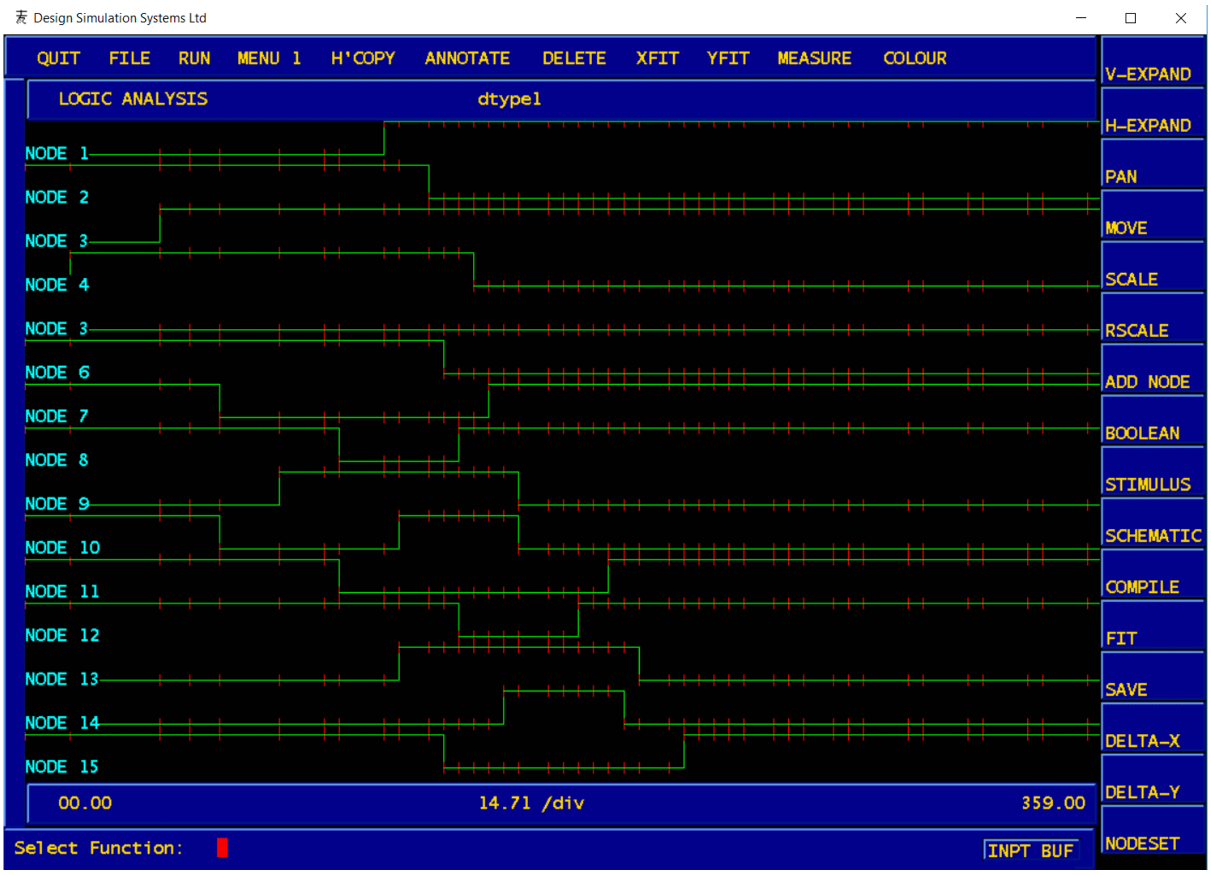
Dsim

High Level Digital Circuit Simulator

Mark Sitkowski
Design Simulation Systems Ltd
http://www.designsim.com.au



Contents

Introduction

Dsim is a high-level, event-driven digital circuit simulator. Most of the outline code was written in Liang Kee's restaurant, in Singapore, during a particularly productive lunch hour as a result of idly wondering if these design goals were achievable:

- The input would be a netlist, modelled on the SPICE netlist, with which the majority of users would be familiar.
- The netlist would be derived from a schematic, created in the schematic editor GEX.
- The simulator would not require a circuit to be broken down to gate level but would have an understanding of complex logic functions, which it would treat as circuit 'primitives'.
- The output would be graphical, and the interface/display comparable to the analogue interface, Vspice. We would call it ‘Vlogic’.

Accordingly, Dsim understands high level primitives, and simulates a circuit at their level. This means that the netlist compiler, which converts the graphical circuit representation into a netlist, can be very fast and memory efficient, since it only has to flatten the circuit hierarchy to reduce the circuit to the primitive level. Having high-level primitives such as these also makes for very efficient simulation since, at each time-step, only one primitive needs to be evaluated per logic function, not a bunch of gates.

Dsim development has been recently resuscitated, on a very low level of priority, due to the feedback we have received.

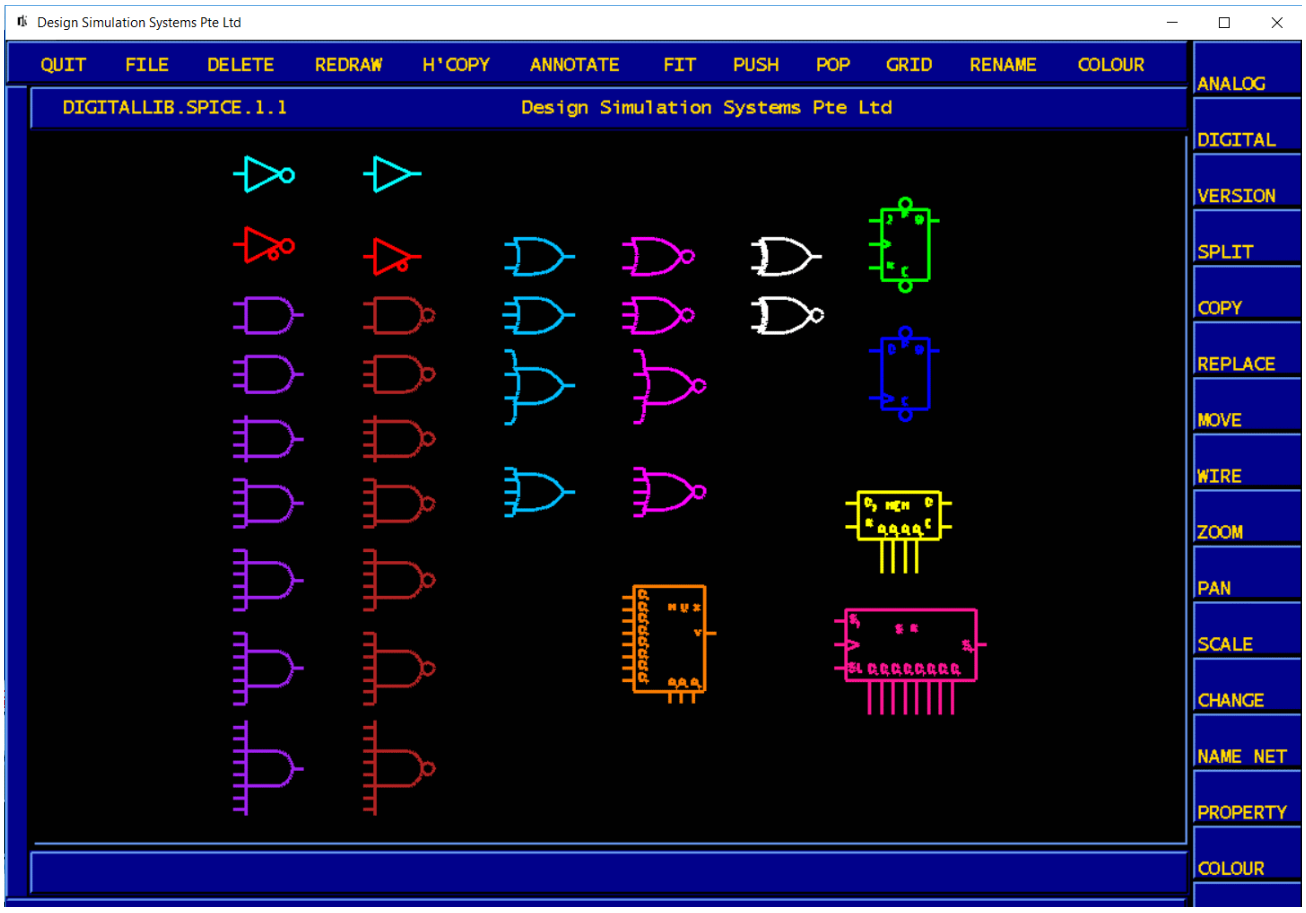
Supported Primitives

- Non-inverting buffer
- Inverter
- Non-inverting Tristate buffer with positive enable
- Non-inverting Tristate buffer with negative enable
- Inverting Tristate buffer with positive enable
- Inverting Tristate buffer with negative enable
- Multiple-input AND gate
- Multiple-input NAND gate
- Multiple-input OR gate
- Multiple-input NOR gate
- Two-input Exclusive-OR gate
- Two-input Exclusive-NOR gate
- Transparent R-S Latch
- Edge-triggered R-S Latch
- D-type Flipflop
- JK Flipflop
- Multiple-input Multiplexer
- Multiple-output Demultiplexer
- Memory (RAM or ROM)

- Parallel-load Shift Register
- Binary Counter (under development)
- Decade counter (under development)
- Multiple-input Full Adder (under development)

All clocked devices are positive edge triggered.

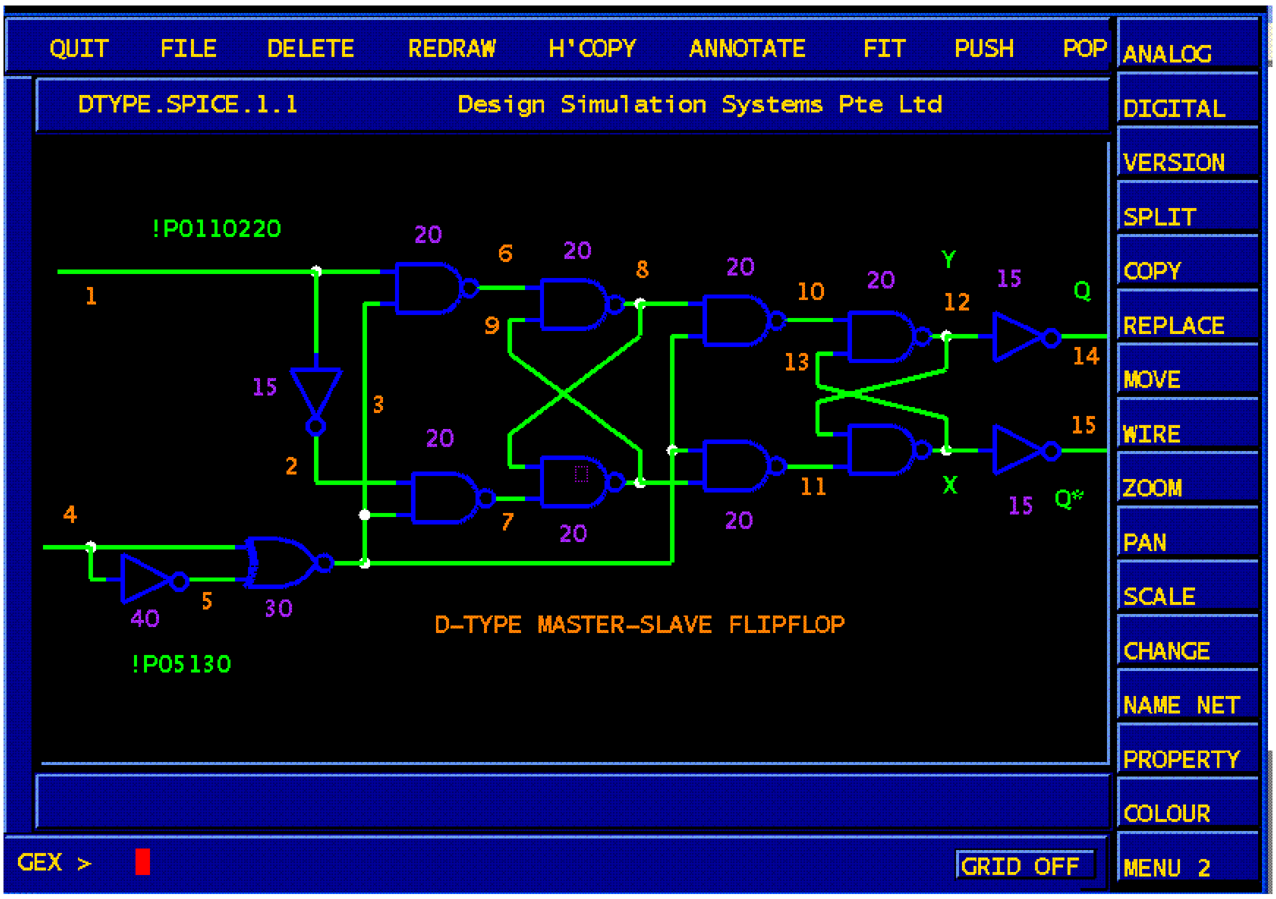
Some of the circuit symbols used by GEX for the creation of schematics are shown below.



The examples which follow carry little explanation, and are only presented to highlight particular features of Dsim.

Example Schematic

Digital circuit schematics are created in GEX, in exactly the same way as analogue schematics, with the obvious exception that they use devices from the 'DIGITAL' menu.



This above shows a D-type flipflop, done the hard way. We've only used it as an example because the path through the logic is a tortuous one, and the events are complicated by multiple feedback paths.

Netlists

The simulator needs to be fed a netlist, which can be created by hand, or from a GEX schematic by invoking the digital netlist compiler, GTL.
The netlist format is totally based on the format used by the SPICE simulator, since most people are familiar with it.
The netlist for the above schematic looks like this:

DTYPE MasterSlave
.sim 1 360 0
.print sim v(1) v(2) v(3) v(4) v(5) v(6) v(7) v(8) v(9) v(10) v(11) v(12) v(13) v(14) v(15)
!C 4 0 75 150
!P 1 0 110
U1 1 2 INV 15

```
U2 4 5 INV 40
U3 4 5 3 XNOR 30
U4 1 3 6 NAND2 20
U5 2 3 7 NAND2 17
U6 6 9 8 NAND2 20
U7 7 8 9 NAND2 20
U8 8 3 10 NAND2 20
U9 9 3 11 NAND2 20
U10 10 13 12 NAND2 20
U11 11 12 13 NAND2 20
U12 12 14 INV 15
U13 13 15 INV 40
.end
```

The first line is a free-form title, while the second line is a simulator directive, specified in a similar way to the SPICE ".tran" directive, as follows:

.sim <step(ns)> <stop(ns)> <start(ns)>

The .print line also works the same as the corresponding SPICE directive, with a list of the node numbers to be displayed.

Components are specified as:

DeviceID input1 input2...inputN output1 output2...outputN name delay1 delay2 delayN(ns)

The **!C** line, is a continuously repeating clock, and the **!P** line, is a non-repeating input signal which, here, drives the 'D' input..

Syntax is as follows:

!C <driven node> initial_value transition1 transition2 (ns)
This waveform will run repeatedly for the full term of the simulation

The **!P** stimulus is, here, a single transition, starting at logic '0', and rising to a logic '1', at 110ns, which is maintained till the end of the simulation. If additional transitions are required, they can be added individually.

!P <driven node> <init_value> transition1 transition2...transitionN (ns)

There is another directive, not used above:

!F <driven node> <state>
This forces a logic value <state> onto a node <node>

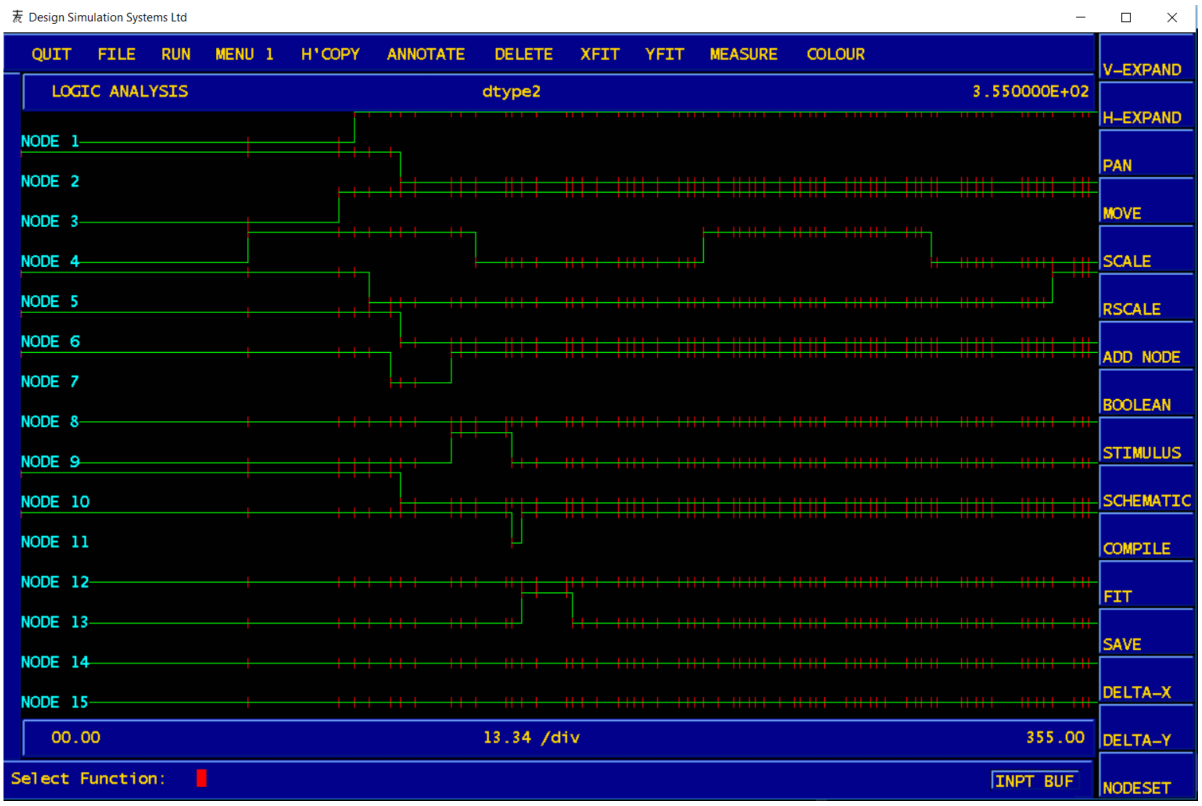
Invocation

Dsim is started either by hand, as

- **dsim < dtype.net**

or from its graphical interface, **'vlogic'**, which is unashamedly copied from the Vspice interface, and which displays the simulation results.

- **vlogic dtype.net**



LOGIC ANALYSIS dtype2 3.550000E+02

NODE 1
NODE 2
NODE 3
NODE 4
NODE 5
NODE 6
NODE 7
NODE 8
NODE 9
NODE 10
NODE 11
NODE 12
NODE 13
NODE 14
NODE 15

00.00 13.34 /div 355.00

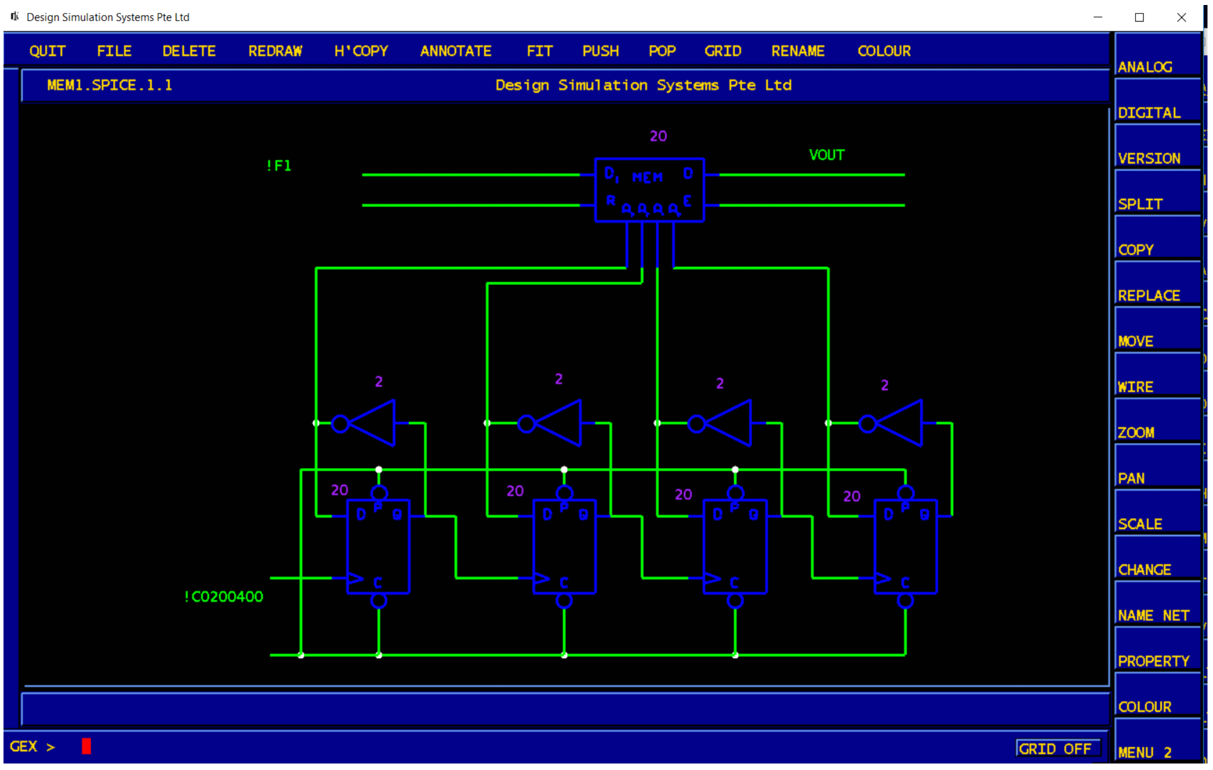
Select Function:

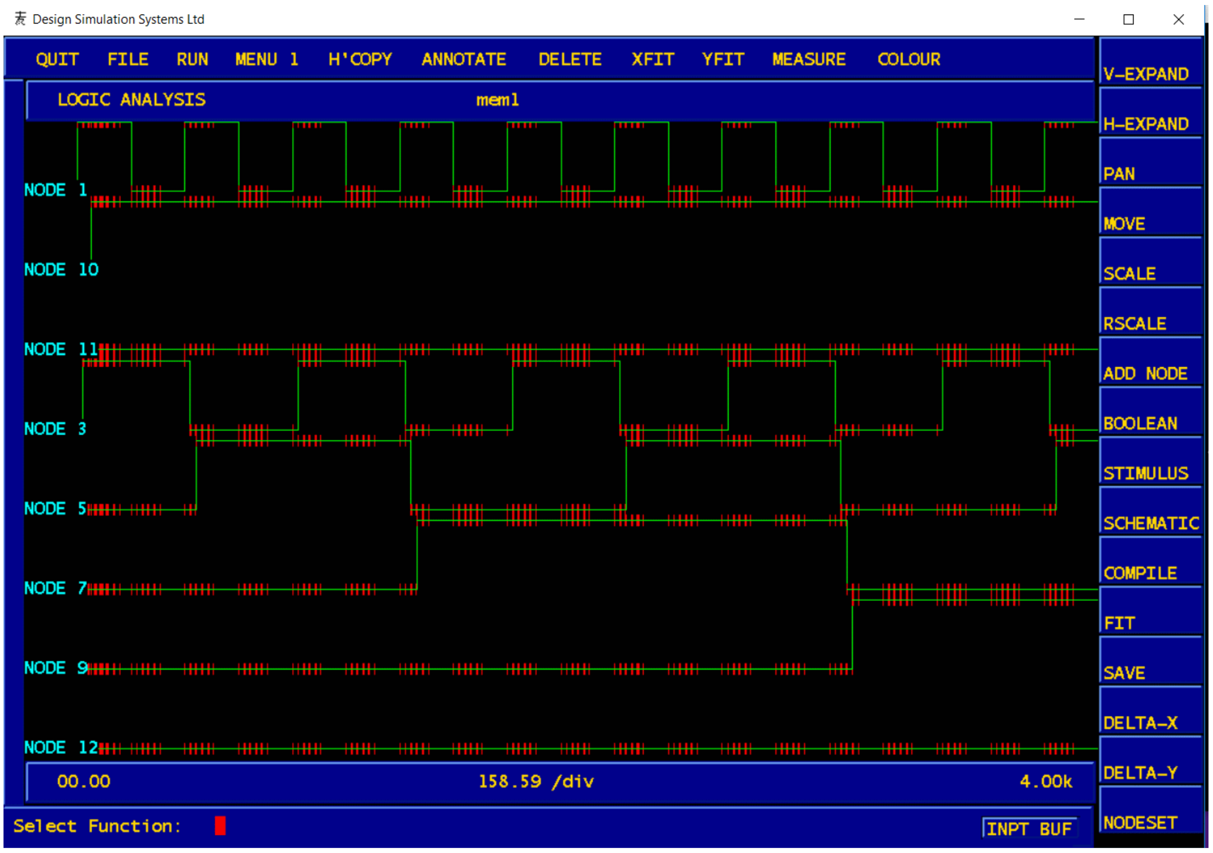
The results of the dtype netlist simlation are shown above. The tick marks on each waveform are the event markers.

Example Using the Memory Primitive

This basically simulates a read-after-write cycle, with the address lines driven by a binary counter

```
MEM
.sim 1 4000 0
.print sim v(1) v(10) v(11) v(3) v(5) v(7) v(9) v(12)
!C 1 0 200 400
!F 11 1
!P 10 0 250
!F 13 1
!F 14 1
U0 9 7 5 3 13 11 14 12 MEM 20
U1 1 2 10 10 3 DFF 20
U2 2 4 10 10 5 DFF 20
U3 4 6 10 10 7 DFF 20
U3 6 8 10 10 9 DFF 20
U4 3 2 INV 2
U5 5 4 INV 2
U6 7 6 INV 2
U6 9 8 INV 2
.end
```





Buffer Toggle

A pushbutton located at the lower right of the graph frame selects between the two waveform buffers, its legend switching to reflect the one currently displayed, as follows.

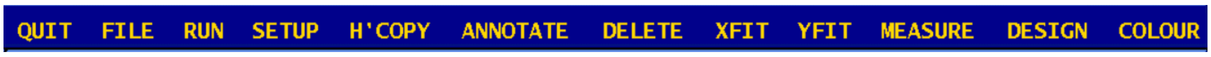
- **PROC BUF**

Displays the current contents of the processing buffer.

- **INPT BUF**

Displays the current contents of the input buffer.

MAIN MENU BAR

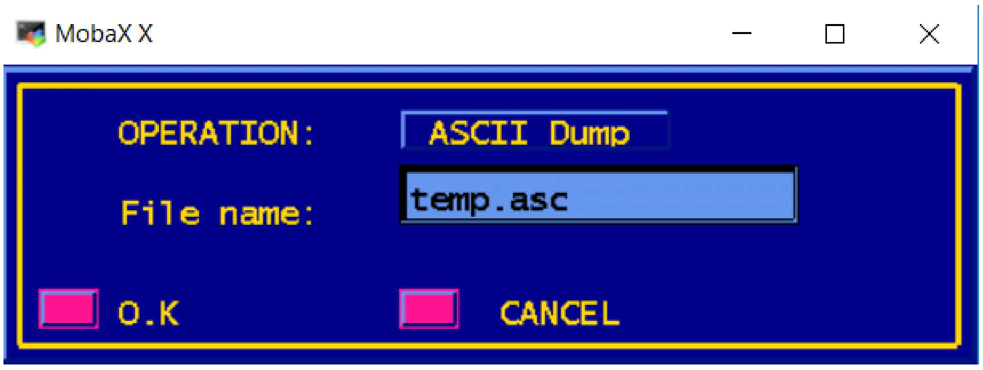


The main menu bar gives access to the following functions:

QUIT
Quits gracefully.

FILE
The ASCII file read function enables more ASCII data to be read into the internal data structures. The file should be in the Berkeley format but, since the first column will also be displayed, it need not contain a sweep variable. This maximises the amount of data read from a 255 character Unix line. Ideally, the number of data points in the new data should be the same as that comprising the waveforms already displayed but, if the two are different, the new data will either be zero-padded or truncated depending on whether there are fewer or more points, respectively. The ability to read in as many ASCII files as required is especially useful when wishing to compare two sets of simulation results (see 'SAVE') The only limit on the amount of extra data read in is that the total number of displayed waveforms should not exceed 40

The ASCII file dump function will dump the first ten waveforms on screen to the named file. If the process buffer is being displayed, it will write a single waveform

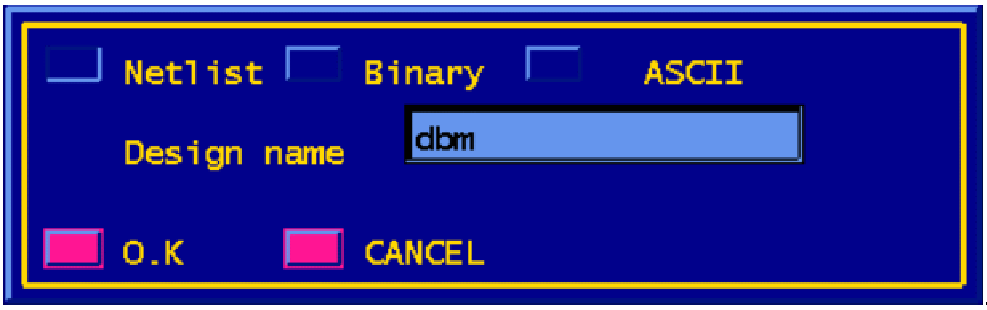


OPERATION: ASCII Dump

File name: temp.asc

.

RUN
Either runs a simulation, reads the binary file of results from a previous simulation, or reads an ascii dump of a previous simulation.


Netlist Binary ASCII

Design name dbm

O.K CANCEL

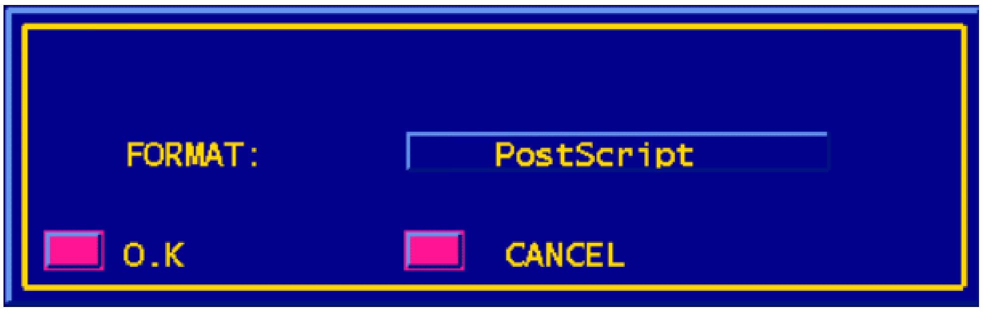
.

MENU 1

Returns to the main menu from the colour menu

H’COPY

Select either a PostScript or HPGL dump of the screen to a file


FORMAT: PostScript

O.K CANCEL

.

ANNOTATE

This command permits annotation of the display area with up to 80 characters per note. The text is glued to the cursor, and may be placed anywhere on the drawing. No case conversion is performed on the text.

DELETE

Removes waveform selected by mouse from display and prompts for 'y' to continue. Typing anything other than a lower-case y aborts the command and redisplays the waveform.
If deletion is confirmed, all waveforms are rescaled to use up the space left by the deleted waveform, the buffer entry for the waveform is cleared and the memory made available for re-use.
.

XFIT

Redisplays the data to fit exactly in the x-axis. All data currently in the buffers is displayed for every waveform.
XFIT does not redisplay signal names relating to waveforms, on the basis that the user may wish to otherwise annotate the display.
On the other hand, YFIT does

YFIT

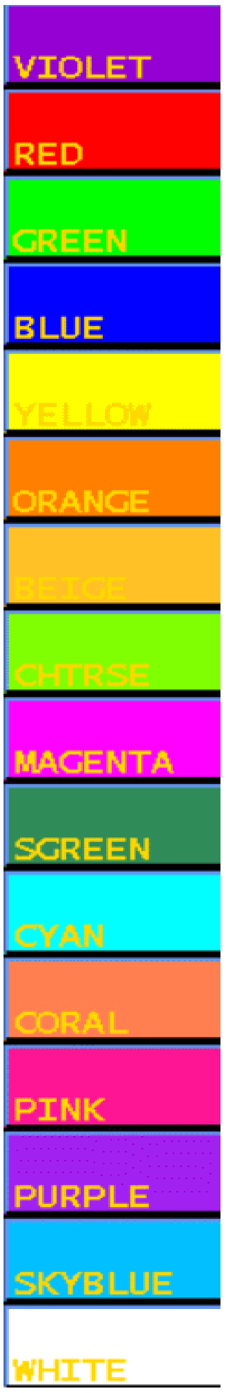
Divides the screen height by the number of waveforms and scales all waveforms to occupy the same height. Waveforms are positioned, in order of buffer number, starting at the top of the screen, to the centre of their allocated space. This is invoked automatically at the end of each simulation, or when waveforms are read in from a file.
All signal names are redisplayed at the lower left of each waveform.

MEASURE

Prints the node number, time value and logic level of the selected waveform. Note that the horizontal axis is actually quantised by the event times. This means that the point displayed will be the next event in time.
Values are displayed in floating point format, since this function is inherited from Vspice. If the file 'name.xref' exists, Vlogic will have read it on start-up and entered the signal names corresponding to the node numbers being plotted into its internal data structures.
In these circumstances, 'MEASURE' will report the name of the signal instead of the node number.
In the event of an ASCII input file, the signal names will have been read directly from the file.

COLOUR

Selecting COLOUR from the menubar displays the colour menu at the right-hand edge of the screen. Selecting it again returns to the default menu..
The colours of all drawing elements may be changed, but body drawings may need special attention, as no nets are formed from the component wires.
Currently available colours are:

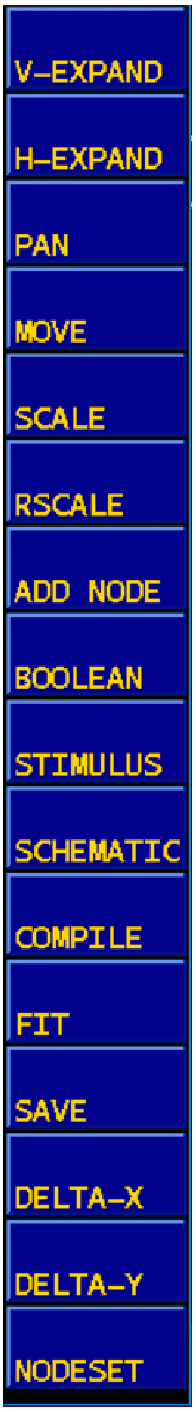


VIOLET
RED
GREEN
BLUE
YELLOW
ORANGE
BEIGE
CHARTREUSE
MAGENTA
SGREEN
CYAN
CORAL
PINK
PURPLE
SKYBLUE
WHITE

Main Menu

V-EXPAND

The command line prompts for two mouse selections. The portion to be expanded should be bracketed top and bottom by two mouse-button presses. The vertical separation of the two mouse entries will determine the segment which will vertically fill the screen, If expansion is required for a derived waveform, such as an FFT, waveform selection is unnecessary, since there is only one waveform in the screen buffer.



H-EXPAND

The command line prompts for two mouse selections. The portion to be expanded should be bracketed left and right by two mouse-button presses. The horizontal separation of the two mouse entries will determine the segment which will horizontally fill the screen, The algorithm first scans the array containing x-axis information until it finds two data points corresponding to the mouse positions.
It then scans y-axis data in opposite directions inward from these positions until it finds two data points corresponding to the selected waveform lying between the y-positions of the mouse; this should uniquely identify a feature on, say, one edge of a pulse, not present on the other edge

PAN

Useable on expanded waveforms only, will reposition the waveform in the x-direction, such that the mouse entry becomes the new screen centre. Three printouts at screen left, centre and right, give the x-axis starting value, scale in nanoseconds per division, and final value respectively, of the displayed waveform

MOVE

The first mouse entry selects the waveform to be moved in the y-direction and the second entry determines the position of the bottom of the waveform. Any points which would be outside the display area are suppressed

SCALE

Prompts for a common Y-scale factor. All waveforms are then displayed using this factor, and positioned such that their troughs are aligned with the bottom of the screen.
This function is inherited from Vspice, and is meaningless for logic waveforms.

RSCALE

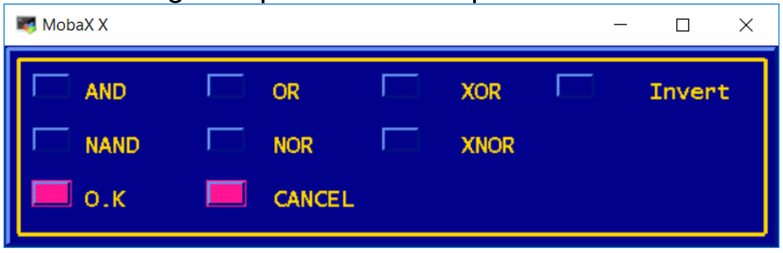
Rescales all waveforms currently in buffer, relative to the reference waveform selected by the mouse.
This function is inherited from Vspice, and is meaningless for logic waveforms.

ADD NODE

This function is only useful when no drawing exists, and the simulation was performed from a netlist, and the results read into Vlogic from a binary output file. If it is desired to plot a node not specified in the netlist's .PLOT/.PRINT command. the command prompts for Dsim node number of the required node It then extracts the appropriate data from the binary file, and adds another waveform to the display.

BOOLEAN

Performs logical operations on a pair of selected waveforms



STIMULUS

Under redevelopment

SCHEMATIC

Starts GEX to display the schematic

COMPILE

Re-invokes GTL, the netlist compiler. It is essential that GTL be located in the directory set by the environment variable $VSPICE_HOME – whether via a soft link or otherwise. It is also necessary that GTL should know the path to its libraries, so if these are located in other than $VSPICE_LIB, there should be a dsim.cmd file, with this information in it.

FIT

Performs an X and a Y fit of all waveforms displayed.

SAVE

Saves the waveform produced by any of the boolean operations, into the buffer containing the input waveforms. Unless this is done, the waveform will be lost as soon as another processing command is executed.

DELTA-X

The printout along the prompt bar displays the difference in value between the two points selected by the mouse. Selection of delta-x requires no waveform to be specified, since the horizontal axis is common. The prompt bar displays the time difference in nanoseconds between two points

DELTA-Y

Function inherited from Vspice, and irrelevant to logic waveforms.

NODESET

Under development

Known Bugs

- Setup and hold only checks hold, not setup in JK flipflops
- DMUX implementation is broken
- ADD NODE appears to work, but doesn't, if there's no xref:
  Node UN$1$MEMORY$9P$A0 not found: Enter by hand
  Adding node : UN$1$MEMORY$9P$A0 No:
  No node number found

- If GEX displays over a popup, the overlap is blank when it refreshes,