## 2.8. Директивы компилятора

Последним из рассматриваемых в данной главе элементов языка являются директивы компилятора.

Все директивы компилятора начинаются со знака «обратный штрих» ( ` ). Директивы предназначены для управления такими параметрами компиляции и моделирования, как, например, разрядность целых чисел по умолчанию, длина шага моделирования и т.п.

Полный список директив компилятора приведен ниже:

**`celldefine** *module_declaration*
**`endcelldefine**
**`default_nettype** *net_type_identifier*
**`define** *macro_name macro_text*
**`define** *macro_name*(*list_of_arguments*) *macro_text*
**`undef** *macro_name*
**`ifdef** *macro_name group_of_lines*
**`else** *group_of_lines*
**`endif**

```
`include "filename"
`resetall
`timescale time_unit / time_precision
`unconnected_drive module_declaration
`nounconnected_drive
```

Подробно каждая из директив будет описана в соответствующих главах при рассмотрении специальных вопросов.

Для примера рассмотрим одну из наиболее широко используемых директив в Verilog-программах – директиву **`include**, предназначенную для включения в текущий файл исходных текстов на языке Verilog из других файлов.

Синтаксис директивы:

```
`include имя_включаемого_файла;
```

Предположим, например, что во внешнем файле modules.v размещены описания модулей *CLK_Generator* и *Shift_Reg*. Детальное описание процедуры использования модулей приведено в главе 3. В дальнейшем сформируем главный модуль, использующий вышеперечисленные подчиненные модули и подключающий их с помощью директивы **`include**:

```
// Заголовок (начало) главного модуля:
module Include_Demo ( CLK, X, Y );

// Раздел описаний главного модуля:
input [7:0] X ;// X – входной порт устройства Include_Demo
wire [7:0] X ;// сигнал типа "цепь"

output CLK ;      // CLK – выходной порт устройства Include_Demo
wire CLK ;        // сигнал типа "цепь"

output [7:0] Y ;// Y – выходной порт устройства Include_Demo
reg [7:0] Y ;   // сигнал типа "регистр"

reg push;         // push – внутренний сигнал устройства
                  // Include_Demo типа "регистр"
. . .
```

```
//Основной текст Verilog-программы
. . .

//Включение подчиненных модулей, не описанных выше в текущем файле
CLK_Generator  Inst00 (CLK);
Shift_Reg      Inst01 (CLK, push, X, Y);

endmodule              //Конец главного модуля Include_Demo

// Директива включения файла modules.v
// (файл в текущем каталоге)
`include modules.v
```

Применение в программе директивы включения файла **`include** аналогично непосредственному копированию в эту программу исходного текста, размещенного во включаемом файле. Использование директивы **`include** (в противовес прямому копированию исходного текста) обладает тем преимуществом, что при изменении часто используемых фрагментов проекта нет необходимости исправлять все программы, куда эти фрагменты входят, так как такие программы ссылаются на один общий источник с помощью директивы **`include**.

Следует отметить, что пользователи программного пакета Active-HDL [23] для включения модулей могут обходиться без применения директивы **`include**. Средства управления проектом и оптимизированный компилятор в указанном пакете позволяют использовать модули, входящие в текущий проект, непосредственно, без указания их физического местоположения на диске. Таким образом, если при работе в пакете Active-HDL оба модуля из перечисленного примера находятся в текущем проекте, то удаление строки с директивой **`include** не приведет к появлению ошибок.