Языки описания схем

(mk.cs.msu.ru → Лекционные курсы → Языки описания схем)

Блок 18
Verilog:
немного синтаксического сахара

лектор:
Подымов Владислав Васильевич

e-mail:
valdus@yandex.ru

Осень 2018

(V) Директивы компилятору, комментарии

```
`include имя_файла
`define имя_макроса текст_макроса
`define имя_макроса(аргументы) текст_макроса
`undef имя_макроса
`ifdef имя_макроса
`ifndef имя_макроса
`elsif имя_макроса
`else
`endif
```

`//` — начало однострочного комментария

`/* ... */` — многострочный комментарий

Всё это работает так же, как и в C/C++

(V) Глобальные параметры

Иногда при разработке схемы хочется описать одним небольшим текстом **много** модулей, отличающихся незначительными деталями

Например, время разработки схемы сократится, если один раз написать модуль “параллельный регистр ширины `W`” и при объявлении экземпляров явно или неявно указывать значение `W`

Так можно делать при помощи параметров

Параметры,

- объявляются локально внутри модуля аналогично проводам и переменным, но не являются точками схемы
- как и порты, доступны для (пере)определения в объявлении экземпляра модуля
- могут быть использованы во многих выражениях в описании модуля

Модуль, содержащий хотя бы один параметр, принято называть параметризованным

(V) Глобальные параметры

```
объявление_параметров ::= parameter список_объявлений;
список_объявлений ::= объявление |
                      объявление, список_объявлений
объявление ::=
    имя_параметра = константное_выражение_по_умолчанию
```

Объявить параметры модуля можно двумя способами, аналогичными объявлению портов:

```
module имя_модуля(имена_портов);
  ... объявление_параметров ...
endmodule
```

```
module имя_модуля #(объявление_параметров)
                   (объявление_портов);
```

(V) Глобальные параметры

Пример: параметризованный модуль параллельного регистра ширины `W`

Первый способ:

```
module parreg(in,out,clk);
  parameter W = 8;
  input [W-1:0] in;
  input clk;
  output reg [W-1:0] out;
  always @(posedge clk) out <= in;
endmodule
```

Второй способ:

```
module parreg #(parameter W = 8)
    (input [W-1:0] in, output [W-1:0] out, input clk);
  always @(posedge clk) out <= in;
endmodule
```

(V) Глобальные параметры

Пример: экземпляры параллельного регистра ширины `W`

```
module top();
  ...
  // регистр ширины 8 (по умолчанию)
  parreg _r1(.in(w1), .out(w2), .clk(w3));
  ...
  // регистр ширины 5, первый вариант
  parreg #(.W(5)) _r2(.in(w4), .out(w5), .clk(w6));
  ...
  // регистр ширины 5, второй вариант
  defparam _r3.W = 5;
  parreg _r3(.in(w7), .out(w8), .clk(w9));
  ...
endmodule
```

(V) Локальные параметры

Единственное отличие объявления локальных параметров от объявления обычных (глобальных) параметров:
вместо слова `parameter` **используется слово** `localparam`

Единственное отличие использования локальных параметров от использования обычных (глобальных) параметров:
запрещено явно переопределять значение по умолчанию при объявлении экземпляра модуля

Использование локальных параметров рекомендуется в двух случаях:

1. значение параметра — это деталь реализации, не интересная тому, кто использует модуль
2. значение параметра однозначно определяется значениями других параметров, и определять её по-другому **нельзя**

(V) Массивы

В языке Verilog можно объявлять и использовать многомерные массивы примерно так же, как это делается в C/C++:

```
// массив из 4-х однобитовых проводов
wire w[0:3];

// массив из 3-х переменных -- шин ширины 5
reg [4:0] r[5:7];

// двумерный массив (3x3) логических значений
reg r2[0:2][1:3];

// двумерный массив (3x3) проводов -- шин ширины 5
wire [4:0] w2[0:2][1:3];
assign w2[1][3] = r[5];
```

Ограничение: порты модуля не могут быть массивами