Section 8
Mixed-signal

8.1 Fundamentals

The most important feature of Verilog-AMS HDL is it puts capabilities of both analog and digital modeling into a single language. This section describes how the continuous (analog) and discrete (digital) domains interact together, as well as the mixed-signal specific features of the language.

8.1.1 Domains

The domain of a value refers to characteristics of the computational method used to calculate it. In Verilog-AMS HDL, a variable is calculated either in the *continuous* (analog) *domain* or the *discrete* (digital) *domain* every time. The potentials and flows described in natures are calculated in the continuous domain, while register contents and the states of gate primitives are calculated in the discrete domain. The values of real and integer variables can be calculated in either the continuous or discrete domain depending on how their values are assigned.

Values calculated in the discrete domain change value instantaneously and only at integer multiples of a minimum resolvable time. For this reason, the derivative with respect to time of a digital value is always zero (`0`). Values calculated in the continuous domain, on the other hand, are continuously varying.

8.1.2 Contexts

Statements in a Verilog-AMS HDL module description can appear in the body of an analog block, in the body of an **initial** or **always** block, or outside of any block (in the body of the module itself). Those statements which appear in the body of an analog block are said to be in the *continuous* (analog) *context*; all others are said to be in the *discrete* (digital) *context*. A given variable can be assigned values only in one context or the other, but not in both. The domain of a variable is that of the context from which its value is assigned.

8.1.3 Nets, nodes, ports, signals, drivers and receivers

In Verilog-AMS HDL, hierarchical structures are created when higher-level modules create instances of lower level modules and communicate with them through input, output, and bidirectional ports. A *port* represents the physical connection between an expression in the instantiating or parent module and an expression in the instantiated or child module. The expressions involved are referred to as *nets*, although they can include registers, variables, and nets of both continuous and discrete disciplines. A port of an instantiated module has two nets, the upper connection (`vpiHiConn`) which is a net in the instantiating module and the lower connection (`vpiLoConn`) which is a net in the instantiated module, as shown in Figure 8-1. The `vpiLoConn` and `vpiHiConn` connections to a port are frequently referred to as the *formal* and *actual connection*s respectively.

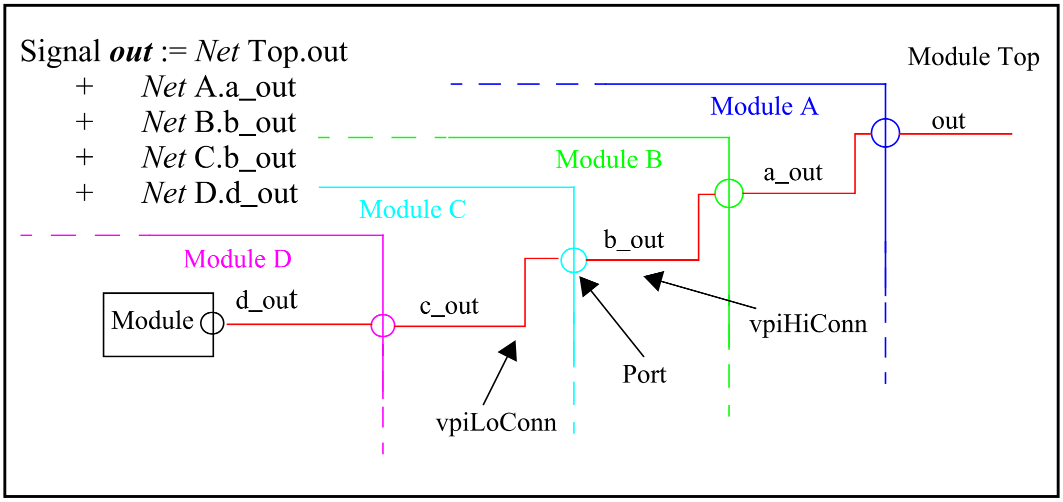

Figure 8-1 Signal "out" hierarchy of net segments

A net can be declared with either a neutral, discrete or analog *discipline*, only digital blocks can drive a discrete net (*drivers*) and only analog blocks can contribute to a analog net (*contributions*). A *signal* is a hierarchical collection of nets which, because of port connections, are contiguous. If all the nets which make up a signal are in the discrete domain, the signal is a *digital signal*. If all the nets which make up a signal are in the continuous domain, the signal is an *analog signal*. A signal which consists of nets from both domains is called a *mixed-signal*.

Similarly, a port whose connections are both analog is an *analog port*, a port whose connections are both digital is a *digital port*, and a port whose connections are analog and digital is a *mixed port*.

~~If a signal is analog or mixed, then it is associated with a node (see 3.4), while a purely digital signal is not associated with a node. Regardless of the number of analog nets in an analog or mixed-signal or how the analog nets in a mixed-signal are interspersed with digital nets, the analog portion of an analog or mixed-signal is represented by a single node. This guarantees a mixed or analog signal has only one value which represents its potential with respect to the global reference voltage (*ground*).~~

Since it is physically one wire in the design, Kirchoff's current law applies to the whole signal, and it forms one node in analog simulation. Drivers in the digital domain are converted to contributions in analog domain using auto-inserted digital-to-analog connection modules, and the signal value is calculated in the analog domain. Resolution of the digital drivers is not performed, instead the digital value of the signal is taken as the value of the analog result converted back by one or more auto-inserted analog-to-digital connection modules. A digital behavioral block that reads the value of a signal is a *receiver*.

The drivers and receivers of a net are associated with the ports of their containing module and derive their discipline from the port declaration; the discipline of the port is used to determine which connection modules to use.The discipline of the signal is found by *disipline resolution* as described in section 8.4, and is used to determine the attributes of the node in simulation.

Nets without drivers, receivers or contributions are treated as neutral interconnect.

8.1.4 Mixed-signal and net disciplines

One job of the discipline of a continuous net is to specify the tolerence (*abstol*) for the potential of the associated node. A mixed-signal can have a number of continuous nets, with different continuous disciplines and different abstols. In this case, the `abstol` of the associated node shall be the smallest of the *abstols* specified in the disciplines associated with all the continuous nets of the signal.

If an undeclared net segment has multiple comparable disciplines connected to it, a connect statement shall specify which discipline to use during discipline resolution.