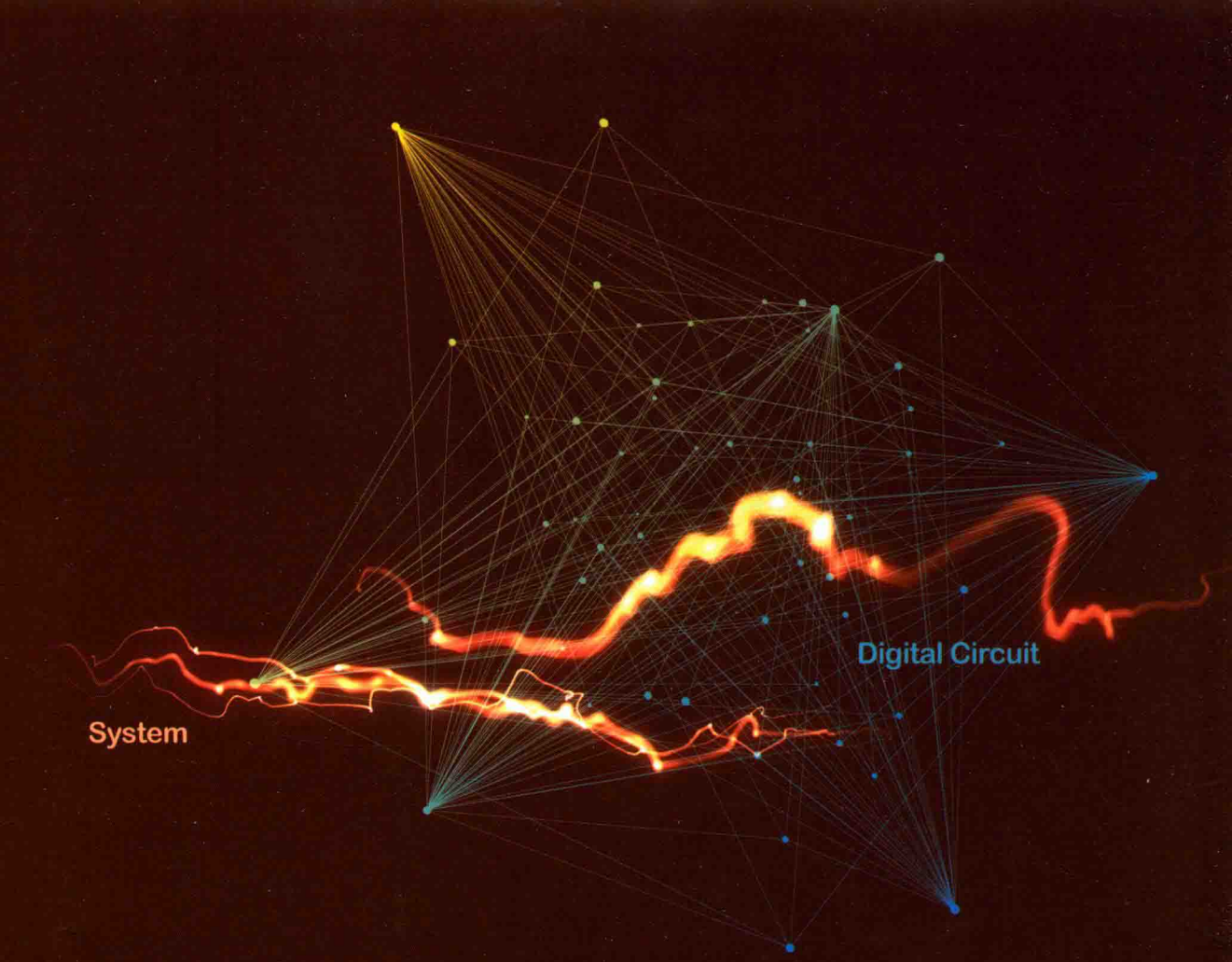



# 现代数字电路与系统设计（VHDL版）

◎ 江国强　编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

# 现代数字电路与系统设计

## （VHDL 版）

江国强　编著

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

## 内 容 简 介

本书是基于电子设计自动化（EDA）技术编写的，全书共 8 章，包括 VHDL、门电路的设计、组合逻辑电路的设计、触发器的设计、时序逻辑电路的设计、存储器的设计、数字系统的设计和常用 EDA 软件。数字电路与系统设计都是基于 VHDL 完成的，每个设计都经过了 EDA 软件的编译和仿真，或经过 EDA 实验开发系统平台的验证，确保无误。

本书图文并茂、通俗易懂，可作为高等学校工科相关专业数字逻辑电路、EDA 技术与应用、可编程逻辑器件等课程的教学参考书，或课程设计、实训和毕业设计的参考书，也可作为从事数字电路与系统设计的工程技术人员的参考书。



**图书在版编目（CIP）数据**

现代数字电路与系统设计：VHDL 版/江国强编著. —北京：电子工业出版社，2018.2
ISBN 978-7-121-33384-2
Ⅰ. ①现… Ⅱ. ①江… Ⅲ. ①数字电路－系统设计 Ⅳ. ①TN79

中国版本图书馆 CIP 数据核字（2017）第 325743 号

责任编辑：韩同平　　特约编辑：邹凤麒　王博　段丹辉
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　1/16　印张：17　字数：544 千字
版　　次：2018 年 2 月第 1 版
印　　次：2018 年 2 月第 1 次印刷
定　　价：49.90 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：hantp@phei.com.cn。

# 前　言

在 20 世纪 90 年代，国际上电子和计算机技术先进的国家，一直在积极探索新的电子电路设计方法和设计工具，并取得巨大成功。在电子设计技术领域，可编程逻辑器件 PLD（Programmable Logic Device）的应用，已得到很好的普及，这些器件为数字系统的设计带来极大的灵活性。该器件可以通过软件编程而对其硬件结构和工作方式进行重构，使得硬件的设计可以如同软件设计那样方便快捷，极大地改变了传统的数字系统设计方法、设计过程和设计观念。随着可编程逻辑器件集成规模不断扩大、自身功能不断完善，以及计算机辅助设计技术的提高，使现代电子系统设计领域的电子设计自动化 EDA（Electronic Design Automation）技术应运而生。传统的数字电路设计模式，如利用卡诺图的逻辑化简手段、布尔方程表达式设计方法和相应的中小规模集成电路的堆砌技术正在迅速地退出历史舞台。

本书是基于硬件描述语言 HDL（Hardware Description Language）编写的。目前，国际最流行的、并成为（美国）电机及电子工程师学会 IEEE（Institute of Electrical and Electronics Engineers）标准的两种硬件描述语言是 VHDL 和 Verilog HDL，两种 HDL 各具特色。VHDL 是超高速集成电路硬件描述语言（Very High Speed Integrated Circuit Hardware Description Language）的缩写，在美国国防部的支持下于 1985 年正式推出，是目前标准化程度最高的硬件描述语言。VHDL 经过 30 多年的发展、应用和完善，以其强大的系统描述能力、规范的程序设计结构、灵活的语言表达风格和多层次的仿真测试手段，在电子设计领域受到了普遍的认同和广泛的接受，成为现代 EDA 领域的首选硬件描述语言。本书以 VHDL 作为数字电路与系统的设计工具。

本书共 8 章，首先介绍 VHDL，然后介绍基于 VHDL 的常用数字电路和一些专用数字电路的设计。所谓常用数字电路是指用途比较广泛并形成集成电路产品的电路，例如 TTL 系列和 CMOS 系列的集成电路产品。专用数字电路是指具有特定功能的电路，例如序列序号发生器、序列序号检测器等，但它们没有现成的集成电路产品。另外，还介绍了一些通俗易懂的数字系统设计和一些常用的 EDA 软件。

第 1 章 VHDL，介绍 VHDL 的语法规则、语句和仿真方法，为基于 VHDL 的数字电路及系统的设计打下基础。

第 2 章门电路的设计，介绍普通门、三态输出门和三态驱动门的设计。

第 3 章组合逻辑电路的设计，介绍算术运算电路、编码器、译码器、数据选择器、数据比较器、奇偶校验器和码转换器等组合逻辑电路的设计。

第 4 章触发器的设计，介绍基本 RS 触发器、钟控 RS 触发器、D 触发器和 JK 触发器的设计。

第 5 章时序逻辑电路的设计，介绍数码寄存器、移位寄存器和计数器等常用时序逻辑电路的设计，还介绍顺序脉冲发生器、序列序号发生器、伪随机信号发生器、序列序号检测器、码转换器和串行数据检测器等专用数字电路的设计。

第 6 章存储器的设计，介绍只读存储器 ROM 和随机存储器 RAM 的设计。

第 7 章数字系统设计，首先介绍数字系统的设计方法，然后介绍串行加法器、24 小时计时器、万年历、倒计时器、交通灯控制器、出租车计费器、波形发生器、数字电压表和数字频率计等系统电路的设计。

第 8 章常用 EDA 软件，介绍 Quartus II 13.0、ModelSim、Matlab/DSP Builder 和 Nios II 等常用的 EDA 软件，供读者在进行数字电路及系统设计时参考。

本书中的所有 VHDL 程序都经过美国 Altera 公司的 Quartus II 软件的编译和仿真，或经过 EDA 实验开发系统平台验证，确保无误。为了使读者看清楚仿真结果，大部分设计的仿真结果是用 Quartus II 9.0 版本软件中的自带仿真工具（Waveform Editor）或 Quartus II 13.0 版本软件中的大学计划仿真工具（university program vwf）实现的。

本书由桂林电子科技大学江国强教授编著，如有不足之处，恳请读者指正。

E-mail：hmjgq@guet.edu.cn
地 址：桂林电子科技大学（541004）
电 话：（0773）5601095，13977393225

编著者

# 目　录

# 第 1 章　VHDL

本章介绍硬件描述语言 VHDL 的语言要素、程序结构及描述风格，并介绍最基本、最典型的数字逻辑电路的 VHDL 描述，作为 VHDL 工程设计的基础。

## 1.1　VHDL 设计实体的基本结构

一个完整的 VHDL 程序，或者说设计实体，是指能被 VHDL 综合器接受，并能作为一个独立的设计单元，即以元件形式存在的 VHDL 程序。这里所谓的“综合”，是将给定电路应实现的功能和实现此电路的约束条件（如速度、功耗、成本及电路类型等），通过计算机的优化处理，获得一个满足上述要求的设计方案。简单地说，“综合”就是依靠 EDA 工具软件，自动完成电路设计的整个过程。因此，VHDL 程序设计必须完全适应 VHDL 综合器的要求，使 VHDL 程序能够在 PLD 或专用集成电路 ASIC（Application Specific Integrated Circuit）中得到硬件实现。这里所谓的“元件”，既可以被高层次的系统调用，成为系统的一部分，也可以作为一个电路的功能块，独立存在和独立运行。

VHDL 设计实体的基本结构如图 1.1 所示。它由库（LIBRARY）、程序包（PACKAGE）、实体（ENTITY）、结构体（ARCHITECTURE）和配置（CONFIGURATION）等部分构成。其中，实体和结构体是设计实体的基本组成部分，它们可以构成最基本的 VHDL 程序。

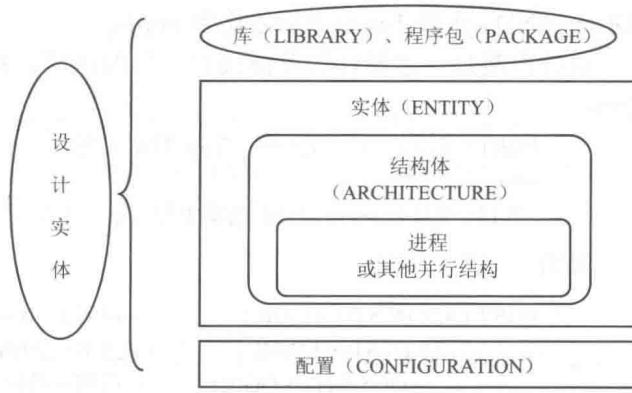



图 1.1　VHDL 设计实体的基本结构图

### 1.1.1　库、程序包

IEEE 于 1987 年和 1993 年先后公布了 VHDL 的 IEEE STD 1076-1987（即 VHDL 1987）、IEEE STD 1076-1993（即 VHDL 1993）和 IEEE STD 1076-2008（即 VHDL 2008）语法标准。根据 VHDL 语法规则，在 VHDL 程序中使用的文字、数据对象、数据类型都需要预先定义。为了方便用 VHDL 编程，IEEE 将预定义的数据类型、元件调用声明（Declaration）及一些常用子程序收集在一起，形成程序包，供 VHDL 设计实体共享和调用。若干个程序包则形成库，常用的库是 IEEE 标准库。因此，在每个设计实体开始都有打开库和程序包的语句。例如，语句：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
```

表示设计实体中被描述器件的输入/输出端口和数据类型将要用到 IEEE 标准库中的 STD_LOGIC_1164 程序包。

### 1.1.2 实体

实体（ENTITY）是设计实体中的重要组成部分，是一个完整的、独立的语言模块。它相当于电路中的一个器件或电路原理图上的一个元件符号。实体由实体声明部分和结构体组成。实体声明部分指定了设计单元的输入/输出端口或引脚，它是设计实体对外的一个通信界面，是外界可以看到的部分。结构体用来描述设计实体的逻辑结构和逻辑功能，它由 VHDL 语句构成，是外界看不到的部分。一个实体可以拥有一个或多个结构体。

实体声明部分的语句格式为（语句后面用“--”引导的是注释信息）：

```
ENTITY 实体名 IS
        GENERIC(类属表);        --类属参数声明
        PORT(端口表);           --端口声明
END     实体名;
```

其中，类属参数声明必须放在端口声明之前，用于指定如矢量位数、器件延迟时间等参数。例如：

```
GENERIC(m: TIME:=1 ns):
```

声明 m 是一个值为 1ns 的时间参数。这样，在程序中，语句

```
tmp1<=d0 AND sel AFTER m:
```

表示 d0 AND sel 经 1ns 延迟后才送到 tmp1。

端口声明是描述器件的外部接口信号的声明，相当于器件的引脚声明。端口声明语句格式为：

```
PORT (端口名,端口名,……: 方向  数据类型名;
……
端口名,端口名,……: 方向  数据类型名);
```

例如：

```
PORT ( a,b: IN STD_LOGIC;          --声明 a、b 是标准逻辑位类型的输入端口
       s: IN STD_LOGIC;            --声明 s 是标准逻辑位类型的输入端口
       y: OUT STD_LOGIC);          --声明 y 是标准逻辑位类型的输出端口
```

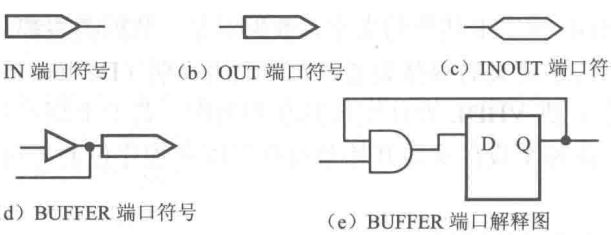
端口方向包括：

IN——输入，原理图符号如图 1.2（a）所示。

OUT——输出，原理图符号如图 1.2（b）所示。

INOUT——双向，既可作为输入也可作为输出，原理图符号如图 1.2（c）所示。

BUFFER——具有读功能的输出，原理图符号如图 1.2(d)所示。图 1.2(e)给出一个 BUFFER 端口的图例子，它是一个触发器的输出，同时可将它的信号读出送到与门的输入端。




（a）IN 端口符号　（b）OUT 端口符号　（c）INOUT 端口符号

（d）BUFFER 端口符号　（e）BUFFER 端口解释图

图 1.2　各种端口的原理图符号及解释图

计数器设计时，一般需要使用 BUFFER 类型输出端口。对于加法计数器来说，当计数脉冲到来时，输出状态加 1，即 Q=Q+1（Q 为计数器的输出端口），表示计数器的输出应具有读功能。

### 1.1.3 结构体

结构体（ARCHITECTURE）用来描述设计实体的内部结构和实体端口之间的逻辑关系，在电路上相当于器件的内部电路结构。结构体由信号声明部分和功能描述语句部分组成。信号声明部分用于描述结构体内部使用的信号名称及信号类型的声明；功能描述部分用来描述实体的逻辑行为。

结构体语句格式为：

```
ARCHITECTURE 结构体名 OF 实体名 IS
[信号声明语句];              --为内部信号名称及类型声明
BEGIN
[功能描述语句]
END ARCHITECTURE 结构体名;
```

例如，设 a、b 是或非门的输入端口，z 是输出端口，y 是结构体内部信号，则用 VHDL 描述的两输入端或非的结构体为：

```
ARCHITECTURE nor1 OF temp1 IS
SIGNAL y: STD_LOGIC;
BEGIN
    y<=a OR b;
    z<=NOT y;
END ARCHITECTURE nor1;
```

说明："nor1"是结构体名，用于区分设计实体中的不同结构体，结构体结束语句"END ARCHITECTURE nor1;"可以省略为"END nor1;"或"END;"。另外，VHDL 程序中的标点符号全部是半角符号，使用全角标点符号被视为非法。

### 1.1.4 配置

配置（CONFIGURATION）用来把特定的结构体关联到（指定给）一个确定的实体，为一个大型系统的设计提供管理和工程组织。

### 1.1.5 基本逻辑器件的 VHDL 描述

在对 VHDL 的设计实体结构有一定了解后，通过以下几个基本逻辑器件的 VHDL 描述的设计示例，使读者对 VHDL 程序设计有初步的理解。

**【例 1.1】** 设计 2 输入端的或门电路。

2 输入端或门的逻辑符号如图 1.3 所示，其中 a、b 是输入信号，y 是输出信号，输出与输入的逻辑关系表达式为：

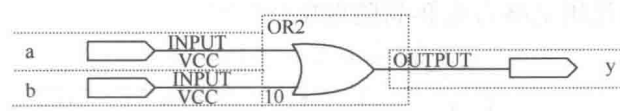



图 1.3 或门逻辑符号

$$y = a + b$$

在 VHDL 语法中，或运算符号是"OR"，赋值符号是"<="，因此在 VHDL 程序中，或逻辑关系表达式为：

```
y <= a OR b
```

下面是按照 VHDL 语法规则编写出来的或门设计电路的 VHDL 源程序，或者称为"或门的 VHDL 描述"。它是一个完整的、独立的语言模块，相当于电路中的一个"或"器件或电路

原理图上的一个“或”元件符号。它能够被 VHDL 综合器接受，形成一个独立存在和独立运行的元件，也可以被高层次的系统调用，成为系统中的一部分。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;          --IEEE 库使用声明
ENTITY or1 IS
PORT (a,b: IN STD_LOGIC;              --实体端口声明
      y: OUT STD_LOGIC);
END or1;
ARCHITECTURE example1 OF or1 IS
BEGIN
    y<=a OR b;                        --结构体功能描述语句
END example1;
```

【例 1.2】设计半加器电路。

半加器的逻辑图如图 1.4 所示，其中 a、b 是输入信号，so、co 是输出信号。用 VHDL 语法规则推导出输出信号与输入信号之间的逻辑表达式为：

```
so<=a XOR b
co<=a AND b
```

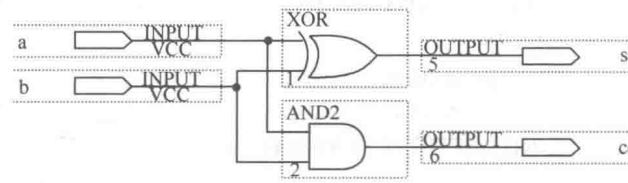



图 1.4　半加器的逻辑图

半加器的 VHDL 描述为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY h_adder IS
PORT (a,b: IN STD_LOGIC;
      so,co: OUT STD_LOGIC);
END h_adder ;
ARCHITECTURE example2 OF h_adder IS
  BEGIN
     so<=a XOR b;
     co<=a AND b;
END example2;
```

VHDL 有多种描述风格，按照原理图的结构进行的描述属于 VHDL 的结构描述风格。结构描述可以从最基本的元件描述开始，然后用结构描述方式将这些基本元件组合起来，形成一个小系统元件，再用结构描述或其他描述方式将一些小系统元件组合起来，形成复杂数字系统。半加器电路的功能仿真波形如图 1.5 所示。在仿真波形中，输入波形的变化是需要经过一定的延迟时间后，才能到达输出端的。另外，从 co 和 so 的输出波形可以看到极窄的脉冲，这是组合逻辑电路的竞争-冒险现象。

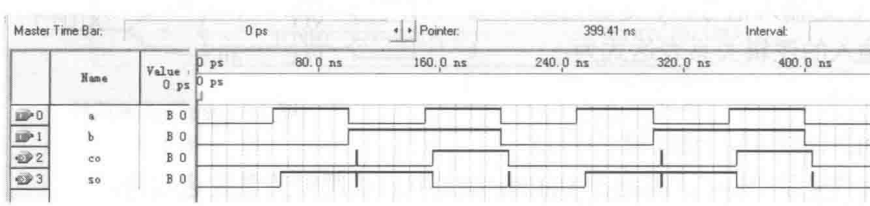



图 1.5　半加器电路的功能仿真波形

【例 1.3】设计 2 选 1 数据选择器。2 选 1 数据选择器的逻辑符号如图 1.6 所示，其中 a、b 是数据输入信号，s 是控制输入信号，y 是输出信号。2 选 1 数据选择器的功能由表 1.1 给出。

表中反映出数据选择器的功能是：如果 s=0 则 y=a，否则（s=1）y=b。用 VHDL 描述 y 与 s 和 a、b 之间的功能关系语句为：

```
y<=a WHEN s=0 ELSE
     b;
```

这是 VHDL 另一种描述风格，称为行为描述。行为描述只描述所设计电路的功能或电路行为，而没有直接指明或涉及实现这些行为的硬件结构。完整的 2 选 1 数据选择器的 VHDL 描述为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY mux21 IS
PORT (a,b: IN STD_LOGIC;
            s: IN STD_LOGIC;
            y: OUT STD_LOGIC);
END mux21;
ARCHITECTURE example3 OF mux21 IS
  BEGIN
    y<=a WHEN s='0' ELSE
            b;
END example3;
```

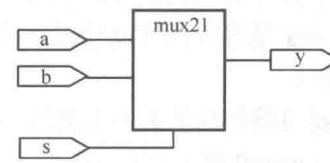



图 1.6　2 选 1 数据选择器的逻辑符号

**表 1.1　2 选 1 数据选择器功能表**

| s | y |
|---|---|
| 0 | a |
| 1 | b |

2 选 1 数据选择器的功能仿真波形如图 1.7 所示。

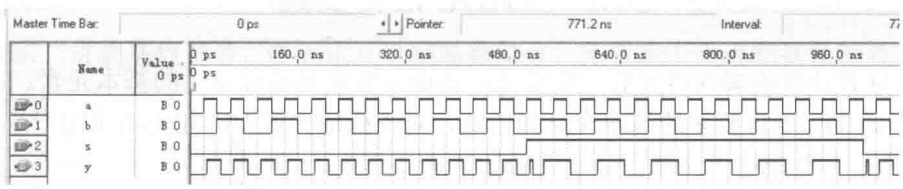



图 1.7　2 选 1 数据选择器的功能仿真波形

【例 1.4】设计锁存器。

上面列举了组合逻辑电路的 VHDL 描述示例，下面以锁存器为例，让读者对时序逻辑电路的 VHDL 描述有一定的了解。1 位数据锁存器的逻辑符号如图 1.8 所示，其中 d 是数据输入信号，ena 是使能信号（或称时钟信号），q 是输出信号。锁存器的功能是：如果 ena=1，则 q=d；否则（即 ena=0）q 保持原来状态不变。

用 VHDL 描述锁存器功能的语句是：

```
IF ena='1' THEN
q<=d:
END IF:
```

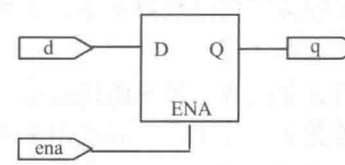



图 1.8　1 位数据锁存器的逻辑符号

完整的锁存器 VHDL 描述如下：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY latch1 IS
PORT   (  d :IN STD_LOGIC;
          ena :IN STD_LOGIC;
            q :OUT STD_LOGIC);
END latch1;
ARCHITECTURE example4 OF latch1 IS
BEGIN
```

```
    PROCESS (d,ena)
        BEGIN
          IF ena='1' THEN
              q<=d;
          END IF;
    END PROCESS;
END example4;
```

在这个程序的结构体中，用了一个进程（PROCESS ）来描述锁存器的行为，其中，输入信号 d 和 ena 是进程的敏感信号，当它们中的任何一个信号发生变化时，进程中的语句就要重复执行一次。

锁存器电路的仿真波形如图 1.9 所示。在仿真波形中，输入 ena 是进程的敏感信号，当 ena=1 时 q=d；当 ena=0 时 q 保持不变；仿真结果验证了设计的正确性。

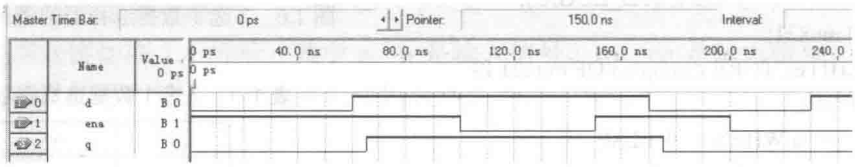



图 1.9　锁存器电路的仿真波形

## 1.2　VHDL 语言要素

VHDL 具有计算机编程语言的一般特性，其语言要素是编程语句的基本元素。准确无误地理解和掌握 VHDL 语言要素的基本含义和用法，对正确地完成 VHDL 程序设计十分重要。

### 1.2.1　VHDL 文字规则

任何一种程序设计语言都规定了自己的一套符号和语法规则，程序就是用这些符号按照 语法规则写成的。在程序中使用的符号若超出规定的范围或不按语法规则书写，都视为非法，计算机不能识别。与其他计算机高级语言一样，VHDL 也有自己的文字规则，在编程中需要认真遵循。

**1．数字型文字**

数字型文字包括整数文字、实数文字、以数制基数表示的文字和物理量文字。

（1）整数文字

整数文字由数字和下画线组成。例如，5、678、156E2 和 45_234_287（相当于 45 234 287）都是整数文字。其中，下画线用来将数字分组，便于读出。

（2）实数文字

实数文字由数字、小数点和下画线组成。例如，188.993 和 88_670_551.453_909（相当于 88 670 551.453 909）都是实数文字。

（3）以数制基数表示的文字

在 VHDL 中，允许使用十进制、二进制、八进制和十六进制等不同基数的数制文字。以数制基数表示的文字的格式为：

数制#数值#

例如：

```
10#170#;              --十进制数值文字
```

```
16#FE#;          --十六进制数值文字
2#11010001#;     --二进制数值文字
8#376#;          --八进制数值文字
```

（4）物理量文字

物理量文字用来表示时间、长度等物理量。例如，60s、100m 都是物理量文字。

### 2. 字符串文字

字符串文字包括字符和字符串。字符是以单引号括起来的数字、字母和符号。例如，'0'，'1'，'A'，'B'，'a'，'b'都是字符。字符串包括文字字符串和数值字符串。

（1）文字字符串

文字字符串是用双引号括起来的一维字符数组。例如，"ABC"，"A BOY. "，"A"都是文字字符串。

（2）数值字符串

数值字符串也称为矢量，其格式为：

```
数制基数符号 "数值字符串";
```

例如：

```
B"111011110";    --二进制数数组，位矢量组长度是 9
O"15";           --八进制数数组，等效 B"001101"，位矢量组长度是 6
X"AD0";          --十六进制数数组，等效 B"101011010000"，位矢量组长度是 12
```

其中，B 表示二进制基数符号，O 表示八进制基数符号，X 表示十六进制基数符号。

### 3. 关键词

VHDL 有 97 个（详见附录 A）关键词，它们是预先定义的单词，在程序中有不同的使用目的，例如，ENTITY（实体）、ARCHITECTURE（结构体）、TYPE（类型）、IS、END 等都是 VHDL 的关键词。VHDL 的关键词允许用大写字母或小写字母书写，也允许大、小写字母混合书写。

### 4. 标识符

标识符是用户给常量、变量、信号、端口、子程序或参数定义的名字。标识符命名规则是：以字母（大、小写均可）开头，后面跟若干个字母、数字或单个下画线，但最后不能为下画线。例如：

```
h_adder，mux21，example          --合法标识符；
2adder，_mux21，ful_ _adder，adder_    --错误的标识符。
```

VHDL1993 标准支持扩展标识符，即以反斜杠来定界，允许以数字开头，允许使用空格及两个以上的下画线。例如，\74LS193\，\A BOY\等为合法的标识符。

### 5. 下标名

下标名用于指示数组型变量或信号的某一元素。下标名的格式为：

```
标识符(表达式);
```

例如，b(3)，a(m)都是下标名。

### 6. 段名

段名是多个下标名的组合。段名的格式为：

```
标识符（表达式 方向 表达式）
```

其中，方向包括：

```
TO                          --表示下标序号由低到高
DOWNTO                      --表示下标序号由高到低
```

例如：

```
D(7 DOWNTO 0)               --可表示数据总线 D7 到 D0
D(0 TO 7)                   --可表示数据总线 D0 到 D7
```

### 1.2.2 VHDL 数据对象

VHDL 数据对象是指用来存放各种类型数据的容器，包括变量、常数和信号。

#### 1. 变量

在 VHDL 语法规则中，变量（VARIABLE）是一个局部量，只能在进程（PROCESS）、函数（FUNCTION）和过程（PROCEDURE）中声明和使用。变量不能将信息带出对它定义的当前设计单元。变量的赋值是一种理想化的数据传输，即传输是立即发生的，不存在任何延时的行为。

任何变量都要声明后才能使用，变量声明的语法格式为：

```
VARIABLE 变量名:数据类型[:=初始值];
```

例如，变量声明语句：

```
VARIABLE a: INTEGER;
VARIABLE b: INTEGER:=2;
```

分别声明变量 a、b 为整型变量，变量 b 赋有初值 2。

变量在声明时，可以赋初值，也可以不赋值，到使用时才用变量赋值语句赋值，因此，变量语句中的“:=初始值”部分内容用方括号括起来表示任选。变量赋值语句的语法格式为：

```
目标变量名:=表达式;
```

例如，下面在变量声明语句后，列出的都是变量赋值语句：

```
VARIABLE x,y: INTEGER;
VARIABLE a,b: BIT_VECTOR(0 TO 7);
x:=100;
y:=15+x;
a:= "10101011";
a(3 TO 6):= ('1','1','0','1');
a(0 TO 5):=b(2 TO 7);
```

#### 2. 信号

信号（SIGNAL）是描述硬件系统的基本数据对象。它作为一种数值容器，不仅可以容纳当前值，也可以保持历史值，这一属性与触发器的记忆功能有很好的对应关系。信号又类似于连接线，可以作为设计实体中各并行语句模块间的信息交流通道。

信号要在结构体中声明后才能使用。信号声明语句的语法格式为：

```
SIGNAL 信号名: 数据类型[:=初值];
```

例如，信号声明语句：

```
SIGNAL temp: STD_LOGIC:=0;
SIGNAL flaga,flagb: BIT;
```