《EDA技术实用教程(第五版)》习题

1　习　　题

1-1 EDA技术与ASIC设计和FPGA开发有什么关系？FPGA在ASIC设计中有什么用途？P3~4

EDA技术与ASIC设计和FPGA开发有什么关系？答：利用EDA技术进行电子系统设计的最后目标是完成专用集成电路ASIC的设计和实现；FPGA和CPLD是实现这一途径的主流器件。FPGA和CPLD的应用是EDA技术有机融合软硬件电子设计技术、SoC（片上系统）和ASIC设计，以及对自动设计与自动实现最典型的诠释。

FPGA在ASIC设计中有什么用途？答：FPGA和CPLD通常也被称为可编程专用IC，或可编程ASIC。FPGA实现ASIC设计的现场可编程器件。

1-2　与软件描述语言相比，VHDL有什么特点？　P4~6

答：编译器将软件程序翻译成基于某种特定CPU的机器代码，这种代码仅限于这种CPU而不能移植，并且机器代码不代表硬件结构，更不能改变CPU的硬件结构，只能被动地为其特定的硬件电路结构所利用。

综合器将VHDL程序转化的目标是底层的电路结构网表文件，这种满足VHDL设计程序功能描述的电路结构，不依赖于任何特定硬件环境；具有相对独立性。综合器在将VHDL(硬件描述语言)表达的电路功能转化成具体的电路结构网表过程中，具有明显的能动性和创造性，它不是机械的一一对应式的“翻译”，而是根据设计库、工艺库以及预先设置的各类约束条件，选择最优的方式完成电路结构的设计。

l-3　什么是综合?有哪些类型?综合在电子设计自动化中的地位是什么？　P6

什么是综合？答：在电子设计领域中综合的概念可以表示为：将用行为和功能层次表达的电子系统转换为低层次的便于具体实现的模块组合装配的过程。

有哪些类型？答：(1)从自然语言转换到VHDL语言算法表示，即自然语言综合。(2)从算法表示转换到寄存器传输级(RegisterTransport Level，RTL)，即从行为域到结构域的综合，即行为综合。(3)从RTL级表示转换到逻辑门(包括触发器)的表示，即逻辑综合。(4)从逻辑门表示转换到版图表示(ASIC设计)，或转换到FPGA的配置网表文件，可称为版图综合或结构综合。

综合在电子设计自动化中的地位是什么？答：是核心地位（见图1-3）。综合器具有更复杂的工作环境，综合器在接受VHDL程序并准备对其综合前，必须获得与最终实现设计电路硬件特征相关的工艺库信息，以及获得优化综合的诸多约束条件信息；根据工艺库和约束条件信息，将VHDL程序转化成电路实现的相关信息。

1-4　在EDA技术中，自顶向下的设计方法的重要意义是什么？P8~10

答：在EDA技术应用中，自顶向下的设计方法，就是在整个设计流程中各设计环节逐步求精的过程。

1-5 IP　在EDA技术的应用和发展中的意义是什么？　P23~25

答：IP核具有规范的接口协议，良好的可移植与可测试性，为系统开发提供了可靠的保证。

1-6 叙述EDA的FPGA/CPLD设计流程，以及涉及的EDA工具及其在整个流程中的作用。　(P12~14)

答：1.设计输入(原理图/HDL文本编辑)(EDA设计输入器将电路系统以一定的表达方式输入计算机)；2.综合(EDA综合器就是将电路的高级语言(如行为描述)转换成低级的，可与FPGA/CPLD的基本结构相映射的网表文件或程序。)；3.适配

(EDA适配器的功能是将由综合器产生的网表文件配置于指定的目标器件中，使之产生最终的下载文件，如 JEDEC、JAM格式的文件。)；4. 时序仿真(EDA时序仿真器就是接近真实器件运行特性的仿真，仿真文件中已包含了器件硬件特性参数，因而，仿真精度高。)与功能仿真(EDA功能仿真器直接对 VHDL 原理图描述或其他描述形式的逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，仿真过程不涉及任何具体器件的硬件特性。)；5. 编程下载(EDA编程下载把适配后生成的下载或配置文件，通过编程器或编程电缆向 FPGA或 CPLD下载，以便进行硬件调试和验证(Hardware Debugging) 。)；6. 硬件测试(最后是将含有载入了设计的FPGA或 CPLD的硬件系统进行统一测试，以便最终验证设计项目在目标系统上的实际工作情况，以排除错误，改进设计。其中 EDA的嵌入式逻辑分析仪是将含有载入了设计的 FPGA的硬件系统进行统一测试，并将测试波形在 PC机上显示、观察和分析。)。

2 习 题

2-1 OLMC（输出逻辑宏单元）有何功能?说明 GAL是怎样实现可编程组合电路与时序电路的。 P34~36

OLMC有何功能？答：OLMC单元设有多种组态，可配置成专用组合输出、专用输入、组合输出双向口、寄存器输出、寄存器输出双向口等。

说明 GAL是怎样实现可编程组合电路与时序电路的？答：GAL（通用阵列逻辑器件）是通过对其中的 OLMC（逻辑宏单元）的编程和三种模式配置（寄存器模式、复合模式、简单模式），实现组合电路与时序电路设计的。

2-2 什么是基于乘积项的可编程逻辑结构？ P33~34，40 什么是基于查找表的可编程逻辑结构？ P40~42

什么是基于乘积项的可编程逻辑结构?答：GAL、CPLD之类都是基于乘积项的可编程结构；即包含有可编程与阵列和固定的或阵列的 PAL（可编程阵列逻辑）器件构成。

什么是基于查找表的可编程逻辑结构?答：FPGA（现场可编程门阵列）是基于查找表的可编程逻辑结构。

2-3 FPGA 系列器件中的 LAB有何作用? P42~44

答：FPGA（Cyclone/Cyclone II ）系列器件主要由逻辑阵列块 LAB、嵌入式存储器块（EAB）、I/O 单元、嵌入式硬件乘法器和 PLL等模块构成；其中 LAB（逻辑阵列块）由一系列相邻的 LE（逻辑单元）构成的；FPGA可编程资源主要来自逻辑阵列块 LAB。

2-4 与传统的测试技术相比，边界扫描技术有何优点？ P47~50

答：使用 BST（边界扫描测试）规范测试，不必使用物理探针，可在器件正常工作时在系统捕获测量的功能数据。克服传统的外探针测试法和“针床”夹具测试法来无法对 IC 内部节点无法测试的难题。

2-5 解释编程与配置这两个概念。 P51~56

答：编程：基于电可擦除存储单元的 EEPROM或 Flash 技术。CPLD一般使用此技术进行编程。CPLD被编程后改变了电可擦除存储单元中的信息，掉电后可保存。电可擦除编程工艺的优点是编程后信息不会因掉电而丢失，但编程次数有限，编程的速度不快。

配置：基于 SRAM查找表的编程单元。编程信息是保存在 SRAM中的，SRAM在掉电后编程信息立即丢失，在下次上电后，还需要重新载入编程信息。大部分 FPGA采用该种编程工艺。该类器件的编程一般称为配置。对于 SRAM型 FPGA来

说，配置次数无限，且速度快；在加电时可随时更改逻辑；下载信息的保密性也不如电可擦除的编程。

2-6　请参阅相关资料，并回答问题：按本章给出的归类方式，将基于乘积项的可编程逻辑结构的 PLD器件归类为 CPLD，将基于查找表的可编程逻辑结构的 PLD器件归类为 FPGA，那么，APEX系列属于什么类型 PLD器件? MAX II 系列又属于什么类型的 PLD器件?为什么？　P47~51

答：APEX(Advanced Logic Element Matrix)　系列属于 FPGA类型 PLD器件；编程信息存于 SRAM中。MAX II系列属于 CPLD类型的 PLD器件；编程信息存于 EEPROM中。

3　习　　题

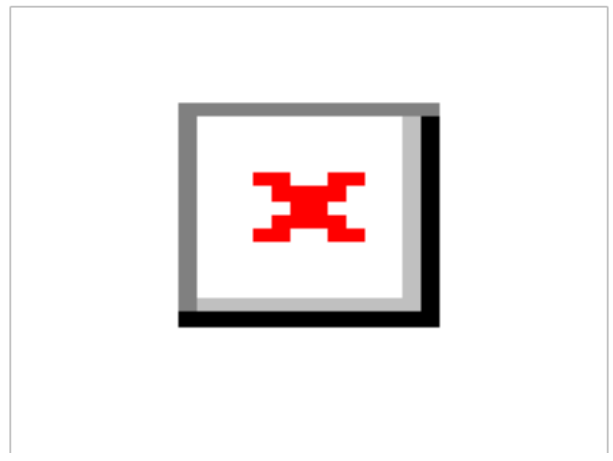
3-1 说明端口模式 INOUT和 BUFFER有何异同点。P60

INOUT：具有三态控制的双向传送端口

BUFFER:具有输出反馈的单向东湖出口。

3-2 画出与以下实体描述对应的原理图符号元件:

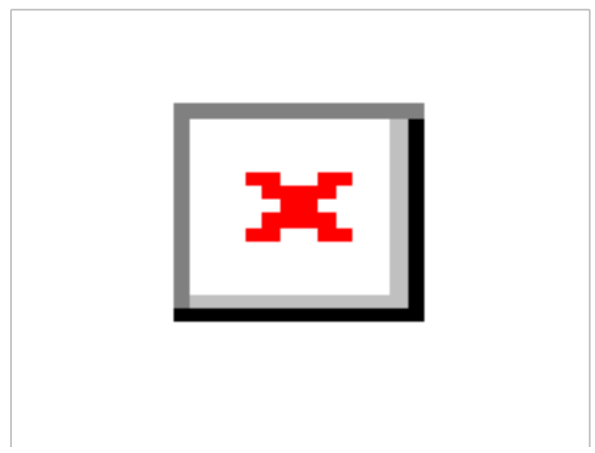
```
ENTITY buf3s IS                --实体 1:三态缓冲器
PORT(input:IN STD_LOGIC;          --输入端
enable:IN STD_LOGIC;              --使能端
output:OUT STD_LOGIC);     --输出端
END buf3s ;
```



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);



3-3 试分别用 IF_THEN 语句和 CASE 语句的表达方式写出此电路的 VHDL 程序，选择控制信号 s1 和 s0 的数据类型为 STD_LOGIC_VECTOR，即 s1='0',s0='0'□ s1='0',s0='1'□s1='1',s0='0'□ s1='1',s0='1'□，分别执行 y<=a、y<=b、y<=c、y<=d。

--解 1：用 IF_THEN 语句实现 4 选 1 多路选择器

LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY mux41 IS

PORT　(a,b,c,d: IN STD_LOGIC;

```
s0:             IN STD_LOGIC;

s1:             IN STD_LOGIC;

y:          OUT STD_LOGIC);

END ENTITY mux41;

ARCHITECTURE rtl_mux41      OF mux41     IS

SIGNAL s0s1 : STD_LOGIC_VECTOR(1 DOWNTO 0);--定义标准逻辑位矢量数据

BEGIN

s0s1<=s1&s0;            --s1   相并 s0, 即 s1 与 s0 并置操作

PROCESS(s0s1,a,b,c,d)

BEGIN

IF                      THEN   y     <=    a;

ELSIF                   THEN   y     <=    b;

ELSIF                       THEN   y     <=    c;

ELSE   y     <=    d;

END IF;

END PROCESS;

END ARCHITECTURE rtl_mux41;
```

--解 2：用 CASE语句实现 4 选 1 多路选择器

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY mux41 IS

PORT   (a,b,c,d: IN STD_LOGIC;
        s0:              IN STD_LOGIC;
        s1:              IN STD_LOGIC;
        y:               OUT STD_LOGIC);
        END ENTITY mux41;
        ARCHITECTURE case_mux41      OF mux41     IS
        SIGNAL s0s1 : STD_LOGIC_VECTOR(1 DOWNTO 0);--定义标准逻辑位矢量数据
类型
        BEGIN
        s0s1<=s1&s0;                 --s1   相并 s0, 即 s1 与 s0 并置操作
        PROCESS(s0s1,a,b,c,d)
        BEGIN
        CASE s0s1 IS            --类似于真值表的 case  语句
```

图 3-18　全减器结构图

--解(1.1)：实现 1 位半减器 h_suber(diff=x-y　；s_out=1,x<y)

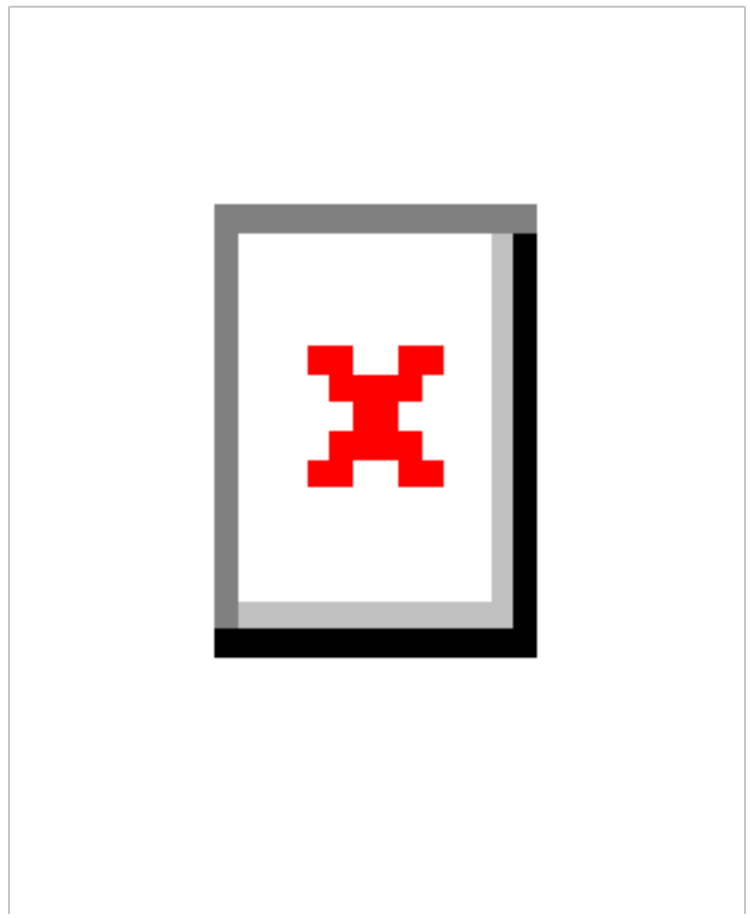
```
LIBRARY IEEE;  --半减器描述(1):布尔方程描述方法

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY h_suber IS

PORT(              x,y:       IN STD_LOGIC;

diff,s_out: OUT STD_LOGIC);

END ENTITY  h_suber;

ARCHITECTURE hs1      OF h_suber IS

BEGIN

Diff      <= x XOR (NOT y);

s_out <= (NOT x) AND y;

END ARCHITECTURE hs1;
```

--解(1.2)：采用例化实现图 4-20 的 1 位全减器

```
                IEEE;      --1 位二进制全减器顺层设计描述

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY f_suber IS

PORT(xin,yin,sub_in:            IN STD_LOGIC;

sub_out,diff_out: OUT STD_LOGIC);

END ENTITY  f_suber;

ARCHITECTURE fs1       OF f_suber IS

COMPONENT h_suber       --调用半减器声明语句

PORT(x, y:                      IN STD_LOGIC;

diff,s_out: OUT STD_LOGIC);

END COMPONENT;

SIGNAL a,b,c: STD_LOGIC;     --定义 1 个信号作为内部的连接线。

      BEGIN

      u1:     h_suber PORT
MAP(x=>xin,y=>yin,     diff=>a,                    s_out=>b);

      u2:     h_suber PORT MAP(x=>a,     y=>sub_in, diff=>diff_out,s_out=>c);

      sub_out <= c OR b;
```

fs1;

(2)以 1 位全减器为基本硬件,构成串行借位的 8 位减法器,要求用例化语句来完成此项设计(减法运算是 x-y-sun_in=difft)　　。



--解(2)：采用例化方法，以 1 位全减器为基本硬件；实现串行借位的 8 位减法器(上图所示)。

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.ALL;

ENTITY suber_8 IS

PORT(x0,x1,x2,x3,x4,x5,x6,x7:                IN STD_LOGIC;

y0,y1,y2,y3,y4,y5,y6,y7,sin:          IN STD_LOGIC;

diff0,diff1,diff2,diff3:                  OUT STD_LOGIC;
```