

第2章

程序结构与数据对象

2.1 VHDL程序结构

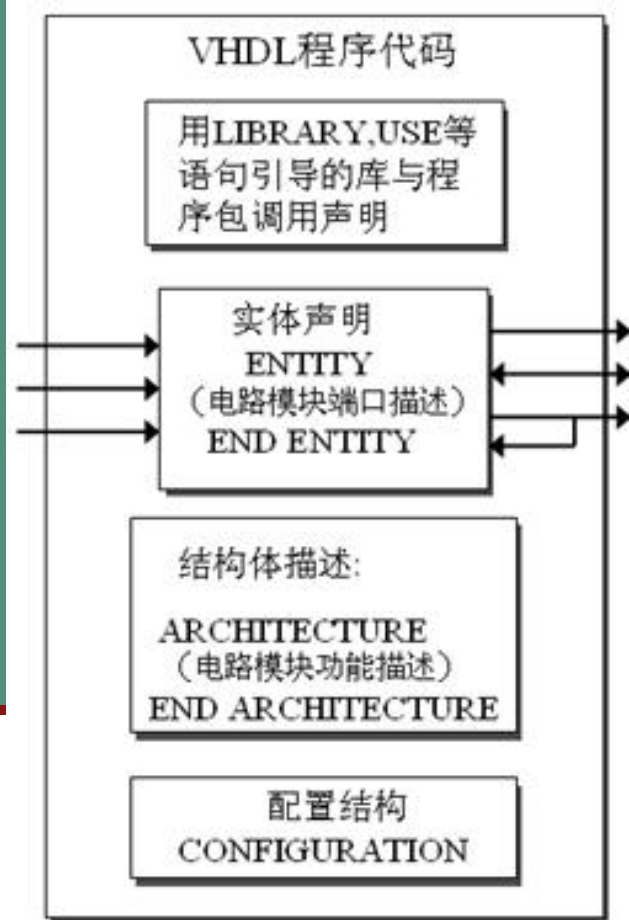


图 2-1 VHDL 程序结构模型图

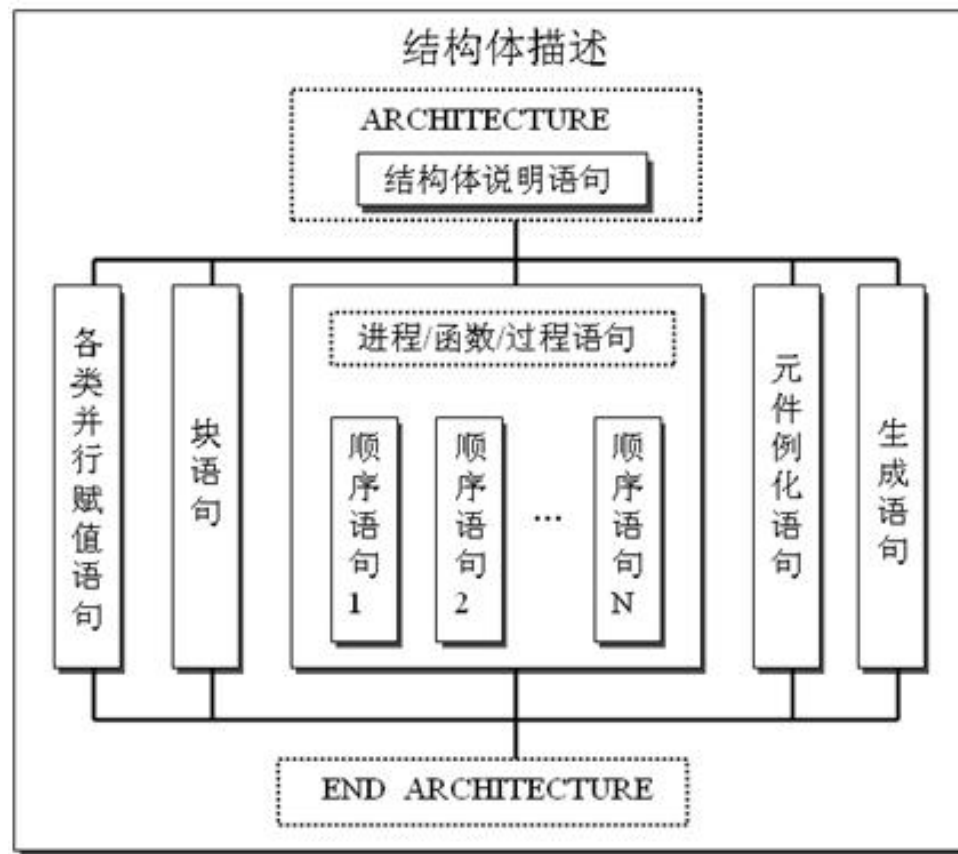
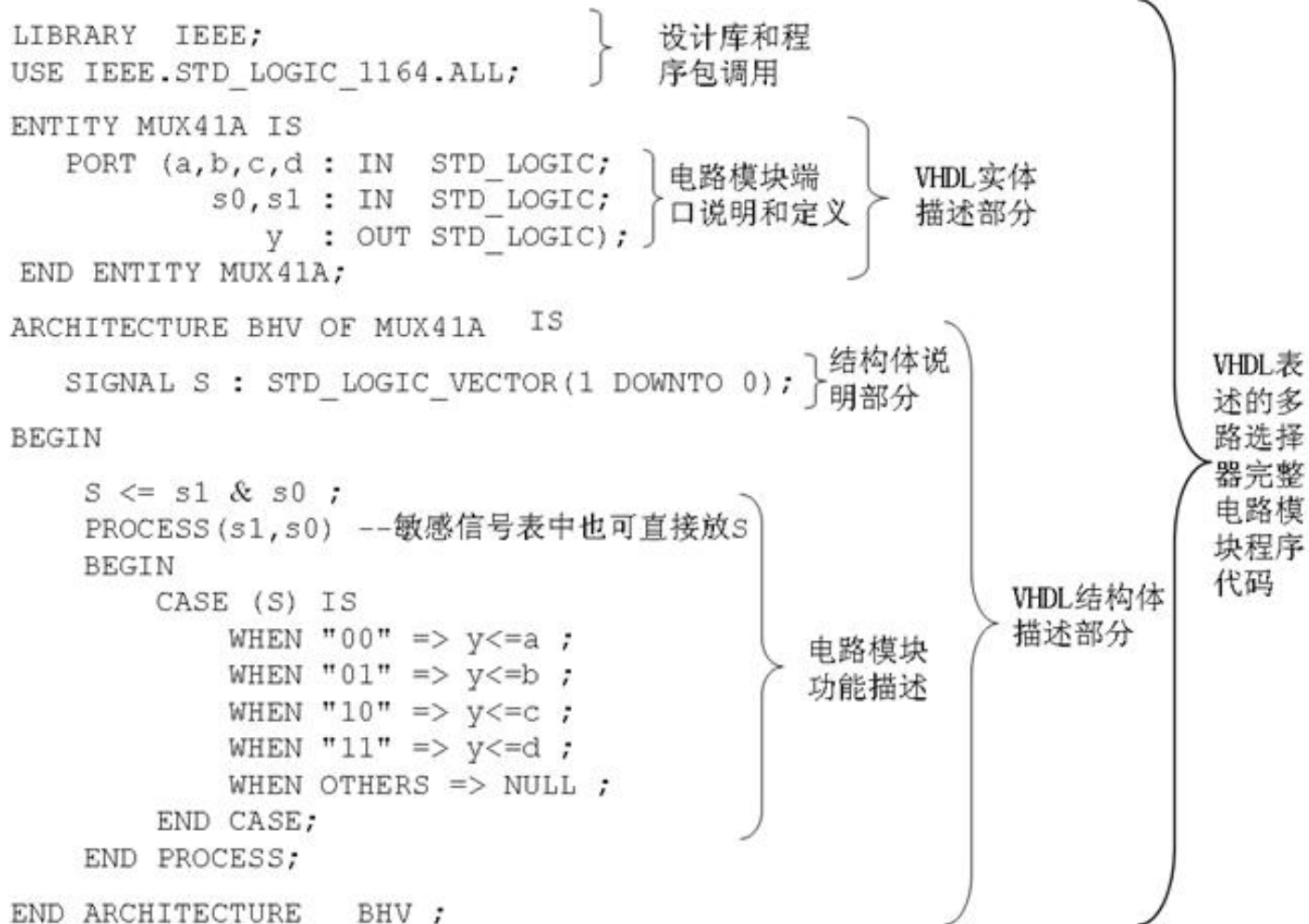


图 2-2 VHDL 结构体模型图

2.1 VHDL程序结构

【例 2-1】



2.1 VHDL程序结构

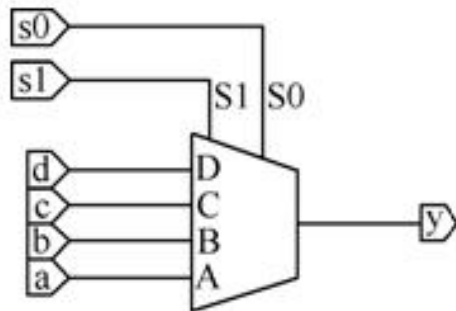


图 2-3 四选一多路选择器

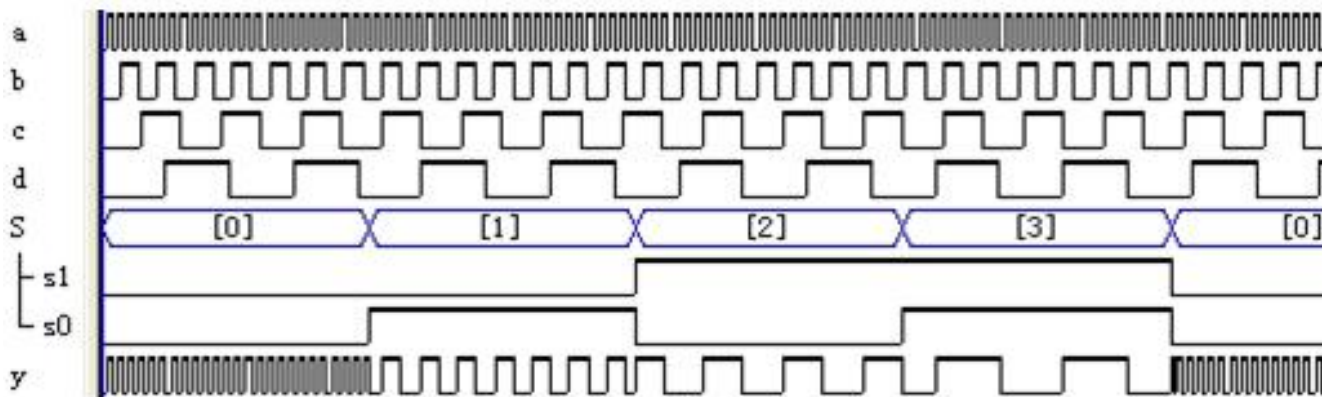


图 2-4 四选一多路选择器 MUX41a 的时序波形

2.2 VHDL程序基本构建

2.2.1 实体和端口模式

```
ENTITY 实体名 IS  
    [GENERIC ( 参数名: 数据类型 );]  
    [PORT ( 端口表: 数据类型);]  
END ENTITY 实体名;
```

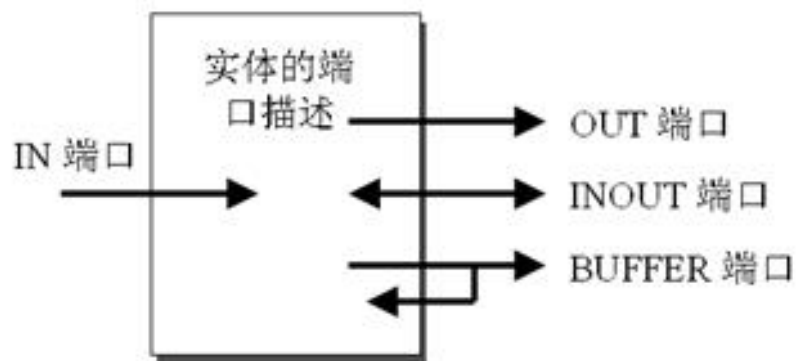


图 2-5 实体中 4 种类型的端口模型图

2.2 VHDL程序基本构建

2.2.2 结构体

```
ARCHITECTURE 结构体名 OF 实体名 IS  
    [说明语句]  
BEGIN  
    [功能描述语句]  
END ARCHITECTURE 结构体名;
```

2.2 VHDL程序基本构建

2.2.3 库和库的种类

IEEE库

STD库

```
LIBRARY WORK ;
```

```
LIBRARY STD ;
```

```
USE STD.STANDARD.ALL ;
```

--打开 STD 库

--能够使用 STD 库中的所有内容

WORK库

VITAL库

2.2 VHDL程序基本构建

2.2.4 库和程序包的调用方法

```
LIBRARY 库名;
USE 库名.程序包名.ALL ;

LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;

USE 库名.程序包名.项目名 ;
USE 库名.程序包名.ALL ;

LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.STD_ULONGIC ;
USE IEEE.STD_LOGIC_1164.RISING_EDGE ;

USE WORK.STD_LOGIC_1164.ALL;
```

2.2.5 配置

2.3 VHDL文字规则

2.3.1 数字

整数

5, 678, 0, 156E2(=15600), 45_234_287 (=45234287)

实数

1.335, 88_670_551.453_909(=88670551.453909), 1.0, 44.99E-2(=0.4499)

2.3 VHDL文字规则

2.3.1 数字

```
SIGNAL d1,d2,d3,d4,d5, : INTEGER RANGE 0 TO 255;--全部定义为整数类型
d1 <= 10#170# ;           --向d1赋值10#170#(十进制表示, 等于 170)
d2 <= 16#FE# ;           -- (十六进制表示, 等于 254)
d3 <= 2#1111_1110#;      -- (二进制表示, 等于 254)---也是整数类型!
d4 <= 8#376# ;           -- (八进制表示, 等于 254)
d5 <= 16#A#E3 ;         -- (十六进制表示, 等于16#A000#)
```

文字示例

物理量文字

60s (60秒), 100m (100米), k (千欧姆), 177A (177安培)

2.3 VHDL文字规则

2.3.2 字符串

```
data1 <= B"1_1101_1110"      -- 二进制数数组，位矢数组长度是 9 位
data2 <= O"15"                -- 八进制数数组，位矢数组长度是 6 位
data3 <= X"AD0"               -- 十六进制数数组，位矢数组长度是 12
data4 <= B"101_010_101_010"  -- 二进制数数组，位矢数组长度是 12
data5 <= "101_010_101_010"   -- 表达错误，缺 B。
data6 <= "101010101010"      -- 表达正确。这里可以略去 B，但不可加下划线
data6 <= "0AD0"              -- 表达错误，缺 X。
```

2.3.3 关键词

2.3 VHDL文字规则

2.3.4 标识符及其表述规则

- ★ 有效的字符：包括 26 个大小写英文字母，数字包括 0~9 以及下划线 “_”。
- ★ 任何标识符必须以英文字母开头。
- ★ 必须是单一下划线 “_”，且其前后都必须有英文字母或数字。
- ★ 标识符中的英语字母不分大小写。
- ★ 允许包含图形符号(如回车符、换行符等)，也允许包含空格符。

Decoder_1 , FFT , Sig_N , Not_Ack , State0 , Idle

<code>_Decoder_1</code>	-- 起始为非英文字母
<code>74LS164</code>	-- 起始为数字
<code>Sig_#N</code>	-- 符号“#”不能成为标识符的构成
<code>Not-Ack</code>	-- 符号“-”不能成为标识符的构成
<code>RyY_RST_</code>	-- 标识符的最后不能是下划线“_”
<code>data__BUS</code>	-- 标识符中不能有双下划线
<code>return</code>	-- 关键词

2.3 VHDL文字规则

2.3.5 文件取名和存盘

2.3.6 规范的程序书写格式

2.4 VHDL数据对象

2.4.1 常数

```
CONSTANT  常数名:数据类型 := 表达式 ;
```

```
CONSTANT  FBT : STD_LOGIC_VECTOR := "010110" ; --定义常数为标准位矢类型  
CONSTANT  DATAIN : INTEGER := 15 ; --定义常数为整数类型
```

2.4 VHDL数据对象

2.4.2 变量

```
VARIABLE 变量名 : 数据类型 := 初始值 ;
```

```
VARIABLE a : INTEGER RANGE 0 TO 15; --变量a定义为整数类型，取值范围是0~15  
VARIABLE d : STD_LOGIC := '1'; --变量d定义为标准逻辑位数据类型，初始值是1
```

```
目标变量名 := 表达式 ;
```

```
VARIABLE x, y : INTEGER RANGE 15 DOWNT0 0; --分别定义变量 x 和 y 为整数类型  
VARIABLE a, b : STD_LOGIC_VECTOR(7 DOWNT0 0) ;  
x := 11 ; --整数直接赋值，这是因为 x 的类型是整数类型  
y := 2 + x ; --运算表达式赋值，y 也是整数变量  
a := b; --b 向 a 赋值  
a(5 DOWNT0 0) := b(7 DOWNT0 2) ; --位矢量类型赋值
```

2.4 VHDL数据对象

2.4.3 信号

```
SIGNAL 信号名: 数据类型 := 初始值 ;
```

目标信号名 <= 表达式 AFTER 时间量; -- AFTER是关键词

```
PROCESS (a, b, c) BEGIN
    y <= a + b ;
    z <= c - a ;
    y <= b ;
END PROCESS ;
```

```
PROCESS (a, b) BEGIN
    y <= a + b ;
END PROCESS ;
PROCESS (c, d ) BEGIN
    z <= c - d ;
    y <= d ;
END PROCESS ;
```



```
SIGNAL e : STD_LOGIC;
```


习 题

2-2

```
ENTITY buf3s IS                                -- 实体 1: 三态缓冲器
    PORT (input : IN STD_LOGIC ;              -- 输入端
          enable : IN STD_LOGIC ;            -- 使能端
          output : OUT STD_LOGIC ) ;         -- 输出端
END buf3x ;

ENTITY mux21 IS                                -- 实体 2: 2 选 1 多路选择器
    PORT (in0, in1, sel : IN STD_LOGIC;
          output : OUT STD_LOGIC) ;
```