

Verilog-HDL

实用设计与工程制作

刘卫玲 常晓明 编著

Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL Verilog-HDL



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

Verilog-HDL 实用设计与 工程制作

刘卫玲 常晓明 编著

北京航空航天大学出版社

内 容 简 介

本书从实践的角度出发,全面介绍硬件描述语言 Verilog-HDL,通过与具体电路实验的结合,使读者能够轻松地掌握 Verilog-HDL 的语法、结构、功能及简单应用。

全书共分 8 章,第 1~5 章,通过应用 Verilog-HDL 描述的各种逻辑电路实例,详细讲解该语言的语法结构和 FPGA 的开发流程;第 6 章,介绍硬件开发应具备的条件;第 7 章,讲解数字电路系统的设计思路;第 8 章,通过列举 12 个简单的应用实例,详细介绍工程应用系统的设计与实现的全过程。书中给出的全部仿真结果和硬件实现均经过验证。全书的所有 Verilog-HDL 实例文件可在北京航空航天大学出版社网站 www.buaapress.com.cn 的“下载专区”进行下载。

本书可作为学习数字设计的初学者和工程技术人员入门书、工具书和参考资料。

图书在版编目(CIP)数据

Verilog-HDL 实用设计与工程制作 / 刘卫玲,常晓明
编著. -- 北京:北京航空航天大学出版社,2016.5

ISBN 978-7-5124-2117-2

I. ①V… II. ①刘… ②常… III. ①硬件描述语言—
程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 101149 号

版权所有,侵权必究。

Verilog-HDL 实用设计与工程制作

刘卫玲 常晓明 编著

责任编辑 杨 昕

*

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(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

北京泽宇印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:25 字数:533 千字

2016 年 7 月第 1 版 2016 年 7 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2117-2 定价:59.00 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。在我们的生活中,已经几乎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,到处都有数字电子技术的应用。与过去的模拟电子技术相比,数字电子技术具有更广阔的应用前景。

数字电子技术的飞速发展得益于大规模集成电路技术的发展和硬件成本的不断下降。在 20 世纪 70 年代,若要买两只用于制作放大器的功率放大晶体管,大约需要花费 10% 的工资,并且还是分立元件。而现在,即使是买一个成品的功率放大器也十分便宜了。如今,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,大量书籍的问世,海量知识的出现,使得当今的人们有时不知从何处下手来获取知识,甚至被知识的海洋冲昏了头脑。这些年,由于社会对人才的需求不断增加,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,我们也应该注意到:由于存在着学校硬件设施还不能满足需要等问题,使得受教育者仅仅掌握了一些理论知识,而缺乏实践能力。

读书就像吃饭的过程,而实践就像消化的过程。如果学过的东西不付诸实践,那么就不能使学到的知识得到巩固和发展。在学习的过程中,如果能够做到边学习边实践,将会获得更好的学习效果。

正是出于这样一种愿望,作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、理论和特点,而是以动手为主,通过动手实践,体验 Verilog-HDL 的语法、结构和功能等内涵。第 1~5 章,通过应用 Verilog-HDL 描述的各种逻辑电路实例,详细讲解该语言的语法结构和 FPGA 的开发流程;第 6 章,介绍硬件开发应具备的条件;第 7 章,讲解数字电路系统的设计思路;第 8 章,通过列举 12 个简单的应用实例,详细介绍工程应用系统的设计与实现的全过程。书中给出的全部仿真结果和硬件实现均经过验证。作者力图通过几个实际的例子,给读者以启发,使读者能够通过阅读本书增强系统设计和开发的专业技能,提高获取知识的能力。

本书的第 1 章由常晓明负责编写,第 2~8 章由刘卫玲编写。在全书的编写过程中,常晓明教授给予了审核与指导。全书的所有 Verilog-HDL 实例文件可在北京航空航天大学出版社网站 www.buaapress.com.cn 的“下载专区”进行下载,所有例程均通过实验验证。

在本书的编写过程中,得到了北京航空航天大学出版社的大力支持,在此表示衷心的感谢! 晓明研究室的安浩和师夏珍同学对书中的实验做了大量的验证工作,陈璐、张超颖和刘竖威同学也参与了编辑工作,在此,对他们表示诚挚的谢意!

由于作者水平有限,书中难免存在不足和疏漏,恳请广大读者给予批评指正。

作者邮箱:liuweiling0036@link.tyut.edu.cn,网址:<http://www.xiaoming-lab.com>。

作 者

2016 年 3 月

目 录

第 1 章 硬件描述语言	1
1.1 什么是硬件描述语言 HDL	1
1.2 基本逻辑电路的 Verilog-HDL 描述	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	5
1.2.3 非门逻辑电路的描述	6
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	7
1.2.6 缓冲器逻辑电路的描述	8
1.3 逻辑仿真	8
1.3.1 顶层模块的编写	9
1.3.2 寄存器类型定义	10
1.3.3 线网类型定义	10
1.3.4 底层模块的调用	10
1.3.5 输入端口波形的描述	10
1.3.6 二与门逻辑电路的逻辑仿真结果	11
第 2 章 Altera FPGA 开发板及开发流程简介	12
2.1 FPGA 开发板及开发环境	12
2.1.1 FPGA 开发板简介	12
2.1.2 FPGA 开发环境	17
2.2 二与门逻辑电路的开发实例	20
2.2.1 工程文件的建立	20
2.2.2 源文件的建立	23
2.2.3 综合分析	27
2.2.4 ModelSim 仿真	27

目 录

2.2.5	引脚配置	32
2.2.6	编译与下载	32
2.2.7	硬件测试	34
第3章	组合逻辑电路	35
3.1	数据选择器	35
3.1.1	2-1 数据选择器	35
3.1.2	2-1 数据选择器的 Verilog-HDL 描述	36
3.1.3	4-1 数据选择器	37
3.1.4	4-1 数据选择器的 Verilog-HDL 描述	37
3.1.5	条件操作符的使用方法	38
3.1.6	数据选择器的行为描述方式	39
3.1.7	case 语句的使用方法	40
3.1.8	if-else 语句的使用方法	41
3.1.9	function 函数的使用方法	42
3.1.10	用于仿真的顶层模块	42
3.1.11	数据选择器的逻辑仿真结果	43
3.2	数据比较器	44
3.2.1	最简单的数据判断方法	45
3.2.2	2 位数据比较器	45
3.2.3	2 位数据比较器的 Verilog-HDL 描述	48
3.2.4	2 位数据比较器的逻辑仿真结果	49
3.2.5	数据比较器的数据宽度扩展	50
3.2.6	4 位数据比较器的 Verilog-HDL 描述	52
3.2.7	4 位数据比较器的逻辑仿真结果	54
3.3	编码器	55
3.3.1	2 位二进制编码器	55
3.3.2	2 位二进制编码器的 Verilog-HDL 描述	56
3.3.3	2 位二进制编码器的逻辑仿真结果	57
3.4	译码器	58
3.4.1	BCD 码译码器	58
3.4.2	非完全描述的逻辑函数和逻辑表达式的简化	60
3.4.3	BCD 码译码器的 Verilog-HDL 描述	61
3.4.4	BCD 码译码器的逻辑仿真结果	63

第 4 章 触发器	64
4.1 异步 RS 触发器	64
4.1.1 异步 RS 触发器的逻辑符号	64
4.1.2 异步 RS 触发器的 Verilog-HDL 描述	65
4.1.3 异步 RS 触发器的逻辑仿真结果	66
4.1.4 always 块语句的使用方法	67
4.2 同步 RS 触发器	67
4.2.1 同步 RS 触发器的逻辑符号	67
4.2.2 同步 RS 触发器的 Verilog-HDL 描述	68
4.2.3 同步 RS 触发器的逻辑仿真结果	69
4.3 异步 T 触发器	69
4.3.1 异步 T 触发器的逻辑符号	69
4.3.2 异步 T 触发器的 Verilog-HDL 描述	70
4.3.3 异步 T 触发器的逻辑仿真结果	71
4.4 同步 T 触发器	72
4.4.1 同步 T 触发器的逻辑符号	72
4.4.2 同步 T 触发器的 Verilog-HDL 描述	72
4.4.3 同步 T 触发器的逻辑仿真结果	73
4.5 同步 D 触发器	74
4.5.1 同步 D 触发器的逻辑符号	74
4.5.2 同步 D 触发器的 Verilog-HDL 描述	75
4.5.3 同步 D 触发器的逻辑仿真结果	76
4.6 带有复位端的同步 D 触发器	76
4.6.1 带有复位端的同步 D 触发器的逻辑符号	76
4.6.2 带有复位端的同步 D 触发器的 Verilog-HDL 描述	77
4.6.3 带有复位端的同步 D 触发器的逻辑仿真结果	78
4.7 同步 JK 触发器	79
4.7.1 同步 JK 触发器的逻辑符号	79
4.7.2 同步 JK 触发器的 Verilog-HDL 描述	80
4.7.3 同步 JK 触发器的逻辑仿真结果	82
第 5 章 时序逻辑电路	83
5.1 寄存器	83
5.1.1 寄存器的组成原理	83
5.1.2 寄存器的 Verilog-HDL 描述	85

目 录

5.1.3	寄存器的逻辑仿真结果	86
5.2	移位寄存器	86
5.2.1	串行输入/并行输出移位寄存器的组成原理	86
5.2.2	并行输入/串行输出移位寄存器的组成原理	87
5.2.3	移位寄存器的 Verilog-HDL 描述	89
5.2.4	移位寄存器的逻辑仿真结果	91
5.3	计数器	92
5.3.1	二进制非同步计数器	92
5.3.2	四进制非同步计数器	93
5.3.3	下降沿触发型计数器及 2^N 进制非同步计数器的组成原理	94
5.3.4	非同步计数器的 Verilog-HDL 描述	96
5.3.5	多层次结构的 Verilog-HDL 设计	98
5.3.6	非同步计数器的逻辑仿真结果	99
5.3.7	四进制同步计数器	100
5.3.8	四进制同步计数器的 Verilog-HDL 描述	100
5.3.9	任意进制同步计数器的 Verilog-HDL 描述	101
5.3.10	同步计数器的逻辑仿真结果	103
第6章	硬件开发应具备的条件	105
6.1	贴片元件的手工焊接	105
6.1.1	什么是贴片元件?	105
6.1.2	为什么要采用贴片元件?	108
6.1.3	如何进行贴片元件的手工焊接?	108
6.2	一些常用贴片元件的封装	112
6.2.1	贴片电阻	112
6.2.2	贴片电容	113
6.2.3	贴片三极管	115
6.2.4	贴片集成电阻	118
6.2.5	贴片集成电路	118
6.3	硬件开发应具备的工具和材料	122
6.3.1	必备的工具和材料	122
6.3.2	更方便工作的工具和材料	126
6.4	硬件开发应具备的仪器仪表	131
6.4.1	必备的仪器仪表	131
6.4.2	更方便工作的仪器仪表	132
6.5	硬件开发应具备的基本常识	134

6.5.1 常用电路符号的表示方法	134
6.5.2 电子电路的基本单位	135
6.5.3 逻辑门的正确描述法	136
6.5.4 其他知识	137
第7章 数字电路系统的实用设计	139
7.1 简单的可编程单脉冲发生器	139
7.1.1 由系统功能描述时序关系	139
7.1.2 流程图的设计	140
7.1.3 系统功能描述	140
7.1.4 逻辑框图	141
7.1.5 延时模块的详细描述及仿真	142
7.1.6 功能模块 Verilog-HDL 描述的模块化方法	146
7.1.7 输入检测模块的详细描述及仿真	147
7.1.8 计数模块的详细描述	151
7.1.9 可编程单脉冲发生器的系统仿真	151
7.1.10 电路设计中常用的几个有关名词	156
7.2 脉冲计数	162
7.2.1 脉冲计数器的设计	162
7.2.2 parameter 的使用方法	165
7.2.3 repeat 循环语句的使用方法	165
7.2.4 系统函数 \$random 的使用方法	165
7.2.5 特定脉冲序列的发生	166
7.3 脉冲频率的测量	171
7.3.1 脉冲频率测量的原理	172
7.3.2 频率测量模块的设计	172
7.3.3 while 循环语句的使用方法	178
7.4 脉冲周期的测量	178
7.4.1 脉冲周期测量的原理	179
7.4.2 周期测量模块的设计(一)	179
7.4.3 forever 循环语句的使用方法	185
7.4.4 disable 禁止语句的使用方法	185
7.4.5 周期测量模块的设计(二)	186
7.4.6 两种周期测量模块设计的对比	191
7.5 脉冲高电平和低电平持续时间的测量	192
7.5.1 脉冲高电平和低电平持续时间测量的工作原理	192

目 录

7.5.2	高低电平持续时间测量模块的设计	192
7.5.3	改进型高低电平持续时间测量模块的设计	201
7.5.4	begin 声明语句的使用方法	208
7.5.5	initial 语句和 always 语句的使用方法	209
第 8 章	实用设计与工程制作	211
8.1	手脉单脉冲发生器	211
8.1.1	手脉单脉冲发生器的功能描述及系统构建	212
8.1.2	输入检测模块的设计与实现	214
8.1.3	计数模块的设计与实现	219
8.1.4	时标信号发生模块的实现	222
8.1.5	手脉单脉冲发生器的硬件实现	231
8.2	手脉脉冲串发生器	234
8.2.1	手脉脉冲串发生器的功能描述及系统构建	234
8.2.2	反馈模块的设计与实现	236
8.2.3	手脉脉冲串发生器的硬件实现	239
8.3	手脉有效沿和转向识别	242
8.3.1	手脉有效沿和转向识别模块的功能描述	242
8.3.2	手脉有效沿和转向识别模块的设计与仿真	243
8.3.3	手脉有效沿和转向识别模块的硬件实现	246
8.4	手脉脉冲串计数器	247
8.4.1	手脉脉冲串计数器的功能描述及系统构建	247
8.4.2	计数模块的设计与仿真实现	249
8.4.3	手脉脉冲串计数器的仿真实现	252
8.5	具有 LCD 显示单元的手脉脉冲串计数器	254
8.5.1	LCD 显示单元的工作原理	254
8.5.2	系统硬件实现	257
8.6	频率可调的方波发生器	267
8.6.1	频率可调的方波发生器的功能描述及系统构建	267
8.6.2	分频模块的设计与实现	270
8.6.3	频率可调的方波发生器的 Verilog-HDL 描述	274
8.6.4	频率可调的方波发生器的硬件实现	276
8.7	脉宽可调的方波发生器	276
8.7.1	脉宽可调的方波发生器的功能描述及系统构建	276
8.7.2	高电平持续时间调节模块的设计与实现	279
8.7.3	PWM 发生器的 Verilog-HDL 描述	283

8.7.4 PWM 发生器的硬件实现	285
8.8 电动窗帘的控制	285
8.8.1 电动窗帘控制系统的设计原理	285
8.8.2 FPGA 控制电机驱动系统的仿真实现	288
8.8.3 FPGA 控制电机驱动系统的硬件测试	289
8.9 基于 FPGA - IP 核的正弦波发生器	290
8.9.1 系统设计与时序分析	290
8.9.2 分频模块的详细描述	292
8.9.3 寻址模块的详细描述	292
8.9.4 数据存储模块的详细描述	293
8.9.5 正弦波发生器的 Verilog-HDL 描述	302
8.9.6 正弦波发生器的硬件实现	302
8.10 具有数码管显示单元的 A/D 转换系统	304
8.10.1 A/D 转换系统的功能描述	304
8.10.2 A/D 采样时钟发生模块	305
8.10.3 数码管显示模块	306
8.10.4 A/D 转换系统的 Verilog-HDL 描述	313
8.10.5 A/D 转换系统的硬件实现	314
8.11 串口通信	316
8.11.1 串口接收模块的设计与实现	316
8.11.2 串口发送模块的设计与实现	343
8.11.3 串口通信的硬件实现	360
8.12 磁致伸缩位移传感器数据采集系统的应用设计与开发	362
8.12.1 磁致伸缩位移传感器数据采集系统的构建	362
8.12.2 1 μ s 单脉冲输出模块的设计与实现	366
8.12.3 信号处理模块的设计与实现	370
8.12.4 串口发送部分的设计与实现	376
8.12.5 系统集成及功能实现	385
参考文献	388

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。硬件描述语言至少可列出 4 种,如表 1.1.1 所列,其中最常用的是 VHDL 和 Verilog-HDL。

表 1.1.1 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的,是最早的标准化(IEEE 1067)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—1955 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 那么强
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 Verilog-HDL 描述

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 1.2.1 为例来说明,它是一个二与门逻辑电路的模块定义,设其模块名为“AND_G2”,输入为“A”和“B”,输出为“F”。

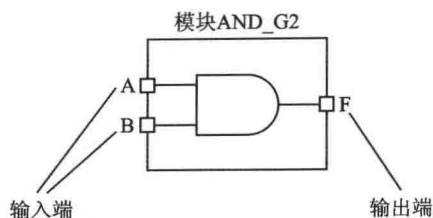


图 1.2.1 二与门逻辑电路的模块定义

2. 模块的描述

模块的结构如图 1.2.2 所示, module 与 endmodule 总是成对出现。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。

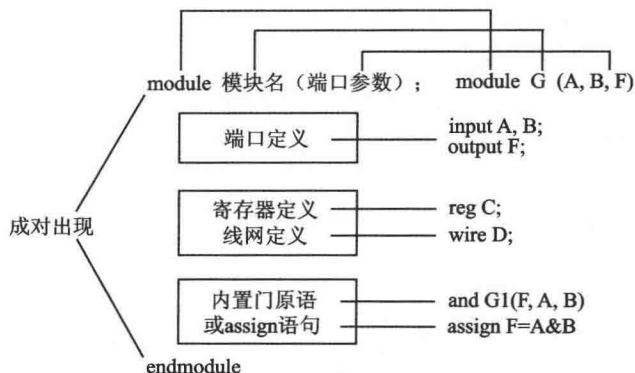


图 1.2.2 模块的结构

用 Verilog-HDL 来描述图 1.2.1, 可以有如下两种描述方法 (在例 1.2.1 中):

【例 1.2.1】 二与门逻辑电路的描述 (exp1-2-01.v, exp1-2-02.v)

方法一:

```
/* AND_G2 */
module AND_G2 (A, B, F); //模块名 AND_G2 及端口参数定义,范围至 endmodule
    input A, B; //输入端口定义
    output F; //输出端口定义

    and U1 (F, A, B); //门实例语句,此行实现输出为 F,输入为 A 和 B 的二
                    //输入与门

endmodule //模块 AND_G2 结束
```

方法二:

```
/* AND_G2 */
module AND_G2 (A, B, F); //模块名 AND_G2 及端口参数定义,范围至 endmodule
    input A, B; //输入端口定义
```

```

output    F;                                //输出端口定义
assign    F = A & B;                        //assign 语句,此行实现二输入与门的功能
endmodule                                   //模块 AND_G2 结束

```

例 1.2.1 中的方法一称为门级描述方式或结构级的建模,方法二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 1.2.2 给出了一个门级描述方式的 HDL 结构,这是一个二与门逻辑的实例。

【例 1.2.2】 门级描述方式

```

语句 1: /* AND_G2 */
语句 2: module AND_G2 (A, B, F);
语句 3: input A, B;           //输入端口定义
语句 4: output F;            //输出端口定义
语句 5: and U2 (F, A, B);
语句 6: endmodule

```

语句说明:

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,仅起注释的作用。

语句 2: 该语句中的“AND_G2”是所定义的模块名,模块名可用下画线“_”,但开始不可使用数字,如可以写为“AND_G2”,而不能写为“2AND”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”和“and_g2”表示的不是同一模块。

模块名后紧跟的是端口参数,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“;”。

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: 描述了输入端口 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: 描述了输出端口 F,以保留字 **output** 说明,行末写入“;”。

端口参数的书写顺序不受限制,既可以是本例中的顺序,也可以是如下顺序:

```

output F;
input A, B;

```

第1章 硬件描述语言

语句 5: 括号内是二与门的输出端口及输入端口参数, 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: 内置门实例语句与实例名之间要留有空格。

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

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

语句 6: 结束语句, 与语句 1 的 **module** 相呼应, 要写为 **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 1.2.2 和例 1.2.2 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

还可以用数据流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连续赋值语句, 即 **assign** 语句。例 1.2.3 给出了这种例子。

【例 1.2.3】 数据流描述方式

```
/*    AND_G2    */
module AND_G2 (A, B, F);    //模块名 AND_G2 及端口参数定义, 范围至 endmodule
    input  A, B;            //输入端口定义
    output F;               //输出端口定义

    assign F = A & B;        //assign 语句, 此行实现二输入与门的功能
endmodule                  //模块 AND_G2 结束
```

本例中, 仅有第 5 条语句与门级描述方式不同, 这里是用 **assign** 语句来描述电路的逻辑功能, 即输出信号“F”的逻辑表达式为“ $A \cdot B$ ”。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“ * /”,或在注释前加“//”。
- ② 标识符(如例 1.2.2 中的模块名和实例名均属于标识符)可用英文及下画线“_”,标识符的开始不可用数字,对标识符的长度没有限制,字母大小写有区别。
- ③ **module** 与 **endmodule** 互相呼应,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** 定义输入端口。
- ⑤ **output** 定义输出端口。
- 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 在数据流描述形式中,以保留字 **assign** 和位运算符来描述逻辑表达式。
- ⑧ 最后写入 **endmodule**。**注意:**行末没有“;”。
- ⑨ 为阅读方便,在本书中保留字全部用小写加粗的形式,其余均采用大写的形式。

1.2.2 与非门逻辑电路的描述

以二与非门逻辑电路为例,其模块定义如图 1.2.3 所示。在此,定义模块名为“NAND_G2”,输入为“A”和“B”,输出为“F”。

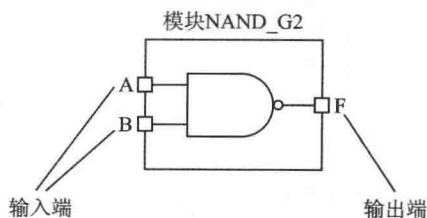


图 1.2.3 与非门逻辑电路的模块定义

图 1.2.3 的门级描述方式和数据流描述方式如例 1.2.4 所示,位运算的优先顺序如下(由高到低): $\sim$, $\&$, $\wedge$, $\sim\wedge$, $|$ 。

例如:“ $F = \sim(A \& B)$;”与“ $F = \sim A \& B$;”的含义是完全不同的。

【例 1.2.4】 与非门逻辑电路的描述(expl-2-03.v,expl-2-04.v)

门级描述方式:

```
/*    NAND_G2    */
module NAND_G2  ( A, B, F );    //模块名 NAND_G2 及端口参数定义,范围至 endmodule
    input  A, B;                //输入端口定义
    output F;                   //输出端口定义

    nand    U4  ( F, A, B );     //门实例语句,此行实现输出为 F,输入为 A 和 B 的二
                                //输入与非门

endmodule                      //模块 NAND_G2 结束
```

数据流描述方式:

```
/*    NAND_G2    */
```