第二章 VHDL基本结构

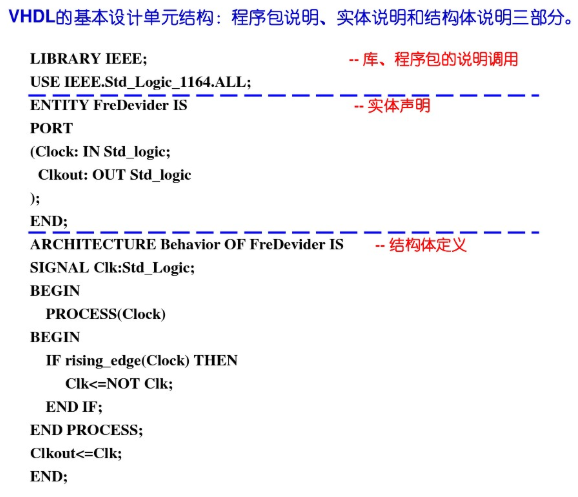
实体（Entity）：描述所设计的系统的外部接口信号，定义电路设计中所有的输入和输出接口；

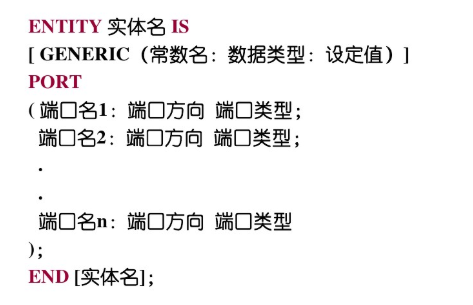
结构体（Architecture）：描述系统内部的结构和行为；

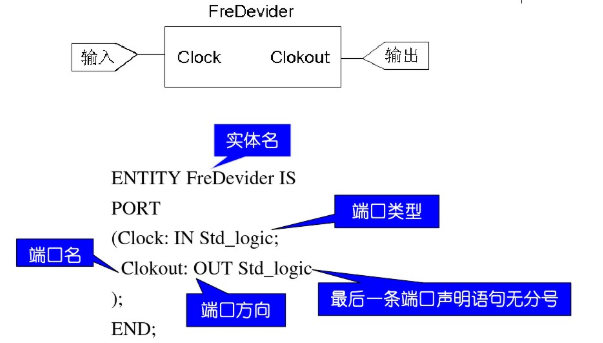
包集合（Package）：存放各设计模块能共享的数据类型、常数和子程序等；

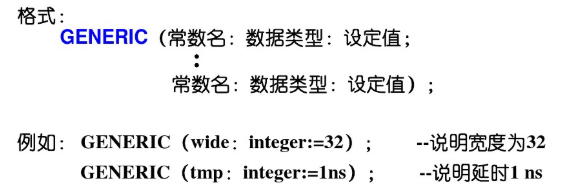
配置（Configuration）：指定实体所对应的结构体；

库（Library）：存放已经编译的实体、结构体、包集合和配置。

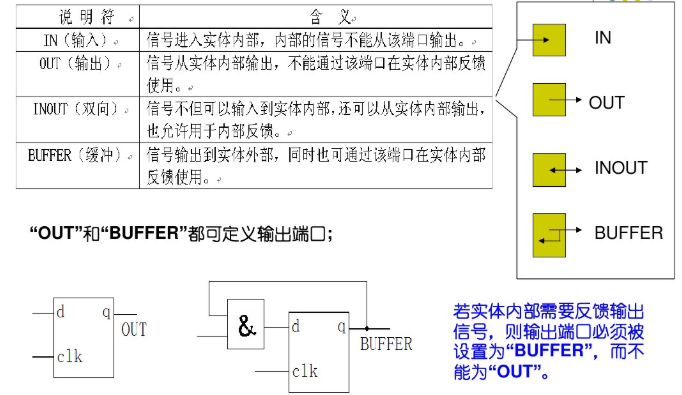
例：

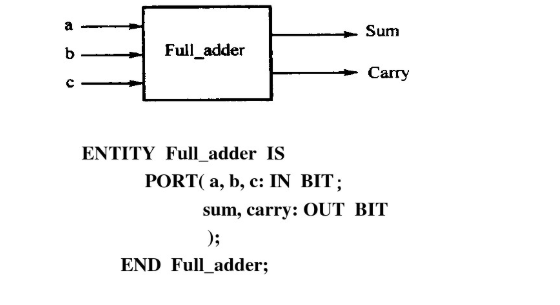
2.1 实体

实体名由设计者自由命名，用来表示被设计电路芯片的名称，但是必须与VHDL程序的文件名称相同。要与文件名一致：

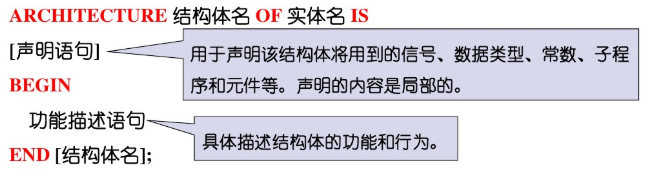
类属为设计实体与外界通信的静态信息提供通道，用来规定端口的大小、实体中子元件的数目和实体的定时特性等。

端口方向：IN, OUT, INOUT, BUFFER



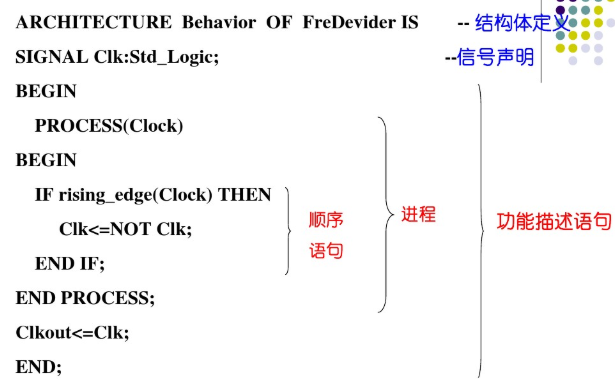
同方向、同类型的端口可以放在同一个说明语句中。

* 1. 结构体

结构体定义了设计单元的具体功能，描述了该基本单元的行为、元件和内部的连接关系。

一个实体可对应多个结构体，每个结构体代表该实体功能的不同实现方案或不同实现方式。同一时刻只有一个结构体起作用，通过CONFIGURATION决定用哪个结构体进行仿真或者综合。

在结构体描述中，具体给出了输入、输出信号之间的逻辑关系。

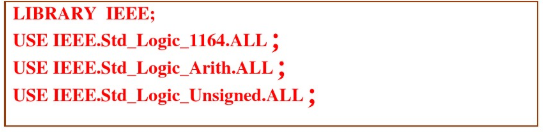


* 1. 库、程序包的调用

LIBRARY 库名；

USE 库名.程序包名.项目名；

LIBRARY IEEE;

USE IEEE.Std\_Logic\_1164.ALL;（调用此程序包中所有的资源）

VHDL程序测试文件的编写技巧：  
先来看一个例子（六进制计数器）

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

--use ieee.std\_logic\_unsigned.all;

entity cnt6 is

port

(clr,en,clk :in std\_logic;

q :out std\_logic\_vector(2 downto 0)

);

end entity;

architecture rtl of cnt6 is

signal tmp :std\_logic\_vector(2 downto 0);

begin

process(clk)

-- variable q6:integer;

begin

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

if(clr='0')then

tmp<="000";

elsif(en='1') then

if(tmp="101")then

tmp<="000";

else

tmp<=unsigned(tmp)+'1';

end if;

end if;

end if;

q<=tmp;

-- qa<=q(0);

-- qb<=q(1);

-- qc<=q(2);

end process;

end rtl;

测试文件：  
library ieee;

use ieee.std\_logic\_1164.all;

entity cnt6\_tb is

end cnt6\_tb;

architecture rtl of cnt6\_tb is

component cnt6

port(

clr,en,clk :in std\_logic;

q :out std\_logic\_vector(2 downto 0)

);

end component;

signal clr :std\_logic:='0';

signal en :std\_logic:='0';

signal clk :std\_logic:='0';

signal q :std\_logic\_vector(2 downto 0);

constant clk\_period :time :=20 ns;

begin

instant:cnt6 port map

(

clk=>clk,en=>en,clr=>clr,q=>q

);

clk\_gen:process

begin

wait for clk\_period/2;

clk<='1';

wait for clk\_period/2;

clk<='0';

end process;

clr\_gen:process

begin

clr<='0';

wait for 30 ns;

clr<='1';

wait;

end process;

en\_gen:process

begin

en<='0';

wait for 50ns;

en<='1';

wait;

end process;

end rtl;

VHDL语言的testbench有一套固定的格式，总结如下：  
--测试平台文件的基本结构

library ieee;

use ieee.std\_logic\_1164.all;

entity test\_bench is --测试平台文件的空实体（不需要端口定义)

end test\_bench;

architecture tb\_behavior of test\_bench is

component entity\_under\_test --被测试元件的声明

port(

list-of-ports-theri-types-and-modes

);

end component;

begin

instantiation:entity\_under\_test port map

(

port-associations

);

process() --产生时钟信号

……

end process;

process() --产生激励源

……

end process;

end tb\_behavior;

激励信号产生方式：

1. 以一定的离散时间间隔产生激励信号的波形；
2. 基于实体的输出响应产生激励信号

两种常用的复位信号

--1.周期性的激励信号，如时钟

--2.时序变化的激励型号，如复位

--eg.产生不对称时钟信号

w\_clk<='0' after period/4 when w\_clk='1' else

'1' after 3\*period/4 when w\_clk='0' else

'0';

--eg.产生堆成时钟信号，process语句

clk\_gen1:process

constan clk\_period := 40 ns;

begin

clk='1';

wait for clk\_period/2;

clk='0';

wait for clk\_period/2;

end process;