
单片机与嵌入式系统丛书
http://www.phei.com.cn
VHDL
数字控制系统
设计范例
● 林明权　等编著
● 马维旻　改编
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

单片机与嵌入式系统丛书

VHDL 数字控制系统设计范例

林明权　等编著
马维旻　　改编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书在简要介绍 VHDL 的语法和基本数字逻辑电路设计技巧的基础之上，完整地给出了七个较为复杂的数字控制系统设计范例，包括自动售货机、电子钟、红绿灯交通信号系统、步进电机定位控制系统、直流电机速度控制系统、计算器以及点阵列 LED 显示控制系统。通过学习这些典型的实例，读者可以学会自己利用 VHDL 设计实用的数字控制系统。

本书适合从事数字控制系统设计的技术人员和高校相关专业的师生阅读。

中文繁体字版书名：《数位控制系统设计——使用 VHDL》
ISBN：957-21-3342-X，林明权等编著，2001 年 10 月

本书中文简体字版由台湾全华科技图书股份有限公司独家授权，仅限于中国大陆地区出版发行。

版权贸易合同登记号：01-2002-2744

图书在版编目(CIP)数据

VHDL 数字控制系统设计范例/林明权等编著；马维旻改编. —北京：电子工业出版社，2003.1
(单片机与嵌入式系统丛书)
ISBN 7-5053-8386-8

Ⅰ.V... Ⅱ.①林... ②马... 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 104591 号

责任编辑：赵丽松 邓小瑜
印 刷：北京李史山胶印厂
出版发行：电子工业出版社 http://www.phei.com.cn
北京市海淀区万寿路 173 信箱 邮编 100036
经 销：各地新华书店
开 本：787×980 1/16 印张：18.5 字数：373 千字 附光盘：1 张
版 次：2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷
印 数：5 000 册 定价：32.00 元(含光盘)

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010)68279077

前　言

电子技术的发展，特别是专用集成电路（ASIC）设计技术的日趋进步和完善，推动了数字系统设计的迅猛发展。电子设计自动化（EDA）工具给电子设计带来了巨大变革，尤其是硬件描述语言的出现和发展，解决了传统用电路原理图设计大系统工程时的诸多不便，成为电子电路设计人员的最得力助手。学习 VHDL 已日益成为我国高校学生和工程技术人员的迫切需求。

本书是在台湾繁体版图书《数位控制系统设计——使用 VHDL》的基础上，结合当前集成电路设计领域的新内容和典型实例改编而成的。既简要地介绍了 VHDL 的语法，又以基本逻辑电路为例，阐述 VHDL 的设计技巧。随后列举了七个数字控制系统设计的完整范例，包括自动售货机、电子钟、红绿灯交通信号系统、步进电机定位控制系统、直流电机速度控制系统、计算器以及点阵列 LED 显示控制系统。目的是使读者通过学习这些典型实例的设计过程，循序渐进地掌握 VHDL 应用在复杂数字控制系统设计中的要领。本书所附的光盘中收录了书中列举的程序代码，供读者学习使用。

本书在改编过程中得到魏永昌等老师的大力帮助，在此表示感谢。由于理论水平和实践经验有限，书中难免存在错误或不妥之处，敬请读者批评指正。

改编者

序　言

由于 CPLD/FPGA 等 IC 硬件与 VHDL/Verilog 等硬件描述语言的发展，传统数字逻辑与数字系统设计的课程应全面简化与调整。学生不必再将大量时间用于逻辑函数的推导以及烦琐的卡诺图化简上，因为 VHDL 与 Verilog 等硬件描述语言（HDL，Hardware Description Language）能帮你完成全部工作。学生只需具有基本逻辑概念与时间序列分析概念即可。既不必去考虑各式各样的 TTL IC，也不用拿着数据手册按照引脚图在印刷电路板上一根一根地画线。这些简单劳动，Xilinx/Altera 公司的 EDA 软件环境都会在 CPLD/FPGA 芯片上帮用户完成，而且随时可更改设计。这两大 CPLD/FPGA 厂商免费提供 EDA 软件设计环境，学生可以很容易地利用它们开发出数字控制集成电路。

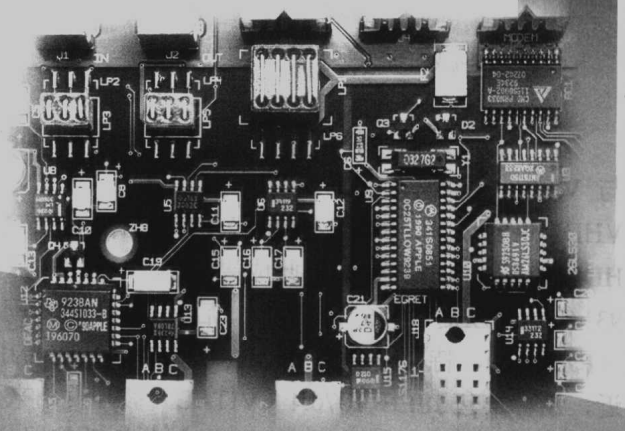
本书采用个案讨论的方式，每一章节介绍一个完整的数字控制系统 IC。一边引进各个数字控制电路的内部构造，一边将 VHDL 语法渐进地导入，让读者在完成各个实际电路设计的同时，自然地掌握 VHDL 语言的设计要点。一般 CPLD/FPGA 的应用有两类，一是用做下线制作 ASIC 前的 IC 雏形制作与验证，从而节省下线制作的验证时间与投资风险，此时电路的设计完全以 VHDL 或 Verilog 语言的形式编写；二是直接将电路以最佳化形式置入 CPLD/FPGA 内，以缩短产品上市时间，抢占产品上市初期的高利润，此时则可以将 HDL 与 Schematic 两种设计方式并用，以取得面积与速度功能的最优化，在本书各章的范例中，两种方式兼而有之。

每一章的完整范例都以 ASIC 格式（*.vhd）存在本书所附光盘中，而且均以 Xilinx Foundation 3.1i 进行编辑、合成与仿真。

本校二技部的陈博玮与方侨立同学对本书所有范例的软件仿真与验证以及硬件下载测试与验证做了大量的工作，同时他们还对文稿的校对付出了辛勤的劳动。在此一并表示感谢。

台湾昆山科技大学电子系

目　录



第1章

VHDL语法概要

VHDL

System Design Of Digital control

1.1 概述

目前最通用的硬件描述语言（HDL，Hardware Description Language）有 VHDL 与 Verilog 两种。1982 年美国国防部的一个分支专案要求所有的数字电路必须用 VHDL 语言设计。1983 年 IBM 及 TI 等公司在此专案规划下，开始开发 VHDL（VLSI HDL）。随后 IEEE 也认可此成果为 1076 号 IEEE 标准（1987），1993 年又修订为 93 年版 IEEE 1076 号标准。后来将一种可配合集成工具的 VHDL 程序包（Package），特别命名为 IEEE 1076.3 ，并成为 1076 号 IEEE 标准的一部分。最近，新标准包 IEEE 1076.4（VITAL）被开发出来，成为建立 ASIC 及 FPGA 的模型函数库。VHDL 的设计层面可以划分为系统层（System）、算法层（Algorithm）、寄存器传输层（Register-Transfer）、逻辑层（Logic）以及电路层（Circuit）。另一支 HDL 语言的主流是 Verilog，其建模能力可以涵盖所有范围，但不在本书论述范围内。Synopsys 公司所提供的 FPGA Express 集成软件同时支持 VHDL 与 Verilog 语言的 PC 仿真与集成环境，非常适合学生使用。

1.2 语言特性

VHDL 语言可描述一个数字电路的输入、输出以及相互间的行为与功能。而其硬件关联性的语法与形式虽类似于一般程序语言，但是涵盖许多与硬件关联的语法构造。其特有的层次性——由上而下的结构式语法结构适合大型设计项目的团队合作。在主要的系统结构、组件及相互间的连接方式决定以后，就能将工作分包下去，各自独立进行，例如使用主程序外的组件（Component）、函数（Function）以及程序内的块（Block）程序。从抽象的层次而言，VHDL 的语句分成以下 4 个大类。

1. 行为式（Behavior）

采用语言逻辑方式直接描述硬件电路的工作，表示一个设计的功能或算法，描述 IC 内部电路行为。在结构定义中可以同时包含并行描述与顺序语句。

2. 数据流（Data flow）

从数据输入与输出的观点，大部分的并行语句都用于数据转换工作。

3．结构式（Structural）

允许设计者以树状形式调用内置电路组件。通常以引脚图方式调用并连接。从硬件的角度说，调用组件就像在组合与连接电路元器件一样。

4．寄存器传输式（Register transfer logic）

VHDL 是一种类型化的语言，一种数据类型的数据内容不能指定给其他类型的数据，而且不同数据类型的数据需经过转换才能相互运算。每一个电路的 VHDL 码都是实体（Entity）与结构（Architecture）的成对组合，先用实体来定义一个 IC 电路引脚规格及基本参数，然后再用结构定义 IC 内部电路的功能运作，即构成一个完整的电路模块。

1.3　VHDL 语法规则

1.3.1　标识符（Identifiers）

（1）标识符中允许的合法字符仅包含 26 个英文大、小写字母及下划线。

（2）标识符的第一个字符必须是英文字母。

（3）“_”不能是标识符的最后一个字符。在标识符中不能同时有 2 个下划线相连。

（4）标识符中的英文字母不分大小写。

（5）VHDL 程序中的说明文字开头一律为 2 个连续的连接线“--”。可以出现在任一语句后面，也可以出现在独立行。

（6）保留字（Reserved words）或关键词不能用做标识符。

1.3.2　数据对象（Data objects）

1．信号（Signal）

信号定义的格式如下：

signal 信号名 [，信号名…]：数据类型 [：=表达式]；

信号包括输出/输入引脚信号以及 IC 内部缓冲信号，有硬件电路与之相对应，故信号之间的传递有实际的附加延时。在实体中定义为外接输出/输入引脚信号，在 Architecture 语句与 begin 语句之间定义为全局（Global）信号。信号与变量的形式几乎相同，只有一项重要的差异：信号可以用来存储或传递逻辑值，因此可被集成为存储器件或数据总线，而变量则不能。信号能在不同进程之间传递，变

量则不能。

2. 变量（Variable）

变量定义的格式如下：

variable 变量名 [, 变量名…]: 数据类型 [: =表达式];

变量并不对应到 IC 的任何输出/输入引脚，而只是用来作为指针或存储程序中计算用的暂时值，故变量之间的传递是瞬时的，并无实际的附加延时。变量仅能出现在 process，if_loop，function 等语句中，用于暂时运算，其值仅局部（Local）有效，要先指定给信号，才能将其最终值传出进程。

变量可以在定义时即赋初始值，也可在程序任何合法的地方重新赋值。

3. 常数（Constant）

常数定义的格式如下：

constant 常数名 [, 常数名…]: 数据类型 : =表达式;

常数一旦被赋值，在整个程序中将不再改变，例如：

```
constant bus_width : integer : =8;
```

此例定义 bus_width 为一个整型数常量，值为 8。

4. 计数指针（Index）

计数指针本身是一种整数类型，但不对应任何信号，只是在程序中用做计数器。故不需说明数据类型（正整数），如下面例子中的指针 I。

```
for I in 7 downto 0 loop
  reg(I) <= "00000000";
end loop;
```

但如果用于运算则应加以说明。

```
process
  variable I:integer;
  begin
  I:=0;
  while I<8 loop
      reg(I) <= "00000000";
      I := I +1;
  end loop;
end process;
```

5. 数据对象定义的例外

VHDL 中大部分的数据类型必须事先说明，才能使用，而且可以用“:=”运算符赋初值。但是有些数据例外，如：

（1）IC 器件的输出/输入引脚已隐含定义为信号，不用再说明。

（2）IC 器件的参数值（Generics）已隐含定义为常数，不用再说明。

（3）格式化的函数（Function）参数必须是常数或信号，而且已在函数说明中隐含定义了。

（4）循环（Loop）或生成（Generate）语句开始时，其指针（Index）的数据类型已隐含定义了，而在语句结束后，则自动消失。

6. 属性（Attribute）

属性用来表示信号的某种特性。例如：

```
signal A : std_logic_vector(3 to 12);
```

则有

```
A'left     = 3
A'right    = 12
A'high     = 12
A'low      = 3
A'length   = 10
A'range    = 3 to 12
```

“event”属性表示信号内涵改变。例如：

```
if (clk'event and clk='1')
```

该语句表示假如 clk 信号发生变化且现值为 1，即 clk 信号发生上升沿触发，亦可改写成：

```
if rising_edge(clk)
```

1.3.3　数据类型（Data types）

1. 预定义的数据类型（Pre-defined types）

1）位（bit）

位允许的数值为 0 或 1。

2）布尔量（boolean）

布尔量只有 true 与 false 两种状态。

3）位矢量（bit-vector）

位矢量由一串位组成，可以按递增或递减方式排列，用来表示总线的状态。

例 1：

```
signal a, b: bit_vector(0 to 3);
```

此例说明 a 与 b 为 4 位的数据总线。

例 2：

```
signal c: bit_vector(3 downto 0): = "1100";
```

此例以递减方式定义：c(3)= '1'，c(2)= '1'，c(1)= '0'，c(0)= '0'。在 VHDL 中，单一位用单引号（' '）标明，而位矢量的常数则用双引号（" "）标出。若程序中赋值语句为 c <= O"8"，则表示 c="001000"，而 O 代表八进制表示。此时在定义信号 c 时，必为 6 位的位矢量，否则会出现错误。若程序中赋值语句为 a <= X "34"，则表示 a="00110100"，而 X 代表十六进制表示，此时在定义信号 a 时，必为 8 位的位矢量，否则会出现错误。

4）字符（character）

字符允许的数据内容为 128 个标准 ASCII 字符。

5）字符串（string）

字符串由一串字符构成。例如：

```
signal ksut: string(1 to 9):=Kung-Shan;
```

此例定义字符串 ksut= Kung-Shan，其中 ksut(1)=K，ksut(9)=n。

6）标准逻辑（std_logic）

程序包（Packages）std_logic_1164 允许标准逻辑的数据内容为'u'，'x'，'0'，'1'，'Z'，'w'，'L'，'H'或'_'。然而在实际 IC 集成时，只允许'0'，'1'，'Z'和'_'4 种数据内容。'0'与'1'出现在程序中的任何地方，不受限制。'_'表示不关心输出，不能用在 if_then_else 和 case 语句中。一个矢量的最左位被认定为最高位（MSB）。IEEE 1164 中定义'Z'为高阻输出。

7）标准逻辑矢量（std_logic_vector）

标准逻辑矢量跟位矢量几乎相同，只是数据内容多了'Z'与'_'两种。'Z'表示电路的高阻状态，仅能指定给最外层电路输出，并且必须对应到 IC 的实际引脚上。

例 1：

```
signal bus: std_logic_vector(7 downto 0);
begin
bus <= "ZZZZZZZZ"
```

此例说明并指定输出总线 bus 全为高阻状态。'Z'可以指定给变量，而且可用在比较中，如：if a = 'Z' then …。'Z'亦可以放在函数中调用。