Situations where the **`ifdef**, **`else**, **`elsif**, **`endif**, and **`ifndef** compiler directives may be useful include:

— Selecting different representations of a module such as behavioral, structural, or switch level
— Choosing different timing or structural information
— Selecting different stimulus for a given run

The **`ifdef**, **`else**, **`elsif**, **`endif**, and **`ifndef** compiler directives have the syntax shown in Syntax 0-1.

```
conditional_compilation_directive ::=
        ifdef_directive
      | ifndef_directive
ifdef_directive ::=
      `ifdef text_macro_identifier
      ifdef_group_of_lines
      { `elsif text_macro_identifier elsif_group_of_lines }
      [ `else else_group_of_lines ]
      `endif
ifndef_directive ::=
      `ifndef text_macro_identifier
      ifndef_group_of_lines
      { `elsif text_macro_identifier elsif_group_of_lines }
      [ `else else_group_of_lines ]
      `endif
```

Syntax 0-1—Syntax for conditional compilation directives

The `text_macro_identifier` is a Verilog HDL *simple_identifier*. The `ifdef_group_of_lines`, `ifndef_group_of_lines`, `elsif_group_of_lines` and the `else_group_of_lines` are parts of a Verilog HDL source description. The **`else** and **`elsif** compiler directives and all of the groups of lines are optional.

The **`ifdef**, **`else**, **`elsif**, and **`endif** compiler directives work together in the following manner:

— When an **`ifdef** is encountered, the `ifdef` text macro identifier is tested to see if it is defined as a text macro name using **`define** within the Verilog HDL source description.
— If the `ifdef` text macro identifier is defined, the `ifdef` group of lines is compiled as part of the description and if there are **`else** or **`elsif** compiler directives, these compiler directives and corresponding groups of lines are ignored.
— If the `ifdef` text macro identifier has not been defined, the `ifdef` group of lines is ignored.
  — If there is an **`elsif** compiler directive, the `elsif` text macro identifier is tested to see if it is defined as a text macro name using **`define** within the Verilog HDL source description.
  — If the `elsif` text macro identifier is defined, the `elsif` group of lines is compiled as part of the description and if there are other **`elsif** or **`else** compiler directives, the other **`elsif** or **`else** directives and corresponding groups of lines are ignored.
  — If the first `elsif` text macro identifier has not been defined, the first `elsif` group of lines is ignored.
  — If there are multiple **`elsif** compiler directives, they are evaluated like the first **`elsif** compiler directive in the order they are written in the Verilog HDL source description.
  — If there is an **`else** compiler directive, the `else` group of lines is compiled as part of the description.

Although the names of compiler directives are contained in the same name space as text macro names, the names of compiler directives are considered not to be defined by **`ifdef**, **`ifndef**, and **`elsif**.

The **`ifndef**, **`else**, **`elsif**, and **`endif** compiler directives work together in the following manner:

- When an **`ifndef** is encountered, the `ifndef` text macro identifier is tested to see if it is defined as a text macro name using **`define** within the Verilog HDL source description.
- If the `ifndef` text macro identifier is not defined, the `ifndef` group of lines is compiled as part of the description and if there are **`else** or **`elsif** compiler directives, these compiler directives and corresponding groups of lines are ignored.
- If the `ifndef` text macro identifier is defined, the `ifndef` group of lines is ignored.
  - If there is an **`elsif** compiler directive, the `elsif` text macro identifier is tested to see if it is defined as a text macro name using **`define** within the Verilog HDL source description.
  - If the `elsif` text macro identifier is defined, the `elsif` group of lines is compiled as part of the description and if there are other **`elsif** or **`else** compiler directives, the other **`elsif** or **`else** directives and corresponding groups of lines are ignored.
  - If the first `elsif` text macro identifier has not been defined, the first `elsif` group of lines is ignored.
  - If there are multiple **`elsif** compiler directives, they are evaluated like the first **`elsif** compiler directive in the order they are written in the Verilog HDL source description.
  - If there is an **`else** compiler directive, the `else` group of lines is compiled as part of the description.

Although the names of compiler directives are contained in the same name space as text macro names, the names of compiler directives are considered not to be defined by **`ifdef**, **`ifndef**, and **`elseif**.

Nesting of **`ifdef**, **`ifndef**, **`else**, **`elsif**, and **`endif** compiler directives shall be permitted.

NOTE—Any group of lines that the compiler ignores still has to follow the Verilog HDL lexical conventions for white space, comments, numbers, strings, identifiers, keywords, and operators.

Examples:

Example 1—The example below shows a simple usage of an **`ifdef** directive for conditional compilation. If the identifier `behavioral` is defined, a continuous net assignment will be compiled in; otherwise, an **and** gate will be instantiated.

```
module and_op (a, b, c);
output a;
input b, c;

`ifdef behavioral
    wire a = b & c;
`else
    and a1 (a,b,c);
`endif

endmodule
```

