

# VOIDSIM

*By Void Yazılımevi*

## User Manual

**v2.0**

29/05/2006

METU

**Table of Contents**

Getting Started..... 2  
Using the Program..... 2  
Circuit Editing ..... 3  
Digital Simulation ..... 6  
Adding Custom Components ..... 6  
Scripting ..... 8

## Getting Started

VoidSim is a digital circuit editing and simulation system, which could be used for editing and verifying the validity of the designed circuits with satisfying graphical user interface features.

The rest of the document will explain how to use this tool with maximum efficiency.

To report bugs, send e-mail to:

*voidyazilimevi@gmail.com*

## Using the Program

The VoidSim program is mainly controlled with the mouse, and also from the keyboard.

There are three basic mouse motions you need to learn:

**CLICK:** Press and release the left button on a menu item/tool bar item/left menu tree items. The required action is specified by clicking on it.

**PRESS/DRAG.** Press down and hold the left button, move the cursor somewhere else, and release. This motion is especially used in select mode, when you press on a gate and drag , it is moved to the point where you release the button.

**RIGHT CLICK** Press the right button and release. This is used to cancel operations like drawing wires. In addition it is used to reach the pop-up menu in the working area.

Some of the functionalities used frequently such as cut/copy/paste or start/stop/pause simulation are reachable through the toolbar. Other functions are in Menu Bar.

There are also keyboard commands, though everything you really need as a novice can be found in on-screen menus. There are several features of the keyboard to notice:

DEL key is used to delete selected components.

<CTRL>-xxx. Press the Control key and the specified letter in order to take specified action.

CTRL+A	Select All
CTRL+C	Copy
CTRL+V	Paste
CTRL+X	Cut
CTRL+Y	Redo
CTRL+Z	Undo
CTRL+O	Open Project
CTRL+W	Close
CTRL+SHIFT+W	Close All
CTRL+S	Save
CTRL+SHIFT+S	Save All
CTRL+P	Pdf Convert
CTRL+N	New Project
CTRL+I	New Circuit
CTRL+R	New Script File
CTRL+F11	Run Simulation
CTRL+E	Execute Script

## Circuit Editing

After running the VoidSim program, a user can start his/her design by either

- Creating a new project

User can select New Project from the File menu, and by specifying a name to it, a new project is created at the working directory.

Moreover, user can create a new project by clicking the folder icon at the toolbar.

Or,

- Opening an existing project.

User can select Open from the File menu and open an existing project. By default a Circuit Design File will be opened. User can add new circuit design files to the existing project. From the File ->New menu or from the toolbar, clicking on the circuit option, a new circuit is added to the project.

- Add Component

To begin drawing a circuit, choose the circuit symbol you want, click on it from the left tree views and then click on the point in the working area, where you desired to put the gate.

There are two tree views at the left menu for selecting components. In first one, the one below project explorer tree view, the gates such as and/or/not are placed. The other tree view, namely the JHDL components tree view, has the components reflected from the JHDL classes.

- Draw Wire

In order to draw a wire, user should select Wire from the component tree view. Then clicking on the working area specifies the starting point of the wire. The wire follows the motion of the mouse, drawing two-dashed-lines. When user clicks, the first part of the wire is drawn, if he/she wants too draw whole wire, clicks one more time. This automatically starts a new wire at the endpoint, so you can make a chain of connected wires. To finish the chain, just tap the right mouse button. The whole sequence of tap-tap-rest takes a few minutes to learn but soon feels very natural.

The aim of this two step wire drawing is to manage the wires more easily, since cross wiring is not permitted in VoidSim, wires are drawn with two parts, one is horizontal and other is vertical.

- Move Component

Components can be moved inside the working area easily by dragging and dropping the component at Select Mode. User can switch to Select Mode by clicking on the hand icon on the toolbar.

- Cut/Copy/Paste

These operations can be done by three ways:

1. From the left popup menu, when you are on a component
2. From the toolbar icons
3. From the menu bar:   Edit->Copy  
                              Edit->Cut  
                              Edit->Paste

- Move Wire

In order to move a wire, one should select it. When user selects a wire, at the editable nodes, (the midpoints of the wire parts) a red circle is appeared. The wires can be moved, left/right or up/down by clicking on the editable node, and moving the mouse while pressing it. When you release the button, the wire segment will be crossing through the point you specified.

- Connect Wires to Components

As user can see by drawing any component, all the gates has some part of wire, in their input and output ports. When user draws a wire near a gate or any other wire, the start/end point is snapped to that component or wire.

- Delete component/components

After selecting the component/components to be deleted, one can right click and select Delete or simply push the Del button from the keyboard.

- Rotate Component

By right clicking on a component one can reach the properties of each component and modify them. A popup menu is appeared by right clicking on a component at select mode. From this menu, a component can be rotated 90/180 or 270 degrees.

- Undo/redo

User can undo or redo his/her actions by clicking the icons at the toolbar/selecting undo/redo from Edit->Undo or Edit->Redo from the Menu Bar.

## Digital Simulation

VoidSim supports on the fly simulation, which means, while you are editing your circuit, it is being simulated. By default the simulation is on. You can stop or pause simulation from the icons at the toolbar, or from the Menu Bar. Simulation->Run

Simulation->Pause

Simulation->Stop

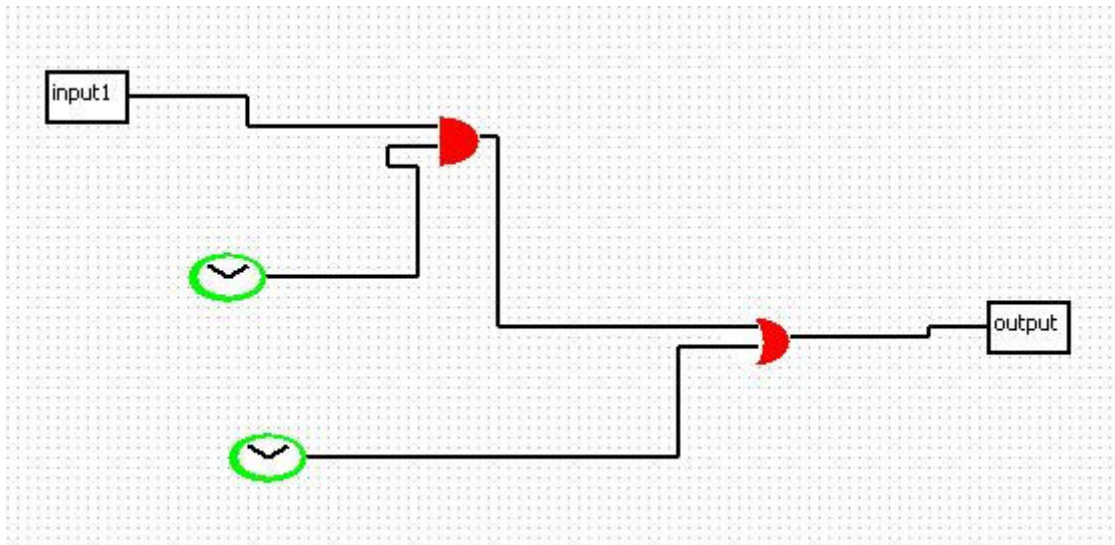
The black wires indicate the value 0, whereas the red wires show that the value of wire is 1. The ground component supplies the input 0 to the circuit, and the Vcc component supplies input value 1. Moreover the clock component supplies 1 and 0 interchangeably according to the frequency set.

## Adding Custom Components

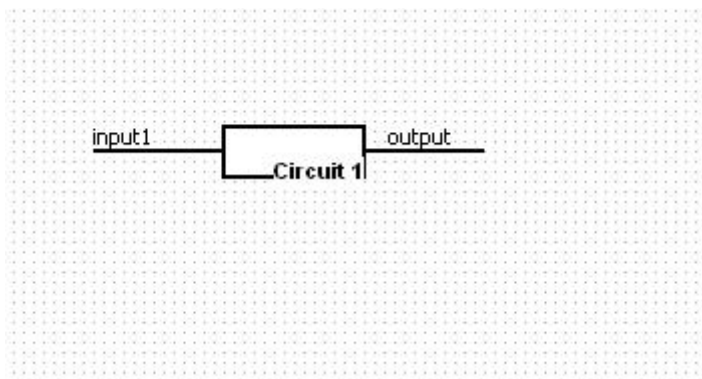
In addition to the components from JHDL library user is free to design his/her own circuit, then using this circuit as a component in a different circuit.

If you want to have your own component, add a new circuit in your project and design your own component. You should put input and output components, which will be used as the input and output ports of your own component. Then select Project->Add Custom Component from Menu Bar.

For example, if you create a circuit as follows you should rename input and outputs by right clicking on the component, then from the popup menu, select Edit Name, and specify a name. Select the Add Custom Component from the menu bar to set your custom component..



Then you will see at the Component Tree view a new component named Circuit\*. When you select this component from the tree view and add to your design, you will see a component as follows, with the input and output names as you specified previously.





## Scripting

Scripting support in VoidSim is an advanced concept, which was designed primarily to help the professionals, run, test and debug the projects from within java code. Also we expect that it will also be useful for the teaching community by means of testing the student's assignments. The rest of the section assumes that you are familiar with java. However knowing some iterative language such as C or C++ is sufficient to use the scripting facility. For more information, please see <http://java.sun.org>.

Just like adding a new circuit, the user can add a script file to the current project from the toolbar or from the file->new menu. Then a new text file is added to the project and can be viewed in the main window. The user can add as many script files as he wants.

Let's look at a new Script file as it is just created. :

```
package design.batch.user;

//import decerations come below
import design.batch.library.*;
import design.batch.Executable;
import customLibrary.*;

public class Script1 implements Executable {

    private final String name = "Sample Project1";
    private final String path = "C:\\Documents and
Settings\\Ben\\Desktop";
    /**
     * Write your statments inside this fn. The system will call this
     * function when execute script command is
     * received.
     */
    public void execute() {

        Project p = new Project(name, path);

    }

}
```

As can be realized, the script file is indeed a java file, which is to be executed by VoidSim. The essence in using java for the script is that, we expect the intended user to be familiar with the language so that the burden of learning a new scripting language is rendered out, and the user can benefit all the features of the language without any restrictions. Having explained the essentials to use the java language, let's look at the particular usage of it for our needs.

When a new script file is created, a new class file is created which implements the Executable interface which has only the public void execute() method. The statements to be performed should be written inside this function. It is an entry point to the code. Except from that the user is free to edit the file to suit his needs.

The user can use the classes inside the package design.batch.library. And also he can benefit to use his own classes which are in the customLibrary package whose path is also supplied by the user. The name and path fields give the name and the path of the project that the file currently belongs to. Now let us consider an example to understand the usage.

```
public void execute() {  
  
    Project p = new Project(name, path);  
    for(int c= 0;c < 10;c++) {  
        p.cycle();  
        Vector<Output> v = p.getOutputs();  
        for(int i=0;i < v.size();i++)  
            System.out.print(v.get(i).getName() + " : " )  
            System.out.print(v.get(i).getValue()+ " \n");  
        }  
  
    }  
}
```

In this example a new Project object is created by supplying the name and the path of it(any project other than the current can be supplied.) Then for 10 times the project is given a cycle, and at each cycle the values of the output wires are displayed. Although simple, the example clearly demonstrates the power of using scripting in the program. For further details please refer to the API for the design.batch.library package in the documentations.

After the script file is finished, the user can press the execute script button in the toolbar, or alternatively use Ctrl + E to run the execute function in the class.