



沈阳工业大学

SHENYANG UNIVERSITY OF TECHNOLOGY

数字电子技术

沈阳工业大学
电子技术教研室

21

知识点21 Verilog HDL基本结构



沈阳工业大学

SHENYANG UNIVERSITY OF TECHNOLOGY



Verilog HDL简介

Verilog HDL作为一种硬件描述语言，用于从算法级、门级到开关级的多种抽象设计层的数字系统建模。被建模的数字系统复杂性可以从简单的门电路到完整的数字系统，数字系统能够按层次描述，并可在相同描述中显式地进行时序建模。

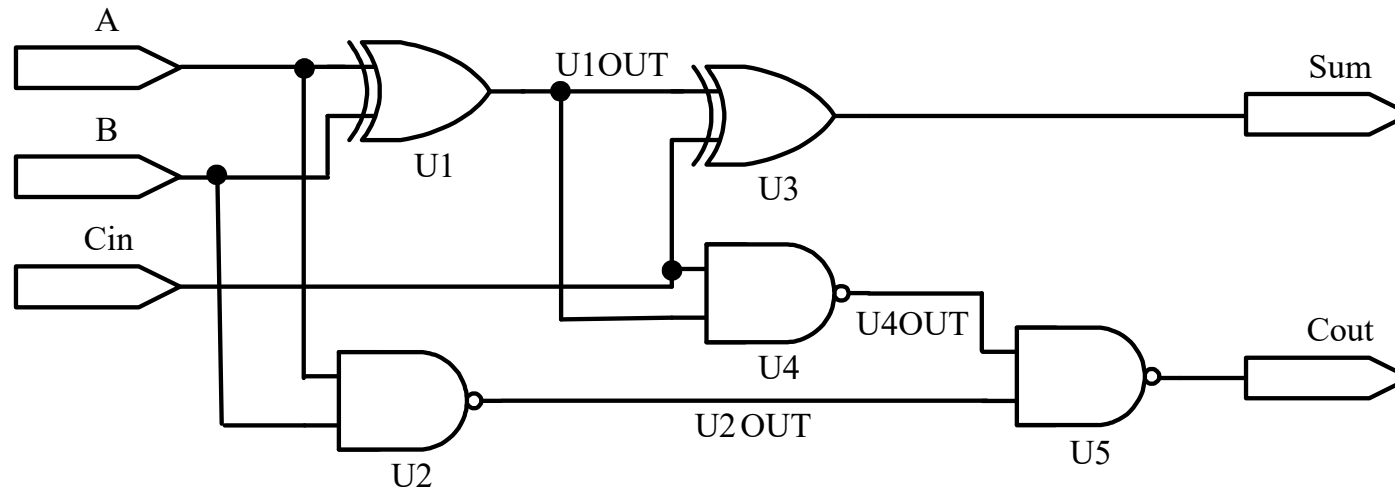
Verilog HDL语言具有以下描述能力：行为建模、数据流建模、结构建模以及包含响应监控和设计验证方面的时延和波形产生机制。在利用Verilog HDL进行数字系统设计过程中，可以单独使用其中的一种建模方式，也可以是多种建模方式混合使用。



Verilog HDL语言程序的基本结构

VerilogHDL结构

1. 模块表达





```
module FullAdder (A, B, Cin, Sum, Cout) ;  
  input  A, B , Cin;  
  output Sum, Cout ;  
  wire  U1OUT, U2OUT, U4OUT;  
    assign U1OUT = A ^ B ;  
    assign U2OUT = !(A & B) ;  
    assign U4OUT = !(U1OUT & Cin) ;  
    assign Sum = U1OUT ^ Cin ;  
    assign Cout = !(U2OUT & U4OUT) ;  
endmodule
```




2. 模块名

3. 端口列表

4. 端口方向模式

`input output inout`

5. 数据类型



关键词

关键词是语言本身预定义的，具有特定用法和含义的符号，例如module、if等，关键词只能用小写符号。

部分关键词列表如下。

序号	关键词	序号	关键词	序号	关键词	序号	关键词
1	always	27	for	53	output	79	supply0
2	and	28	force	54	parameter	80	supply1
3	assign	29	forever	55	pmos	81	table
4	begin	30	fork	56	posedge	82	task
5	buf	31	function	57	primitive	83	time
6	bufif0	32	highz0	58	pull0	84	tran
7	bufif1	33	highz1	59	pull1	85	tranif0
8	case	34	if	60	pullup	86	tranif1
9	casex	35	ifnone	61	pulldown	87	tri
10	casez	36	initial	62	rcmos	88	tri0



Verilog HDL语言的基本要素

1. 标识符

标识符是由用户自主定义的，用来表示用户指定的特定含义符号，例如模块名称、变量名称、端口名称等。

标识符可以由任意一个或多个字母、数字、\$符号和下划线符号的组合。

标识符的第一个字符必须是字母或者下划线。另外，标识符是区分大小写的，同一字符的大写形式和小写形式在Verilog HDL语言中被认为是两种不同的字符。

以下是标识符的几个例子：

Counter

COUNTER // 与Counter不同

_m1u2 _f432

FILE\$



2.注释

注释的含义与C语言等的注释一样，可以用来对某个变量、信号或模块等进行必要的解释说明，提高程序代码的可读性。在Verilog HDL 语言中有两种形式的注释符，分别为：

(1) /* */ 此注释符可以用来注释一行、多行或部分行

(2) // 此注释符仅能注释符号本身右侧的内容，范围至行尾结束



编译指令

以`（反引号）开始的标识符是编译器指令。在Verilog HDL语言编译中，特定的编译器指令在整个编译过程中有效，直到遇到其它不同的编译指令。

1. `define和`undef

`define 指令用于文本替换，类似于C语言中的#define指令，如：

```
`define DataBus 32
```

```
...
```

```
Reg[DataBus -1:0] Data;
```

`define编译指令在整个编译过程中都有效。

`undef指令用于终止前面`define编译指令的定义。

`define 编译指令的作用范围自定义位置开始，到`undef指令结束。

即`define 和 `undef共同决定了该编译指令的作用范围。



2. ``ifdef`、``else` 和 ``endif`

这三个编译指令属于条件编译，类似于C语言中的`#ifdef` `#esle`。如下所示：

```
`ifdef WIN16
    Parameter WordSize =16;
`else
    Parameter WordSize = 32;
`endif
```

在编译过程中，如果已经定义了名字为WIN16的文本宏，就选择第一种参数声明，否则选择第二种参数说明。``else` 程序指令对于``ifdef`指令是可选的。



3. ``default_nettype`

该指令用于指定默认的隐式线网类型。即在沒有明确指定类型的情况下自动默认线网类型。在沒有使用该指令的情况下，Verilog HDL语言本身默认的线网类型为wire。

``default_nettype wand`

该实例指令将默认的隐式线网类型修改为wand，即线网类型。从该指令开始后续的所有模块中，沒有经过明确说明的连线都被自动指定为默认的wand类型。



4. ``include`

``include` 编译指令类似于C语言的`#define`。用于将Verilog HDL程序文件添加到当前你的工程中，供当前工程进行功能模块的调用。被添加的文件既可以用相对路径名定义，也可以用全路径名定义，例如：

```
`include "D:/Verilog HDL/model.v"
```

编译时，这一行由文件“D:/Verilog HDL/model.v”的内容替代。



5. ``resetall`

该编译器指令将所有被修改过的编译指令重新恢复到缺省状态。

```
`default_nettype wand
```

```
...
```

```
`resetall
```

在该实例中，``resetall`编译指令之前的默认类型为wand，之后的默认类型恢复为语言默认的wire。



6. `timescale

在Verilog HDL模型中，时间延时不直接使用时间物理量，如5ms，而是所有延时都用单位时间的数量来表述。而单位时间与实际的对应由`timescales编译器指令进行关联。`timescales编译指令用于定义延时的时间单位和延时精度。

 `timescale编译器指令格式为：

 `timescale time_unit/ time_precision

 time_unit 是时间单位，time_precision是延时精度。

time_unit和time_precision由数值和时间的物理量单位组成。

数值只能是1、10、和100。时间物理单位包括s、ms、us、ns、ps和fs。例如：

 `timescale 1ns/100ps



7. ``unconnected_drive`和``nounconnected_drive`

在模块实例化中，出现在这两个编译器指令间的任何未连接的输入端口，默认状态由该指令决定。

```
`unconnected_drive pull1
```

```
...
```

```
/*在这两个程序指令间的所有未连接的输入端口默认为高电平*/
```

```
`nounconnected_drive
```

```
`unconnected_drive pull0
```

```
...
```

```
/*在这两个程序指令间的所有未连接的输入端口默认为低电平*/
```

```
`nounconnected_drive
```



数值表示

Verilog HDL语言有下列四种基本的值：

- 0： 低电平、数值0或逻辑“假”
- 1： 高电平、数值 1或逻辑“真”
- x： 未知
- z： 高阻

Verilog HDL中有三类常量：整型数、实数、字符串型。下划线符号“_”可以随意用在整数或实数中，不影响数值本身的大小，但可以用来提高数值的易读性，需要注意的是下划线符号不能用作为首字符。



1. 整型数

(1) 基本十进制数格式

这种形式的整数定义为带有一个可选的一元“+”或一元“—”操作符的数字序列，这种格式的整型数代表一个有符号数。
例如：

128

十进制数128

—43

十进制数—43

(2) 基数格式

这种格式的整数表示为：

`[size] ' base value`

size定义以数据位（bit）为计量单位的常量所占用的位长；
base是所表示数据的进制数，其中o或O表示八进制，b或B表示二进制，d或D表示十进制，h或H表示十六进制；**value**是基于**base**所示进制的数值。值x和z以及十六进制中的a、b、c、d、e、f不区分大小写。



2.实数

实数可以有两种形式表示，十进制计数法和科学计数法

(1) 十进制计数法 例如：

22.0

15.678

342.234

0.134

5. //非法：小数点两侧必须有1位数字

(2) 科学计数法 例如：

23_3.le1 其值为2331.0；忽略下划线

4.7E2 470.0 (e与E相同)

3E-4 0.0003



3.字符串

字符串是双引号中的字符序列。字符串不能分成多行书写。例如：

```
" Verilog_HDL "  
"Reached->Here"
```

反斜线（\）用于对确定的特殊字符转义。

\n	换行符
\t	制表符
\\	字符\本身
\"	字符"
\203	八进制数203对应的字符



知识要点： VerilogHDL基本结构及各部分的功能要求
知识难点： VerilogHDL基本结构及各部分的功能要求