LogSim®

A Logic Simulation Language

Xunzhou Zhang (xz2025), David Lau (dsl2012), Mukul Khajanchi (mk2603), Nithya S Ganesan (nsg2104)
PLT Fall 2005

1.1 Background

Digital electronics is an indispensable part of our lives today. Televisions, computers, and cars all depend on digital electronics. In the field of digital design, engineers often encounter the issue of design and testing. However, building each circuit physically and testing it can be very costly. Hence, they need tools that enable them to describe their design abstractly. This gives rise to the need for Hardware Description Languages (HDL), that enable the programmers to describe a digital circuit efficiently, verify that it's correct, and perform virtual simulations on the design.

Currently, there are two major HDL's in the market, VHDL and Verilog. VHDL, which was created by the United States Department of Defense in 1981, was the first HDL created, and is still widely used today. Verilog, created 2 years later by Gateway Design Automation Inc., as a direct competition to VHDL, is another popular HDL.

Both VHDL and Verilog are hugely complex languages that require a very steep learning curve. Though they are powerful, programmers often fail to exploit all their features due to their complex syntaxes. Our goal is to build a simple and efficient language that can be used to simulate logic circuits, and yet maintains most of VHDL and Verilog's functionality.

1.2 Features

1.2.1 Portability

One advantage of LogSim is portability. We will use ANTLR as the syntax recognizer, which runs on a JVM. Since we are using ANTLR, LogSim compiler will generate Java code that can be compiled by Java compiler in JDK and run on the JVM.

1.2.2 Ease of Use

LogSim will have a simple and intuitive syntax that is quick to learn and easy to decipher. Programmers will be able take advantage of all the features of LogSim without being overwhelmed by syntax issues. Electrical engineers without too much programming background can easily use it for simulating their designs.

1.2.3 Efficiency

Since this is a small, domain-specific language, it will have a small set of keywords, identifiers and operators. This ensures that implementation can be made as efficient as possible, and program compilation will be quick and with minimal system resources utilization.

1.3 Language Description

A LogSim program consists of operators and components. Operators include basic gates such as AND, OR, NOR, XOR, etc. Components include MUXes, Encoders, Latches, Flipflops, etc, that the user can make use of in the code. In addition to the built-in components, users can define their own components using LogSim syntax. Operators and components together make up a System. A System is the final circuit that the user specifies and all simulations are run on the System.

1.3.1. Sample Program

Consider the implementation of a one-bit Full Adder where the Sum and Carry are calculated as:

Sum = (A xor B) xor Cin

CarryOut = (A and B) or (Cin and (A xor B))

This can be implemented using LogSim as a component. Using this component, a two bit Full Adder can be built easily and the System can be simulated.

Sample code using LogSim:

```
//signal keyword defines a one-bit variable
Component FullAdder(In: Signal A,B,Cin; Out: Signal Sum, Cout)
{
      Sum = (A # B) # Cin;              // # means xor
      Cout = (A * B) + (Cin * (A # B)); // *, + mean and, or
}


Component TwoBitAdder(In: signal A0,A1,B0,B1;
                      Out: signal S0,S1,Cout)
{
      Signal C_temp;
      fullAdder(In: A0, B0, 0; Out: S0, C_temp);
      fullAdder(In: A1, B1, C_temp; Out: S1, Cout);

}

//System keyword denotes the main program, where the simulation occurs
System AdderSim()
{
      Signal A0, A1, B0, B1, S0, S1, Cout;
      A0=0;
      A1=1;
      B0=1;
      B1=0;
      //Display function displays the results of the component, along
      //with inputs
      Display(TwoBitAdder(In: A0,A1,B0,B1; Out: S0,S1,Cout));
}
```

The output of this system would be a Graphical User Interface, which would depict the Inputs and Outputs.

1.4 Summary

LogSim enables easy programming and simulation of logic circuits, and is a lightweight language that can be readily deployed for use.