1-4 在EDA技术中，自顶向下的设计方法的重要意义是什么?

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。

1-6 叙述EDA的FPGA/CPLD设计流程）（Quartus Ⅱ）。

答：1.设计输入(原理图/HDL文本编辑)；

2.综合；

3.适配；

4.时序仿真与功能仿真；

5.编程下载；

6.硬件测试。

1、什么是EDA技术？ EDA的英文全称是什么？

答：EDA技术有狹义和广义之分，狭义EDA技术就是以大规模可编程逻辑器件为设计载体， 以硬件描述语言为系统逻辑描述的主要表达方式，以计算机、大规模可编程逻辑器件的开发 软件及实验开发系统为设计工具，通过有关的开发软件，自动完成用软件的方式设计的电子 系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布网布线、逻辑 仿真，直至完成对于特定U标芯片的适编译、逻辑映射、编程下载等工作，最终形成集成电子系统或专用集成芯片的一门新技术，或称为IES/ASIC 自动设计技术。

2、从使用的角度讲，EDA技术主要包括几个方而的内容？

答：EDA技术的学习主要应掌握四个方而的内容：①大规模可编程逻辑器件；②硬件描述语言；③软件开发工具；④实验开发系统。其中，硬件描述语言是重点-。

3、CPLD和FPGA的区别是什么？

答：（1）**结构不同**：CPLD内部互连结构由固定长度的连线资源组成，布线的延迟固定，逻辑单元主要由“与/或阵列”构成。FPGA 内部由多种长度不同的连线资源组成，每次布线的延迟不同；逻辑单元主要由静态存储器（SRAM）构成的函数发生器，即查找表，通过查找表可以实现函数逻辑功能。（2）**集成度不同**：CPLD一般500～50000门，FPGA一般1K～10M门，现在可达到千万门级。(3**)工艺不同**：CPLD采用PROM工艺，集成度低。FPGA采用SRAM工艺，集成度高。（4）**应用范围不同**：CPLD逻辑功能强而寄存器少，适用控制密集型系统。FPGA逻辑功能弱而寄存器多，适用于数据密集型系统；（5）**使用方法不同**：CPLD是PROM工艺，写上的信息掉电不会丢失；FPGA采用SRAM工艺，掉电信息丢失。需在外部专门加一片ROM，把编程的数据写到 ROM中，用来保存FPGA的编程数据。

1、简述Quartus II的设计流程。

答：Quartus II的设计过程包括设计项目的建立与设计的输入、设计编译、设计校验（仿真和定时分析）、器件编程四个步骤。

**设计输入**：可以采用原理图输入、HDL语言描述及波形输入等几种方式。

**设计编译**：先根据设计要求设定编译参数和编译策略，如器件的选择、逻辑综合方式的选择等。然后根据设定的参数和策略对设计项目进行网表提取、逻辑综合和器件适配，并产生报告文件、延时信息文件及编程文件，供分析仿真和编程使用。

**设计校验（项目仿真）**：包括功能仿真、时序仿真和定时分析，可以利用软件的仿真功能来验证设计项目的逻辑功能是否正确。

**器件编程与验证**：用经过仿真确认后的编程文件通过编程器（Programmer）将设计下载到实际芯片中，最后测试芯片在系统中的实际运行性能。

在设计过程中，如果出现错误，则需重新回到**设计输入阶段**，改正错误或调整电路后重复上述过程。

或者：步骤1：建立工作库文件夹；步骤2：编辑设计文件；步骤3：存盘，注意实体名与文本取名一致；步骤4：创建工程；步骤5：选择目标器件；步骤6：启动编译；步骤7：建仿真波形文件；步骤8：仿真测试和波形分析；步骤9：引脚锁定并编译；步骤10：编程下载/配置；步骤11：硬件测试。

3、结构体有几种描述方法？

结构体的描述方式有：结构描述，数据流描述，行为描述

**结构描述**：描述该设计单元的硬件结构（表示元件之间的互连）

**数据流描述**：描述数据的传输和变换。

**行为描述**：只描述电路的功能或电路的行为（表示输入与输出间转换的行为，不包含任何结构信息。）

4、什么是**自顶向下**的设计方法？（5分）

答：所谓自顶向下的设计，就是设计者首先从整体上规划整个系统的功能和性能，然后对系统进行划分，分解为规模较小、功能较为简单的局部模块，并确立它们之间的相互关系。

15、简述**层次化设计的优点**

答：层次化设计是一种模块化的设计方法，设计人员对设计的描述由上至下逐步展开，符合常规的思维习惯。由于顶层设计与具体的器件和工艺无关，因此易于在各种可编程逻辑器件中间进行移植。

什么是**半定制器件**？

半定制集成电路由厂家提供一定规格的功能块，如门阵列、标准单元、可编程逻辑器件等，按用户要求利用专门设计的软件进行必要的连接，从而设计出所需要的专用集成电路，称为半定制电路。

**全定制集成电路**：按规定的功能、性能要求对电路的结构布局、布线均进行专门的最优化设计，以达到芯片的最佳利用。

一、简答题：

1. Altera公司的MAX7000S系列器件是CPLD还是FPGA？主要由哪几个部分组成？

CPLD。逻辑阵列块LAB，宏单元，扩展乘积项，可编程连线阵列PIA，I/O控制块

2. Altera公司的FLEX10K系列器件是CPLD还是FPGA？主要由哪几个部分组成？

FPGA。逻辑阵列块LAB，嵌入式存储器块，嵌入式硬件乘法器，I/O单元和嵌入式PLL

▲3. 什么是CPLD？CPLD采用了什么技术，什么工艺？

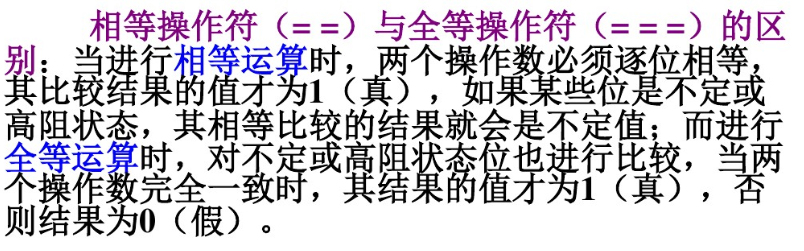
Complex Programmable Logic Device，复杂可编程逻辑器件。采用了EEPROM或FLASH技术，使用了**乘积项工艺**

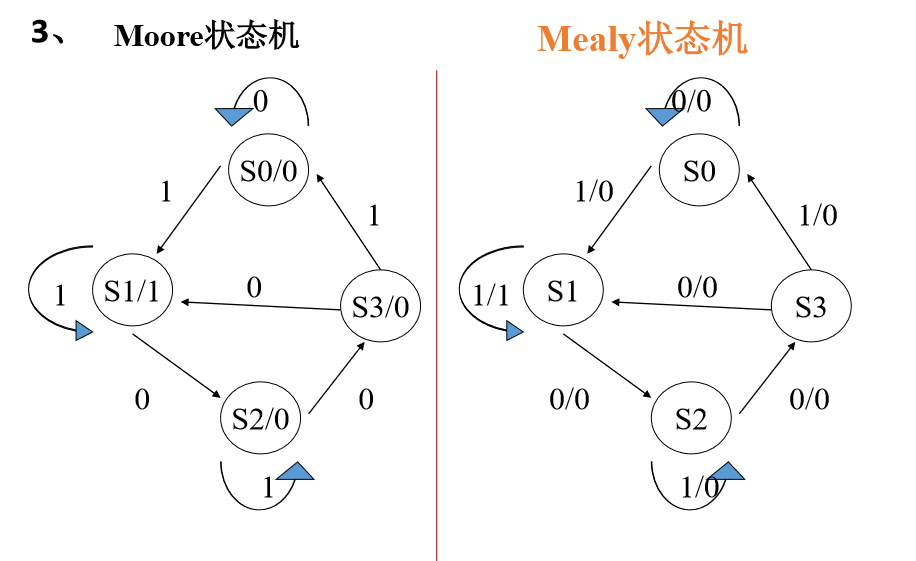
▲4. 什么是FPGA？FPGA采用了什么技术，什么工艺？

Field Programmable Gate Array，现场可编程门阵列。采用了SRAM技术，使用**查找表工艺**

5. CPLD/FPGA芯片主要生产厂家有哪几家？ Xilinx，Lattice，Alter

ASIC Application Specific Integrated Circuit，专用集成电路





module MOORE(clk,reset,din,op);

input clk,reset,din;

output op; reg op;

parameter s0=0,s1=1,s2=2,s3=3;

reg[2:0] pstate,nstate;

always@(posedge clk or negedge reset)

begin

if(!reset)

pstate<=s0;

else pstate<=nstate;

end

always@(pstate or din)

case(pstate)

s0: begin op<=0;

if(din==1'b0) state<=s0;

else nstate<=s1; end

s1: begin op<=1;

if(din==1'b1) nstate<=s1;

else nstate<=s2; end

s2: begin op<=0;

if(din==1'b1) nstate<=s2;

else nstate<=s3; end

s3: begin op<=0;

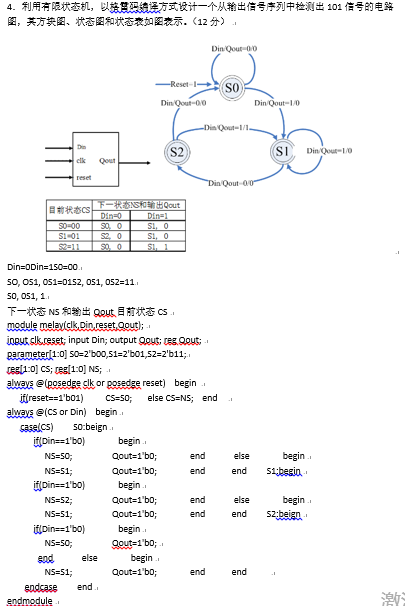
if(din==1'b1) nstate<=s0;

else nstate<=s1; end

default:nstate<=s0;

endcase

endmodule



**半减器**：

module sub\_h(a,b,s,c);//一位二进制减法s=a-b

input a,b;

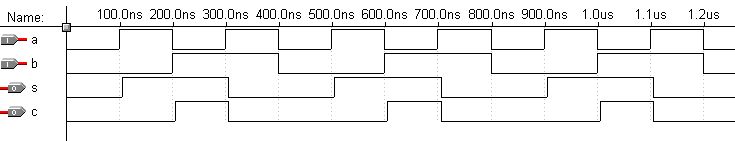
output s; //差

output c;//输出借位标志

assign s=a^b;

assign c=(~a)&b;

endmodule



**全减器**：

module sub(a,b,s,cin,cout);//一位二进制减法s=a-b-cin

input a,b;

input cin;//输入借位标志

output s; //差

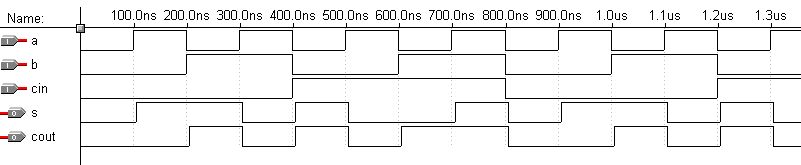
output cout;//输出借位标志

assign {cout,s}=a-b-cin;

//ssign c=a^b;

//assign s=(~a)&b;

endmodule



**除法器**

module div\_rill(

input[31:0] a,

input[31:0] b,

output reg [31:0] yshang,

output reg [31:0] yyushu

);

reg[31:0] tempa;

reg[31:0] tempb;

reg[63:0] temp\_a;

reg[63:0] temp\_b;

integer i;

always @(a or b)

begin

tempa <= a;

tempb <= b;

end

always @(tempa or tempb)

begin

temp\_a = {32'h00000000,tempa};

temp\_b = {tempb,32'h00000000};

for(i = 0;i < 32;i = i + 1)

begin

temp\_a = {temp\_a[62:0],1'b0};

if(temp\_a[63:32] >= tempb)

temp\_a = temp\_a - temp\_b + 1'b1;

else

temp\_a = temp\_a;

end

yshang <= temp\_a[31:0];

yyushu <= temp\_a[63:32];

end

endmodule

**三-八线编码器**

module decode3\_8 (data\_out,data\_in,enable) ;

input [2:0] data\_in;

input enable;

output [7:0] data\_out;

reg [7:0] data\_out;

always @(data\_in or enable)

begin

if (enable==1)

case (data\_in )

3'b000: data\_out=8'b11111110;

3'b001: data\_out=8'b11111101;

3'b010: data\_out=8'b11111011;

3'b011: data\_out=8'b11110111;

3'b100: data\_out=8'b11101111;

3'b101: data\_out=8'b11011111;

3'b110: data\_out=8'b10111111;

3'b111: data\_out=8'b01111111;

default: data\_out=8'bxxxxxxxx;

endcase

else

data\_out=8'b11111111;

end

endmodule

4. 16位循环右移寄存器

module SHIFT(RST,REG\_OUT,LOAD);

reg [3:0]REG\_IN;

reg CLK;

input RST;

input LOAD;

output [3:0]REG\_OUT;

shift\_reg str1(REG\_IN,CLK, REG\_OUT, RST, LOAD); //顺序端口连接

//shift\_reg str1(.reg\_in(REG\_IN),

// .clock(CLK),

// .reg\_out(REG\_OUT),

// .rst\_n(RST),

// .load(LOAD)); //命名端口连接

endmodule

module shift\_reg(reg\_in, clock, reg\_out, rst\_n,load); **//移位寄存器**

input [3:0]reg\_in;

input clock;

input rst\_n;

input load ;

output [3:0]reg\_out;

reg [3:0]data\_r;

always@(posedge clock or negedge rst\_n or posedge load)

begin

if (!rst\_n)

begin

data\_r[0] <= 0;

end

else if (load )

data\_r <= reg\_in;

else

begin

data\_r <= (data\_r<<1);

data\_r[0] <= 0;

end

end

assign reg\_out = data\_r;

endmodule

编程题

1.用七段 LED 数码管显示对应的十六进制值

2.三输入端与非门

3**.4选1数据选择器**

input a,b,c,d;

input s0,s1;

output y;

always@(a or b or c or d or s0 or s1) begin

case({s1,s0})

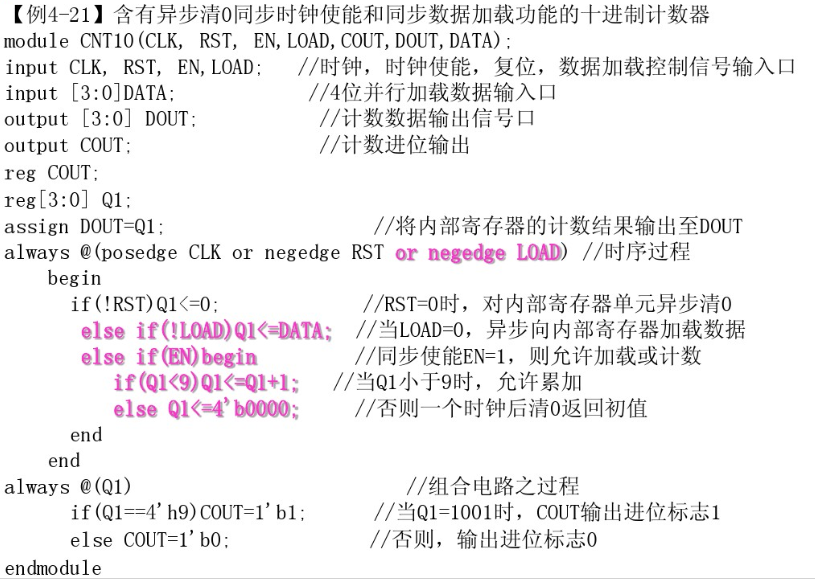
2’b00: y <= a;

2’b01: y <= b;

2’b10: y <= c;

2’b11: y <= d;

endcase end



**//3位计数器**

module counter(clock, q\_out, rst\_n);

input clock;

input rst\_n;

output q\_out;

reg [2:0]q\_out;

//①异步清零：可独立于CLK清零

always@(posedge clock or negedge rst\_n)

begin

if (!rst\_n)

q\_out <= 0;

else

q\_out <= q\_out + 1;

end

//②同步清零：跟时钟同步

always@(posedge clock)

begin

if (!rst\_n)

q\_out <= 0;

else

q\_out <= q\_out + 1;

end

endmodule

12进制加法/减法计数器

repeat(循环次数表达书)

**8位奇偶校验器**

module jiaoyan\_3(a,out);

parameter size = 8;

input[7:0] a;

output out;

reg out;

integer n;

always@(a)

begin

out = 0;

n = 0;

repeat(size)

begin

out – out^a[n];

n = n+1;

end

end

endmodule

3/8线译码器（case语句/if-else语句）

**8/3线优先编码器**（if语句）

module encode8\_3(y,a);

input [7:0] a;

input [2:0] y;

reg[2:0] y;

always@(a)

begin

if(~a[7]) y = 3’b111;

else if (~a[6]) y = 3’b110;

else if (~a[5]) y = 3’b101;

else if (~a[4]) y = 3’b100;

else if (~a[3]) y = 3’b011;

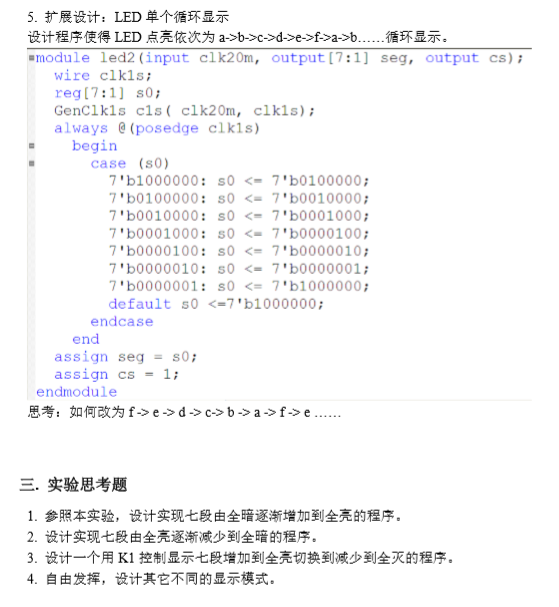
else if (~a[2]) y = 3’b010;

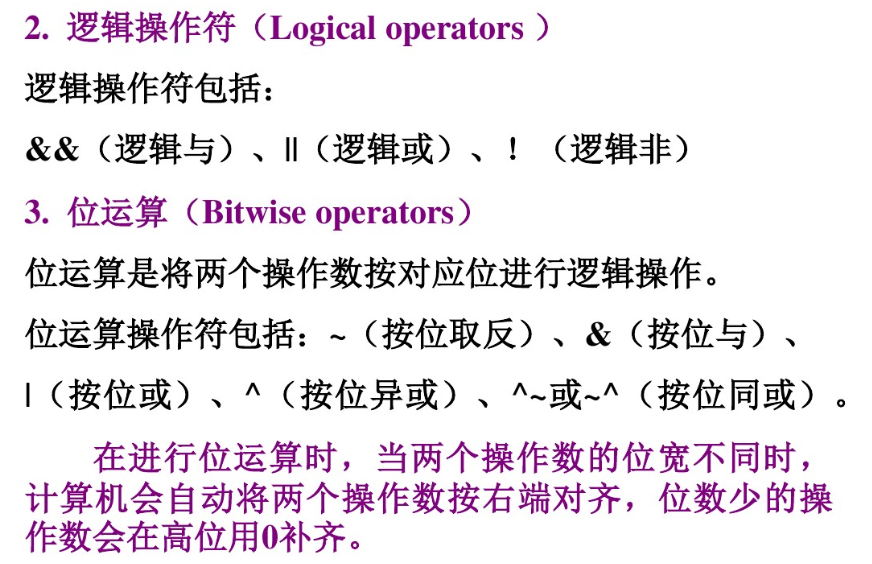
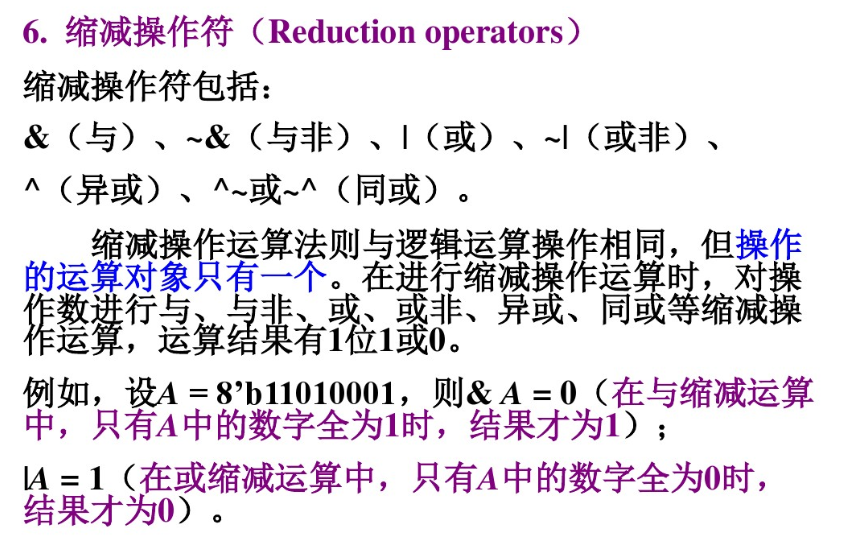
else if (~a[1]) y = 3’b001;

else y = 3’b000;

end

endmodule



编写**序列发生器** 1011011

**比较器**（例：设计一个比较电路，当输入的BCD码大于5时输出1，否则输出0）

设计一个求补码的程序，输入数据是一个有符号的8位二进制数

设置一个格雷码至二进制数的转换器

设计一个4位乘法器，为此首先设计一个加法器，用例化语句调用这个加法器，用移位相加的方式完成乘法。并以此为基础，使用parameter参数传递的功能，设计一个16位乘法器。

利用8个全加器，可以构成一个8位加法器。利用循环语句来实现这项设计。并以此项设计为例，使用parameter参数传递的功能，设计一个32位加法器。

什么是**基于乘积项**的可编程逻辑结构？什么是**基于查找表**的可编程逻辑结构？

乘积项：GAL、CPLD。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。

查找表：FPGA。采用了RAM“数据”查找的方式，并使用多个查找表构成了一个查找表阵列（PGA）。

**综合**：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。

与传统的测试技术相比，**边界扫描技术**有何优点？

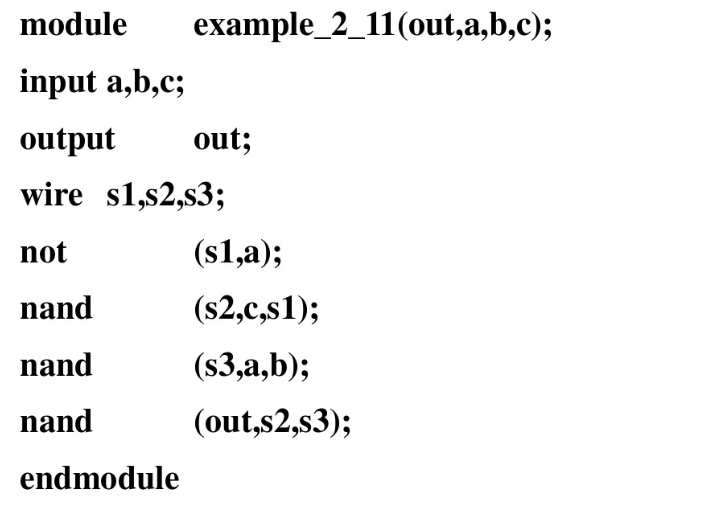
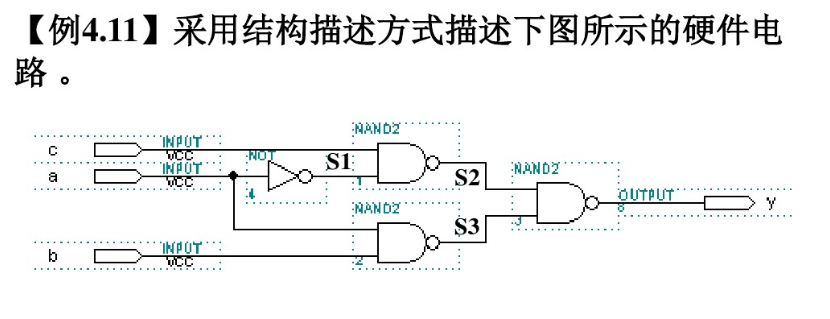
使用 BST（边界扫描测试）规范测试，不必使用物理探针，可在器件正常工作时在系统捕获测量的功能数据。克服传统的外探针测试法和“针床”夹具测试法来无法对 IC 内 部节点无法测试的难题。

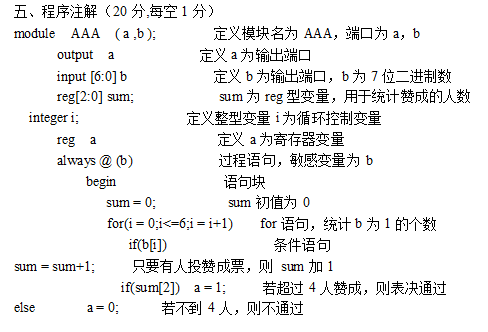
解释**编程**与**配置**这两个概念

通常，对CPLD的下载成为编程对于FPGA中的SRAM进行直接下载的方式成为配置，但对于反熔丝结构和Flash结构的FPGA的下载和对FPGA的专用配置ROM的下载仍称为编程。

程序分析题

程序改错题





 本程序的逻辑功能是：    **7人投票表决器**。