8.7 Connect specification statements

Any number of connect modules can be defined. The designer can choose and specialize those in the design via the connect specification statements. The connect specification statements allow the designer to define:

- specification of which connect module is used, including parameterization, for bridging given discrete and continuous disciplines
- overrides for the connect module default disciplines and port directions
- resolution of incompatible disciplines

The syntax for connect specifications is shown in Syntax 8-6.

```
connect_specification ::=
      connectrules connectrule_identifier;
       {connect_spec_item }
      endconnectrules
connect_spec_item ::=
        connect_insertion
      | connect_resolution
```

Syntax 8-6—Syntax for connect specification statements

The two forms of the connect specification statements and their syntaxes are detailed in the following subsections.

8.7.1 Connect module auto-insertion statement

The connect module insertion statement declares which connect modules are automatically inserted when mixed nets of the appropriate types are encountered, as shown in Syntax 8-7.

This specifies the connect module *connect_module*_identifier is used to determine the mixed-nets of the type used in the declaration of the connect module.

There can be multiple connect module declarations of a given (**discrete - continuous**) discipline pair and the connect module specification statement specifies which is to be used in the auto-insertion process. In addition, parameters of the connect module declaration can be specified via the *connect_attributes*.

connect_insertion ::=
  connect *connect_module*_identifier connect_attributes
  [[direction] *discipline*_identifier **,** [direction] *discipline*_identifier] **;**

connect_attributes ::=
  [connect_mode] [**#(** attribute_list **)**]

connect_mode ::=
  merge
  | **split**

attribute_list ::=
  attribute
  | attribute_list **,** attribute

attribute ::=
  .*parameter*_identifier **(** expression **)**

direction ::=
  input
  | output
  | inout

discipline_list ::=
  *discipline*_identifier
  | discipline_list **,** *discipline*_identifier

Syntax 8-7—Syntax for connect configuration statements

Connect modules can be reused for different, but compatible disciplines by specifying different discipline combinations in which the connect module can be used. The form is

connect *connect_module*_identifier connect_attributes *discipline*_identifier **,** *discipline*_identifier **;**

where the specified disciplines shall be compatible for both the continuous and discrete disciplines of the given connect module.

It is also possible to override the port directions of the connect module, which allows a module to be used both as a unidirectional and bidirectional connect module. This override also aids library based designs by allowing the user to specify the connect rules, rather than having to search the entire library. The form is

connect *connect_module*_identifier connect_attributes direction *discipline*_identifier **,**
  direction *discipline*_identifier **;**

where the specified disciplines shall be compatible for both the continuous and discrete disciplines of the given connect module and the specified directions are used to define the type of connect module.

8.7.2 Discipline resolution connect statement

The discipline resolution connect statement specifies a single discipline to use during the discipline resolution process when multiple nets with compatible discipline are part of the same mixed net, as shown in Syntax 8-8.

connect_resolution ::=
  connect discipline_list **resolveto** *discipline*_identifier **;**
discipline_list ::=
   *discipline*_identifier
  | discipline_list **,** *discipline*_identifier

Syntax 8-8—Syntax for connect configuration resolveto statements

where *discipline_list* is the list of compatible disciplines and *discipline_identifier* is the discipline to be used.

Issue 97: Ambiguity in the connect-resolveTo statement during discipline resolution as defined in 8.7.2. Not very clear as to how the connect rules apply. (Sri)

8.7.2.1 Connect Rule Resolution Mechanism

When there is an exact match for the set of disciplines specified as part of the discipline_list, the resolved discipline would be as per the rule specified in the exact match. When more than one specified rule applies to a given scenario a "Warning" message shall be issued by the simulator and the first match would be used.

When there is no exact fit, then the resolved discipline would be based on the subset of the rules specified. If there is more than one subset matching a set of disciplines, the simulator shall give an "Warning" message and apply the first subset rule that satisfies the current scenario.

The resolved discipline need not be one of the disciplines specified in the discipline list.

The connect-resolveto shall not be used as a mechanism to set the disciplines of simulator primitives but used only for discipline resolution.

Example 1:

```
connect x,y,a resolveto a;
connect x,y resolveto x;
```

For the above set of connect rule specifications;

disciplines x,y would resolve to discipline x.
disciplines x,y,a would resolve to discipline a.
disciplines y, a would resolve to discipline a.

Example 2:

```
connect x,y,a resolveto y;
connect x,y,a resolveto a;
connect x,y,b resolveto b;
```

For the above set of connect rule specifications;

disciplines x,y would resolve to discipline y with a warning.
disciplines x,y,a would resolve to discipline y with a warning.
disciplines y,b would resolve to b.

8.7.3 Parameter passing attribute

An attribute method can be used with the connect statement to specify parameter values to pass into the Verilog-AMS HDL connect module and override the default values. Any parameters declared in the connect module can be specified.

Examples:

```
connect a2d_035u #(.tt(3.5n), .vcc(3.3));
```

Here each parameter is listed with the new value to be used for that parameter.

8.7.4 connect_mode

This can be used to specify additional segregation of connect modules at each level of the hierarchy. Setting *connect_mode* to **split** or **merge** defines whether all ports of a common discrete discipline and port direction share an connect module or have individual connect modules.

Examples:

```
connect a2d_035u split #(.tt(3.5n), .vcc(3.3));
```

Here each digital port has a separate connect module.

8.8 Automatic insertion of connect modules

Automatic insertion of connect modules is performed when signals and ports with continuous time domain and discrete time domain disciplines are connected together. The connect module defines the conversion between these different disciplines.

An instance of the connect module shall be inserted across any mixed port which matches the rule specified by a **connect** statement. Rules for matching connect statements with ports take into account the port direction (see 8.8.1) and the disciplines of the signals connected to the port.