第二章 VHDL代码结构

- VHDL代码基本单元
- 库声明
  - 库的种类
  - 库的声明
- 实体
- 构造体
- 例题

VHDL代码基本单元

一段独立的VHDL代码至少包括三个部分：

- 库（Library）声明：如ieee, std, work等
- 实体（Entity）：I/O Pin
- 构造体（Architecture）或结构体：具体描述电路所要实现的功能

一、 库

库是一些常用代码的集合。

- 将常用代码存放到库中有利于设计的复用和代码共享，也可使代码结构更清晰；
- 库的典型结构：

一个库的基本组成部分

1.1、库的种类

VHDL有3个常用的库：

1）IEEE 库

定义了四个常用的程序包：

- std_logic_1164（std_logic （8值逻辑）& std_ulogic（9值逻辑））
- std_logic_arith（signed、unsigned数据类型的算术、比较运算函数）
- std_logic_signed（std_logic_vector类型数据的一些signed运算操作函数）
- std_logic_unsigned（std_logic_vector类型数据的一些unsigned运算操作函数）

2）STD 库（默认库）

VHDL的标准资源库，包括数据类型和输入/输出文本等内容。库中包集有：standard和textio。

3）WORK库（默认库）

当前工作库，当前设计的所有代码都存放在work库中，无需声明。

1.2、库的声明

在使用库之前，首先需要对库进行声明。

用关健字library说明要使用的库，用关健字 use 说明要使用的库中的程序包。

```
LIBRARY  library_name;
USE library_name.package_name.package_parts;
```

库的声明总是放在实体单元前面，默认库可不作说明。

库的作用范围：仅限于所说明的设计实体。

每一个设计实体都必须有自己完整的库说明语句。

库中常用的3个包集：

- ieee.std_logic_1164(ieee库)
- standard(std库)
- work

例：

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

二、实体（ENTITY）

实体：定义电路的输入输出端口或引脚。

端口声明：确定输入、输出端口的数目和类型。

语法结构：

```
ENTITY <entity_name> IS
    Port （
        port_name1{，port_name2}: signal_mode signal_type;
        port_name3: signal_mode signal_type;
          …）
END <entity_name>;
```

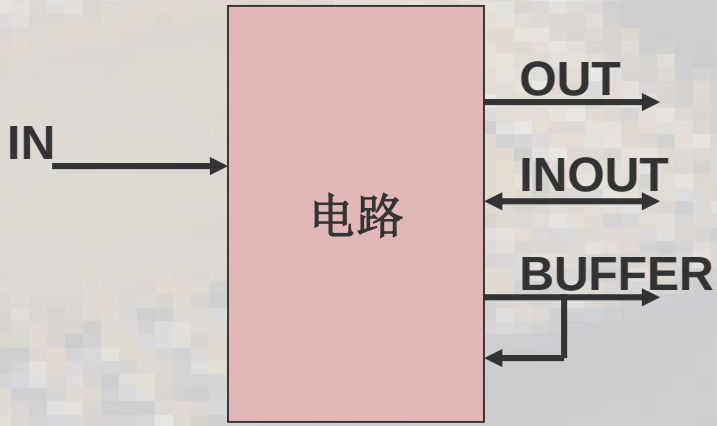
- 端口的信号模式（signal_mode）：

  in：输入型，单向引脚，只读。

  out：输出型，单向引脚，不能供电路内部使用。

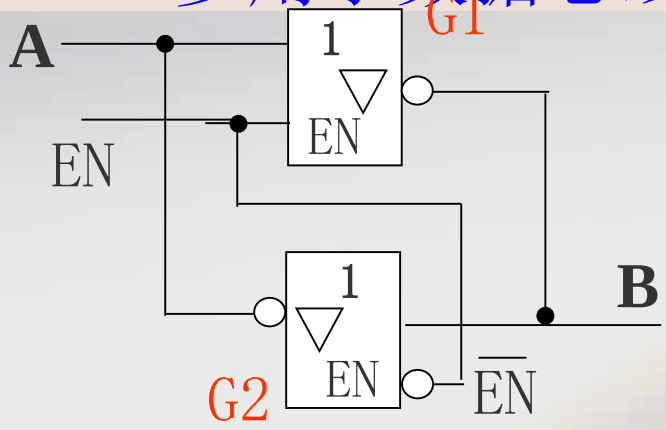
  inout:输入输出型，双向引脚，可读也可写。

  buffer：缓冲型，与 out 相似，但可供电路内部使用。



inout的典型结构：

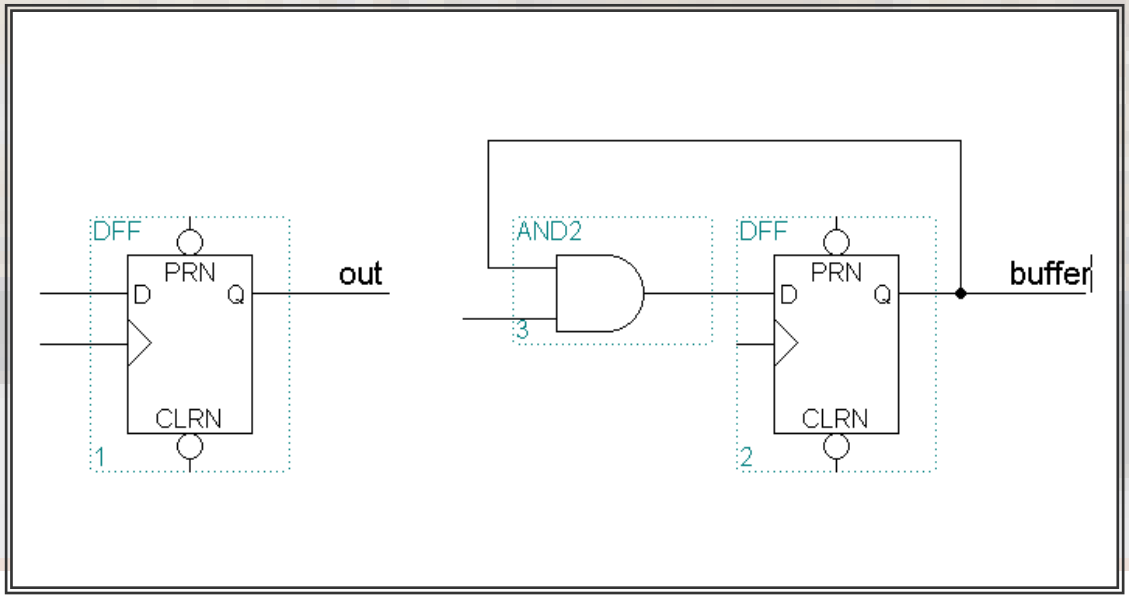
多用于数据总线的设计，节省引脚数目！



工作原理

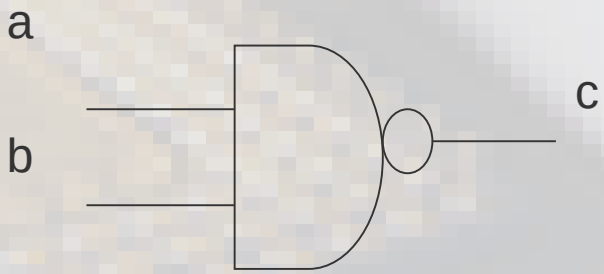
EN	G1	G2	传输结果
1	工作	高阻态	$B=\overline{A}$
0	高阻态	工作	$A=\overline{B}$

out和 buffer 的典型结构：用法如P112页out1/out2的使用



- 信号的类型：BIT、STD_LOGIC、INTEGER等；
- 实体ENTITY的命名：不要与VHDL关键字冲突。
- 例子：　基本的与非门---纯组合逻辑

```
ENTITY nand_gate IS
    PORT (
        a, b: IN BIT;
        x: OUT BIT  );
END nand_gate;
```



三、构造体（ARCHITECTURE）或结构体

作用：描述电路行为和实现功能，如定义元件及内部的连接关系。

两个组成部分：

- 声明部分（可选），对数据类型、常量、信号等元素进行声明。
- 代码部分（BEGIN与END之间）：描述电路的行为或功能。

名称：不能与VHDL关键字冲突，可与ENTITY同名；

构造体的语法：

```
architecture  结构体名称  of  实体名称  is
      [声明语句]内部信号、常数、
       数据类型、子程序（函数、过程）、
       元件等的说明;
  begin
       （代码）;
  end 结构体名称;
```

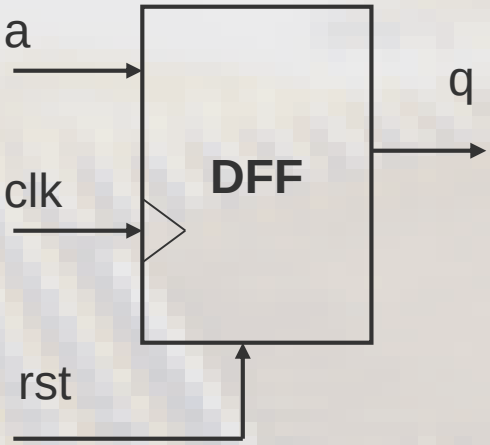
注：同一实体的结构体不能同名。定义语句中的常数、信号不能与实体中的端口同名。

- 构造体举例：

```
ARCHITECTURE myarch OF nand_gate IS
BEGIN
        x<=a NAND b;
END myarch;
```

例题

- 例1 带有异步复位端的D触发器—纯时序逻辑



功能描述：

•rst=‘1’时，输出q置低电平；

•否则，时钟信号上升沿时输入的值传递给输出q；

注意：时序电路→电路随着时钟节拍一步一步地顺序工作→顺序执行的代码

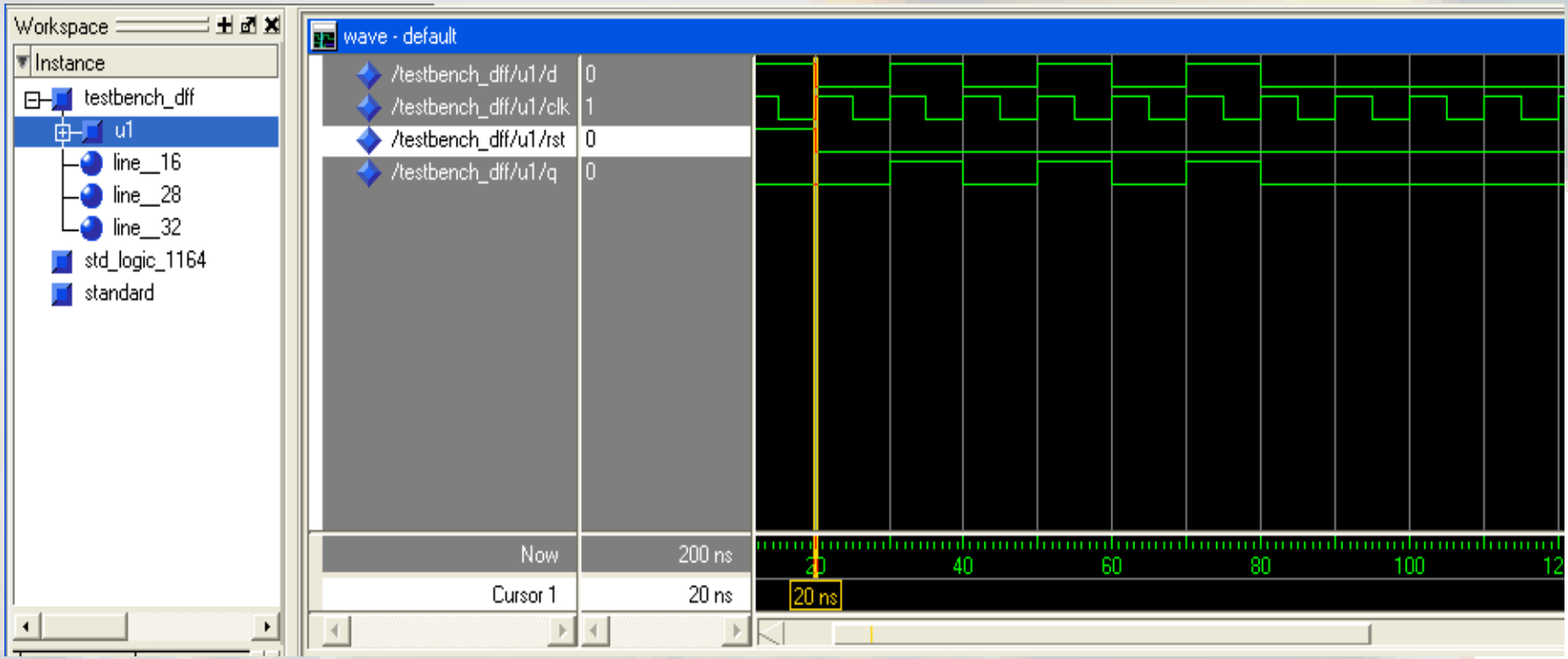
```
1 ----------------------------------------------------------------
2 LIBRARY ieee;
3 USE ieee.std_logic_1164.all;
4 ----------------------------------------------------------------
5 ENTITY dff   IS
6    PORT(d, clk, rst: IN STD_LOGIC;
7                q: OUT STD_LOGIC);
8 END dff;
9 ----------------------------------------------------------------
10 ARCHITECTURE behavior OF dff  IS
11 BEGIN
12      PROCESS (rst, clk)
13      BEGIN
14           IF (rst='1') THEN
15              q<='0';
16          ELSIF ( clk'EVENT AND clk='1' ) THEN
17              q<=d;
18          END IF;
19    END PROCESS;
20 END behavior;
21----------------------------------------------------------------
```

注意：VHDL不区分大小写！

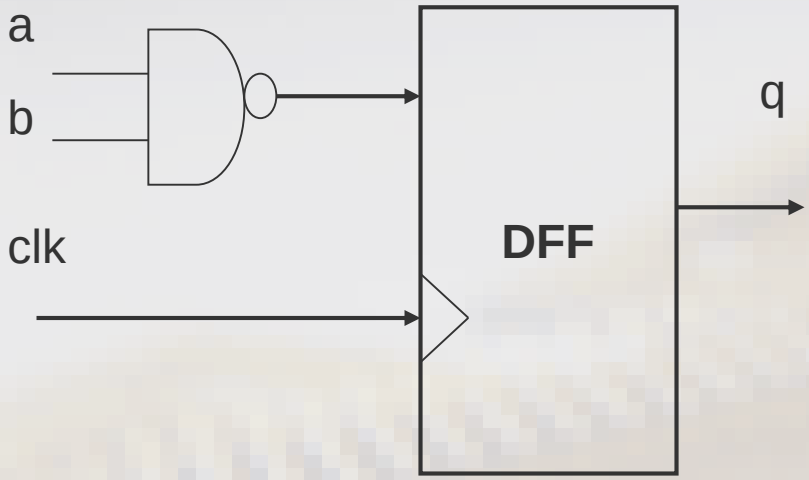
testbench代码：

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

entity testbench_dff is
end testbench_dff;

architecture behavior of testbench_dff is

component dff is
   port(d,clk,rst:in std_logic;
        q:out std_logic);
end component;

signal d,clk,rst,q:std_logic;

begin

   u1: dff port map(d,clk,rst,q);

   process
   begin
      clk<='0';
      wait for 10 ns;
      loop
         clk<=not clk;
         wait for 5 ns;
      end loop;
   end process;

d<='0','1' after 10 ns, '0' after 20 ns,
    '1' after 30 ns, '0' after 40 ns,
    '1' after 50 ns, '0' after 60 ns,
    '1' after 70 ns, '0' after 80 ns;
 rst<='1' after 10 ns,'0' after 20 ns;

end behavior;
```

例子2.1的仿真波形图


Workspace
Instance
testbench_dff
u1
line__16
line__28
line__32
std_logic_1164
standard
wave - default
/testbench_dff/u1/d
/testbench_dff/u1/clk
/testbench_dff/u1/rst
/testbench_dff/u1/q
0
1
0
0
Now
200 ns
Cursor 1
20 ns
20 ns
40
60
80
100

例2 D触发器+与非门---组合逻辑与时序逻辑相结合的电路



功能描述：

- 无复位端rst；
- 时钟信号上升沿时a NAND b的结果传递给输出q；

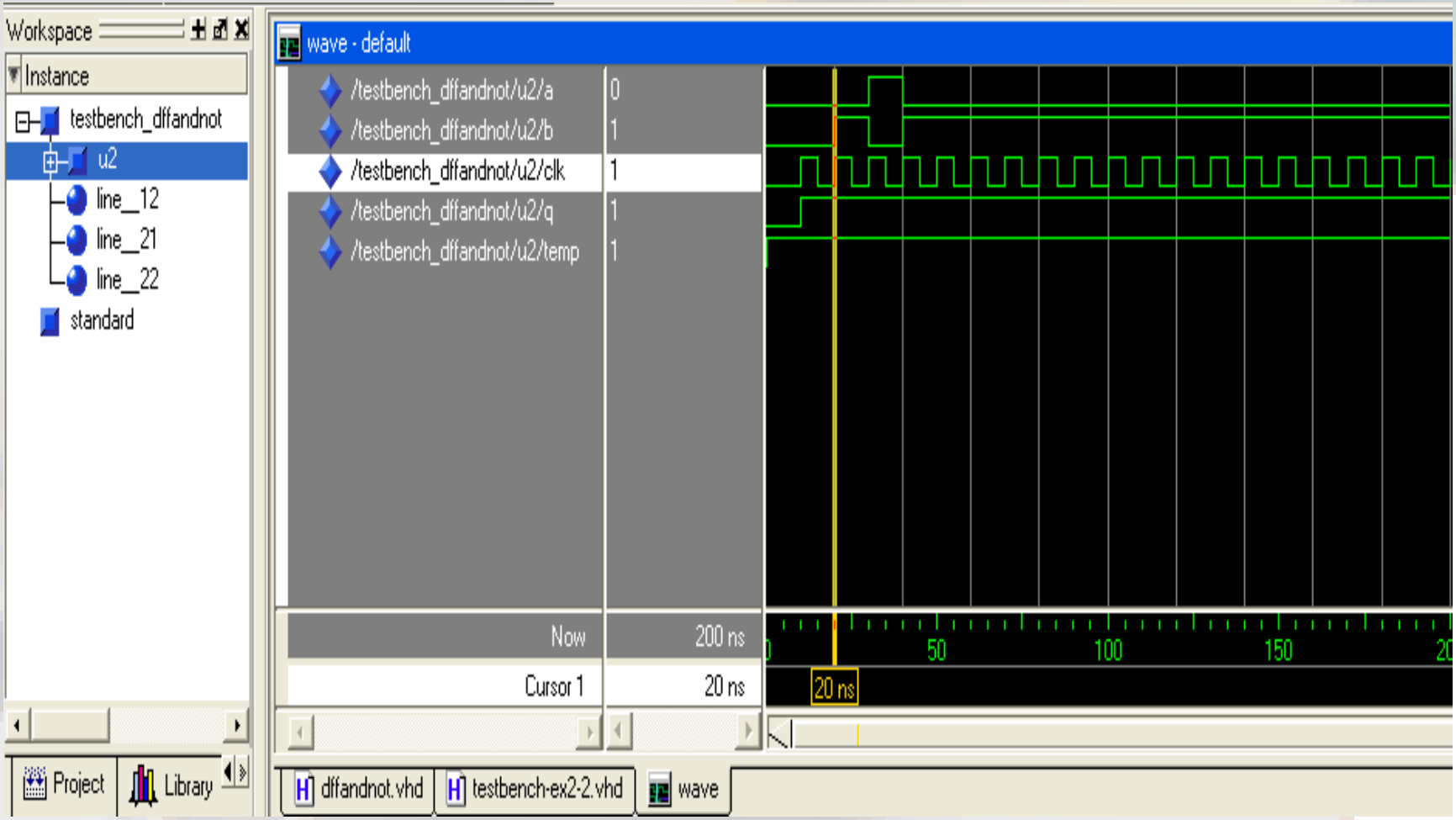
```
-------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.all;
1 -------------------------------------------------------------------
2 ENTITY example   IS
3       PORT (a, b, clk: IN BIT;
4                   q: OUT BIT);
5 END example;
6 -------------------------------------------------------------------
7 ARCHITECTURE example OF example  IS
8      SIGNAL temp: BIT;
9 BEGIN
10     temp<= a NAND b;
11     PROCESS (clk)
12       BEGIN
13          IF (clk'EVENT AND clk='1') THEN     q<= temp ;
14        END IF;
15    END PROCESS;
16 END example;
17 -------------------------------------------------------------------
```

并发执行

testbench代码：

```
entity testbench_dffandnot is
end testbench_dffandnot;
architecture behavior of testbench_dffandnot is
    component dffandnot is
        port(a,b,clk:in bit;
         q:out bit);
    end component;
    signal a,b,clk,q:bit;
begin
   u2: dffandnot port map(a,b,clk,q);
    process
    begin
      clk<='0';
      wait for 10 ns;
      loop
         clk<=not clk;
         wait for 5 ns;
      end loop;    end process;
    a<='0' after 10 ns,'1' after 30 ns,'0' after 40 ns;
    b<='0' after 10 ns,'1' after 20 ns,'0' after 30 ns,'1' after 40 ns;
end behavior;
```

例子2.2的仿真波形图


Workspace
Instance
testbench_dffandnot
u2
line__12
line__21
line__22
standard
Project
Library
wave - default
/testbench_dffandnot/u2/a
/testbench_dffandnot/u2/b
/testbench_dffandnot/u2/clk
/testbench_dffandnot/u2/q
/testbench_dffandnot/u2/temp
0
1
1
1
1
Now
200 ns
Cursor 1
20 ns
20 ns
50
100
150
dffandnot.vhd
testbench-ex2-2.vhd
wave

第2章　思考题

1. VHDL中最基本的结构是什么？其作用各是什么？
2. 说明inout、out 和 buffer有何异同点。

课后作业：

2.1（a）

2.1（b）