



# Using eightolives' WaveformViewer

eightolives WaveformViewer - Mozilla Firefox

eightolives Wavefor... x +

www.eightolives.com/docs/WaveformViewer/WaveformViewer.htm

File Edit Sim Pref Help

eightolives WaveformViewer

ESC I O H L Z X J I ff ↑

CLK U

RST\_N U

DBUS U

Time

100,000 ns 200,000 ns 300,000 ns 400,000 ns 500,000 ns 600,000 ns

Figure 1 Demo Timing

eightolives WaveformViewer

PRELIMINARY Version- In Development

ESC to delete the splash waveforms

Cursor 1 0 Cursor 2 0 Delta 0

Report:

```
Welcome to eightolives WaveformViewer
getService http://www.eightolives.com/docs/
PROJECT_LIB = LOCAL -1 null
wm addSimData CLK 1000000 0
wm addSimData CLK 1000000000 1
wm addSimData CLK 1500000000 0
```

File Edit Sim Pref Help

eightolives WaveformViewer

ESC I O H L Z X J I ff ↑

ff ↑

CLK U

RST\_N U

DBUS U

Time

100,000 ns

Figure 1 Demo Timing

eightolives WaveformViewer

PRELIMINARY Version- In Development

ESC to delete the splash waveforms

# WaveformViewer lets you:

- Draw timing diagrams
- Add timing constraints (delays, setup, hold)
- Import / export Value Change Dump (VCD) files
- Import eightolives Hardware Models from Schematic Mobile to:
  - Get signal names from a schematic design
  - Simulate digital designs that reference eightolives' simulatable models

# WaveformViewer

- Menu Bar
- ESCAPE button
- Set Value buttons
- Zoom In
- Zoo Out
- Pan Left
- Pan Right
- Restart
- Load VCD
- Add Waves to Sim
- Run
- Run For...
- Single Step
- Stop
- Cursor select buttons
- Cursor positions
- Report Window

The screenshot shows the WaveformViewer application running in a browser. The main window displays a timing diagram for three signals: CLK (green), RST\_N (blue), and DBUS (purple). The time axis is labeled 'Time' and has major ticks every 100,000 ns. The text 'Figure 1 Demo Timing' is centered on the waveform. A splash screen with the 'eightolives WaveformViewer' logo is overlaid on the waveform. Below the waveform, there are two cursor position fields: 'Cursor 1' and 'Cursor 2', both set to '0', and a 'Delta' field set to '0'. At the bottom, a 'Report' window shows the following text:

```

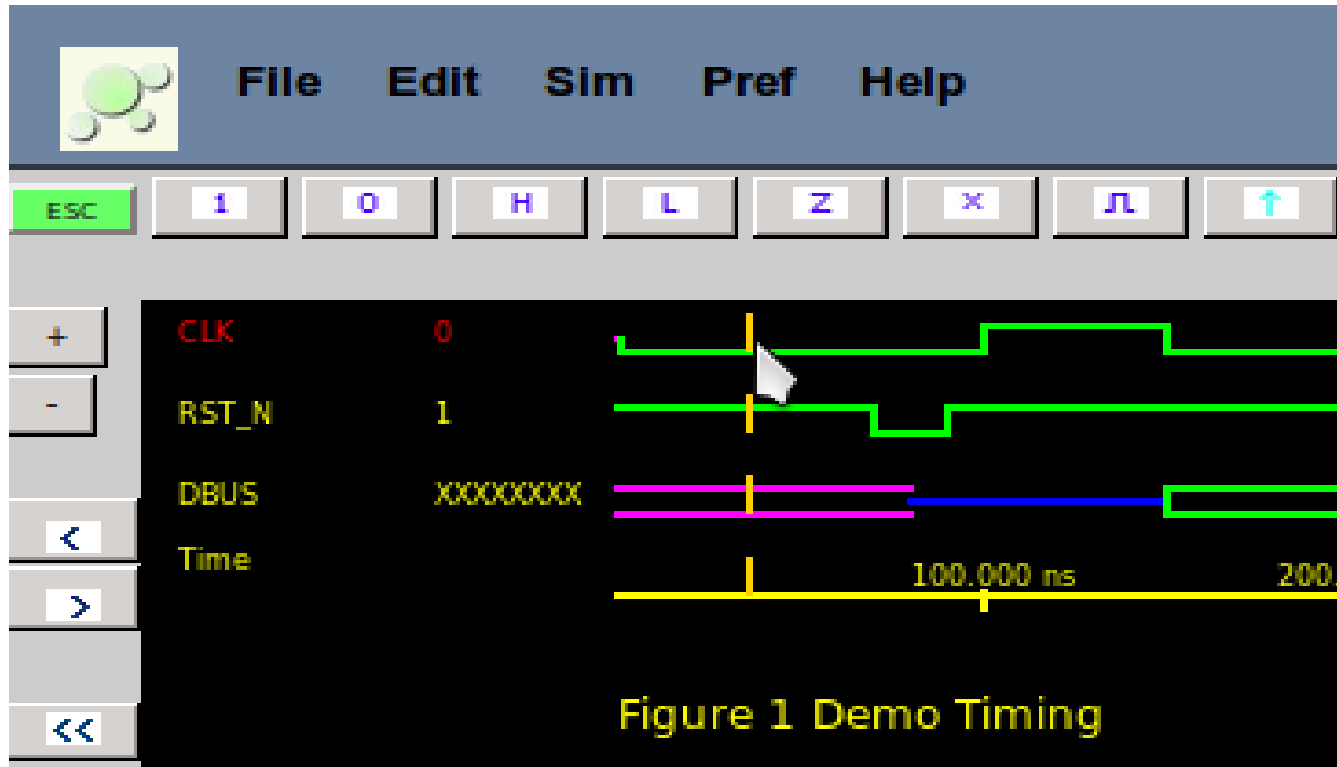
Welcome to eightolives WaveformViewer
getService http://www.eightolives.com/docs/
PROJECT_LIB = LOCAL -1 null
wm addSimData CLK 1000000 0
wm addSimData CLK 100000000 1
wm addSimData CLK 150000000 0
  
```

When you invoke WaveformViewer, the initial splash screen can be deleted if you click the ESC key first. To delete the waveforms later, use menu option **Edit > Delete All**

# Basic Definitions

- A **waveform** is a set of logic value / time pairs associated with a signal that can be graphically displayed.
- A **constraint** represents a timing specification such as delay, setup, hold, pulse width parameters.
- A **model** is coded description of a circuit component that can be examined and executed.
- A **simulation** is the computed behavior of models that attempt to describe the operation of a circuit.
- A **service** is defined as a provider and place of storage.
- A **project** is a defined directory structure that contains files such as Value Change Dump (.vcd file), timing constraints (.json file) or Hardware Model (.json file)

# Click on a waveform to select a signal



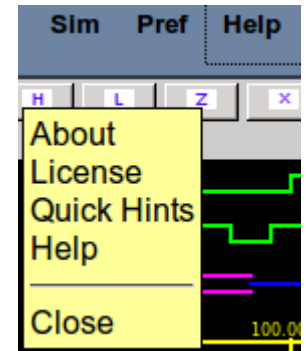
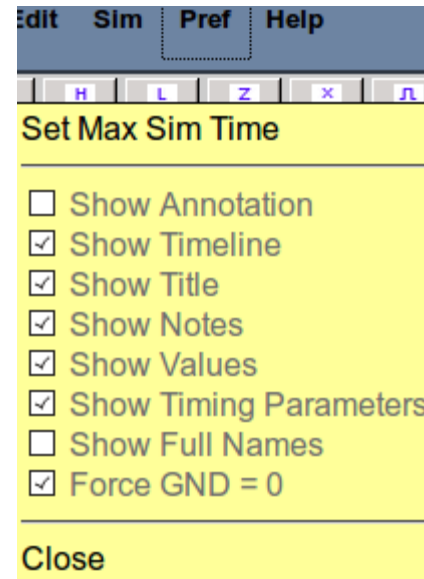
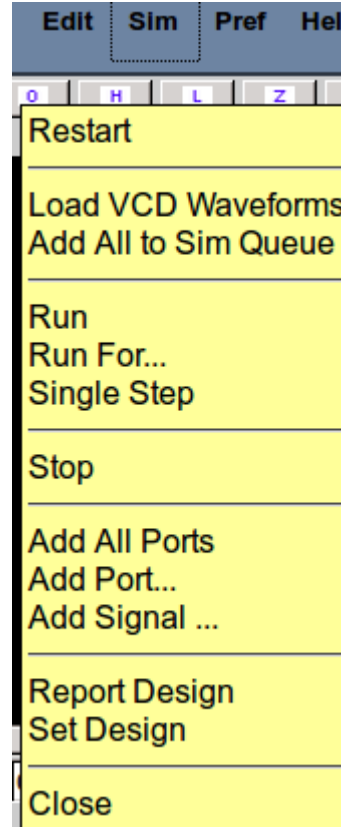
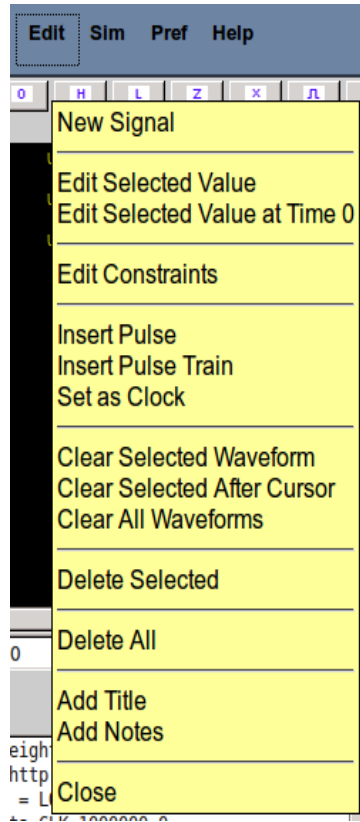
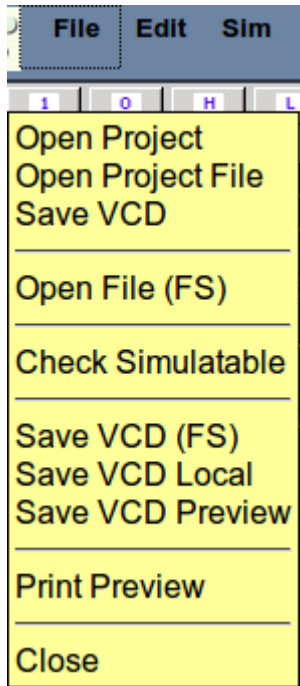
Clicking in the diagram area, positions the active cursor and selects the signal.

Clicking the “1”, “0”, “H”, “L”, “Z” or “X” buttons set the signal to that value at the cursor time.

Clicking the “pulse” button lets you insert a single pulse of a width you can specify.

The Yellow, Cursor 1, is used for most operations. Click the Yellow Cursor 1 button to use Cursor1 or Click the Red Cursor 2 button to use Cursor 2.

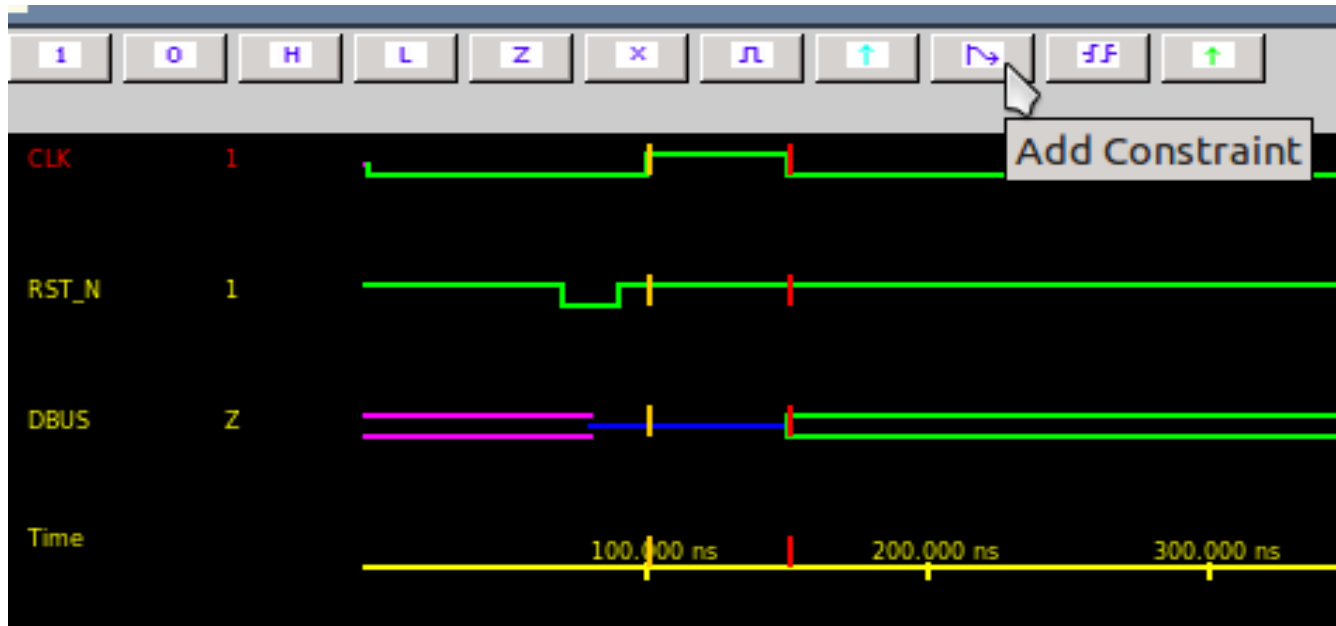
# Menu Bar Options



Use the Edit Menu to add a New Signal, Insert a Pulse Train, specify the signal as a CLOCK or edit the diagrams title or notes.

Use File Menu options to save the waveform as a VCD file. **Save VCD** saves to the current project. **SaveVCD (FS)** saves to your File System. **Save VCD Local** saves to your LOCAL project (cache memory).

# Adding Timing Constraints



- To add a Timing Constraint, first select menu item **Pref > Show Annotation** to space the waveforms.
- Place Cursor 1 on the first edge and Cursor 2 on the second edge.
- Then click the **Add Constraint** button

# Update Info in the Constraint Editor

**Constraint Editor**

Constraint Name:

Description:

Trigger Event:

Constrained Event

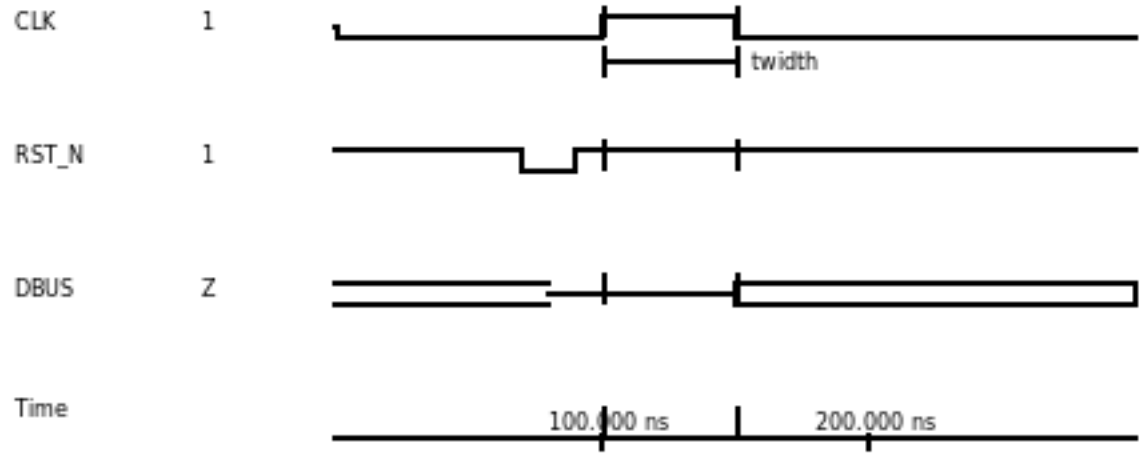
Occurs Sometimes  
 Occurs Always  
 Occurs Never

Min, Setup or Hold Time  
 Display Time  
 Max Time

- You may wish to change the constraint name
- Add a description
- Check correctness of the trigger and constrained events
- Add the appropriated timing specification



# Resulting Image

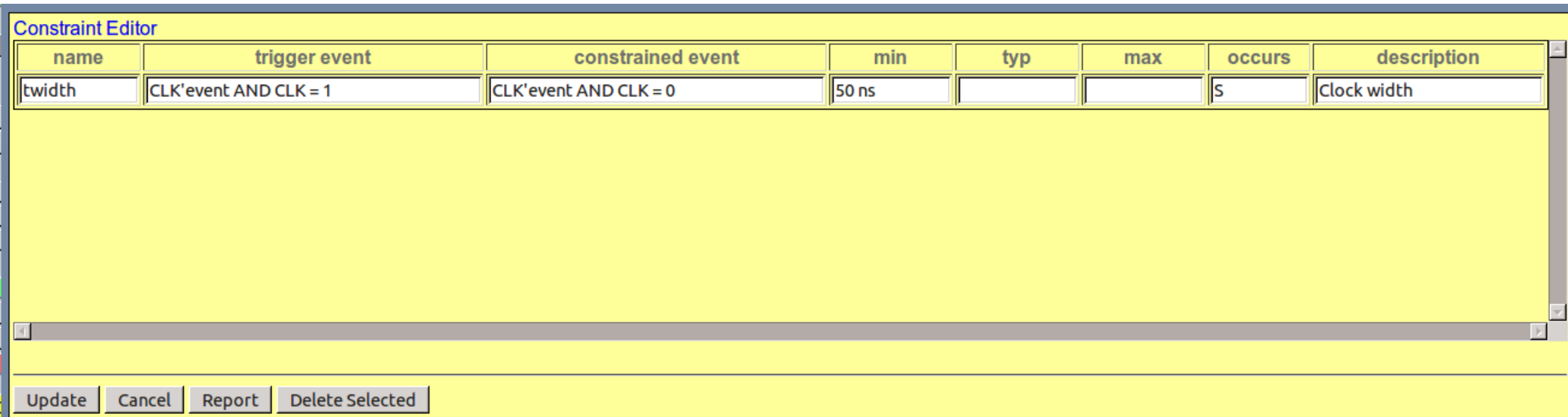


Use **Pref >Set Max Sim Time** to adjust the right edge of the image.

Figure 1 Demo Timing

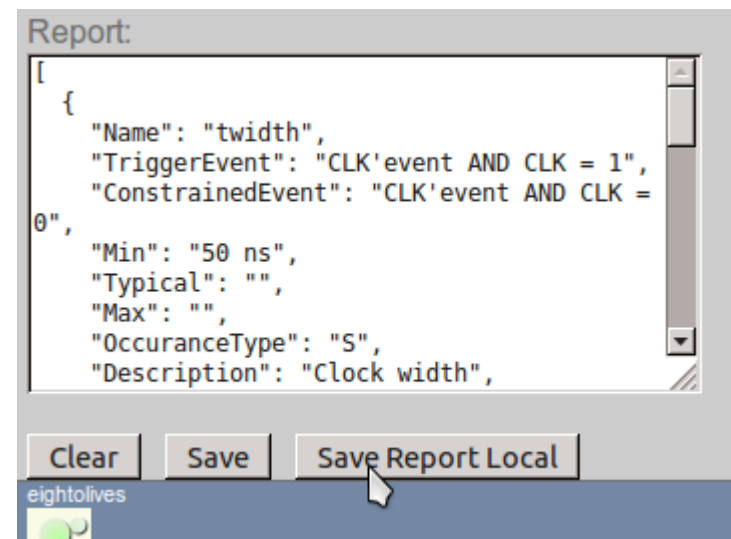
Timing Parameters				
Parameter	Description	Min	Typ	Max
twidht	Clock width	50 ns		

# Edit > Edit Constraints



To save constraints, click the Report button.

Under the Report window, click the Save or Save Report Local button to save the data. Be sure the file is saved with the .json suffix.



# Value Change Dump Files

- The VCD file format is specified in the Verilog specifications.
- It is used as a way to store waveform data from simulators.
- In this manner, you can draw your stimulus waveforms, save them and then load them for use in simulation.

# VCD File Format

```

$comment
  Value Change Dump Removed Signal null - version 1 - W. Kaupinis 10/9/05
$end
$comment
  Value Change Dump Filter - version 1 - W. Kaupinis 10/9/05
$end
$comment
  File created using the following command:
    vcd file TestVectors/temp.vcd -dumpports
$end
$date
  Mon Nov 13 12:40:27 2006
$end
$version
  dumpports ModelSim Version 5.8c
$end
$time scale
  1ps
$end
$scope module testbench_mim_module $end
$scope module top $end
$scope module u1 $end

$var port 1 <0 a0 $end
$var port 1 <1 a1 $end
$var port 1 <2 a2 $end
$var port 1 <3 a3 $end
$var port 1 <4 a4 $end
$var port 1 <92 wr_n $end
$upscope
$end

$upscope
$end

$upscope
$end

$enddefinitions
$end

```

Header information

Node Information

```

#0 Value Changes
$dumpports
pN 6 6 <0
pN 6 6 <1
pN 6 6 <2
pN 6 6 <3
pN 6 6 <4
pN 6 6 <5
pN 6 6 <6
pN 6 6 <84
pN 6 6 <90
pN 6 6 <92
$end

#50000
pU 0 6 <92
pf 0 0 <90
pf 0 0 <84
pf 0 0 <18
#1950000
pD 6 0 <28
pU 0 6 <27
$vcddclose

#1960000
$end

```

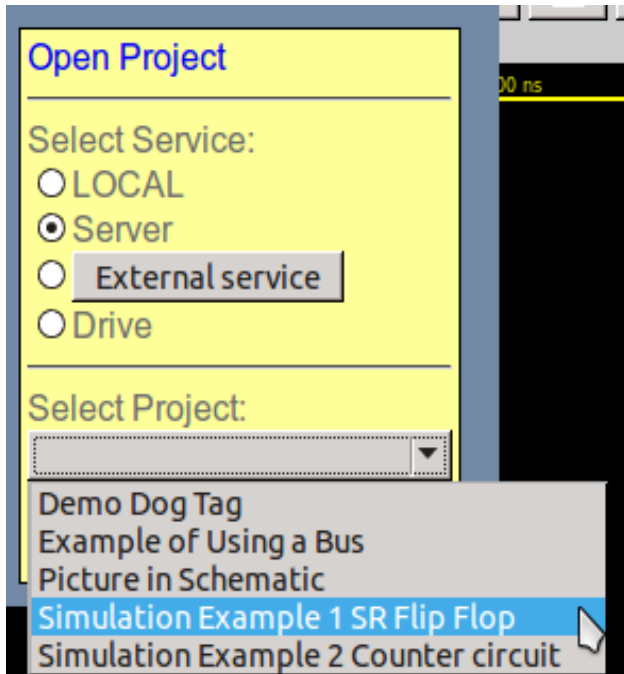
pValue	0_strength	1_strength	node_id
D, L	low		
U, H	high		
N, X	unknown		
Z, T	tri-state		
D, l	low (two or more drivers active)		
U, h	high (two or more drivers active)		
0,	low both input and output are active	0	
1,	high both input and output are	1	
?,	unknown		
F	three state		
A	unknown input 0 output 1		
a	unknown input 0 output X		
B	unknown input 1 output 0		
b	unknown input 1 output X		
C	unknown input X output 0		
c	unknown input X output 1		
f	unknown input and output three-state		

Strength	
0	highz
1	Small
2	Medium
3	Weak
4	Large
5	Pull
6	Strong
7	supply

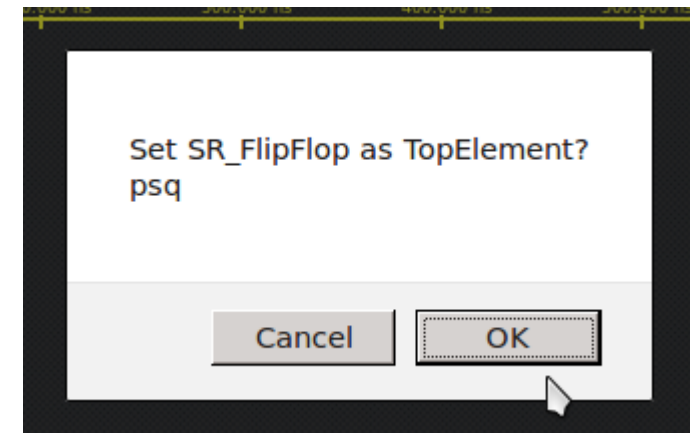
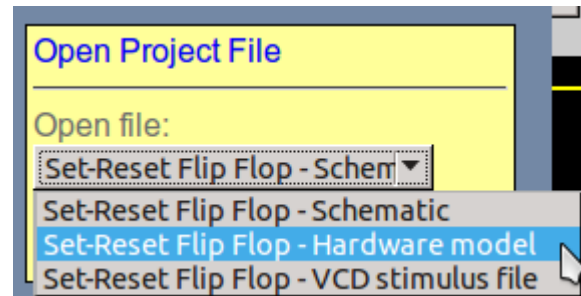
# Simulation

- WaveformViewer can perform “digital” simulation on circuits described with eightolives' Hardware models (written in Javascript).
- A schematic model can be directly created by Schematic Mobile. It calls models for each of it's components.
- Schematic symbols from the **eightolives Simulation Library** all have functional models that are automatically accessed on-line.
- Schematic Mobile can create “**template**” models of any other symbol used. Templates need to have 2 functions completed in order to implement functionality in the sim.
  - uninitialized() - called when a simulation is restarted
  - execute() - called when an input signal to the model changes.
- No model is the same as having no component in the circuit.

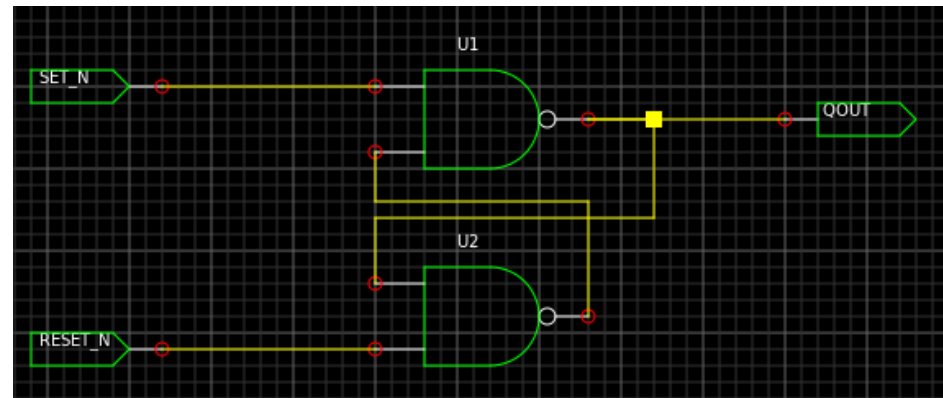
# To run a simulation example



File > Open Project File  
Select to load the Hardware Model  
Set the SR\_FlipFlop as the Top Element

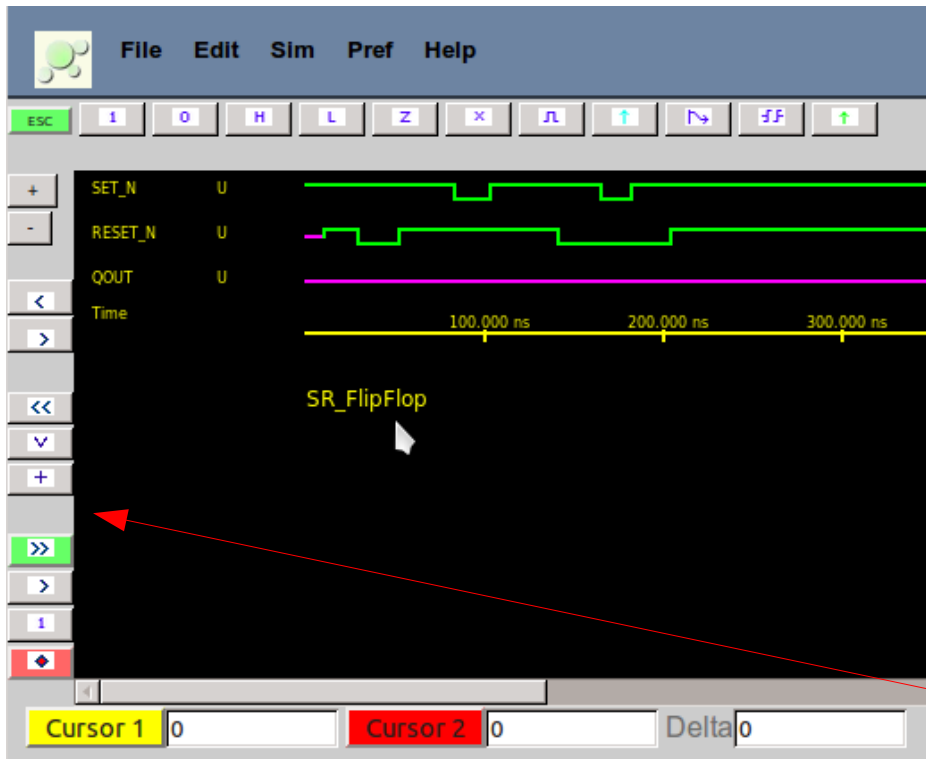


- File > Open Project
- Select **Server**
- Select **Example 1**



Schematic of the SR\_FlipFlop design

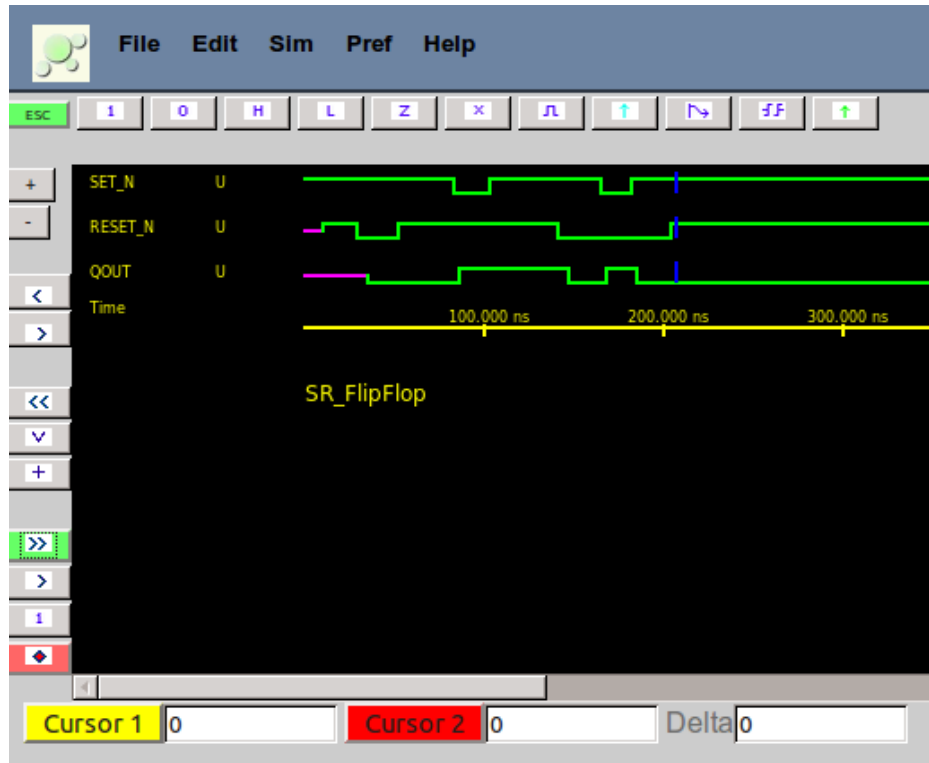
# Step 2 Load the VCD File



- When the model is loaded the name appears below the time line.
- To load the VCD file: File > Open Project File and select the VCD file.
- Click the “+” button to add the waveforms to the simulation queue.

You can modify the stimulus waveform before click the “+” button.

# Step 3 Run the Sim

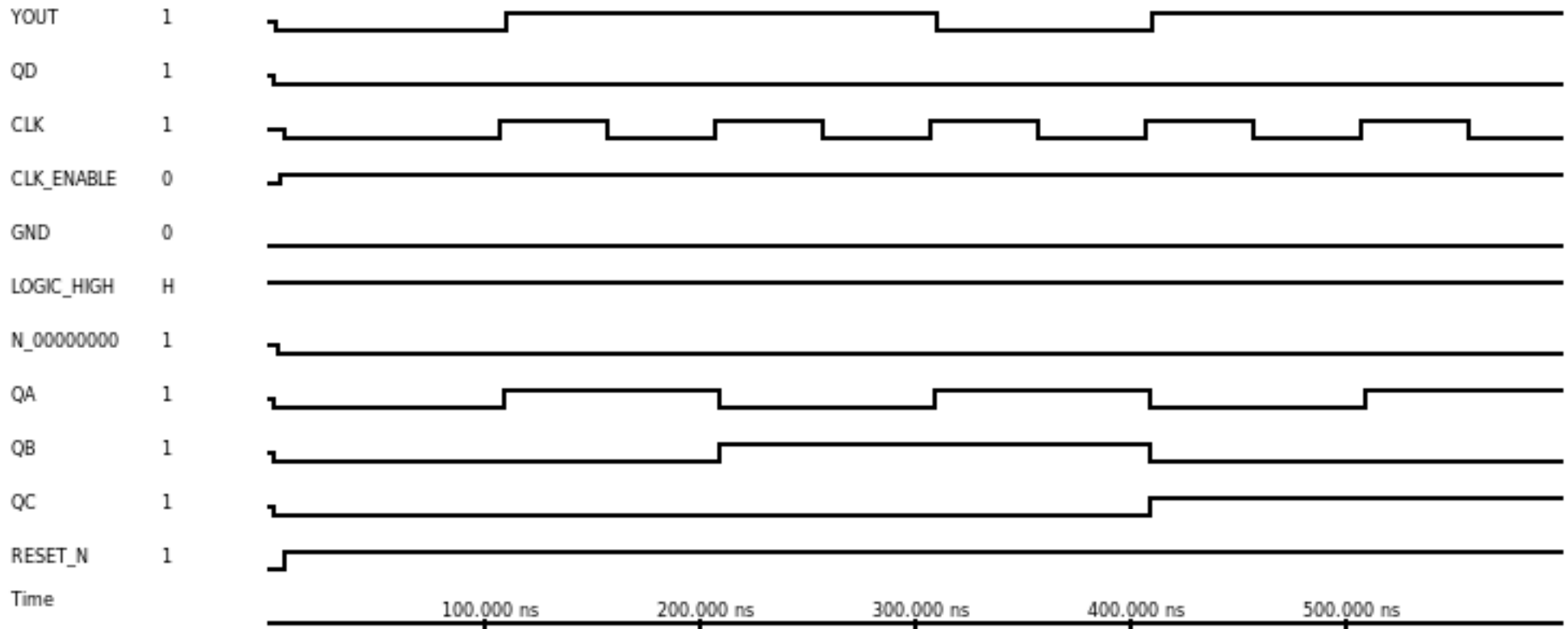


- Click the “>>” button to run the sim.
- The output QOUT is updated and a blue vertical cursor indicates the end of the simulation.
- To normal sim sequence is
  - Restart (“<<” button)
  - Load VCD (“v” button)
  - Add waveforms to sim (“+” button)
  - Run (“>>” button)

The Sim menu has options to select what ports and signals you want to add to the display.



# Try Other Examples



SimDemo 2

Sim example 2 is self-contained in that no external stimulus is needed. The VCD file is used only to add signals to be displayed.

# Making Simulatable Models

- See the separate tutorial on writing simulatable models.
- Best to start with a model template of the component as created from Schematic Mobile
- Schematic Mobile also lets you create a schematic of a component so you can include existing library models before creating your template

# Model of a 2 input NAND gate (1 / 2)

```
// nand2.js 4/14/15 1429016605943

// this is a template hardware model for nand2
// eightolives hardware model version 2

nand2.prototype = new Element();
nand2.prototype.constructor = nand2;

function nand2()
{
  this.name = "nand2";
  this.initialize();

  //define ports
  var Y = null;
  var A = null;
  var B = null;

  //define generics
  var device = this.addStringGeneric("device", "nand2");
  var refdes = this.addStringGeneric("refdes", "U?");
  var logicfunction = this.addStringGeneric("logicfunction", "NAND_2");
  var primitive = this.addStringGeneric("primitive", "true");
  var model = this.addStringGeneric("model", "simlib");
  var Td = this.addTimeGeneric("Td", "3 ns");
```

Generics are defined by the attributes on the symbol.

All the code on this page was created as part of the template created by Schematic Mobile.

# Model of a 2 input NAND gate (2 / 2)

```
this.define = function()
{
  //define ports
  clearArray(this.ports);
  Y = this.addNewStdLogicPort("Y", 1).signal;
  A = this.addNewStdLogicPort("A", 0).signal;
  B = this.addNewStdLogicPort("B", 0).signal;
}
this.define();

// set attributes
setAttributeValue("PRIMITIVE",this, "true");

//define architecture
var a = this.addNewArchitecture("simulatable");
a.initialize();

a.uninitialize = function()
{
  Y.set("U");
}

a.execute = function()
{
  var c = A.nand(B);
  Y.setWithDelay(c, Td.value);
}
}

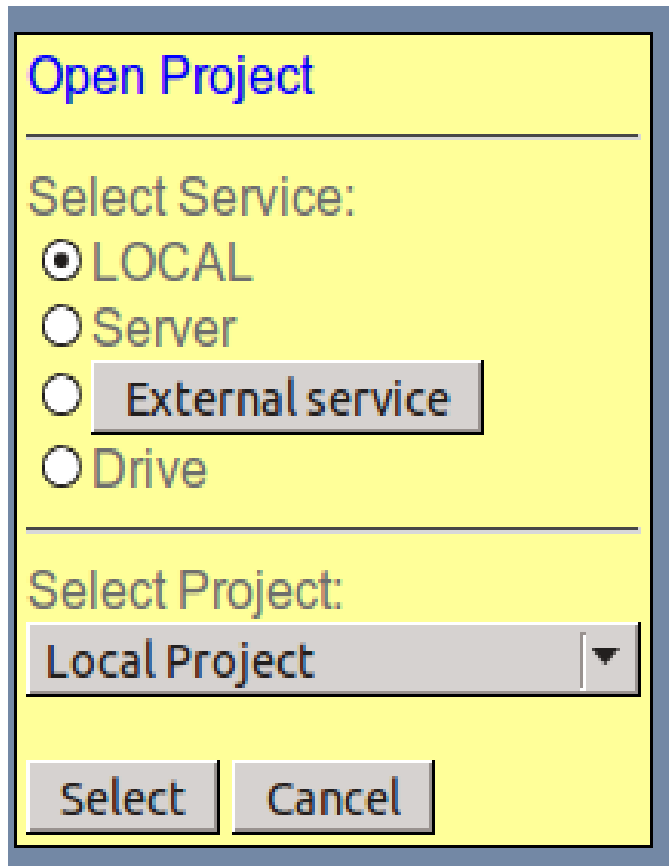
addModelDescription(new ModelDescription("nand2", nand2, []));
```

Only 2 lines of code needed to be added to the template to define the model's functionality in the execute function.

# Mobile Platform Constraints

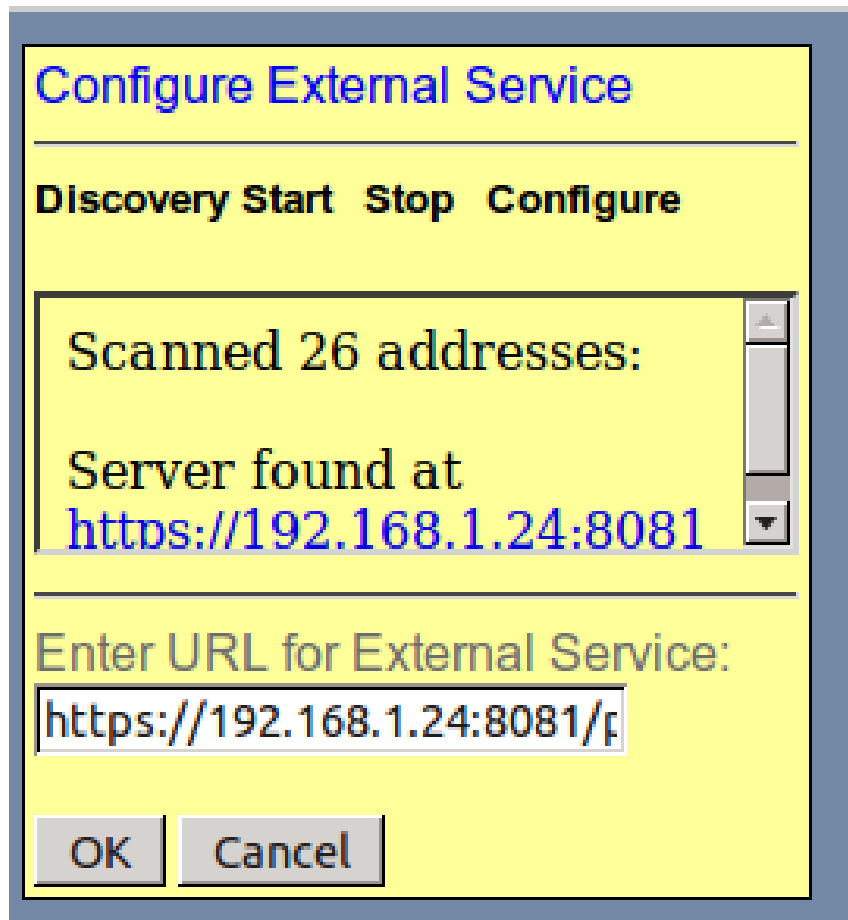
- Limited memory and file storage (not so reliable).
  - Use browser “cookie” memory for LOCAL storage
  - Use a companion Microserver to store files on a wifi-networked PC
  - Use Google Drive to store data for anywhere use
- Security concerns of mobile devices and web browsers
  - Companion Microserver uses secure https link, log-in with password
- No mouse
  - Use simple finger actions to easily create schematics
  - **Using a mouse is a lot easier**

# Services and Projects



- The default Service is LOCAL which lets you save files in browser cache memory.
- With Server selected, various example Projects may be available from the Schematic Mobile server.
- External Service lets you store files on another server such as a computer running Microserver.
- Drive lets you connect to your Google Drive account.

# Configuring External Service



- Currently External Service can be used with an eightolives Microserver operating on your local network.
- You can use the Discovery option to scan for available Microservers
- Clicking the found server enters the URL for External Service

# Using the File System (FS)

- Browsers running on Windows or Linux permit manual access to the platform's File System.
  - You can manually store and retrieve schematic sheets and symbols to the directory of your choice
- The iPod does not support a similar File System
- When reading a schematic sheet that requires a symbol that cannot be found or one that is stored in the FS, you will be prompted to locate the symbol which is then loaded to LOCAL project
  - Reloading the schematic will use symbols from LOCAL



# Getting Started

- WaveformViewer is web page which is a Javascript program that runs in your Firefox, Safari or Chrome browser. It uses “cookie” memory to store your LOCAL project
- You can run the program from a link on the eightolives Home Page or the eightolives QuickApps menu
  - Access to eightolives on-line simulation libraries
  - <http://www.eightolives.com/WaveformViewer/WaveformViewer.htm>
  - Improved file services require access to an external server
  - Google Drive provides an excellent way to access your files wherever you go
  - A companion eightolives Microserver running on a networked computer can provide access to other libraries and store your files on that computer

# FAQ

- What format is used for timing diagram files?  
VCD
- What format is used for timing constraints?  
.json format
- What format is used for hardware models? .js
- Can I run really big simulations? Probably not.  
The simulator is running non-optimized  
Javascript models which is a slow process.  
Memory availability is a limiting factor.

# For more information

- Visit <http://www.eightolives.com>