EDA（Electronic Design Automation）—VHDL及数字电路设计

任课教师：但果

单位：深圳大学医学院

Email: danguo@szu.edu.cn

本课程安排：

学时：54学时（课堂教学18学时，上机实验36学时）

课堂教学内容：

第一课、引言
第二课、VHDL代码结构
第三课、数据类型
第四课、运算操作符和属性
第五课、并发代码
第六课、顺序代码
第七课、信号与变量
第八课、状态机
第九课、包集和元件
第十课、函数和过程
第十一课、复习与答疑

教学目的：

- 了解数字集成电路的结构特点；
- 了解数字集成系统的基本设计方法；
- 掌握VHDL的基本语法和主要编程要点。

实验教学内容及要求：

分5次共10学时。

实验一：学习安装、使用MODELSIM 6.0D，并根据testbench PPT中的两个例子，能够仿真出波形；

实验二：（1）用GENERIC语句改写例4.1，设计成通用译码器，要求书写tb代码并仿出波形；（2）课后习题5.6，要求书写tb代码并仿出波形；（3）课后习题6.1，要求书写tb代码并仿出波形；

实验三：例子7.5、7.8、8.6，要求书写tb代码并仿出波形；

实验四：第9章例题9.2、9.6、9.9，要求书写tb代码并仿出波形；

实验五：选做例子串-并型乘法器、并行乘法器、乘-累加电路中的两个，要求书写tb代码并仿出波形；

实验课要求

- 掌握 Modelsim仿真工具，从简单的电路设计入手，到最后能够设计比较复杂的电子系统，培养设计电路系统的实际动手能力。

- **实验教学目的：**
  - 掌握常用EDA工具的基本使用方法，掌握常用数字电路的设计特点。

考核方式

- 内容：基本概念与基本功能器件的设计编程
- 方式：平时作业、上机与试卷相结合
- 平时作业与上机：50%
- 考试试卷：50%

教材及参考资料

教材：

- VHDL数字电路设计教程，Volnei A. Pedroni著，乔庐峰等译，电子工业出版社

参考资料：

1. Altera FPGA/CPLD设计（基础篇），王诚等著，人民邮电出版社。
2. Altera FPGA/CPLD设计（高级篇），王诚等著，人民邮电出版社。
3. FPGA/VHDL快速工程实践入门与提高，杨恒、卢飞成编著，北京航空航天大学出版社。

相关网址：

www.xinlinx.com

www.altera.com

www.fpga.com.cm

www.edaclub.net

www.edachina.com

第一章 引言

- VHDL的历史
- VHDL的作用
- VHDL的语言特点
- VHDL与其它硬件描述语言的比较
- VHDL设计概述
- 从VHDL代码到电路的转化

VHDL的历史

- VHDL：VHSIC (Very High Speed Integrated Circuit) Hardware Description Language

1. 80年代初由美国国防部在实施超高速集成电路（VHSIC）项目时开发的。
2. 1987年由 IEEE 协会批准为 IEEE 工业标准，称为IEEE1076-1987。 各EDA公司相继推出支持VHDL的设计环境。
3. 1993年被更新为 93 标准，即IEEE1076-1993。进一步提高抽象描述层次，扩展系统描述能力。

VHDL的作用

1. VHDL打破软、硬件的界限。传统的数字系统设计分为：
   - 硬件设计（硬件设计人员）
   - 软件设计（软件设计人员）
2. 采用文本形式进行程序设计，包含许多具有硬件特征的语句，主要用于描述数字系统的结构、功能、行为和接口，能够支持电路硬件的设计、验证、综合和测试；设计与具体工艺无关，适合于多层次大规模设计，具有良好的开放性和并行设计能力、便于交流保存共享。
3. VHDL是电子系统设计者和 EDA工具之间的界面。EDA工具及 HDL的流行，使电子系统向集　成化、大规模和高速度等方向发展。美国硅谷约有80%的 ASIC和 FPGA/CPLD已采用HDL进行设计。

VHDL的语言特点

1. VHDL具有强大的语言结构，系统硬件描述能力强、设计效率高；具有较高的抽象描述能力。
2. VHDL语言可读性强，易于修改和发现错误。
3. VHDL具有丰富的仿真语句和库函数。
4. VHDL源代码进行早期功能仿真，有利于大系统的设计与验证。
5. VHDL设计与硬件电路关系不大。

VHDL的语言特点

6. VHDL设计不依赖于器件，与工艺无关 。
7. 可移植性好。
8. VHDL体系符合TOP-DOWN和CE（并行工程）设计思想。
9. VHDL设计效率高，产品上市时间快，成本低。
10. 易于ASIC实现。

VHDL与其他硬件描述语言的比较

（RTL: Register Translate Level，寄存器传输级）

VHDL：

- 具有较强的系统级抽象描述能力，适合行为级和RTL级的描述。设计者可不必了解电路细节，所作工作较少，效率高。但对综合器的要求高，不易控制底层电路的生成。IEEE标准，支持广泛。

VHDL与其他硬件描述语言的比较

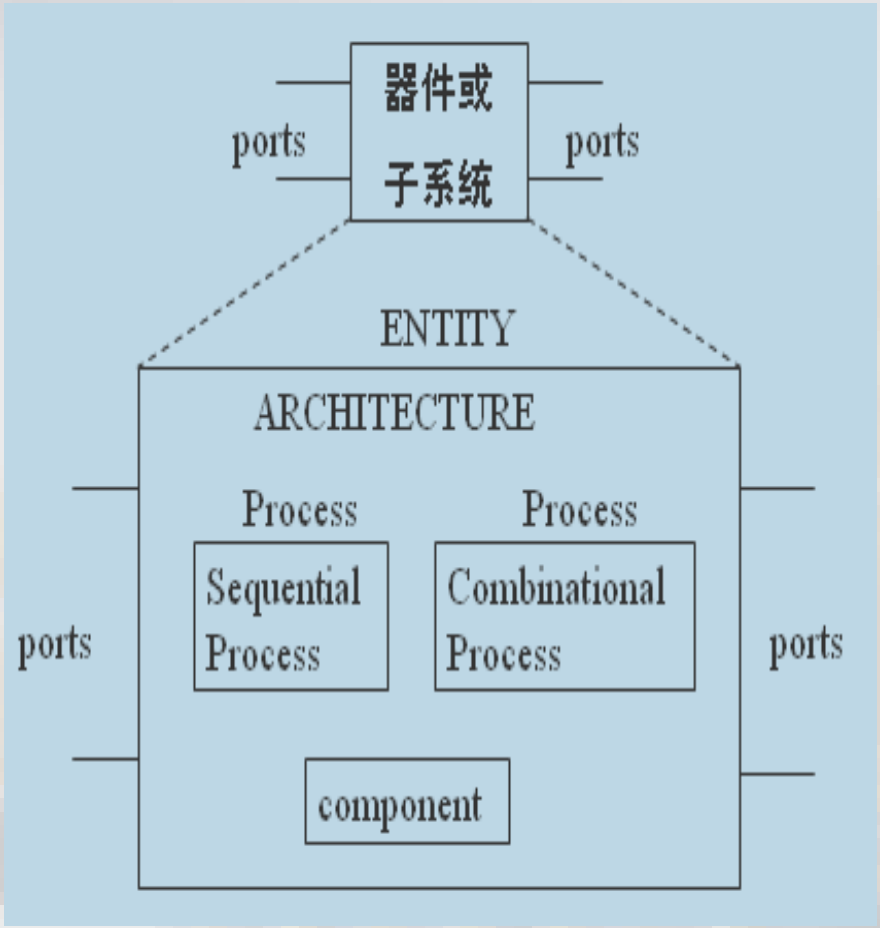
- Verilog HDL：

系统级抽象描述能力比VHDL稍差；门级开关电路描述方面比 VHDL 强。适合 RTL级和门电路级的描述。设计者需要了解电路细节，所作工作较多。IEEE标准，支持广泛。

- ABEL、PALASM、AHDL(Altera HDL)：

系统级抽象描述能力差，一般作门级 电路描述。要求设计者对电路细节有详细的了解。对综合器的性能要求低，易于控制电路资源、支持少。

VHDL设计简述

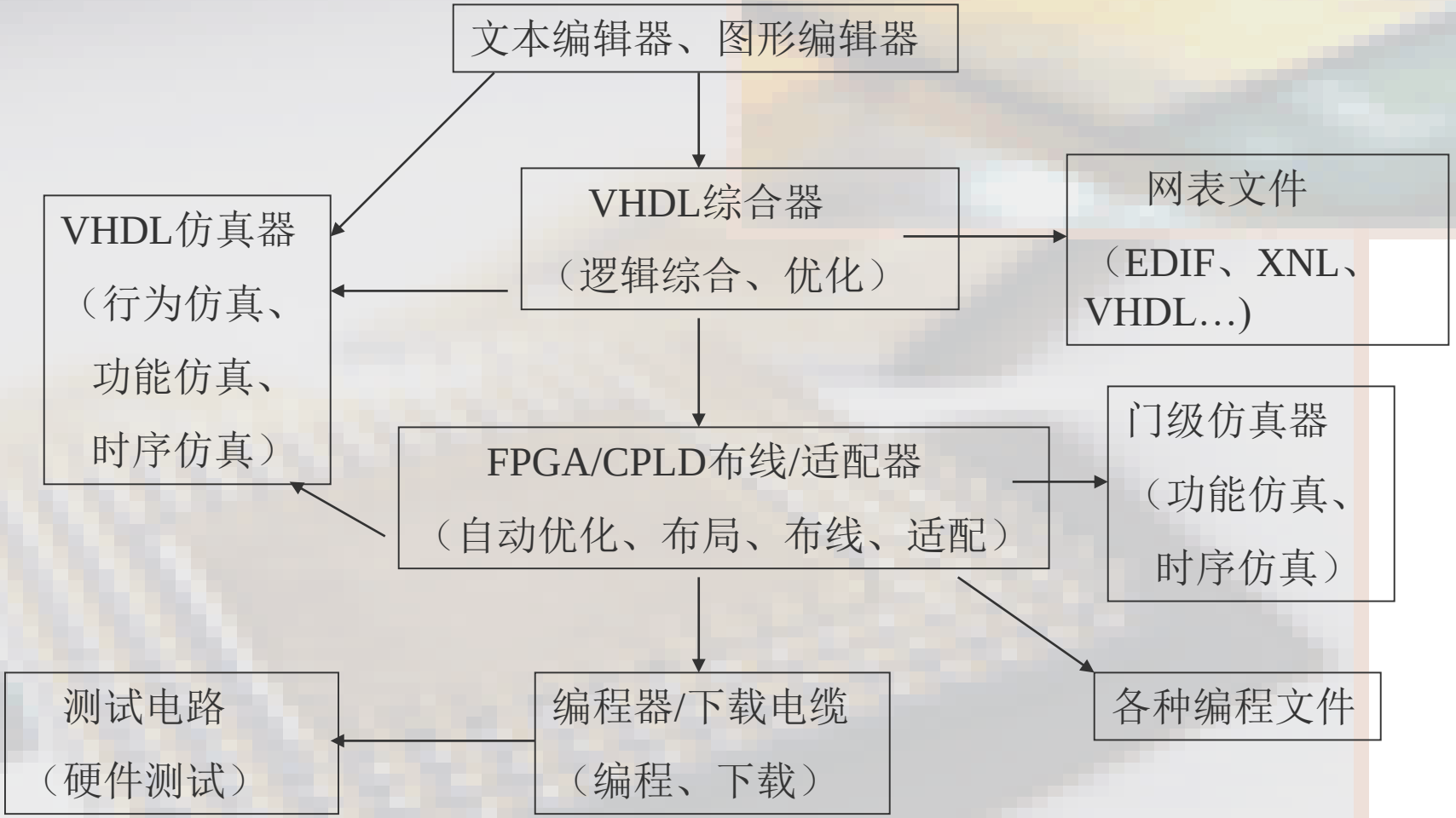


VHDL主要用于描述数字系统的结构、行为、功能和接口。

VHDL将一个设计（元件、电路、系统）分为：

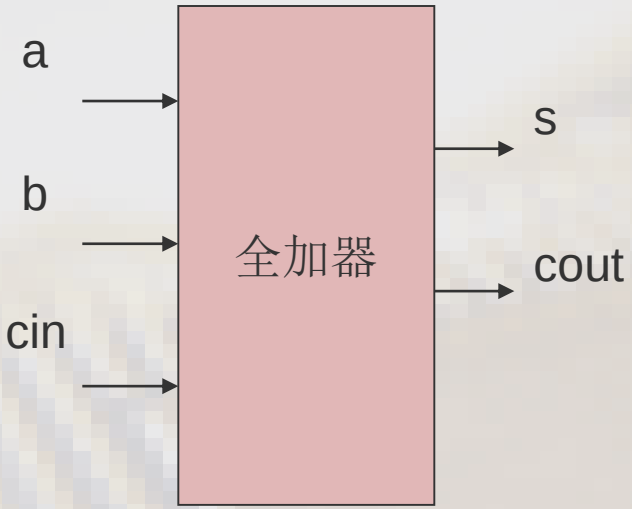
- **外部（可视部分、端口）**
- **内部（不可视部分、内部功能、算法）**

VHDL的FPGA/CPLD工程设计流程



从VHDL代码到电路的转化

- 一位全加器



电路外部框图

a	b	cin	s	cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

功能分析：真值表方法

逻辑表达式：

Cout=a.b+a.cin+b.cin

S=a⊕ b⊕ cin

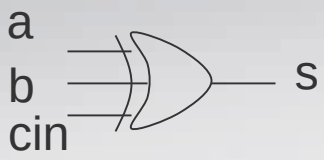
用VHDL描述一位全加器的功能：

（full_adder.vhd）

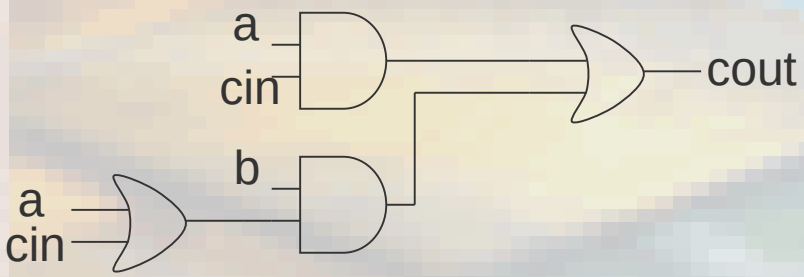
```
ENTITY full_adder IS
PORT (a, b, cin: IN BIT;
          s, cout: OUT BIT);
END full_adder;

ARCHITECTURE dataflow OF full_adder IS
BEGIN
     s<=a XOR b XOR cin;
    cout<=(a AND B) OR (a AND cin) OR (b AND cin);
END dataflow;
```

*文件名、实体名与结构名要一致！

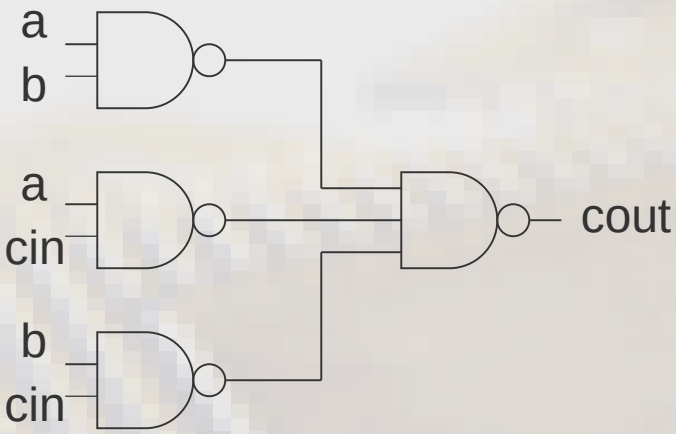
不同的编译器类型、电路优化目标、实现方式都将影响最终的电路结构。



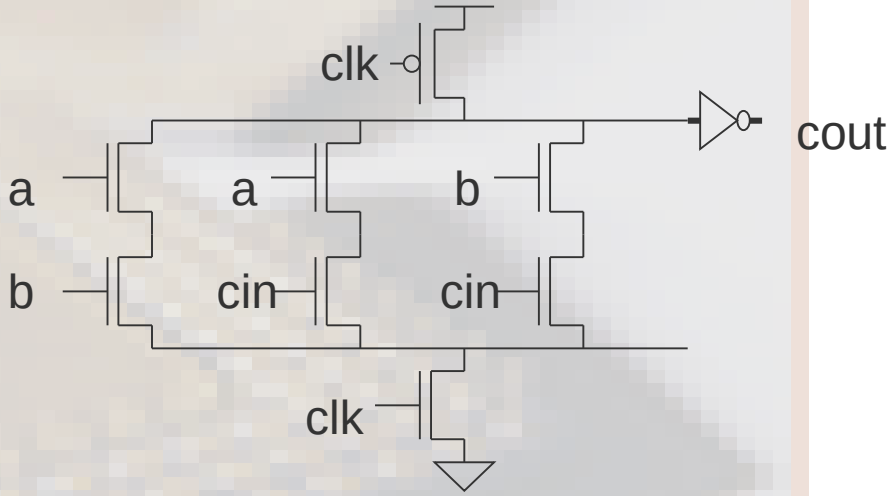
(a)



(b)可编程器件实现方式一



(d)可编程器件实现方式二



(d) 一种ASIC的实现方式

仿真波形见书P6.

课后思考题：

1、数字集成电路的设计特点是什么？

2、数字集成电路的设计流程是怎样的？

3、EDA技术的含义和内容是什么？

4、VHDL的设计流程是什么？

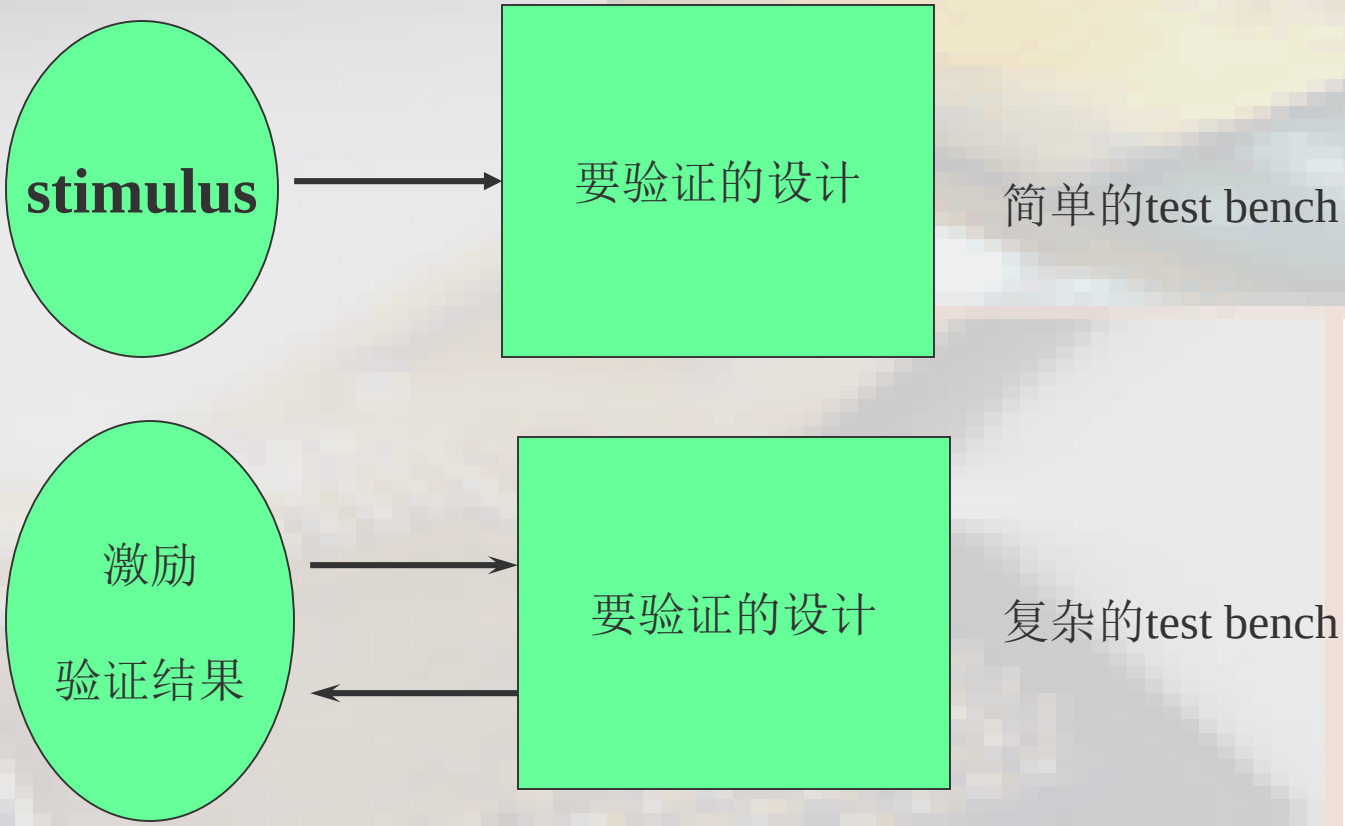
5、在VHDL代码到电路的转化过程中需要做出哪些选择？

6、阅读书后“附录A”！

Modelsim 6.0的使用步骤

- 打开Modelsim 6.0
- 新建一个工程，File->new->project;
- 往project里添加源文件。分为两种：一种是目标代码，另一种是测试代码testbench;
- 添加一个work库，File->new->library;
- 编译源文件，Compile->Compile all;
- 开始仿真，Simulate->star simulation;
- 选择testbench文件作为top-level文件；
- 查看波形，Add->wave
- 单击run图标
- 调试

testbench的组织



- 简单的test bench向要验证的设计提供向量，人工验证输出。
- 复杂的test bench是自检测的，其结果自动验证。

施加激励

产生激励并加到设计的方法：将component实例化，并对其施加一些输入值和时间量。

具体做法：

- 主代码作为一个单独的x_main.vhd文件；
- testbench代码作为另外一个.vhd文件，并且在其中将主代码x_main中的实体声明为component；
- 在testbench代码中对component的输入端口赋一些初值，并用wait for x_time来给出时序关系。

Tb代码的基本特点

1. Tb代码本身不生成具体的电路，仅供仿真之用；
2. 模板固定，记住即可：
   - 其entity中无需定义in/out port，原因：tb代码的输出信号往往是由设计者手工直接提供，而不是由一组tb的输入信号经过功能运算自动生成的；
   - 需使用process语句，因为所提供的测试用例都是时间相关的，是顺序执行的；
   - 由于无需输入信号，因此其process的敏感信号列表为空；
   - 测试用例往往是手工赋值的，且不生成具体电路，因此不需要使用generic语句来设计多种规格的测试用例；

例1：使用元件实例化方法编写的testbench

```
----目标文件fulladder.vhd---
ENTITY fulladder IS
    PORT (
      a ,b,ci: IN bit;
      co,s: OUT bit);
END fulladder;

ARCHITECTURE rtl OF fulladder
IS
BEGIN
  s<=a xor b xor ci;
  co<=(a and b) or (a and ci) or (b
and ci);
end rtl;
```

```
----测试文件testadder.vhd-
-
entity fa_testbench is
end fa_testbench;
architecture beh of
      fa_testbench is
component fulladder
  port (a,b,ci: in bit;
       s,co : out bit);
end component;
signal xt,yt,zt,st,cot:bit;
```

```
begin
 u1: fulladder port map (xt,yt,zt,st,cot);
 process
 begin
   xt<='0';yt<='0';zt<='0'; wait for 10 ns;
   xt<='0';yt<='0';zt<='1'; wait for 10 ns;
   xt<='0';yt<='1';zt<='0'; wait for 10 ns;
   xt<='0';yt<='1';zt<='1'; wait for 10 ns;
   xt<='1';yt<='0';zt<='0'; wait for 10 ns;
   xt<='1';yt<='0';zt<='1'; wait for 10 ns;
   xt<='1';yt<='1';zt<='0'; wait for 10 ns;
   xt<='1';yt<='1';zt<='1'; wait for 10 ns;
   xt<='0';yt<='0';zt<='0'; wait for 10 ns;
  end process;
end beh;
```

同步时序电路的testbench中clk的典型写法

```
        ........
process
    begin
     clk <= '0';
     wait for 12 ns;
     loop
      clk <= not clk;
      wait for 7 ns;
     end loop;
end process;
        ..........
```

等待时钟信号开始工作的时间

半个时钟周期的长度

注意：

- 需要将端口信号全部定义为signal；
- 电路内部的一些连线（信号或变量）的赋值信息不能出现在tb中，否则将导致这些中间信号不能变化；
- 测试向量或矢量的赋值需要出现在process中，否则就被看做是并发赋值，易造成“线与”现象，导致波形错误。

- Wait for语句和after语句的区别：

  1、after表示从仿真周期的起点开始，到当前信号值发生变化时的整个时间区间，当该信号需要再次变化时，时间量只能升序。 例：

  x<=1 after 10ns; x<=2 after 20ns; x<=3 after 30ns;

  2、wait for语句表示一个信号值的保持时间长度，时间量的大小不必升序。例：

  x<=1；wait for 10ns; x<=2; wait for 5ns; x<=5; wait for 2ns;

  此外，每个信号被赋新值之后，必须要wait for 一个时间间隔，否则编译器认为该信号值持续时间为0，不能在波形上反映出来。