

Overview: The purpose of this is to correct inconsistencies between the bind BNF and the text. Specifically:

1. The text after the BNF states that “Possible target scopes include module, program, and interface declarations” but the BNF only shows modules and interfaces, and the text at the top only states modules and instances.
2. Clarify that what is allowed for a bind instantiation is restricted by what is allowed to be instantiated within the bind target scope.

Replace in 17.15 and BNF in A.1.4:

To facilitate verification separate from design, it is possible to specify properties and bind them to specific modules or instances. The following are some goals of providing this feature:

- It allows verification engineers to verify with minimum changes to the design code and files.
- It allows a convenient mechanism to attach verification Internet Protocol (IP) to a module or an instance.
- No semantic changes to the assertions are introduced due to this feature. It is equivalent to writing properties external to a module, using hierarchical path names.

With this feature, a user can bind a module, interface, or program instance to a module or a module instance.

The syntax of the **bind** construct is as follows:

```
bind_directive ::=
    bind bind_target_scope [: bind_target_instance_list] bind_instantiation ;
    | bind bind_target_instance bind_instantiation ;
bind_target_scope ::=
    module_identifier
    | interface_identifier
bind_target_instance ::=
    hierarchical_identifier constant_bit_select
bind_target_instance_list ::=
    bind_target_instance { , bind_target_instance }
bind_instantiation ::=
    program_instantiation
    | module_instantiation
    | interface_instantiation
```

The **bind** directive can be specified in any of the following:

- A module
- An interface
- A compilation-unit scope

There are two forms of bind syntax. In the first form, *bind_target_scope* specifies a target scope into which the *bind_instantiation* should be inserted. Possible target scopes include module, program, and interface declarations. In the absence of a *bind_target_instance_list*, the *bind_instantiation* is inserted into all instances of the specified target scope, designwide. If a *bind_target_instance_list* is present, the *bind_instantiation* is only inserted into the specified instances of the target scope. The *bind_instantiation* is effectively a complete program, module, or interface instantiation statement.

WITH

To facilitate verification separate from design, it is possible to specify properties and bind them to specific modules or interfaces, or instances thereof. The following are some goals of providing this feature:

- It allows verification engineers to verify with minimum changes to the design code and files.
- It allows a convenient mechanism to attach verification Internet Protocol (IP) to a module or an instance.
- No semantic changes to the assertions are introduced due to this feature. It is equivalent to writing properties external to a module, using hierarchical path names.

With this feature, a user can bind a module, interface, or program instance to a module or a module instance. *An interface could also be bound to an interface or interface instance.*

The syntax of the **bind** construct is as follows:

```

bind_directive ::=
    bind bind_target_scope [: bind_target_instance_list] bind_instantiation ;
    | bind bind_target_instance bind_instantiation ;

bind_target_scope ::=
    module_identifier
    | interface_identifier
bind_target_instance ::=
    hierarchical_identifier constant_bit_select
bind_target_instance_list ::=
    bind_target_instance { , bind_target_instance }

bind_instantiation ::=
    program_instantiation
    | module_instantiation
    | interface_instantiation

```

note 1: when the target scope is an interface, the bind instantiation must be an interface instantiation.

The **bind** directive can be specified in any of the following:

- A module
- An interface
- A compilation-unit scope

There are two forms of bind syntax. In the first form, *bind_target_scope* specifies a target scope into which the *bind_instantiation* should be inserted. Possible target scopes include module, ~~program~~, and interface declarations ~~or instances of such~~. In the absence of a *bind_target_instance_list*, the *bind_instantiation* is inserted into all instances of the specified target scope, designwide. If a *bind_target_instance_list* is present, the *bind_instantiation* is only inserted into the specified instances of the target scope. The *bind_instantiation* is effectively a complete program, module, or interface instantiation statement.