Chapter 2

SystemVerilog Literal Values and Built-in Data Types

SystemVerilog extends Verilog’s built-in data types and enhances how literal values can be specified. This chapter explains these enhancements and offers recommendations on proper usage. A number of small examples illustrate these enhancements in context. Subsequent chapters contain other examples that utilize SystemVerilog’s enhanced data types and literal values. The next chapter covers another important enhancement to data types, user-defined types.

The enhancements presented in this chapter include:

- Enhanced literal values
- ‘define text substitution enhancements
- External compilation-unit scope declarations
- Time values
- New data types
- Signed and unsigned types
- Variable initialization
- Static and automatic variables
- Casting
- Constants
2.1 Enhanced literal value assignments

In the Verilog language, a vector can be easily filled with all zeros, all Xs (unknown), or all Zs (high-impedance).

```
reg [127:0] data;

data = 0;    // fills data with 128 bits of zero
data = 'bz;  // fills data with 128 bits of Z
data = 'bx;  // fills data with 128 bits of X
```

However, Verilog does not provide a convenient mechanism to fill a vector with all ones without using a literal value with all bits set to one, or using operators such as the replicate operator, a ones complement operator, or a twos complement operator. The following examples illustrate these styles, showing ways to assign a 128 bit vector to all ones:

```
data_bus=128'hFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF;
data_bus = {128{1'b1}};  // replicate operation
data_bus = ~0;           // ones complement operation
data_bus = ~1;          // twos complement operation
```

SystemVerilog enhances assignments of a literal value in two ways. First, a simpler syntax is added, that allows specifying the fill value without having to specify a radix of binary, octal or hexadecimal. Secondly, the fill value can also be a logic 1. The syntax is to specify the value with which to fill each bit, preceded by an apostrophe ( ' ), which is sometimes referred to as a “tick”. Thus:

- '0 fills all bits on the left-hand side with 0
- '1 fills all bits on the left-hand side with 1
- 'z or 'Z fills all bits on the left-hand side with z
- 'x or 'X fills all bits on the left-hand side with x

Note that the apostrophe character ( ' ) is not the same as the grave accent ( ` ), which is sometimes referred to as a “back tick”.

Using SystemVerilog, a vector of any width can be filled with all ones without hard coding the width of the value to be assigned, or using operations.