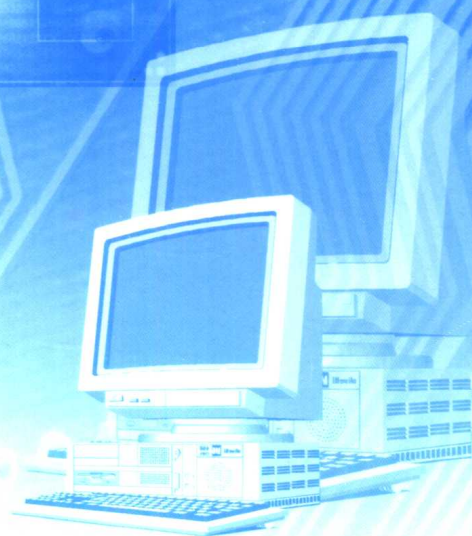


工科高等院校电路系列课程教材

数字电路EDA入门

——VHDL程序实例集

张亦华 延明 编著



北京邮电大学出版社
www.buptpress.com

数字电路 EDA 入门

——VHDL 程序实例集

张亦华 延 明 编著

北京邮电大学出版社
·北京·

内 容 提 要

本书是数字电路电子设计自动化(EDA)入门的工具书。其内容主要包括:用 VHDL 设计的基本组合电路、时序电路、数字综合电路、电路图输入法要领概述、实用 VHDL 语句等;附录部分介绍了 VHDL 基本知识和基本术语,中小规模集成电路等。

本书提供的所有程序代码都经过 MAX + plus II 9.23 软件和 PLD 器件的编译、仿真、下载和实际测量,可以作为进一步开发的参考。大部分实例电路都是在设计数字电路时经常使用的电路。本书为那些想快速步入 EDA 设计大门的读者提供了一个仿制、借鉴、研究、创新的良好工作平台。

本书可作为大学本科和专科院校通信、电子工程类专业的 EDA 实验教材,也可供从事 EDA 工作的技术人员作程序开发的参考书。

图书在版编目(CIP)数据

数字电路 EDA 入门:VHDL 程序实例集/张亦华,延明编著. —北京:北京邮电大学出版社,2003
ISBN 7-5635-0673-X

I. 数... II. ①张...②延... III. ①数字电路—电路设计:计算机辅助设计②硬件描述语言,VHDL—程序设计 IV. TN790.2

中国版本图书馆 CIP 数据核字(2003)第 010620 号

出 版 者:北京邮电大学出版社(北京市海淀区西土城路 10 号)

邮编:100876 发行部电话/传真:62282185/62283578

网址: <http://www.buptpress.com>

经 销:各地新华书店

印 刷:北京忠信诚胶印厂

印 数:4 000 册

开 本:787 mm × 1 092 mm 1/16

印 张:10.5

字 数:275 千字

版 次:2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号:ISBN 7-5635-0673-X/TN·284

定 价:17.00 元

如有印装质量问题请与北京邮电大学出版社发行部联系

前 言

本书的编写遵循“基本、实用、工具、创新”四条原则。其目的,既为学习电子设计自动化(EDA)人员提供一种快速有效的入门工具,一个仿制、借鉴、研究、创新的工作平台,亦为从事 EDA 的技术人员提供一本关于基本数字电路电子设计自动化的检索开发工具书。

本书中的一部分实例是在多年教学、科研的基础上开发整理出来的,另一部分是业内经常使用的 VHDL 程序模块。所有实例都在 MAX + plus II 软件平台上进行过编译、通过仿真,用可编程逻辑器件 EPM7128S(或者 FLEX10K)下载、实际测量,因而均可以作为进一步开发的参考。

如果说学习、掌握电子设计自动化技术有捷径的话,那就应该从仿制、理解、实践、创新开始。学习 VHDL 应遵循边学边练的原则,其语法规则只有在实践中才能完全领悟、掌握。

本书的出版得益于电路中心各个时期从事 EDA 教学工作的老师们的教学成果,没有他们的宝贵经验,也就没有本书的顺利出版。本书的出版也得益于广大同学的 EDA 学习与实践,他们为本书的编写提供了丰富的实践经验。本书在准备、编写过程中得到了学校各级领导,教务处、电路中心教师和工程技术人员的热情支持与帮助,在此向他们表示诚挚的谢意,特别向安德宁、徐惠民、赵廷瑞、赵振纲教授,以及肖冰、郭莉、袁宝库、魏学军、姬红强、袁东明等老师表示衷心的感谢!

本书的编写是我们从事 EDA 教学工作的一个阶段总结,真诚的希望广大读者能对书中存在的问题提出宝贵的批评和建议。

作者

2003 年 1 月

100-513-1

目 录

第一章 VHDL 设计基础	1
第一节 VHDL 设计入门	1
实例 1-1 行为描述模式	1
实例 1-2 数据流描述模式	2
实例 1-3 结构描述模式	3
第二节 VHDL 的框架结构	4
实例 1-4 VHDL 的基本框架结构	4
实例 1-5 VHDL 的较复杂的框架结构	5
第二章 用 VHDL 设计组合电路	9
第一节 建立组合电路的方法	9
实例 2-1 借助真值表设计组合电路	9
实例 2-2 用 VHDL 的逻辑表达式设计组合电路	10
实例 2-3 用 VHDL 的算数表达式设计组合电路	10
第二节 描述组合电路的 VHDL 程序实例	11
实例 2-4 基本逻辑门	11
实例 2-5 用 0c 门实现线与功能	12
实例 2-6 带使能端的三态门	13
实例 2-7 三选一数据选择器	14
实例 2-8 四选一数据选择器(1)	15
实例 2-9 四选一数据选择器(2)	16
实例 2-10 芯片内两个节点相接	17
实例 2-11 3 线-8 线译码器	18
实例 2-12 共阴极七段数码显示驱动电路	20
实例 2-13 优先编码器	22
实例 2-14 比较器	24
实例 2-15 二十进制 BCD 译码器	25
实例 2-16 并行奇校验电路	26
实例 2-17 补码生成电路	27
实例 2-18 全减法器	28
实例 2-19 移位器	29
实例 2-20 只读存储器	30
实例 2-21 并行奇校验发生器	31

第三章 用 VHDL 设计时序电路	35
第一节 时序电路的基本要素	35
实例 3-1 对触发器及时钟信号的 VHDL 描述(1)	35
实例 3-2 对触发器及时钟信号的 VHDL 描述(2)	36
实例 3-3 对触发器及时钟信号的 VHDL 描述(3)	37
实例 3-4 对触发器及时钟信号的 VHDL 描述(4)	38
实例 3-5 对触发器及时钟信号的 VHDL 描述(5)	39
实例 3-6 对触发器及时钟信号的 VHDL 描述(6)	40
实例 3-7 锁存器	42
第二节 移存器	43
实例 3-8 串入串出移存器(1)	43
实例 3-9 串入串出移存器(2)	44
实例 3-10 串入串出移存器(3)	45
实例 3-11 串入串出双向移存器	46
实例 3-12 串入并出移存器	47
实例 3-13 并入串出移存器	49
第三节 计数器	51
实例 3-14 一热态位编码计数器	51
实例 3-15 4 位移存器型扭环计数器	52
实例 3-16 七位移存器型计数器	54
实例 3-17 二进制($M=16$)计数器	55
实例 3-18 60 进计数器	56
实例 3-19 BCD 码 60 进异步计数器	58
实例 3-20 BCD 码 60 进同步计数器	60
实例 3-21 用 VHDL 描述中小规模集成电路 74LS169	62
实例 3-22 模值可变计数器	65
实例 3-23 指定起始状态的计数器	66
第四节 状态机应用	68
实例 3-24 序列信号发生器	68
实例 3-25 告警器	69
实例 3-26 楼梯照明灯控制器	71
实例 3-27 循环彩灯控制器	73
第四章 用电路图输入方法设计数字电路	76
实例 4-1 TFFE 触发器	76
实例 4-2 冒险电路	77
实例 4-3 74169 计数器的应用	78
实例 4-4 与门阵列控制器	79

实例 4-5 参数型宏功能与门	80
实例 4-6 序列发生器	81
第五章 资源调用与特色电路	84
第一节 资源调用	84
实例 5-1 不用调库令调用自制器件	84
实例 5-2 使用调库、调包令调用程序包中的自制器件	85
实例 5-3 使用调库、调包令调用程序包中定义的函数	86
实例 5-4 使用调库、调包命令调用软件包中定义的子程序	88
实例 5-5 为支持不同数据类型进行运算,调用库系统程序包	90
实例 5-6 在结构体内定义一个子程序(过程)	91
实例 5-7 调用 Altera 公司的库元件 DFF(D 触发器)和 74151b(选择器)	91
第二节 特色电路	92
实例 5-8 计数器型防抖动电路(1)	92
实例 5-9 计数器型防抖动电路(2)	94
实例 5-10 采样型防抖动微分电路	96
实例 5-11 积分分频器	98
实例 5-12 4×4 键盘输入电路	100
实例 5-13 串行偶校验器	106
第六章 数字系统课题	108
课题 6-1 带数字显示的秒表	108
课题 6-2 8×8 发光点阵逐点扫描显示装置	111
课题 6-3 彩灯闪烁装置	113
课题 6-4 抢答器	116
课题 6-5 密码锁	118
课题 6-6 数字频率计	122
附 录	132
附录 1 VHDL、MAX + plus II 知识问答	132
附录 2 集成电路	134
附录 3 集成电路主要性能参数	144
附录 4 VHDL 术语汉英对照	148
附录 5 数字电路术语汉英对照	152
参考文献	157

第一章 VHDL 设计基础

第一节 VHDL 设计入门

由于本书大部分设计实例均以 VHDL 语言进行描述,因此在每一个实例的“设计任务”段落中都略去“用 VHDL 语言进行设计”这一限定文字。

用 VHDL 描述数字电路有行为描述、数据流描述和结构描述三种模式,其核心模式是行为描述模式,下面分别加以介绍。

实例 1-1 行为描述模式

一、设计任务

设计一个有高进位 c1、低进位 c0 的 10 位二进制全加器电路。

二、算法设计

若使用几个低位加法器组合求解,描述太繁琐,因而考虑用抽象的上层行为描述模式设计该加法器。

三、VHDL 源程序

1. 文件名: adder1.vhd

2. 源程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity adder1 is
port (a,b: in std_logic_vector(9 downto 0) ;
      co: in std_logic_vector(9 downto 0);
      c1: out std_logic;
      sum: out std_logic_vector (10 downto 0));
end ;

architecture jg of adder1 is
signal a_temp:std_logic_vector(10 downto 0);
signal b_temp:std_logic_vector(10 downto 0);
signal sum_temp:std_logic_vector(10 downto 0);
begin
process
begin
a_temp <= '0' & a;
```

```

    b_temp <= '0' & b;
    sum_temp <= a_temp + b_temp + co;
    sum <= sum_temp(9 downto 0);
    cl <= sum_temp(10);
end process;
end jg;

```

四、程序说明

1. 用行为描述模式设计加法器,可以降低设计难度。行为描述用于表示输入与输出之间转换的行为,不需要包含任何结构方面的信息。
2. 设计者只需编制出源程序,而挑选电路方案的工作则由计算机系统自动完成。
3. 最终选取的电路方案的优化程度,往往取决于综合软件的技术水平和器件的支持能力。也就是说,最终选取的电路方案占用的 PLD 器件资源不一定是最少的。
4. 设计策略,首先考虑用行为描述模式设计电路,如果设计的结果不能满足资源占有率的要求,则应改变描述模式。
5. 本程序用“和(sum)”的最高位作为高进位,由于规定设计的是 10 位加法器,所以应设置“和(sum)”的位数为 11 位。

实例 1-2 数据流描述模式

一、设计任务

设计一个实现逻辑函数: $f = ab + cd$ 的逻辑电路。

二、算法设计

使用数据流描述模式设计电路。

三、VHDL 源程序

```

1. 文件名: and_or.vhd
2. 源程序
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity and_or is
port (a,b,c,d: in std_logic;
      f: out std_logic);
end ;

architecture sjl of and_or is
begin
process
begin
    f <= (a and b) or (c and d);
end process;

```

end;

四、程序说明

1. 用数据流描述模式设计电路与用传统的逻辑方程设计电路很相似, 显见,

$$f = ab + cd$$

和

$$f \leq (a \text{ and } b) \text{ or } (c \text{ and } d)$$

是很相似的。它们的差别仅在于描述逻辑运算的逻辑符号及表达方式略有不同。数据流描述亦表示行为, 但含有结构信息, 如进程间的通信等, 通常用并行语句进行描述。应当说明的是, 有的描述形式究竟属于哪一种模式是难以界定的, 但这绝对不会影响对具体描述的应用。

2. 设计中只要有了布尔代数表达式就很容易将它转换为 VHDL 的数据流表达式。转换方法是用 VHDL 中的逻辑运算符置换布尔逻辑运算符即可。例如, 用 `or` 置换“+”; 用“`<=`”置换“=”。

实例 1-3 结构描述模式

一、设计任务

设计一个实现逻辑函数: $f = ab + cd$ 的逻辑电路。

二、算法设计

使用结构描述模式设计电路。

三、VHDL 源程序

1. 文件名: `and_or_gate.vhd`

2. 源程序

```
entity and_or_gate is
    port (a,b,c,d: in bit ;
          f: out bit);
end;

architecture and_or_gate1 of and_or_gate is
    component ym
        port(
            a,b: in bit;
            c: out bit);
    end component;

    component hm
        port(
            a,b: in bit;
            c: out! bit);
    end component;

    signal temp1,temp2: bit;
begin
    u1: ym
    port map (a=> a , b=> b,c=> temp1);
```

```

u2: ym
port map (a => c , b => d, c => temp2 );
u3: hm
port map (temp1, temp2, c => f);
end;

```

四、程序说明

1. 用结构描述模式设计电路的步骤

- (1) 调用已有元件,通过 component 语句实现。
- (2) 用设计的端口名称替换被调用元件的端口名称,这一步通过例化语句实现。

2. 结构描述模式的特点

(1) 在已有元件的端口之间进行连接,若多个元件的端口被命名为同一名称,则表示这几个端口是并接在一起的。

(2) 结构描述实质是用文字描述电路原理图中各元件的连接关系。

五、被调用元件的 VHDL 源程序

1. 被调用元件 ym 的 VHDL 源程序

```

entity ym is
port (a, b: in bit;
      c: out bit);
end;
architecture ym1 of ym is
begin
  c <= a and b;
end;

```

2. 被调用元件 hm 的 VHDL 源程序

```

entity hm is
port (a, b: in bit;
      c: out bit);
end;
architecture hm1 of hm is
begin
  c <= a or b;
end;

```

第二节 VHDL 的框架结构

实例 1-4 VHDL 的基本框架结构

一、设计任务

描述一个与门: $f = a \text{ and } b$ 。

二、算法设计

使用数据流模式描述。

三、VHDL 源程序

1. 文件名:ym.vhd

2. 源程序

```
entity ym is
port (a,b:in bit;
      c:out bit);
end;
architecture yml of ym is
begin
c <= a and b;
end;
```

四、程序说明

VHDL 语言的最基本结构是由设计实体部分和设计结构体部分组成。无论是简单的设计还是复杂的设计,这两部分都必须存在。

1. 从 entity 开始到关键字 architecture 之前是设计实体部分,主要用于描述器件(无论是复杂还是简单的器件)的外端口(外貌),即输入、输出端口的数量、名称、类型和端口使用的数据类型。

2. 从关键字 architecture 开始到结束是设计结构体部分,主要用于描述元件内部各个逻辑器件或者功能部件的连接关系。

实例 1-5 VHDL 的较复杂的框架结构

一、设计任务

描述一个系统,它含有:

- (1) 1 个 3 位二进制代符号位的减法器。
- (2) 1 个分频系数等于 6 的分频器。
- (3) 1 个 2 输入端“与非门”。
- (4) 1 个只有数据输入端、时钟控制端和输出端的 D 触发器。
- (5) 1 个 1 位二进制数比较器。

二、算法设计

系统分为 5 个电路部分。为了设计整齐,将电路设计文件分为相互独立的 5 个部分进行描述,它们是:

- (1) 由 1 个 block b1 块语句描述 3 位二进制代符号位的减法器。
- (2) 由 1 个 block b2 块语句描述分频系数等于 6 的分频器。
- (3) 由 1 个并发语句描述 2 输入端“与非门”。
- (4) 由 1 个并发语句 process p1 描述 D 触发器。
- (5) 由 1 个并发语句 process p2 描述 1 位比较器。

三、VHDL 源程序

1. 文件名:jg.vhd

2. 源程序

```

library ieee;          -- 调用资源库语句。
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity jg is           -- 开始描述设计实体。
generic(x:integer := 2);    -- 类属参数说明语句。
    port (din1,din2:in std_logic_vector(0 to x);
          clk,d,j,k:in bit;
          yout, qout,f:out bit;
          bjz, fh:out std_logic;
          dout:out std_logic_vector(0 to x));
end ;

architecture zh of jg is    -- 开始描述结构体。
    signal temp :std_logic_vector(0 to x) ;    -- 对结构体内使用的对象或元件作说明。
    signal q ,y :bit;
    signal    r :integer range 0 to 5;
    begin      -- 开始各子电路描述。
        -- 描述减法器模块开始。
    b1:block begin
        p:process
            begin
                if din1 > din2 then
                    fh <= '0';
                    temp <= (din1 - din2 );
                else
                    fh <= '1';
                    temp <= (din2 - din1 );
                end if;
                dout <= temp;
            end process ;
        end block b1;
        -- 描述减法器模块结束,描述分频器开始。
    b2:block
        begin
        process(clk)
            begin
                if clk'event and clk = '1' then
                    if r = 5 then

```

```
r <= 0;
y <= '1';
else
    r <= r + 1;
    y <= '0';
end if;
end if;
    yout <= y;
end process;
end block b2 ;
-- 描述分频器结束。
-- 下为描述与非门的并发语句。
f <= not(j and k);
p1: process(clk)    -- 描述 D 触发器的 process p1 进程。
begin
    if clk'event and clk = '1' then
        q <= d;
        qout <= q;
    end if;
end process p1;
p2: process    -- 描述比较器的 process p2 进程。
begin
    if j > k then
        bjz <= '1';
    else
        bjz <= '0';
    end if;
end process p2;
end;
```

四、程序说明

1. 一个复杂的 VHDL 源文件都需要调用库资源,以减少编程量。同样复杂的 VHDL 源文件也必须有 entity(设计实体)、architecture(结构体)这两个最基本部分。一个 VHDL 源文件容许存在多个结构体,但是在编译时,必须用语句指定一个结构体供软件系统编译使用。使用 MAX + plus II 软件对 VHDL 语言进行编译时,该软件是自动选择几何位置排列在最后的结构体作为当前结构体进行编译的,因此,对多结构体的调用语句不再叙述。

结构体是由若干并发语句组成的,而并发语句可由若干顺序语句构成。

2. 块语句、process 语句都是并发语句,软件系统是异步处理各个不同的并发语句的,在物理硬件上,不同的并发语句描述芯片内部不同路径上传递的信号和电路结构。不同的并发语句之间可以通过全局性信号来通信。

块语句、process 语句既可以用来描述一个独立的电路,也可以用来描述一个大电路中的一个子电路。

并发语句内部可以嵌套并发语句,例如在块语句的内部可以嵌入 process 语句,但是,在 process 语句的内部,各语句却是顺序执行的。

3. 本程序在设计实体说明部分引入一条“类属参数说明语句”,语句以关键字 generic 开头。该语句用来定义一个通用参数,使程序的通用性好,修改更方便。

第二章 用 VHDL 设计组合电路

第一节 建立组合电路的方法

实例 2-1 借助真值表设计组合电路

一、设计任务

设计一个四选一选择器。

二、算法设计

借助真值表进行设计。

三、VHDL 源程序

1. 文件名: xzq4_1.vhd

2. 端口图与真值表

端口图如图 2-1 所示,真值表如表 2-1 所示。

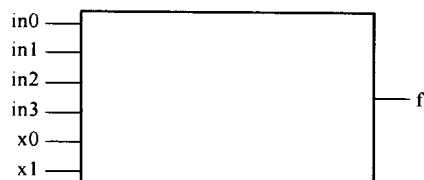


图 2-1 端口图

表 2-1 真值表

in0	in1	in2	in3	x1	x0	f
in0	-	-	-	0	0	in0
-	in1	-	-	0	1	in1
-	-	in2	-	1	0	in2
-	-	-	in3	1	1	in3

3. 源程序

```
entity xzq4_1 is
```

```
    port(in0,in1,in2,in3:in bit;  
          x0,x1:in bit;  
          f2:out bit);
```

```
end;
```

```
architecture a of xzq4_1 is
```

```
begin
```

```
    f <= (in0 and ((not x1 )and(not x0))) or  
          (in1 and ((not x1 )and    x0)) or  
          (in2 and (    x1 and(not x0))) or  
          (in3 and (    x1 and    x0));
```

```
end;
```

四、程序说明

1. bit 数据类型是 VHDL 语言 IEEE 标准的缺省数据类型,不需要调用程序库(包)语句。
2. end 语句中的实体名 xzq4_1 和结构体名 a 可以省略。
3. 按真值表(可用“与或”结构实现)要求,用 VHDL 语言逻辑表达式方式描述四选一选择器,将 $f = '1'$ 的行用 VHDL 最小项表达式表达出来即可。这种描述方法和传统的由真值表变为最小项表达式的设计方法是相同的,只是用 VHDL 语言进行描述无须化简(由计算机进行化简);而用传统设计方法描述时,常常要对最小项表达式进行化简,以使设计电路简化。
4. 源程序中 x0, x1 是选择控制信号。

实例 2-2 用 VHDL 的逻辑表达式设计组合电路

一、设计任务

设计一个函数电路: $y = abc + ef$ 。

二、算法设计

用 VHDL 的逻辑表达式进行描述。

三、VHDL 源程序

```

1. 文件名:hs.vhd
2. 源程序
library ieee;
use ieee.std_logic_1164.all;

entity hs is
    port(a,b,c,e,f:in std_logic;
          y:out std_logic);
end;

architecture a of hs is
begin
    y <= (a and b and c) or (e and f)

end;
```

四、程序说明

使用 VHDL 语言的逻辑表达式设计函数电路是很方便的,只要用 VHDL 语言的逻辑符号置换布尔方程中相应的逻辑符号即可。

实例 2-3 用 VHDL 的算数表达式设计组合电路

一、设计任务

设计 1 位全加器电路。

二、算法设计

用 VHDL 的算数表达式进行设计。