

Verilog-HDL

实践与 应用系统设计

常晓明 编著



北京航空航天大学出版社

<http://www.buaapress.com.cn>

Verilog – HDL 实践与应用系统设计

常晓明 编著

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内容简介

本书从实用的角度介绍了硬件描述语言 Verilog - HDL。通过动手实践,体验 Verilog - HDL 的语法结构、功能等内涵。在前五章,以简单的实例列举了 Verilog - HDL 的用法;在后四章,以应用系统为例详细讲解了系统设计的全过程。书中的全部例子都给出了仿真结果,其源代码都在本书所附的 CD-ROM 中,并均经过验证无误。

本书的前半部分特别适合于初学者,也可作为工程技术人员的参考内容。后半部分很适合工程开发和研究人员参考。本书除了介绍 Verilog - HDL 外,还涉及到 VB 编程、VC++ 和 DLL 的开发等知识,以及接口等领域的问题,书中都进行了详尽的讲解。

图书在版编目(CIP)数据

Verilog - HDL 实践与应用系统设计/常晓明编著.

北京:北京航空航天大学出版社,2003.1

ISBN 7-81077-263-5

I. V... II. 常... III. 硬件描述语言,VHDL—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 088334 号

Verilog - HDL 实践与应用系统设计

常晓明 编著

责任编辑 孔祥燮

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn>

E-mail:pressell@publica.bj.cninfo.net

河北省涿州市新华印刷厂印装 各地书店经销

*

开本:787×1092 1/16 印张:13.25 字数:339 千字

2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-263-5 定价:26.00 元

前 言

近年来科学技术的发展十分迅猛,特别是电子信息技术的发展更是日新月异。在我们生活的周围,已经越来越看不到与电子技术无关的东西了。从各种家电到办公自动化设备、通信设备、多媒体外围设备等领域,无处不有数字技术的应用。与过去的模拟电子技术相比,数字技术更有其广阔的应用前景。

数字技术的飞速发展得益于大规模集成电路技术的发展和其硬件成本的不断下降。著者在20世纪70年代,为了买两只用于放大器的功率放大晶体管,需要花费工资10%的投入,并且还是分立元件。而现在,按一般工资收入考虑,用10%的工资额度已经可以买一台收音机了。更不用说用20~30元即可买一只具有成千上万个元件的集成电路。这些良好的外围环境为我们提供了得天独厚的学习条件,可以使我们在短时间内掌握更多的东西。

但另一方面,大量的信息的涌现,大量的书籍的问世,大量需要学习的东西的出现,使得当今的人们有时不知从何处下手来获取知识。这些年来,由于社会对人才的需求,各大专院校也在不断扩招学生,为社会培养了大量的人才。同时,应该注意到:由于有的地方硬件设施方面还不能满足需要,使得接受教育的人仅仅掌握了些理论知识,而缺乏实践的能力。

读书就像吃饭的过程,而实践就像消化的过程。如果学了的东西不用于实践,就不能使学到的知识巩固和发展。在学习过程中,如果能做到边学习边实践,将会有很好的学习效果。

正是出于这样一种思考,著者力求从实践的角度编写了本书。本书不是讲解Verilog-HDL的历史、理论 and 特点,而是以动手为主。通过动手实践,体验Verilog-HDL的语法、结构、功能等内涵。在前五章,以简单的实例列举了Verilog-HDL的用法,给出了仿真结果。在其后的章节里,以应用系统为例,详细讲解了系统设计的全过程。著者力图通过几个实际例子,给读者以启发,使得读者能够通过阅读本书增长自己的系统设计和开发的能力。

为此,本书所附CD-ROM中将本书的全部实例收录在内,这些实例全部通过了验证。第7章以后的设计实例,不仅有Verilog-HDL的例子,也附了包括VB、VC++等源程序,甚至将DLL的生成方法也详尽地作了说明。

在本书的编写过程中,太原理工大学2002级计算机系研究生李媛媛对本书的全部软件进行了验证,并参加了部分语言的设计和调试工作。在此,著者表示

深切的感谢。在编写过程中,参考了许多有关 Verilog - HDL 和 VHDL 方面的书籍,这些书都为本人提供了十分有益的参考。在此,对这些书的作者表示谢意。

本书存在的不足和错误之处,恳请广大读者给予批评指正。

作者邮件地址:lab@xiaoming-lab.com。

编著者

2002 年 9 月

目 录

1

第 1 章 硬件描述语言

1.1 什么是硬件描述语言 HDL	1
1.2 基本逻辑电路的 HDL	1
1.2.1 与门逻辑电路的描述	1
1.2.2 与非门逻辑电路的描述	4
1.2.3 非门逻辑电路的描述	5
1.2.4 或门逻辑电路的描述	6
1.2.5 或非门逻辑电路的描述	6
1.2.6 缓冲器逻辑电路的描述	7
1.3 逻辑仿真	7
1.3.1 顶层模块的编写	8
1.3.2 寄存器定义	8
1.3.3 线网定义	8
1.3.4 底层模块的调用	9
1.3.5 输入信号波形的描述	9
1.3.6 二与门逻辑电路的仿真结果	10

第 2 章 仿真器的获取、安装及运行

2.1 如何得到 ISE WebPACK ModelSim XE 仿真器?	11
2.2 通过网站下载和安装 ISE WebPACK ModelSim XE 仿真器	11
2.3 建立一个新的工程文件	20
2.4 一个最简单的仿真实例	21

第 3 章 组合逻辑电路

3.1 数据选择器	31
3.1.1 2-1 数据选择器的描述	31
3.1.2 真值表和逻辑表达式	31
3.1.3 2-1 数据选择器的 Verilog - HDL 描述	32
3.1.4 4-1 数据选择器的逻辑电路	33
3.1.5 4-1 数据选择器的 Verilog - HDL 描述	33
3.1.6 条件操作符的使用方法	34
3.1.7 数据选择器的行为描述方式	35
3.1.8 case 语句的使用方法	36

3.1.9 if_else 语句的使用方法	37
3.1.10 function 函数	38
3.1.11 用于仿真的顶层模块	38
3.1.12 数据选择器的仿真结果	39
3.2 数据比较器	41
3.2.1 最简单的数据判断方法	41
3.2.2 2 位数据比较器	41
3.2.3 2 位数据比较器的 Verilog - HDL 描述	42
3.2.4 数据比较器的数据宽度扩展	45
3.2.5 全比较器的 Verilog - HDL 描述	47
3.3 编码器	49
3.3.1 二进制编码器	49
3.3.2 二进制编码器的 Verilog - HDL 描述	50
3.4 译码器	52
3.4.1 BCD 码译码器	52
3.4.2 非完全描述的逻辑函数和逻辑表达式的简化	53
3.4.3 BCD 码译码器的 Verilog - HDL 描述	55
3.4.4 BCD 码译码器的仿真结果	56

第 4 章 触发器

4.1 异步 RS 触发器	57
4.1.1 异步 RS 触发器的逻辑符号	57
4.1.2 异步 RS 触发器的 Verilog - HDL 描述	57
4.1.3 异步 RS 触发器的仿真结果	59
4.1.4 always 块语句	59
4.2 同步 RS 触发器	60
4.2.1 同步 RS 触发器的逻辑符号	60
4.2.2 同步 RS 触发器的 Verilog - HDL 描述	61
4.2.3 同步 RS 触发器的仿真结果	62
4.3 异步 T 触发器	62
4.3.1 异步 T 触发器的逻辑符号	62
4.3.2 异步 T 触发器的 Verilog - HDL 描述	63
4.3.3 异步 T 触发器的仿真结果	64
4.4 同步 T 触发器	64
4.4.1 同步 T 触发器的逻辑符号	64
4.4.2 同步 T 触发器的 Verilog - HDL 描述	65
4.4.3 同步 T 触发器的仿真结果	66
4.5 同步 D 触发器	66
4.5.1 同步 D 触发器的逻辑符号	66

4.5.2 同步 D 触发器的 Verilog - HDL 描述	67
4.5.3 同步 D 触发器的仿真结果	68
4.6 带有复位端的同步 D 触发器	68
4.6.1 带有复位端的同步 D 触发器的逻辑符号	68
4.6.2 带有复位端的同步 D 触发器的 Verilog - HDL 描述	69
4.6.3 带有复位端的同步 D 触发器的仿真结果	70
4.7 同步 JK 触发器	70
4.7.1 同步 JK 触发器的逻辑符号	70
4.7.2 同步 JK 触发器的 Verilog - HDL 描述	72
4.7.3 同步 JK 触发器的仿真结果	73

第 5 章 时序逻辑电路

5.1 寄存器	74
5.1.1 寄存器的组成原理	74
5.1.2 寄存器的 Verilog - HDL 描述	75
5.1.3 寄存器的仿真结果	76
5.2 移位寄存器	77
5.2.1 串行输入并行输出移位寄存器的组成	77
5.2.2 并行输入串行输出移位寄存器的组成	78
5.2.3 移位寄存器的 Verilog - HDL 描述	79
5.2.4 移位寄存器的仿真结果	81
5.3 计数器	82
5.3.1 二进制非同步计数器	82
5.3.2 四进制非同步计数器	82
5.3.3 下降沿触发的计数器及 2^N 进制非同步计数器的组成	84
5.3.4 非同步计数器的 Verilog - HDL 描述	85
5.3.5 多层次结构的 Verilog - HDL 设计	87
5.3.6 非同步计数器的仿真结果	88
5.3.7 同步计数器	89
5.3.8 同步计数器的 Verilog - HDL 描述	90
5.3.9 同步任意进制计数器的 Verilog - HDL 描述	91
5.3.10 同步计数器的仿真结果	92

第 6 章 基于 Verilog - HDL 的硬件电路的实现

6.1 硬件系统设计到实现的基本流程	95
6.2 下载电缆的制作	97
6.2.1 Xilinx 下载电缆的连接方法	97
6.2.2 下载接口电路的组成	97
6.2.3 制作中需要注意的事项	99

6.3 JTAG 标准	99
6.3.1 何为 JTAG	99
6.3.2 JTAG 的信号线及功能	100
6.4 Xilinx 公司的 CPLD	100
6.4.1 何为 CPLD	100
6.4.2 XC9500 系列	101
6.5 WebPACK Project Navigator 的使用方法	103
6.5.1 如何将仿真与硬件联系起来	103
6.5.2 WebPACK Project Navigator 编译实例	103
6.5.3 编译结果的报告	114

第 7 章 应用系统设计实例(多功能测试器)

7.1 多功能测试器的制作	116
7.1.1 在硬件开发中提出的问题	116
7.1.2 多功能测试器的设计思想	117
7.1.3 硬件电路的组成	118
7.2 100 MHz 计数器的制作及其在超声波测量中的应用	121
7.3 可编程单脉冲发生器	124
7.3.1 由系统功能描述时序关系	125
7.3.2 流程图的设计	126
7.3.3 系统功能的描述	127
7.3.4 逻辑框图	128
7.3.5 延时模块的描述及仿真	128
7.3.6 功能模块 Verilog - HDL 描述的模块化方法	132
7.3.7 输入检测模块的描述及仿真	133
7.3.8 计数模块的描述	136
7.3.9 可编程单脉冲发生器的系统仿真	137
7.4 可编程单脉冲发生器的硬件实测	141

第 8 章 应用系统设计实例(直接数字频率合成器)

8.1 直接数字频率合成器 DDS	143
8.2 数字式波形生成的基础知识	143
8.2.1 存储器与波形数据	143
8.2.2 波形发生器的系统组成	144
8.2.3 采用 DDS 方式的波形发生器	144
8.2.4 DDS 设计中的参数选择	146
8.3 基于 XC9572 的 DDS 设计	147
8.3.1 基于 XC9572 的 DDS	147
8.3.2 加法器的 Verilog - HDL 描述	149

8.3.3 DDS 的 Verilog - HDL 描述	155
8.3.4 DDS 的仿真结果	158
8.3.5 目标文件的下载与硬件调试	160
8.3.6 基于 VB 的波形数据生成方法	161

第 9 章 Verilog HDL 的系统设计实例(并行接口电路)

9.1 打印口数据传送接口电路的设计	165
9.1.1 打印口接口电路的应用问题	165
9.1.2 微机打印口的基本结构	165
9.1.3 打印口的数据格式	166
9.2 基于打印口的数据传送	167
9.2.1 用打印口实现数据传送的基本方案	167
9.2.2 并行接口电路的系统组成	167
9.2.3 时序设计方法	171
9.2.4 代码分配时应考虑的问题	173
9.3 数据传送电路的 Verilog - HDL 描述	174
9.3.1 译码器的 Verilog - HDL 描述	174
9.3.2 并行接口电路的 Verilog - HDL 描述	174
9.3.3 与 8255 有数据交换情况下的仿真方法	176
9.3.4 并行接口电路读写操作的仿真结果	178
9.4 系统调试方法	181
9.4.1 印刷电路板及其连接电缆	181
9.4.2 并行接口板的安装及系统调试技术	182
9.4.3 并行接口板与微机间的通信	184
9.4.4 DLL 库的生成方法	188

参考文献

第 1 章 硬件描述语言

1.1 什么是硬件描述语言 HDL

简单地说,硬件描述语言 HDL(Hardware Description Language)是一种记述数字电路的功能和结构的语言。硬件描述语言可列出 4 种,如表 1.1 所列,其中最常用的是 VHDL 和 Verilog - HDL。

表 1.1 硬件描述语言 HDL 的种类

种 类	功 能
VHDL	1981 年以美国国际部为中心提出,是最早标准化(IEEE1067)的 HDL,语法丰富且严谨
Verilog - HDL	1985 年由 Gateway Design Automation 公司(现在的 Cadence 公司)开发,1995 年 12 月作为 IEEE 1364-1995 被承认。它类似于 C 语言的语法体系,库文件丰富,已被广泛使用 在日本与美国,该语言的应用要比 VHDL 广泛。但 Verilog - HDL 的描述能力并不像 VHDL 那么高
UDL/I	1990 年起由日本电子工业振兴协会开发,但因没有实用的仿真及合成工具,故未能达到实用化
SFL	由日本 NTT 开发,仅在 NTT 及一部分大学的研究室中使用

1.2 基本逻辑电路的 HDL

本节将通过最简单的例子来讨论 Verilog - HDL 的用法。

1.2.1 与门逻辑电路的描述

1. 模块的定义

硬件描述语言 HDL 一开始所要做的就是模块(module)定义。所谓模块可以理解为是 Verilog - HDL 的基本描述单位。以图 1.1 为例来说明。图 1.1 是一个二与门模块,设其模块名为[AND_G2],输入为[A]和[B],输出为[F]。

2. 模 块

模块的结构如图 1.2 所示,module 与 end-module 总是成对出现的。此外,还有端口参数定义、寄存器定义、线网定义和行为功能调用及定义等。

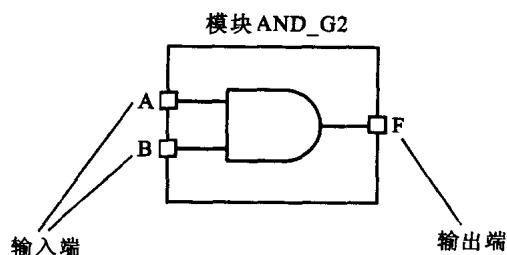


图 1.1 二与门模块定义

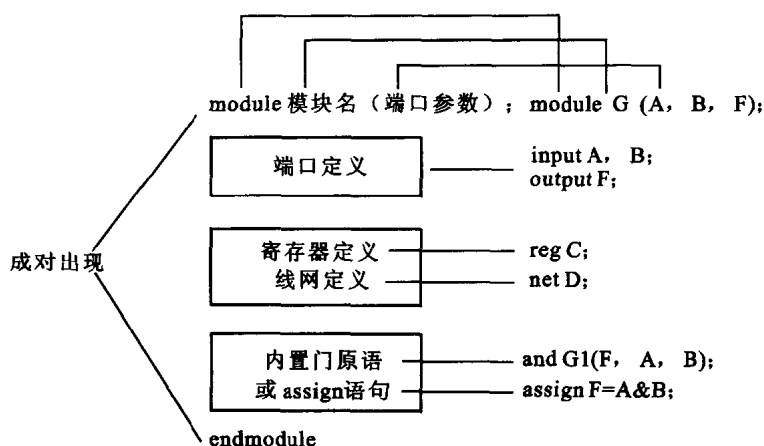


图 1.2 模块的结构

用 Verilog - HDL 来描述图 1.1, 可以有如下两种描述方法:

例 1.1 二与门逻辑电路的描述(expl-1.v, expl-2.v)

```
/* AND_G2 */
module AND_G2 (A, B, F);
    input A, B;           //输入信号定义
    output F;             //输出信号定义

    and U1 (F, A, B);
endmodule
```

或

```
/* AND_G2 */
module AND_G2 (A, B, F);
    input A, B;           //输入信号定义
    output F;             //输出信号定义

    assign F = A & B;
endmodule
```

例 1.1 中的第一个例子称为门级描述方式⁽¹⁾或结构级的建模, 第二个例子称为数据流描述方式或数据流级的建模。对一个逻辑电路, 用硬件描述语言对其描述, 或者说对其用一个模型来描述, 这个过程称为建模。

3. 门级描述方式

例 1.2 给出了一个门级描述方式的 HDL 结构, 其是一个二与门逻辑的实例。

例 1.2 门级描述方式

语句 1: `/* AND_G2 */`

注释

语句 2: `module AND_G2 (A, B, F);`

模块名 端口参数

语句 3: `input A, B;` //输入信号定义

语句 4: `output F;` //输出信号定义

语句 5: `and U2 (F, A, B);`

实例名

语句 6: `endmodule`

语句 1: 注释行。注释语句要写在“/*”和“*/”之间,或在行后加“//”,如语句 3 和语句 4。注释行不被编译,仅起注释作用。

语句 2: 该语句中的“AND_G2”是所定义的模块名,模块名可用下划线“_”,但开始不可使用数字。例如可以写成“AND_G2”,而不能写为“2AND”之类的形式。此外,模块名的字母可以是大写,也可以是小写。例如写为“AND_G2”或“and_g2”都可以,但“AND_G2”与“and_g2”不表示同一模块。

模块名后紧接着的是端口参数,即括号所包含的部分,参数间以逗号“,”来区分。在此,对参数的顺序没有规定,先后自由。在端口参数行的最后要写入分号“;”。要注意在保留字(HDL 中已规定使用的字被称为保留字,如 `module` 即为保留字,本例及全书中的保留字都用小写字母)与模块名之间要留有空格。

语句 3: 描述了入口参数 A 和 B,由保留字 `input` 说明,参数间以逗号“,”区分,行末写入分号“;”,该行的另一种描述形式可写为:

```
input A;
```

```
input B;
```

语句 4: 描述了出口参数 F,以保留字 `output` 说明,行末写入“;”。

端口参数的记述顺序不受限制,即可以是本例中的顺序,也可以是如下顺序:

```
output F;
```

```
input A,B;
```

语句 5: 括号内是二与门的出口及入口参数,由于内置门实例语句规定其顺序必须是(输出,输入,输入)的形式,所以必须写成(F,A,B)的形式,最后以“;”结束该语句。

在门级描述方式中,调用了 Verilog - HDL 所具有的内置门实例语句,例如语句 5 的 `and` 即为内置门实例语句。它调用 AND 逻辑功能。其之后的“U2”称为实例名。实例名在具有行为功能的描述行里也可以省略,即可将语句 5 表现为如下两种形式。注意:内置门实例语句要与实例名之间留有空格。

```
and G1(F,A,B);
```

```
and (F,A,B);
```

语句 6: 结束语句,与语句 2 的 `module` 相呼应,要写成“`endmodule`”的形式。注意:行末不要加写分号“;”。

以上讲述了门级描述方式。比较图 1.2 和例 1.2 的语法结构,即可初步理解端口定义和内置门实例语句的用法。

在 Verilog - HDL 中,属于内置门实例语句的有以下 8 种:

`and, nand, or, nor, not, xor, xnor, buf`。

4. 数据流描述方式

我们还可以用数据流的描述方式来对前述同样功能的逻辑门进行描述,用数据流描述方式对一个逻辑门描述的最基本的方法就是连续使用持续赋值语句。例 1.3 给出了这种例子。

例 1.3 数据流描述方式(exp1-3.v)

```
/* AND_G2 */
module AND_G2 (A, B, F);
    input  A, B;           //输入信号定义
    output F;             //输出信号定义
    assign F = A & B;
endmodule
```

在此,仅有第 5 条语句与门级描述方式不同,在这里是以 assign 来描述电路的逻辑功能的。本例中的输出信号[F]的逻辑表达式是 $[A \cdot B]$,在此,用了位运算符“&”。位运算符的种类及功能有以下几种:

```
~:NOT;
|:OR;
~^:XNOR;
&:AND;
^:XOR。
```

5. Verilog - HDL 的语法总结

(1) 注释要用“/*”与“*/”,或在注释前用“//”。

(2) 标识符(如例 1.2 中的模块名和实例名均属于标识符)可用英文及下划线“_”,标识符的开始不可用数字,对标识符的长度没有限制,大小写文字有区别。

(3) “module”与“endmodule”相呼应,成对出现。其间有端口定义,寄存器定义以及后述的线网定义等。

(4) input 定义输入信号变量。

(5) output 定义输出信号变量。

(6) 在门级描述方式中,调用 Verilog - HDL 具有的内置门实例语句,描述顺序为“(输出,输入 1,输入 2,……);”的形式。注意:输出在前,输入在后。实例名也可省略。

(7) 在数据流的描述方式中,以保留字 assign 和位运算符来描述逻辑表达式。

(8) 最后写入 edmodule,注意行末没有“;”。

(9) 为阅读方便,在本书中的保留字全部用小写,其余均采用大写的形式。

1.2.2 与非门逻辑电路的描述

以二与非门为例,其符号如图 1.3 所示。在此,定义其模块名为[NAND_G2],输入为[A]和[B],输出为[F]。

图 1.3 的门级描述方式和数据流描述方式如例 1.4 所示,位运算的优先顺序如下(由高到低):

~, &, ^, ~^, |

为此,需将“ $F=\dots\dots$ ”写成“ $F=\sim(A \& B);$ ”的形式,如果写成“ $F=\sim A \& B;$ ”的形式,则含义就完全改变。

例 1.4 与非门电路的描述(expl-4.v)

门级描述方式:

```
/* NAND_G2 */
module NAND_G2 (A,B,F);
    input  A,B;
    output F;

    nand U4 (F,A,B);
endmodule
```

数据流描述方式:

```
/* NAND_G2 */
module NAND_G2 (A,B,F);
    input  A,B;
    output F;

    assign F = ~(A & B);
endmodule
```

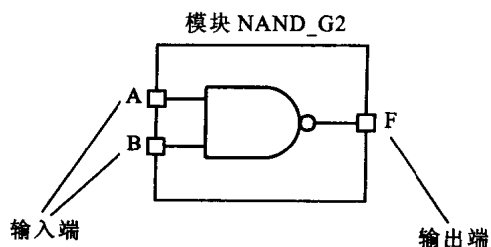


图 1.3 与非门的模块定义

1.2.3 非门逻辑电路的描述

非门的模块定义如图 1.4 所示。在此,定义模块名为[NOT_G],输入为[A],输出为[F]。
例 1.5 为其门级描述方式和数据流描述方式。

例 1.5 非门电路的描述(expl-5.v)

门级描述方式:

```
/* NOT_G */
module NOT_G (A, F);
    input  A;
    output F;
```

```
not U5 (F, A);
endmodule
```

数据流描述方式:

```
/* NOT_G */
module NOT_G (A, F);
    input  A;
    output F;
```

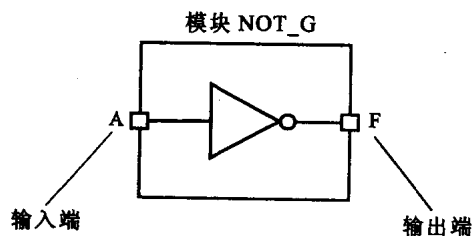


图 1.4 非门的模块定义

```

    assign    F = ~A;
endmodule

```

1.2.4 或门逻辑电路的描述

或门的模块定义如图 1.5 所示。在此,定义其模块名为[OR_G2],输入变量名为[A]和[B],输出变量名为[F]。

对该逻辑电路可用门级和数据流两种方式描述,如例 1.6 所示。与前述的二与门相比,在此只是用到了“或”的逻辑表达“or”。

例 1.6 或门电路的描述(expl-6.v)

门级描述方式:

```

/*    OR_G2    */
module  OR_G2    ( A, B, F );
    input  A, B;
    output F;

```

```

    or    U6    ( F, A, B );
endmodule

```

数据流描述方式:

```

/*    OR_G2    */
module  OR_G2    ( A, B, F );
    input  A, B;
    output F;

    assign    F = A | B;
endmodule

```

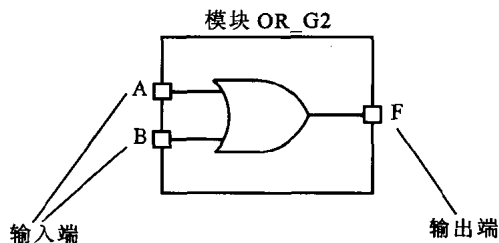


图 1.5 或门的模块定义

1.2.5 或非门逻辑电路的描述

或非门的模块定义如图 1.6 所示。在此,定义其模块名为[NOR_G2],输入为[A]和[B],输出为[F]。例 1.7 为其门级描述方式和数据流描述方式。

例 1.7 或非门电路的描述(expl-7.v)

门级描述方式:

```

/*    NOR_G2    */
module  NOR_G2    ( A, B, F );
    input  A, B;
    output F;

```

```

    nor    U7    ( F, A, B );
endmodule

```

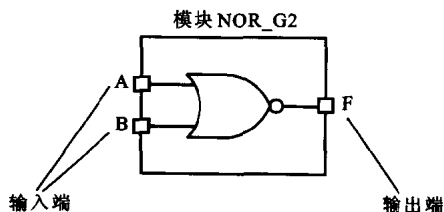


图 1.6 或非门的模块定义

数据流描述方式:

```
/*    NOR_G2    */
module  NOR_G2  ( A, B, F );
    input  A, B;
    output  F;

    assign  F = ~( A | B );
endmodule
```

1.2.6 缓冲器逻辑电路的描述

缓冲器的模块定义如图 1.7 所示。在此,定义其模块名为[BUF_G]。例 1.8 为其门级描述方式和数据流描述方式。

例 1.8 缓冲器(BUF)电路的描述(expl-8.v)

门级描述方式:

```
/*    BUF_G    */
module  BUF_G  ( A, F );
    input  A;
    output  F;

    buf    U8  ( F, A );
endmodule
```

数据流描述方式:

```
/*    BUF_G    */
module  BUF_G  ( A, F );
    input  A;
    output  F;
    assign  F = A;
endmodule
```

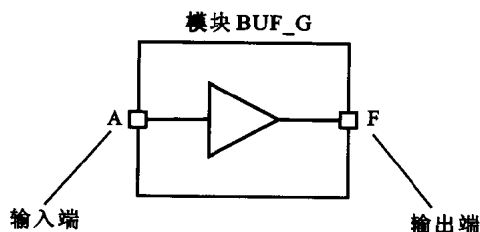


图 1.7 缓冲器的模块定义

1.3 逻辑仿真

假设制作了一个电子设备,例如一个信号放大器。那么,在放大器装完之后一定会想到以下几个问题。这个放大器的放大倍数如何?放大器频响特性如何?放大器瞬态特性如何?放大器的失真度如何?等。为了得到这些数据,需要用各种信号发生装置去测试这个放大器,如图 1.8(a)所示。

同样的道理,某一功能的模块用硬件描述语言写好后,还并不能保证它的完整性,应该用某种方法对其进行测试和验证,如图 1.8(b)所示给出了对硬件描述语言的测试验证方法。与放大器相对应,具有某种功能的硬件描述语言模块要接收来自验证测试程序的信息,然后将输出信息传递给显示模块。