LogSim®

Language Reference Manual

October 17, 2005

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

1.1 Lexical Conventions

1.1.1 Comments

1.1.1.1 Multi-line comments

The /* characters introduce a comment; the */ characters terminate a comment. Once a /* is seen, all other characters are ignored until the ending */ is encountered.

1.1.1.2 Single-line comments

All the text after // until the end of line are ignored

1.1.2 Identifiers

An *identifier* is an unlimited-length sequence of letters, digits and underscores, the first of which must be a letter. An identifier cannot be the same as a keyword.

`Abcd_123;` // `Abcd_123` is a valid identifier
`12_abc` // 12_abc is an invalid because identifiers cannot start with a number
`component;` // component is an invalid identifier because it is a keyword

1.1.3 Keywords

The following character sequences are reserved for use as *keywords* and cannot be used as identifiers:

```
in out import
```

1.1.4 Constants

1.1.4.1 Constants

1.1.4.2.1 Decimal Constants

Of the format 0d12, which means a Vector 1100

1.1.4.2.2 Binary Constants

Of the format 1100

1.1.5 Operators

An operator specifies an operation to be performed. The operators ( ) and [ ] must occur in pairs, possibly separated by expressions.
Operator can be one of the following:

( ) [ ]
+ * # !
= := $

1.1.6 Punctuators

A punctuator is a symbol that has semantic significance but does not specify an operation to be performed. The punctuators ( ), [ ], and { } must occur in pairs, possibly separated by expressions, declarations, or

statements. Following punctuators are used in LogSim
: ; , () {} [ ]

1.2 Expressions and Operators

1.2.1 Precedence and Associativity

( ) []	Component Instantiation, Vector indexing	Post-fix	L-R
… $	Vector slicing, bit-allocation	Unary	L-R
!	NOT	Pre-fix Unary	L-R
*	And	Binary	L-R
# +	Xor, OR	Binary	L-R
:= =	Sequential Assignment, Combinational Assignment	Binary	R-L

1.2.2 Parenthesized Expressions

For user-defined precedence such as `!(A*B)`

1.3 Types

1.3.1 Vectors

Vectors are the only data type in LogSim. A single bit signal is simply a one member vector. Vector members can be accessed using an index within brackets. For example A[2] refers to the second element in the vector A. To declare a vector, simply write the vector name followed by an equal sign and the value of the vector. For example, A=101 defines a vector of the binary value '101'. By default, if a vector is referred to without an index, it points to the first member of the vector.

1.3.1.1 Vector Slicing

Any continuous subset of a vector can be accessed using the following syntax:
`A[n...n+k]` returns all the bits from the nth element to the (n+k)th element in A.

1.4 Components

Components are the basic building blocks in LogSim. A Component with the name System is the one on which simluations will be run.

1.4.1 Declaring a Component

```
Component Component_Name (In: I0,I1,I2...;
                          Out: O1,O1,O2...;)
```

I0,I1,I2,.... are inputs to the component, and *O1,O2,O3...* are outputs to the component.

1.5 Library

A library is a logsim source file that does not contain a System component. It is used in conjunction with the `import` keyword to specify that all components inside the library file can be directly accessed in the main program. The library file name should end with an "lib" extension. For example:

```
import MyComponent.lib
```

This statement specifies that all components defined in the MyComponent.lib file are accessible in the main program.

1.6 Statements

1.6.1 Sequential Assignment

To assign values sequentially, use the ":=" operator.

For example:
```
A$8:=B;
B$8:=C;
```
will assign B to A in clock cycle 1, and C to B in clock cycle 2, so that A in does not immediately take the value of C, as it would in combinational assignments. The “$8” after the variable name allocates 8 bits for the variable.

Note: If the result has more than 8 bits, it’ll be right-truncated. If the result has fewer than 8 bits, the most significant bits will be filled with 0’s.

1.6.2 Combinational Assignment

To assign values combinationally, use the "=" operator.

For example: `A=11`will assign the binary value '11' to the vector A.

Note: all combinationally assignments in LogSim are concurrent.

For example:
```
A$8=B;
B$8=C;
```
will assign the value of C to A, as all assignments are evaluated concurrently.

1.6.3 Component Instantiation

A component is instantiated from within another component. Upon each instantiation, a new copy of the component is created.

1.7 Output

Upon running the program, the program will print to the screen the values of all inputs and outputs to the System Component for one clock cycle. If the user wishes to see values for more than one clock cycle, he may enter the name of the program followed by two integers that specify the starting and ending clock cycles that the user wants to see.

For example:

```
MyProgram.log 0 100 //Prints inputs and outputs in the System
                    Component from the 0th clock cycle to
                    the 100th clock cycle
```

1.8 Namespaces

LogSim has two namespaces, one for Vectors and one for Components. A Vector can have the same name as a Component.

1.9 Scopes

All components are accessible from everywhere within the same file. All vectors are accessible only from within the component that they are defined in. Within a component, only vectors declared inside and passed through are accessible. All vectors are passed by reference.