Verilog HDL语言规范

主 讲：何宾

Email：hebin@mail.buct.edu.cn

2014.06

Verilog HDL分配

分配(也称赋值）是最简单的机制，用于给网络和变量设置相应的值。Verilog HDL提供了两种基本形式的分配：

- 连续分配

  用于给网络分配值。

- 过程分配

  用于给变量分配值。

Verilog HDL分配

Verilog HDL还额外提供了两种分配形式

- assign /deassign
- force/release

称之为过程连续分配。

一个分配由两部分构成，包括：

- 左侧和右侧，它们通过“=”隔开。
- 或者在非阻塞过程赋值中，使用“<=”隔开。

  右侧可以是任意表达式。

Verilog HDL分配

根据连续分配或者过程分配，左边的赋值（分配）类型遵守下表的规则。

描述类型	左侧
连续分配	1．网络（标量或者矢量） 2．一个向量网络的常数位选择 3．一个向量网络的常数部分选择 4．一个向量网络的常数索引的部分选择 5．以上任何左侧的连接或者嵌套的连接
过程分配	1．变量（标量或者矢量） 2．一个向量寄存器、整数或者时间变量的比特选择 3．一个向量寄存器、整数或者时间变量的部分选择 4．一个向量寄存器，整数或者时间变量索引的部分选择 5．存储器字 6．以上任何左侧的连接或者嵌套的连接

Verilog HDL分配
--连续分配

连续分配包括：

- 网络声明分配
- 连续分配描述

Verilog HDL分配
--连续分配

□网络声明分配

前面讨论了声明网络的两种方法，这里给出第三种方法，即：网络声明分配。在声明网络的相同描述中，允许在网络上使用连续分配。

Verilog HDL分配
--连续分配

连续分配的网络声明格式Verilog HDL描述的例子

```
wire (strong1, pull0) mynet = enable ;
```

注：

由于一个网络只能声明一次，所以对于一个特定的网络来说，只能分配一个网络声明。这和连续分配描述是不一样的。连续分配描述中，一个网络可以接受连续分配形式的多个分配。

Verilog HDL分配
--连续分配

连续分配将为一个网络数据类型设置一个值。网络可能明确的声明，或者根据隐含声明规则继承一个隐含声明。

给一个网络进行分配是连续的和自动的。换句话说，任何时候，只要右边的一个操作表达式的操作数发生变化，则将改变整个右边表达式。

如果新的值和以前的值不一样，则将给左边分配新的值。

Verilog HDL分配
--连续分配

连续分配描述的语法格式如下：

assign variable=expression；

其中：

- **variable为网络数据类型。**
- **expression为赋值表达式。**

Verilog HDL分配
--连续分配

使用连续分配实现带进位的四位加法器Verilog HDL描述的例子。

```
module adder (sum_out, carry_out, carry_in, ina, inb);
output wire[3:0] sum_out;
output wire carry_out;
input wire[3:0] ina, inb;
input wire carry_in;
assign {carry_out, sum_out} = ina + inb + carry_in;
endmodule
```

Verilog HDL分配
--连续分配

使用连续分配实现4:1的16位总线的多路选择Verilog HDL描述的例子。

```
module select_bus(busout, bus0, bus1, bus2, bus3, enable, s);
parameter n = 16;
parameter Zee = 16'bz;
output [1:n] busout;
input [1:n] bus0, bus1, bus2, bus3;
input enable;
input [1:2] s;
tri [1:n] data;     // 声明网络
```

Verilog HDL分配
--连续分配

```
tri [1:n] busout = enable ? data : Zee; // 带有连续分配的网络声明

//带有四个连续分配的分配描述
assign
data = (s == 0) ? bus0 : Zee,
data = (s == 1) ? bus1 : Zee,
data = (s == 2) ? bus2 : Zee,
data = (s == 3) ? bus3 : Zee;
```

Verilog HDL分配
--连续分配

在连续分配中，延迟用于指定将右边操作数的变化分配到左边的时间间隔。

如果左边是一个标量网络，这种分配的效果和门延迟是一样的。即：可以为输出上升、下降和改变为高阻指定不同的延迟。

Verilog HDL分配
--连续分配

如果左边是一个向量网络，则可以应用最多三个延迟。下面的规则用于确定哪个延迟用于控制分配：

- 如果右边从非零变化到零，则应该使用下降延迟(falling delay)。
- 如果右边变化到高阻，则应该使用关闭延迟(turn-off delay)。
- 对于其它情况，应该使用上升延迟(rising delay)。

注：在连续分配中指定延迟，是网络声明的一部分，它不仅指定一个网络延迟，而且，可以为这个网络进行连续分配。在一个网络声明中，可以将一个延迟值应用到一个网络中。

Verilog HDL分配
--连续分配

一个延迟值应用到一个网络中Verilog HDL描述的例子。

```
wire #10 wireA;
```

这描述了任何变换的值，需要延迟10个时间单位后，才能应用到wireA网络。

注：对于一个向量网络的分配，当在声明中包含分配时，不能将上升延迟和下降延迟应用到单个的位。

Verilog HDL分配
--连续分配

- 强度

用户可以在一个连续分配中指定驱动强度。这只应用于为下面类型的标量网络进行分配的情况：

wire	tri	trireg
wand	triand	tri0
wor	trior	tri1

Verilog HDL分配
--连续分配

连续分配驱动强度可以在网络声明中指定，也可以通过使用assign关键字在一个单独的分配中指定。

如果提供了强度描述的话，应该紧跟关键字（用户网络类型的关键字或者assign），并且在任何指定的延迟前面。当连续分配驱动网络时，应该按照指定的值进行仿真。

Verilog HDL分配
--连续分配

一个驱动强度描述应该包含一个强度值，当给网络分配的值是1的时候使用第一个强度值；当给分配的值是0的时候使用第二个强度值。

- 下面的关键字，用于为分配1指定强度值：

  supply1 strong1 pull1 weak1 highz1

- 下面的关键字，用于为分配0指定强度值:

  supply0 strong0 pull0 weak0 highz0

Verilog HDL分配
--连续分配

两个强度说明的顺序是任意的。下面两个规则将约束强度说明：

- 强度描述（highz1，highz0）和（highz0，highz1）认为是非法的结构。
- 如果没有指定驱动强度，则默认为（strong1，strong0）。

Verilog HDL分配
--过程分配

连续分配驱动网络的行为类似于逻辑门驱动网络。右边的分配表达式可以认为是连续驱动网路的组合逻辑电路。

不同的是，过程分配为变量赋值。分配没有连续性，变量一直保存着上一次分配的值，直到下一次为变量进行了新的过程分配为止。

Verilog HDL分配
--过程分配

过程分配发生在下面的过程中，比如：

- always
- initial
- task
- function

Verilog HDL分配
--过程分配

变量声明分配是过程分配的一个特殊情况，用于给变量分配一个值。它允许在相同描述中，给一个变量设置一个初始值。

- 过程分配应该是一个常数表达式。
- 该分配没有连续性。取而代之的是，变量一直保持该分配值，直到下一个分配到来。
- 不能对一个数组使用变量声明分配。此外，变量声明分配只能用于模块级。如过在initial模块和变量声明分配中，为相同的变量分配了不同的值，没有定义分配的顺序。

Verilog HDL分配
--过程分配

定义一个4位变量且分配初值Verilog HDL描述的例子。

```
reg[3:0] a = 4'h4;
```

这等价于：

```
reg[3:0] a;
initial a = 4'h4;
```

为数组分配初值是非法的

```
reg [3:0] array [3:0] = 0;
```

Verilog HDL分配
--过程分配

声明两个整数，第一个分配值为0的Verilog HDL描述的例子。

```
integer i = 0, j;
```

声明两个实数变量，为其分配值2.5和300,000的Verilog HDL描述的例子。

```
real r1 = 2.5, n300k = 3E6;
```

声明一个时间变量和一个实时时间变量，并分配初值的Verilog HDL描述的例子。

```
time t1 = 25;
realtime rt1 = 2.5;
```