y-axis
    relative to mouse

^U: update
^o: Open file
^R or AR':
     redraw
'AL'  'AR'  'AU'  'AD':
     move to the specified
     direction by 1/4th
',': centre to the mouse
     position
----------------------
Mouse operations

Mouse1:
 Select this panel

^-Mouse1:
 Add this panel to
 the selected panels

Drag-Mouse2 (Without SH):
 Zoom in the X or Y
 axis

Drag-Mouse2 (With SH):
 Zoom in a box
```

```
----------------------
In the Measurement mode
the following keys operate

Mouse3:
 Set the position of the
 first cursor
Mouse1:
 Set the position of the
 second cursor
----------------------
In the Text Paste mode
the following keys operate

Mouse1: Add or relocate
 text
Mouse3: Delete text
```

14 Known Bugs

- In some cases if you zoom-in too much you will see many vertical lines in the panel: This is a bug related to the particular graphical library (XForms) that has been used for developing the interface. Hopefully, when the bug is fixed in that library, this bug will go away.
- The points do not seem to be on their exact position: This is caused by the fact that the data is represented by floating point numbers, while the points on the display are represented by `short` integers. The data truncation causes a difference between the real and displayed values. However, still in the measurement mode the values that are printed out are the exact values of the data.

15 Bug reports & Suggestions

In the case that an error happens, first try to see if the error is repeatable. If it is, then use the debug option to get a very verbose output. This output can then be used to locate the position of the error. Some parts of the code are not error proof. From example, if the parameters in the configuration file are not correct, then HSVIEW may crash.

In any case if you find a persistent bug, please report them to "moini@eleceng.adelaide.edu.au".

16 Obtaining HSVIEW

Currently HSVIEW binary files are free of charge to any individual, organization, institution, companies, and universities. The binaries are available for Solaris 2.5.1, SunOS 4.1, and DEC OSF 4.0, and can be obtained through the WWW in
http://www.eleceng.adelaide.edu.au/Personal/moini/hsview/

A Raw data format (RDF)

RDF is a free ascii file format for specifying numeric data to visualization and manipulation in HSVIEW . This chapter specifies the syntax of RDF. In order to keep the format very simple, the syntax has been kept very simple.

A.1 Header

In the header of RDF several parameters may be specified. Some of these parameters are optional and some are compulsory. Each parameter is described below.

- **Raw_data**
  This is the first line in the file. This line should be specified as it is typed here. As this is used for detecting the format of the file, any variations will result in not being able to read the file in.
-
  Any line starting with a is considered to be a comment line, and ignored.
- **Title:***the title*
  This specifies the title used for file.
- **No. Variables:** *no_variables*
  This specifies the number of variables that is expected to be read in. As this is used in the calculation of the amount of memory allocated for reading the data, if it is less than the actual value, the program may crash.

- **No. Sweep:** *no_sweep*
  This specifies the number of sweep points. In fact this specifies the second dimension of the data, when a two-dimensional data is specified. By default the number of sweeps is zero. If *no_sweep* is greater than one, for each data point there should be at least *no_sweep* columns of data.

- **No. Points:** *no_point*
  This specifies the number of data points for each signal. As this is used in the calculation of the amount of memory allocated for reading the data, if it is less than the actual value, the program may crash. This line should be the last line of the header.

A.2 Data Body

Signals (variables) and data are specified using a very simple syntax.

- **Signal:** *signal_name*
  A new signal with the name *signal_name* is added to the list of signals.

- *data1 data2 ....*
  This allocates the data for the signal.

The following gives an example of a simple RDF file.

```
*Raw_data
**************************************************
* Header starts from here
**************************************************
Title: A just for fun file,
No. Variables: 3
No. Sweep: 2
* No. Points: should be the last line of the header
No. Points: 10
**************************************************
* Data starts from here
**************************************************
Signal: ali
1 5
2 10
3 15
4 20
5 25
6 30
7 35
8 40
9 45
10 50
Signal: ali2
2  2
4  4
6  6
8  8
10 10
12 12
14 14
16 16
```

```
18 18
20 20
Signal: alix
3   3
6   6
9   9
12  12
15  15
18  18
21  21
24  24
27  27
30  30
```