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);
        c1 <= 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 <= (a and b)or(c 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 中的逻辑运算符号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 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 architectu。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。

三、VHDL 源程序

1. 文件名：full_adder.vhd

2. 源程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;        -- 为了进行加法运算必须调用该库。

entity full_adder is
   port(a,b,c0:in std_logic;
        sum,c1:out std_logic);
end ;

architecture a of full_adder is
signal sum1: std_logic_vector(1 downto 0);
signal a1,b1,c01: std_logic_vector(1 downto 0);

begin
a1 <= '0' & a;
b1 <= '0' & b;
c01 <= '0' & c0;          -- 必须先进行并位运算才能进行下式运算。
sum1 <= a1 + b1 + c01;
sum <= sum1(0);           -- 求和。
c1  <= sum1(1);           -- 求高进位。
end ;
```

四、程序说明

1. 使用 VHDL 语言的算数运算表达式设计全加器是很方便的。请注意：在 VHDL 描述中，要用并位运算解决求进位值 c1，否则对高进位 c1 的描述就要采用其他较复杂的方法。

2. 在程序中，用并位运算是为了满足代入符 "<=" 左、右两侧运算对象的位数相同。

3. 一般来说，组合电路可以使用行为描述、数据流描述和结构描述中的一种或者几种描述模式进行描述。设计时，只要描述方便、易于理解、便于记忆，程序经编译、仿真、测试效果好的描述模式就是好的描述模式。

有时，程序代码虽然简单，但是占用器件资源却多；相反，有时，程序代码虽然复杂，但是占用器件资源却少。因此，不能简单的用程序代码简单还是复杂来判断程序代码的优劣，而是要对程序代码进行综合评价，才能得出比较科学的结论。

第二节　描述组合电路的 VHDL 程序实例

实例 2-4　基本逻辑门

一、设计任务

构成基本逻辑门。

二、算法设计

用 VHDL语言的逻辑标识符（赋值语句）描述。

三、VHDL 源程序

1. 文件名 jbm.vhd

2. 源程序

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

entity jbm is
   port( a,b : in bit;
   f1,f2,f3,f4,f5,f : out bit);
end ;

architecture a of jbm is
begin
 --以下部分构造基本逻辑门。
        f1 <= a and b;          -- 构成与门。
        f2 <= a or b;           --构成或门。
        f <= not a ;            -- 构成非门。
        f3 <= a nand b;         --  构成与非门。
        f4 <= a nor b;          -- 构成异或门。
        f5 <= not(a xor b);     --构成异或非门，即同门。
end ;
```

四、程序说明

1. 应注意 VHDL语言的各种逻辑标识符号的含义。

2. 由基本逻辑门可以实现任意函数功能。

3. 注意 not 逻辑标识符号的正确使用方法。例如：'1' <= not '0' 表达式是错误的，这是因为逻辑表达式中的运算对象必须是信号或者变量，不能是常量，而 '1'、'0' 是常量。

实例 2-5　用 Oc 门实现线与功能

一、设计任务

用一个 Oc 门设计一个实现线与功能的电路，令 d0、d1、d2 为输入信号，f 为输出信号。

二、算法设计

用 if 顺序语句描述。

三、VHDL 源程序

1. 文件名：xian_yu.vhd

2. 端口图

线与电路端口图如图 2-2所示。

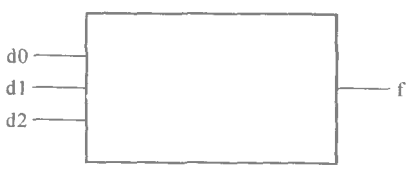

图 2-2　线与电路端口图

3. 源程序

```
library ieee;
```

```
use ieee.std_logic_1164.all;
entity xian_yu is
  port(d0,d1,d2 : in std_logic;
    f: out std_logic);
end ;

architecture a of xian_yu is
begin
  process
  begin
    if ((d0 and d1 and d2) = '1') then
      f<='Z';
    else
      f<='0';
    end if;
  end process;
end a;
```

四、程序说明

1. 文件描述用一个 Oc 门实现信号 d0、d1、d2 的线与功能。

2. 实际应用时，应在输出端接一个上拉电阻，若用 Altera 的 PLD 芯片，推荐使用的电阻值为 1 kΩ。

3. 字符'Z'中的 Z 必须大写。

4. if 顺序语句置于并发语句 process 的内部。if 顺序语句按书写顺序执行。

实例 2-6　带使能端的三态门

一、设计任务

设计一个三态门，令该三态门的输入信号为 data，使能信号为 oe，输出信号为 tri_output。

二、算法设计

用 if 顺序语句描述。

三、VHDL 源程序

1. 文件名：tri.vhd

2. 端口图

带使能端的三态门端口图如图 2-3 所示。

图 2-3　带使能端的三态门端口图

3. 源程序

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

entity tri is
  port (oe : in std_logic;
```

```
        data: in std_logic;
 tri_output: out std_logic);
end ;

architecture a of tri is
begin
  process (oe, data)
  begin
    if oe = '0' then
     tri_output <= 'Z';
    else
     tri_output <= data;
    end if;
  end process;
end a;
```

四、程序说明

1. **文件描述一个三态门。**

2. **实际应用时，若** output **为高阻，且在输出端接一个上拉电阻，则输出端逻辑值为**'1'**如果在输出端接一个下拉电阻，则输出端逻辑值为**'0'**。若用** Altera **的** PLD **芯片，推荐使用的电阻值为** 1 kΩ。

3. **字符**'Z'**中的** Z **必须大写。**

实例 2-7　三选一数据选择器

、设计任务

描述一个三选一数据选择器电路，令 a、b、c **为输入信号，**e1、e2 **为选择信号，**fout **为输出信号。**

二、算法设计

用 if **顺序语句描述。**

三、VHDL **源程序**

1. **文件名：**mux3.vhd

2. **端口图**

三选一数据选择器端口图如图 2-4 **所示。**

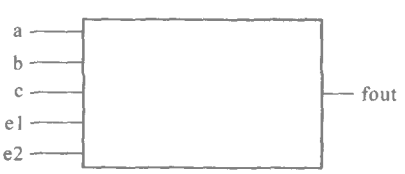

图 2-4　**三选一数据选择器端口图**

3. **源程序**

```
library ieee;
use ieee.std_logic_1164.all;
entity mux3 is
  port(a,b,c,e1,e2:in bit;
                fout:out bit);
end ;
```

```
architecture a1 of mux3 is
begin
  process (a,b,c,e1,e2)
  begin
    if (e1 = '1') then
      fout <= a;
    elsif (e2 = '1') then
      fout <= b;
    else
      fout <= c;
    end if;
  end process;
end ;
```

四、程序说明

程序中 e1、e2 **为选择信号,当** e1e2 = "10" **或者** e1e2 = "11", a **输出 当** e1e2 = "01". b **输出 当** e1e2 = "00", c **输出。**

实例 2-8　四选一数据选择器(1)

、设计任务

描述一个四选一数据选择器电路，令 din **为输入矢量信号，**a、b **为选择信号，**q **为输出信号。**

二、算法设计

用 case **顺序语句描述。**

三、VHDL **源程序**

1. **文件名**：mux4.vhd

2. **端口图**

四选一数据选择器(1**)端口图如图** 2-5 **所示。**

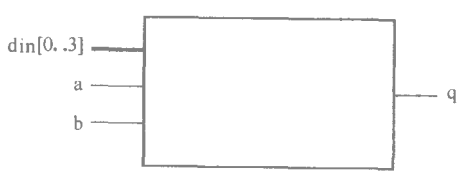

图 2-5　**四选一数据选择器(**1**)端口图**

3. **源程序**

```
library ieee;
use ieee. std_logic_1164.all;
entity mux4_case is
  port( a,b: in std_logic;
        din: in std_logic_vector( 0 to 3);
          q: out std_logic);
end ;
architecture a of mux4_case is
signal sel : std_logic_vector( 0 to 1);
```

```
begin
  process
    begin
    sel <= a & b;
    case sel is
      when "00" => q <= din(0);
      when "01" => q <= din(1);
      when "10" => q <= din(2);
      when "11" => q <= din(3);
      when others => q <= 'Z';        -- 见程序说明。
    end case;
  end process ;
end a;
```

四、程序说明

1. std_logic 数据类型,除'0'、'1'值外,还有其他值,用 others 穷尽所有可能的组合值。

2. case 顺序语句置于并发语句 process 的内部。case 顺序语句按书写顺序执行。

3. 本程序中的 din 为输入矢量信号,这是一个 4 位矢量信号。矢量信号在图中用黑粗线表示。

实例 2-9 四选一数据选择器(2)

一、设计任务

描述一个四选一数据选择器电路，令 input 为输入矢量信号,a、b 为选择信号,s 为输出信号。

二、算法设计

用并行条件信号赋值语句及选择信号赋值语句描述。

三、VHDL 源程序

1. 文件名:mux4_2.vhd

2. 端口图

四选一数据选择器(2)端口图如图 2-6所示。

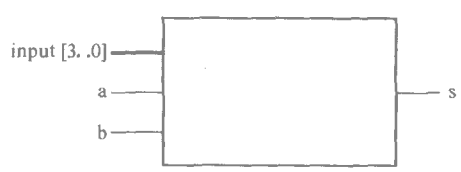

图 2-6 四选一数据选择器(2)端口图

3. 源程序

```
library ieee;
use ieee.std_logic_1164.all;
entity mux4_2 is
port(input:in std_logic_vector(3 downto 0);
       a,b:in std_logic ;
         s:out std_logic) ;
end ;
```

```
architecture with_when of mux4_2 is
  signal sel : std_logic_vector(1 downto 0);
  begin
    sel <= a & b;
    s <= input(0) when sel = "00" else
         input(1) when sel = "01" else
         input(2) when sel = "10" else
         input(3) ;        --见程序说明。
end ;
architecture with_select of mux4 is
  signal sel : std_logic_vector (1 downto 0);
  begin
    sel <= a & b;
    with sel select
      s <= input(0) when "00" ,
           input(1) when "01" ,
           input(2) when "10" ,
           input(3) when "11" ,
           'Z' when others;
end ;
```

四、程序说明

1. 本程序中含有两个结构体 with_when 和 with_select，MAX + plus Ⅱ 软件系统自动执行几何位置处于最后的结构体 with_select。

2. 结构体 with_when 是用并行条件信号赋值语句描述四选一数据选择器。注意，最后一个输出 input(3)不含 when 子句 在 s 表达式中只有一个分号(;)。

3. 结构体 with_select 是用并行选择信号赋值语句描述四选一数据选择器。注意，选择信号赋值语句中的选择条件与 case 语句相似，不允许条件重叠和涵盖不全。由于 a、b 的值除了'1'、'0'外还有其他 7 个值，所以要用 when others 代表其他值，以穷尽所有可能值。

4. 由本例可知，VHDL 语言的描述能力很强，同一个设计任务，可以用不同的语句进行描述，这种强描述功能对设计者选择描述方法是很方便的。

5. 本程序中的 input 为输入 4 位矢量信号。

实例 2-10　芯片内两个节点相接

一、设计任务

将芯片的两个节点 a 和 b 相接。

二、算法设计

用信号赋值语句描述。