1. 单**项选择题（本大题共15小题，**每小题1分，共15分）

1.一般把EDA技术的发展分为几个阶段？ B

A.2

B.3

C.4

D.5

2.将设计的系统按照EDA开发软件要求的某种形式表示出来,并送入计算机的过程,称为( A )。

A.设计输入

B.设计输出

C.设计处理

D.综合

3.FPGA/CPLD有如下设计步骤：①原理图/HDL文本输入、②适配、③功能仿真、④综合、⑤编程下载、⑥硬件测试，正确的设计顺序是（ B ）。

A.①②④③⑤⑥

B.①③④②⑤⑥

C.①④③②⑤⑥

D.①②④⑤③⑥

4.基于硬件描述语言的现代数字系统设计采用的设计方法是（ B ）。

A.自底向上

B.自顶向下

C.积木式

D.顶层

5.在以下ISE Foundation的集成工具中，可以进行设计仿真的工具是（ D ）。

A.iMPACT

B.System Generator

C.XST

D.ISE Simulator

6.根据Verilog HDL语法，`timescale 1ns/100ps中1ns,100ps分别代表（ A ）。

A.时间单位、时间精度

B.时间精度、时间单位

C.总时间、阶段时间

D.仿真时间、基本时间

7.根据Verilog HDL语法，语句 assign c=(a>b)? a:b；若a=4，b=3,则c=（ D ）。

A.1

B.2

C.3

D.4

8.设a = 4´b1010，b=4´b0001，则根据Verilog HDL语法，下列式子的值为1的是（ D ）。

A.a - b

B.a << 1

C.a ^ b

D.2 – (a>b)

9.IP核在EDA技术和开发中具有十分重要的地位,IP是指( A )

A.知识产权

B.互联网协议

C.网络地址

D.智能外设

10.在Xilinx FPGA的开发集成环境中，IP 核的开发工具是（ B ）

A.HDL Editor

B.Core Generator

C.PlanAhead

D.ECS

11.根据配置数据线数，串行配置一般以什么为单位向可编程器件载入配置数据？ A

A.比特（bit）

B.字节(Byte)

C.帧结构

D.通用IP方式

12.把基于电可擦除存储单元的E2PROM或Flash 技术的CPLD 的在系统下载称为( D )。

A.配置

B.ICR

C.JTAG

D.编程

13.FPGA中嵌入式处理器可以有三种不同的使用模式，分别是：状态机模式，单片机模式和（ A ）

A.定制嵌入模式

B.PicoBlaze模式

C.MicroBlaze模式

D.无线通信模式

14.关于JTAG边界扫描测试，下面说法错误的是 D 。

A.JTAG标准可以用于芯片的测试与配置。

B.JTAG接口由4个必需的信号，以及1个可选信号构成。

C.JTAG在芯片I/O端上增加移位寄存器，并将这些寄存器连接起来，配合时钟复位、测试方式选择以及扫描输入和输出端口，形成边界扫描通道。

D.JTAG边界扫描测试需要物理探针配合捕捉数据。

15.下面关于Chipscope Pro的描述，错误的是（ D ）

A.它是ISE集成套件中的在线片内逻辑分析工具。

B.它通过JTAG接口将FPGA内部信号实时读出。

C.它将采集到的数据通过JTAG电缆传送到计算机。

D.它可以完成芯片的手动布局布线操作。

1.从设计输入文件到生成编程数据文件的编译过程，称为（ C ）。

A.设计准备

B.设计输入

C.设计处理

D.设计校验

2.基于EDA软件的FPGA / CPLD设计流程为:原理图/HDL文本输入→( )→综合→适配→( )→编程下载→硬件测试。请选择如下操作步骤，按顺序填空：①功能仿真 ②时序仿真 ③逻辑综合 ④配置 ⑤引脚锁定 D

A.③①

B.⑤②

C.④⑤

D.①②

3.将硬件描述语言的代码描述转化为硬件电路原理图的EDA工具软件称为（ B ）。

A.编辑器

B.HDL综合器

C.适配器

D.下载器

4.在ISE Foundation的集成工具中，可以实现FPGA/CPLD的配置和编程的工具是（ A ）。

A.iMPACT

B.System Generator

C.XST

D.ISE Simulator

5.根据Verilog HDL语法，下列标识符不合法的是（ A ）。

A.08lymoon

B.next\_state

C.tempdata

D.system$

6.在Verilog HDL的逻辑运算中，设 a = 4´b1100，b = 4´b0010，则a | b结果是( C )。

A.4´b1010

B.4´b0010

C.4´b1110

D.4´b0000

7.Verilog HDL的模块分为两种类型：其中一种描述某种电路系统结构，功能，以综合或者提供仿真模型为设计目的，这一类型的模块被称为（ B ）。

A.结构模块

B.功能模块

C.综合模块

D.仿真模块

8.在Verilog HDL端口声明语句中，声明输出端口的关键词是（ D ）。

A.OUTPUT

B.INOUT

C.input

D.output

9.在现代数字系统设计中，IP模块的使用是常用的有效设计方法之一。下面对IP模块的描述，错误的是 C 。

A.IP模块是知识产权产品。

B.IP有时候也称为虚拟元件或宏功能块，是满足特定规范，并能在设计中复用的功能模块。

C.IP即为网络协议。

D.IP模块的重用是使得数字电路系统赢得迅速上市时间的主要策略之一。

10.CPLD可编程逻辑是基于( B )

A.查找表结构

B.乘积项结构

C.PLD

D.CLB

11.IP核在EDA技术和开发中具有十分重要的地位，以HDL方式提供的IP核被称为（ C）。

A.硬核

B.固核

C.软核

D.通用IP核

12.下列对FPGA的描述中,正确的是( C )。

A.FPGA是基于乘积项结构的可编程逻辑器件

B.FPGA是全称是复杂可编程逻辑器件

C.基于SRAM结构的FPGA器件,在每次上电后必须进行一次配置

D.在Xilinx公司生产的器件中,XC9500系列属于FPGA结构

13.CPLD的可编程逻辑是基于( B )。

A.LUT结构

B.乘积项结构

C.PLD结构

D.CLB结构

14.嵌入式系统的架构可分为哪四部分？A

A.处理器、存储器、输入/输出接口、软件

B.处理器、存储器、嵌入式操作系统，应用软件

C.嵌入式微控制器，存储器，输入/输出接口、应用软件

D.状态机模块、存储器、嵌入式操作系统，应用软件

15.在ISE集成套件中，作为在线逻辑分析仪使用的工具软件是( B )。

A.XST

B.ChipScope Pro

C.Synplify Pro

D.FPGA Advantage

1.正确的现代数字系统设计的流程是 B

A.设计准备→设计输入→适配→综合→ 功能仿真一→编程下载一时序仿真→硬件测试

B.设计准备→设计输入一功能仿真→综合一适配一时序仿真→编程下载→硬件测试

C.设计准备一设计输入一功能仿真→综合→时序仿真 编程下载一适配→硬件测试

D.设计准备一设计输入一功能仿真→适配 时序仿真编程下载一综合→硬件测试

2. FPGA 的可编程结构是主要基于A

A.查找表(LUT) B. ROM可编程

C. PAL可编程 D. 与或阵列乘积项

3.在EDA工具中，能将硬件描述语言转化为硬件电路的重要工具软件是B

A.仿真器 B.综合器

C.适配器 D.下载器

4.在Verilog HDL的always块语句本身是 B

A.顺序执行 B.并行执行

C.顺序执行或并行执行 D.串行执行

5.下列标识符中，不合法的标识符是 A

A. 9moon B. State0

C. Not\_ Ack 0 D. signall

6.在现代数字系统设计中，IP 模块的使用是常用的有效设计方法之一。下面对IP模块的描述，错误的是C

A. IP 即知识产权产品

B. 调用IP核能避免重复劳动，大大减轻工程师的负担

C. IP 即为网络协议

D. IP模块的重用是使得数字电路系统赢得迅速上市时间的主要策略之一

7. 用Verilog HDL的assign语句赋值的方法一般称为A

A.连续赋值 B. 并行赋值

C.串行赋值 D. 函数赋值

8.数字系统中的ASIC可以分为哪两大类? A

A.全定制ASIC、半定制ASIC B. FPGA, CPLD

C.门阵列、PLD D.简单低密度PLD,复杂高密度PLD

9.大规模可编程器件主要有 FPGA、CPLD两类，下列对FPGA结构与工作原理的描述中，正确的是C

A.FPGA全称为复杂可编程逻辑器件

B.FPGA是基于乘积项结构的可编程逻辑器件

C.基于SRAM的FPGA器件，在每次上电后必须进行一次配置

D.在Xilinx公司生产的器件中，Spartan-3E系列属CPLD结构

10.在Verilog HDL的逻辑运算中，设A＝8＇b11010001，B＝8＇b00011001，则表达式“A＆B”的结果为 A

A. 8'b00010001

B.8'b11011001

C.8'b11001000

D.8’b00110111

11.下列语句中，不属于并行语句的是 D

A.非阻塞赋值语句

B.assign 语句

C.元件例化语句

D.case 语句

12.关于可编程逻辑器件的分类，下面说法正确的是 C

A.PLA、PAL、GAL属于高密度可编程逻辑器件

B.CPLD基本上采用熔丝方式编程

C.FPGA基本上采用SRAM配置存储器方式编程

D.FPGA基本上采用浮栅编程技术方式编程

13.在下列选项中，XILINX公司提供的SOPC嵌入式解决方案的硬核处理器是 D

A. PicoBlaze

C. MicroBlaze

B.NIOS II

D.PowerPC

14.在ISE Fundation 集成开发环境下，属于第三方的综合工具的是 B

A. Modelsim B.Synplify

C.XST D. ChipScope

15.基于FPGA的DSP系统进行数字信号处理，不具有的因素特点是 D

A.高度的并行性 B. 重构的灵活性

C.性价比高 D. 信号处理能力较低

1．状态机编码方式中，其中占用触发器较多的方式是C

A.状态位直接输出型编码 B.顺序编码

C.一位热码编码 D.BCD编码

2．关于XC95288XL-6TQ144C器件，下面说法错误的是 B

A.这是一款XILINX公司的FPGA芯片

B．-6表示器件的温度等级

C.TQ表示该器件的封装类型

D.它是一款商业用器件

3．关于将编程信息下载到FPGA或CPLD器件，下面说法正确的是C

A．基于电可擦除存储单元的E2PROM或Flash技术的CPLD在系统下载称为配置

B．基于SRAM查找表结构的FPGA的在系统下载称为编程

C．可编程逻辑器件的配置方式分为主动配置和从动配置两类

D. 在从动配置模式下，是由可编程器件引导配置过程

4．嵌入式系统的构架包括处理器、输入／输出（I／0）接口、软件和A

A.存储器 B.嵌入式微处理器

C.嵌入式微控制器 D.嵌入式数字信号处理器

5．CPLD的可编程结构是主要基于D

A.查找表（LUT） B.ROM可编程

C.PAL可编程 D.与或阵列乘积项

6．关于FPGA／CPLD，下面说法正确的是 D

A.FPGA较CPLD集成度低，触发器的数量相对较少

B.FPGA采用连续式互连结构，使其系统布线延时可以预测

C.在编程工艺方面，CPLD编程信息掉电后丢失

D.FPGA可以达到比CPLD更高的集成度，同时也具有更复杂的布线结构和逻辑实现

7．可以无外设、无总线结构、无实时操作系统（RTOS）的FPGA 嵌入式处理器的使用模式是A

A.状态机模式 B.单片机模式

C.定制嵌入模式 D.IP模式

8．基于EDA软件的FPGA／CPLD设计流程为: 原理图／HDL文本输入→综合→ → →适配→编程下载→硬件测试。正确的是

①功能仿真 ②时序仿真 ③逻辑综合 ④配置 ⑤分配管脚 B

A.③① B.①⑤

C.④⑤ D.④②

9．设a=1’b1,,b=3’b101,c=4’b1010，则X={a,b,c}的值的D

A.7’b1101100 B.8’b10101011

C.8’b 11010101 D. 8’b11011010

10．IP核在EDA技术和开发中具有十分重要的地位，已经验证的具有特定电路功能的集成电路版图方式提供的IP核被称为 A

A.硬核 B.固核

C.软核 D.都不是

11．不完整的IF语句，其综合结果可实现A

A.时序逻辑电路 B.组合逻辑电路

C.双向电路 D.三态控制电路

12．下面对利用原理图输入设计方法进行数字电路系统设计，说法正确的B

A.原理图输入设计方法适合完成大规模的电路系统设计

B.原理图输入设计方法容易实现仿真，便于信号观察和电路调整

C.原理图输入设计方法输入效率最高

D.原理图设计对于复杂系统的设计、阅读和修改都十分方便

13．在EDA工具中，能完成在目标系统器件上布局布线的软件称为 C

A.仿真器 B.综合器

C.适配器 D.下载器

14．IP核在EDA技术和开发中具有十分重要的地位，IP是指A

A.知识产权 B.互联网协议

C.网络地址 D.软件工具

15．针对FPGA生产厂商开发的EDA软件中，属于第三方公司的逻辑仿真软件是D

A.ISE B. Synplify

C.Quartus II D.ModelSim

1．用户通过一些具有通用性的单元元件或半成品硬件来制作，并且在使用时仅需要考虑电路逻辑功能和各功能模块之间的合理连线的芯片是B

A.全定制ASIC B.半定制ASIC

C.晶体管芯片 D.存储器芯片

2．设计处理的过程包括：逻辑优化和综合、映射、布局布线及 D

A.逻辑化简 B.网表文件合并

C.逻辑元件互连 D.生成编程数据文件

3．EDA软件工具大致分为五个模块：设计输入编辑器、HDL综合器、仿真器、适配器及C

A.优化器 B.布局布线器

C.下载器 D.转换器

4．在ISE Foundation集成工具中，实现FPGA／CPLD的配置和通信的工具是D

A.ECS B.XST

C.HDL Bench D.iMPACT

5．设a=4‘b1101,b=4‘b1010，按照 Verilog HDL语法，执行语句 assign c＝a＆b；则c=D

A.4’b0001 B.4’b0010

C.4’b0100 D.4’b1000

6．根据IP Core使用的方式不同，IP Core 设计者可以按三种形式设计供集成选择的IP Core：可再用、可重定目标以及A

A.可配置 B.可移植

C.可集成 D.可分析

7．已完成对性能、尺寸和功耗的优化，并对一个特定的工艺技术进行映射，以集成电路版图形式提供的IP模块是A.

A.硬核 B.固核

C.软核 D.嵌入式核

8．在Modelsim 中编译 Xilinx 器件的仿真库时，添加的 Simprim 仿真库用于D

A.综合后仿真 B.编译后仿真

C.映射后仿真 D.布局布线后仿真

9．在进行时序仿真前，需产生的支持时序仿真的文件有：综合布局布线生成的网表文件，测试激励、元件库及A

A.具有时延信息的反标文件 B.HDL程序

C.引脚锁定信息 D.功能仿真波形

10．互连结构上归类为统计型的PLD是C

A.CPLD B.GAL

C.FPGA D.PLA

11．FPGA的配置按照主/被动以及串/并行可以使用的配置模式有：从动串行，从动并行，主动串行，主动并行和A

A.JTAG模式 B.MS模式

C.SS模式 D.MSAP模式

12．Xilinx公司的三类SOPC嵌入式内核分别是：PicoBlaze、MicroBlaze和B

A.CoreConnect B.powerPC

C.LMB D.PLB

13．下面哪项属于基于EDK的嵌入式设计中的软件设计工作范畴？C

A.使用EDK工具进行硬、软核集成

B.将EDA生成的HDL代码用ISE进行综合、布局布线、仿真

C.将编译、链接后生成的可执行程序下载到目标器件，进行硬件级的调试、测试和优化

D.使用ISE编程器将生成的配置文件下载到FPGA，进行后端验证和测试

14．Xilinx 公司提供的在线逻辑分析仪是 B

A.JTAG B.ChipScope Pro

C.Floorplanner D.PROM File Formatter

15．Synplicity 公司提供的针对FPGA和CPLD实现的逻辑综合工具是B

A.XST B.Synplify

C.ISE simulator D.Xpower

1.设a=4’b1010,b＝4’b0100,则按照 Verilog HDL语法,（a＞4＇b1000) &&((b==4’b0100)的运算结果是C

A.0 B.4’b0000

C.1 D.4’b1110

2.以可综合的寄存器传输级（RTL）描述或通用库元件的网表形式提供的可重用IP模块，称为C

A.硬核 B.固核

C.软核 D.嵌入式IP Core

3.在ISE Foundation的集成工具中，可以用于辅助生成测试激励文件的是C

A. ISE Simulator B.XST

C.HDL Bench D.iMPACT

4.IP核按功能方面的划分，可以分为：嵌入式IP Core和C

A.可重用IP核 B.可重定目标IP核

C.通用IP模块 D.存储器IP核

5.Core Generator是哪个公司提供的IP核开发工具？C

A.Altera B. Aldec

C. Xilinx D. Synplicity

6.将多个模块化设计文件合并为一个网表文件，使层次设计平面化，并可以施加合理的用户约束，以满足设计要求。这是设计处理中 A

A.逻辑优化综合环节 B.映射环节

C.布局布线环节 D.生成编程数据文件环节

7.关于设计仿真，下面描述错误的是D

A.功能仿真用于验证设计的逻辑功能，没有延时信息

B.时序仿真是与实际器件工作情况基本相同的仿真

C.时序仿真应在选择了具体器件并完成布局、布线之后进行

D.时序仿真较功能仿真耗时短，对硬件库和综合器没有任何要求

8.哪一类IP核是着眼于按各种再使用标准定义的格式和快速集成的要求而建立的？A

A.可再用IP核

B.可重定目标IP核

C.可配置IP核

D.嵌人式

9.ModelSim是哪个公司开发的独立仿真器？A

A. Mentor Graphics B. Aldec

C.Xilinx D. Synplicity

10.从互联结构上分类，下面哪个器件属于统计型PLD？D

A.PROM B.PLA

C.CPLD D.FPGA

11．在可编程逻辑器件的配置方式中，由可编程器件引导的配置过程是B

A.从动配置 B.主动配置

C.同步配置 D.串行配置

12．以下哪一个软件工具可以提供MATLAB与System Generator或ISE之间的直接链接，并自动产生可综合的RTL模型和测试床？ C

A. Simulink B. Platform Studio

C.DSP TOOLS D. Accelchip

13.使用System Generator进行基于模型设计的大体流程分为：算法与系统模型开发、自动生成代码以及D

A.系统架构和波形设计

B.HDL代码的仿真、调试和分析

C.建立模型并运用MATLAB仿真验证是否符合系统要求

D.Xilinx 实现

14.下面哪一种方法可以在不使用物理探针的情况下，通过在芯片内部建立边界扫描通道，捕获其核心逻辑内容，实现芯片功能的测试？A

A.JTAG B. JPEG

C.MPEG D.VTEG

15.下面哪一款软件是Mentor Graphics公司开发的FPGA全流程设计工具？ C

A. Synplify Pro B. Active HDL

C.FPGA Advantage D. ISE Foundation

1．电子设计自动化技术的发展大致分为CAD阶段、CAE阶段和C

A.FPGA 阶段 B.CPLD阶段

C.EDA阶段 D.HDL阶段

2．设计者将所设计的系统或电路按照EDA开发软件要求的某种形式表示出来，并送入计算机的过程，这是现代数字系统设计开发流程的哪个环节？ B

A.设计准备 B.设计输入

C.设计处理 D.器件编程

3．按照仿真是否考虑硬件电路的延时信息来分类，HDL仿真器可以分为功能仿真器和D

A.VHDL仿真器 B.编译型仿真器

C.解释型仿真器 D.时序仿真器

4. ISE Foundation提供了两种测试激励文件的编写方法，一种是利用HDL编写，另一种采

用的编写工具是A

A.TestBench B.XST

C.ECS D.HDL Editor

5．设a=4’b0101 , b=4’b1010，按照 Verilog HDL语法，执行语句 assign c=(a>b)?a:b;则D

A.4’b0001 B.4’b0010

C.4’b0101 D.4’b1010

6．IP Core 按照功能方面划分可分为嵌入式IPCore和C

A.硬IPCore B.固IP Core

C.通用IP Core D.DSP IP Core

7．为合理选择IPCore，应主要从三个方面考虑：IPCore的功能，性能可靠性和A

A.实现技术 B.开发环境

C.设计时间 D.核的验证结果

8．下面关于Modelsim的描述，错误的是D

A.它是Mentor Granpics 公司开发的独立仿真器

B.它是唯一的单内核支持VHDL和Verilog HDL 混合仿真的仿真器

C.它能够对程序进行调试，测试代码覆盖率，对波形进行比较

D.它只能单独使用，不能在ISE Foundation 中被调用

9．在单独使用Modelsim时，首先建立工程，添加源文件，然后编译并导入测试文件,执行仿真的方式是C

A.时序仿真 B.编译方式

C.工程方式 D.基本方式

10．可编程逻辑器件从可编程特性分类，被认为可实现无限次编程的技术是B

A.紫外线擦除技术 B.SRAM配置存储器技术

C.闪速存储器技术 D.熔丝和反熔丝编程技术

11．基于查找表结构的可编程逻辑器件是B

A.CPLD B.FPGA

C.PAL D.GAL

12．FPGA中的嵌入式处理器一般情况下可以使用状态机模式，单片机模式和A

A.定制嵌入模式 B.寄存器模式

C.总线模式 D.编译模式

13．下面哪项是属于基于EDK的嵌入式设计中的硬件设计工作范畴？A

A.使用EDK工具进行硬、软核集成

B.编写独立于器件的C／C＋＋应用程序

C.根据EDK对系统的配置脚本生成对应的HAL库

D.将编译、链接后生成的可执行程序下载到目标器件，进行硬件级的调试、测试和优化

14．Aldec公司提供的HDL仿真器是C

A.Analysit B.Synplify Pro

C.Active HDL D.iMPACT

15．FPGA Advantage 是Mentor Graphic 公司开发的D

A.设计管理工具 B.高级仿真调试工具

C.后仿真工具 D.FPGA全流程工具

**二、名词解释题（本大题共3小题，每小题2分，共6分）**

16.PLD 可编程逻辑器件

17.FPGA 现场可编程门阵列

18.LUT 查找表

16.HDL 硬件描述语言

17.RTL 寄存器传输级

18.Hard IP CORE硬核

16.ASIC专用集成电路

17.RTL

18.SOPC 可编程片上系统

16.HDL

17.PLD

18.SOC 片上系统

16.PLD

17.RTL

18.FPGA

16.CPLD 复杂可编程逻辑器件

17.SOPC

18. IP CORE 知识产权核

16.EDO 电子设计最优化

17.SOPC

18.CPLD

**三、判断改错题（本大题共4小题，每小题4分，共16分）**

19.在EDA工具中，能将硬件描述语言转化为硬件电路原理图的重要工具软件为适配器。综合器

20.原理图输入设计方法直观便捷，但不适合完成较大规模的电路系统设计。√

21.FPGA的配置模式有从动串行模式、从动并行模式、主动串行模式、主动并行模式、以及JTAG模式。√

22.嵌入式系统的构架可分为：处理器，存储器，输入/输出(I/0)接口和应用软件四部分。软件

19.功能仿真是在选择了具体器件并完成布局，布线后进行的时序检验，可对设计性能做整体上的分析，它与实际器件工作情况基本相同。时序仿真

20.IP核在EDA技术和开发中具有十分重要的地位；其中嵌入式IP核主要是CPU与DSP,对比通用IP模块，嵌入式IP核的技术门槛相对较高，有比较大的利润空间。√

21.基于SRAM查找表结构的FPGA的在系统下载称为配置。√

22.基于电可擦除存储单元的E2PROM或Flash技术的CPLD的在系统下载称为配置,这个过程就是把编程数据写入E2CMOS单元阵列。编程

19．硬件软化是指硬件的设计使用软件编程的方式进行，整个设计和修改过程如同完成软件设计一样方便和高效。√

20．功能仿真是在选择了具体器件并完成布局、布线之后进行的快速时序检验。时序仿真

21． ISE Design Suite 10.1整合了Xilinx的嵌入式、DSP 和逻辑设计设计工具。√

22．ChipScope Pro是ISE集成套件中的片外逻辑分析工具。 片内逻辑分析红菊

19．ChipScope 是基于JTAG边界扫描测试方法的在线逻辑综合工具。逻辑分析或测试工具

20．ISE Foundation集成了输入、综合、实现、验证和配置五大功能的一套完整的FPGA设计工具。√

21．IP核在提供形式上可以分为硬核、软核和通用IP核。固核

22．Xilinx 公司的FPGA器件是利用嵌入的乘法器和DSP处理核具有了高性能的乘法和算术能力。√

19．EDA技术中的硬件软化是指硬件的设计使用软件编程的方式进行，整个设计和修改过程如同完成软件设计一样方便和高效。√

20．ISE Foundation 集成工具中的 Floorplanner 可以编辑与1／O引脚和面积约束相关的用户约束文件。错 PACE

21．通用型IP Core除IP Core本身的设计外，还需要有良好的开发环境，软件支持及完善的服务体系，其技术门槛较高，利润空间较大。嵌入式IP Core

22．通过改变Xilinx的FPGA芯片的M2，M1，M0管脚的连接，可以实现FPGA下载配置模式的切换。√

19.ISE软件对层次化设计提供了很好的支持，例如：绘制好的原理图可以很方便地转换为原理图符号，并被添加到模块符号库中，作为子模块供更高层次的设计调用。（√ ）

20.硬IP核在功能和时序上有非常可靠的保证，应用硬IP核进行设计可以显著地节省设计时间，但应用硬IP核进行设计时不能按设计需要修改和调整时序，缺少使用的灵活性。

( 对 )

21.进行功能仿真时需要综合工具生成的网表文件、测试激励文件、元件库、布局布线生成的具有时延信息的文件等。( 时序仿真前 )

22.FPGA采用EPROM或E2CMOS工艺，断电后编程数据不会丢失，因此不需要外部存储器，而且这种器件中设置有加密单元，加密后可以防止编程数据被读出。( CPLD )

19．“自底向上”设计方法首先是从系统设计入手，在顶层进行功能划分和结构设计，然后再逐级设计底层结构，最后再用EDA工具生成具体的门级逻辑电路网表。（ 自顶向下 ）

20．ISE Foundation 在综合工具上还可以内嵌Mentor Graphic公司、Synplicity 公司的产品，实现无缝链接。（ 对 ）

21．在一个完整的功能模块中，可能同时存在软IP Core部分和硬IP Core部分，软核部分提供某种程度的可调整性和适应性，硬核部分节省了芯片设计时间。（ 对 ）

22．时序仿真是直接对HDL、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的过程，仿真过程不涉及任何具体器件的硬件特性。（ 功能仿真 ）

**四、简答题（本大题共4小题，每小题5分，共20分）**

23.什么是功能仿真，时序仿真？简述它们系统设计中的应用。

24.什么是IP核?按照IP核的功能方面划分可将IP核分为哪几类？

25.在Verilog HDL赋值语句中，过程赋值与连续赋值语句有哪些区别？

26.简述可编程逻辑器件按器件集成度、编程元件这两个角度的分类情况。

23.请简要描述在EDA设计中，自顶向下设计方法的思路。

24.根据Verilog HDL语言描述的电路抽象程度不同，可以划分为哪几种层次？

25.按提供方式的不同，IP核是如何分类的？简述这几种提供方式在使用中的各自优缺点。

26.简述FPGA有哪五种配置方式？

23．FPGA中的嵌入微处理器一般情况下可以有3种不同的使用模式，分别是哪三种使用模式？

24．简述Verilog HDL语句中阻塞赋值与非阻塞赋值的不同。

25．什么是IP核？IP核有哪三种有效形式？

26．传统设计方法和EDA设计方法有何不同？

27．FPGA设计中需要完成功能仿真和时序仿真，简述时序仿真和功能仿真的不同点。

23．简述现代数字系统设计流程。

24．Verilog HDL 语言有过程赋值和连续赋值两种赋值方式，简要说明过程赋值和连续赋值的区别。

25．简述FPGA和CPLD在逻辑单元结构上的特点和差异。

26．PLD器件按照编程元件的不同，可以分为哪几类编程方式？哪一种编程方式当PLD器件掉电后配置数据会丢失？

27．简述两类有限状态机Mealy状态机和Moore 状态机的特点。

23．简述现代数字系统的设计流程。

24．什么是IPCore的复用？在现代数字系统设计中采用IP Core 复用的优点是什么？

25．简述功能仿真和时序仿真各自的目的。如果功能仿真波形显示设计出错，应进行哪些设计文件的修改？

26．FPGA和CPLD分别是基于什么结构的可编程逻辑器件？FPGA和CPLD的基本结构由哪几种可编程部分组成？

23.简述“自顶向下”设计方法的特点。

24.简述HDL综合器的作用是什么。在这一过程中，包括了哪三个步骤？

25.单独使用ModelSim有哪两种使用方式？这两种方式各自的仿真步骤是什么？

26.简述为实现数字电路系统合理选择CPLD或FPGA，应从哪几个方面进行考虑？

23．可综合的VHDL／Verilog HDL转化成硬件电路时，包含了哪三个过程？简述每个过程实现的工作。

24．简述软核与硬核在应用中各自的优缺点。

25．在Modelsim 中编译 Xilinx 器件仿真库时，需要添加 Simprims、Unisims、Xilinxcorelib三个库，简述以上三个库函数编译后的各自用途。试简要写出在Modelsim 仿真器中编译仿真库的操作步骤。

26．简述FPGA和CPLD在内部互连资源与连线结构上的特点与差异。

**五、程序分析题（本大题共2小题，第27题12分，第28题12分，共24分）**

27.分析下面的tes1.v程序。

1）完成程序填空。

2）分析程序，根据输入信号得到相应功能仿真的输出信号，完成表一的填空。（12分）

module test1 (data1,data2,sel,dout1,dout2,dout3);

input data1,data2;

（空1） sel;

output dout1,dout2,dout3;

（空2） dout1;

reg dout2,dout3;

assign dout1 = data1 ^ data2;

always @ (data1 or data2 or sel)

if(sel==1'b1)

begin

dout2 = data1 & data2; dout3 = data1 | data2;（空3）

else

begin

dout2 = data1; dout3 = data2;end

endmodule

表一

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| data1 | data2 | sel | dout1 | dout2 | dout3 |
| 1’b1 | 1’b0 | 1’b1 | 空4 | 空5 | 空6 |

28.分析以下Verilog HDL代码，画出test2.v代码描述的电路图。

module test2(d,clk,q);

input d;

input clk;

output q;

reg q;

reg data1,data2,data3

always @ (posedge clk)

begin

data1 <= d;

data2 <= data1;

data3 <= data2;

q <= data3;

end

endmodule

27.分析下面的tes1.v程序。

1）完成程序填空。

2）分析程序，根据输入信号得到相应功能仿真的输出信号，完成表一的填空。（14分）

module test1(in1,in2,in3,in4,dout1,dout2);

input [1:0] in1,in2,(空1),in4;

output[1:0] dout1,（空2）;

reg （空3） dout1,temp;

（空4） [1:0] dout2;

function [1:0] mytest;

input [1:0] a,b;

if(a > b)

mytest = a;

（空5） mytest = b;

endfunction

always @ (in1 or in2 or in3 or in4)

begin

dout1 = mytest(in1,in2);

temp = mytest(in3,in4); end

assign dout2 = dout1 & temp;

endmodule

表一

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| in1 | in2 | in3 | in4 | dout1 | dout2 |
| 2’b11 | 2’b10 | 2’b01 | 2’b00 | 空6 | 空7 |

28、分析以下Verilog HDL代码，画出代码描述的状态机转移图。(12分)

module fms(clk,reset,data\_in,data\_out);

input clk,reset;

input [1:0] data\_in;

output [3:0] data\_out;

reg [1:0] cur\_state,next\_state;

reg [3:0] data\_out;

always @ (posedge clk)

if(reset)

cur\_state <= 2'b00;

else cur\_state <= next\_state;

always @ (cur\_state or data\_in)

case(cur\_state)

0: begin

data\_out = 4'b1011;

if(data\_in==2'b00)

next\_state = 0;

else next\_state = 1;end

1: if(data\_in==2'b10)

begin data\_out = 4'b1011; next\_state = 1;end

else

begin data\_out = 4'b1010; next\_state = 2;end

2: begin

data\_out = 4'b1011;

if(data\_in==2'b11)

next\_state = 3;

else next\_state = 1;end

3: begin

data\_out = 4'b1011;

if(data\_in==2'b01)

next\_state = 0;

else next\_state = 2;end

default :begin data\_out = 4'b1011; next\_state = 0;end

endcase

endmodule

28．分析test1程序模块，完成下列题目：

（1）完成程序填空。

（2）分析程序，根据输入信号得到功能仿真相应的输出信号，完成题28表填空。

module test1 (data\_in1,data\_in2,data\_in3,data\_out);

input data\_in1,data＿in2,空（1）；

output data\_out;

wire data\_out;

reg [1:0] data\_1

task example;

input in1in2;

output [1:0] out1

begin

out1[1]=in1& in2;

out1[0]=in1 | in2;

end空（2）

always @ (data\_in1 or data\_in2)

example (data\_in1,data\_in2,data\_1);

assign data\_out =(data\_in3)?data\_1[1]:data\_1[0];

endmodule

题28表

|  |  |  |  |
| --- | --- | --- | --- |
| data\_in1 | data\_in2 | data\_in3 | data\_out |
| 1’b1 | 1’b0 | 1’b1 | 空（3） |
| 1’b0 | 1’b1 | 1’b0 | 空（4） |

29．分析test2程序模块，画出该模块综合后输出的电路原理图。

module test2(clk, reset, din1, din2, din3, dout1, dout2);

input clk, reset;

input din1,din2,din3;

output dout1,dout2;

reg dout1,dout2;

reg data\_temp;

always@(posedge clk)

if(reset)

begin

data\_temp <=1’b0;

dout1<=1‘b0;

dout2<=1’b0;

end

else

begin

data\_temp <= din1 & din2;

dout1<out1 <= data\_temp | din3;

dout2 <= ~dout1;

end

endmodule

28．分析count模块功能，完成下列题目：

（1）说明reset，load在模块中分别起什么作用？

（2）根据题28图的仿真输入波形，画出0到500纳秒的输出信号的功能仿真波形（注：时钟周期是20ns）。

module count(data,clk,reset,load,cout,qout);

output cout;

output[3:0] qout;

reg[3:0] qout;

input[3:0] data;

input clk,reset,load;

always @(posedge clk)

begin

if (!reset) qout= 4'h00;

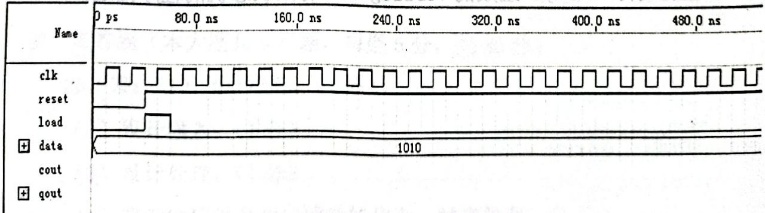
else if (load) qout= data;

else qout=qout+1'b1;

end

assign cout=(qout==4'hf)?1'b1:1'b0;

endmodule



题28图

29．下列程序分别给出名为dff2底层D触发器模块和名为top顶层模块，请画出综合后的顶层模块的电路图。

D触发器模块：

module dff2(clk,d,q);

input clk,d;

output q;

reg q:

always @(posedge clk)

q <= d;

endmodule

顶层模块：

module top(A,CLK,C,B);

input A,CLK;

output C,B;

wire s1,s2;

assign s2=s1&A;

dff2 U1(CLK,A,s1);

dff2 U2(CLK,s2,C);

dff2 U3(CLK,s1,B);

endmodule

27．分析下面的Verilog HDL 源程序，回答问题。

module test1(in1,in2,in3,out1, );

input[2:0] in1,in2,in3;

output[2:0] ,out2;

reg out1,out2;

always @ (in1 or in2 or in3)

begin

out1=in1 & in2;

out2=out1 lin3;

end

endmodule

（1）完成程序填空。

（2）如果输入信号值如题27表所示，写出功能仿真后相应的输出信号值，完成表格填空。

题27表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| in1 | in2 | in3 | out1 | out2 |
| 3’b001 | 3’b011 | 3’b100 |  |  |

28．分析下面的 Verilog HDL源程序，回答问题。

module  test2\_1(clk,q.d):

input clk,d;

output q;

always @ (posedge clk)

q<=d;

endmodule

module test2 (a,b,clk,data\_out);

input a,b;

input clk;

output data\_out;

wire data\_out;

temp;

assign temp = a & b;

test2\_1 u1(.clk(clk),.q( ),.d(temp));

endmodule

（1）完成程序填空。

（2）画出test2.v综合后的电路图。

27．分析下面的Verilog HDL程序，回答问题。

module testl(cout,sum,a,b,cin);

output cout;

output[2:0] ；

input cin;

input[2:0] a,b;

{cout,sum}=a+b+cin;m}=a+b+cin

endmodule

（1）完成程序填空。

（2）分析该模块的功能，根据输人信号得到相应的输出信号，完成题27表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入信号 | | | 输出信号 | |
| a | b | cin | cout | sum |
| 3'010 | 3'b011 | 1'b0 |  |  |
| 3'b101 | 3'b100 | 1'b1 |  |  |

题27表

28．分析下面的三段Verilog HDL程序，画出该程序综合后的电路原理图。

子模块1：

module test2(a,b,c);

input a,b;

output c;

assign c = a & b;

endmodule

子模块2：

module test3(clk,d,q);

input clk;

input d;

output q;

reg q;

always @ (posedge clk)

q <= d;

endmodule

顶层模块

module test4(ain,bin,clk,data\_out);

input ain,bin;

input clk;

output data\_out;

wire data\_temp\_1,data\_temp\_2,data\_temp3;

test3 u1(.clk(clk),.d(ain),.q(data\_temp\_1));

test3 u2(.clk(clk),.d(bin),.q(data\_temp\_2));

test2 u3(.a(data\_temp\_1),.b(data\_temp\_2),.c(data\_temp3));

test3 u4(.clk(clk),.d(data\_temp3),.q(data\_out));

endmodule

27．分析下面的 Verilog HDL程序，回答问题。

module testl (in1,in2,in3,out1,out2);

input[1:0] in1,in2,in3;

output [1:0] out1, ;

[1:0] out1;

[1:0] out2;

always @ (in1 or in2)

outl=in1 & in2;

assign out2=in2 | in3;

endmodule

（1）完成程序填空。

（2）如果输入信号值如题27表所示，写出功能仿真后相应的输出信号值，完成填表。

题27表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| in1 | in2 | in3 | out1 | out2 |
| 2’b00 | 2’b01 | 2’b10 |  |  |

28．分析下面的Verilog HDL程序，回答问题。

module test2(data\_in,data\_out,clk);

input data\_in;

input clk;

output data\_out;

data\_templ,data\_temp2,data\_out;

@(posedge clk)

data\_templ <=data\_in;

data\_temp2<=data\_templ;

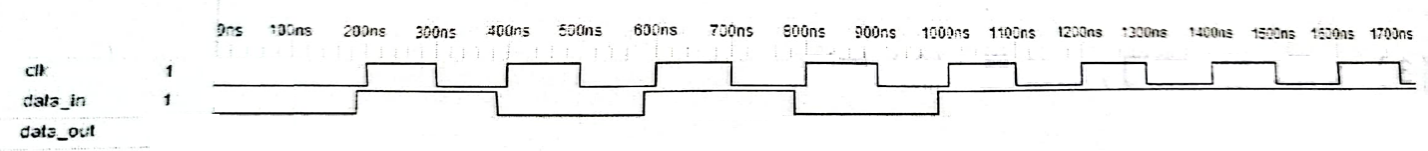
data\_out<=data\_temp2;

end

endmodule

（1）完成程序填空。

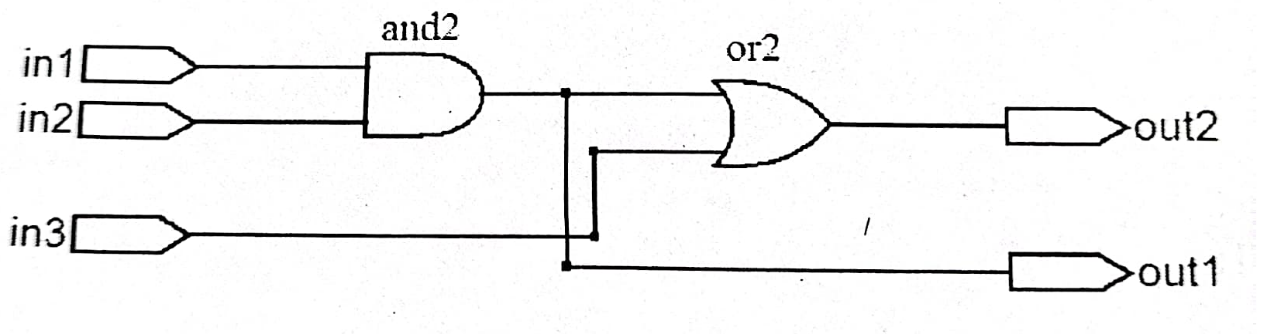
（2）根据题28图给出的测试文件波形，画出输出信号的功能仿真波形。



题28图

六、编程題（本大题共2小题，第29小题8分，第30小题11分，共19分）

29．根据题29图所示原理图写出相应的Verilog HDL程序。



题29图

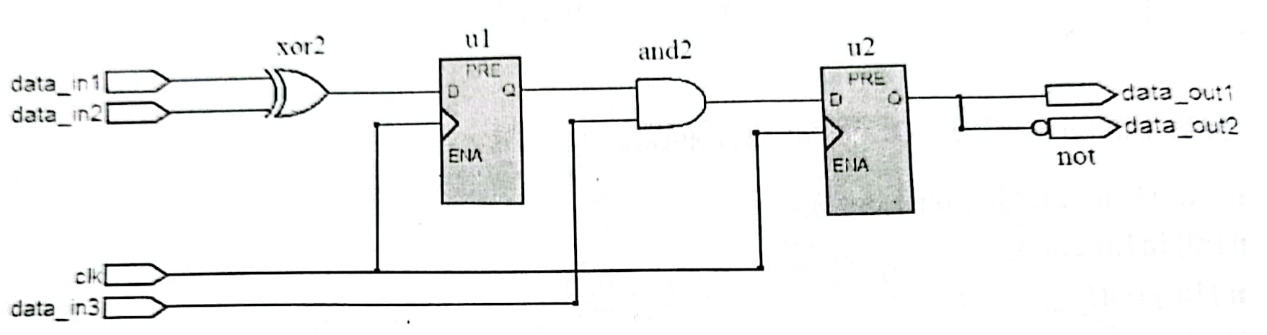
说明：

（1）模块输入端口：in1，in2，in3。

（2）模块输出端口：out1，out2。

（3）or2是二输入或门，and2是二输入与门。

30．根据题30图中所示原理图写出相应的Verilog HDL程序。



题30图

说明：

例化名为u1，u2的模块均是时钟clk上升沿触发的触发器。

xor2是二输入异或门，and2是二输入与门，not是一输入非门。

输入信号：

data＿in1，data＿in2，data＿in3：均是位宽为1bit的信号。

clk：系统时钟。

输出信号：

data＿out1，data＿out2：均是位宽为1bit的信号。

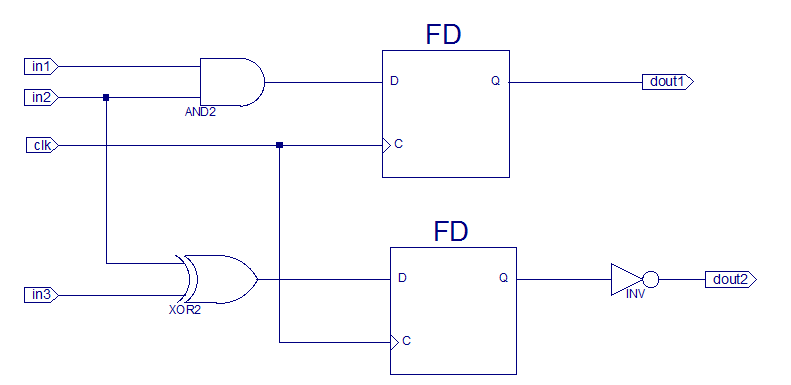
**六、编程题（本大题共2题，第29题9分，第30题10分，共19分）**

29.用Verilog HDL设计一个组合逻辑电路系统，按表二所示关系，输出信号驱动至共阳极七段LED以显示相应信息。（9分）

表二 输入信号与输出信号的转换关系

|  |  |  |
| --- | --- | --- |
| **系统输入**  **data\_in[2:0]** | **系统输出到七段LED的数据**  **data\_out[6:0]（共阳极）** | **七段LED图例** |
| 3′b000 | 7′b1000000 | 7led |
| 3′b001 | 7′b1111001 |
| 3′b010 | 7′b0100100 |
| 3′b011 | 7′b0110000 |
| 3′b100 | 7′b0011001 |
| 3′b101 | 7′b0010010 |
| 3′b110 | 7′b0000010 |
| 3′b111 | 7′b1111000 |
| 其它情况 | 7′b1000000 |

30.图一是一个含有上升沿触发的D触发器的时序电路，试写出设计此电路的Verilog HDL程序。



图一

29.按表二所示真值表，用Verilog HDL设计一个38译码器（9分）

表二 真值表

|  |  |  |
| --- | --- | --- |
| 使能信号ena | 输入数据din[2:0] | 输出数据dout[7:0] |
| 1 | XXX | 8’b11111111 |
| 0 | 3’b000 | 8’b01111111 |
| 0 | 3’b001 | 8’b10111111 |
| 0 | 3’b010 | 8’b11011111 |
| 0 | 3’b011 | 8’b11101111 |
| 0 | 3’b100 | 8’b11110111 |
| 0 | 3’b101 | 8’b11111011 |
| 0 | 3’b110 | 8’b11111101 |
| 0 | 3’b111 | 8’b11111110 |

端口说明：

输入端口：

din-------输入数据，位宽3比特

ena-------使能控制信号，当ena=1时，系统输出恒定为8’b1111\_1111，当ena=0时，根据输入数据，按表二对应关系输出数据。

输出端口：

dout-------输出数据，位宽8比特

30.用Verilog HDL设计对输入时钟进行四分频的模块，具体要求如下：

输入端口：

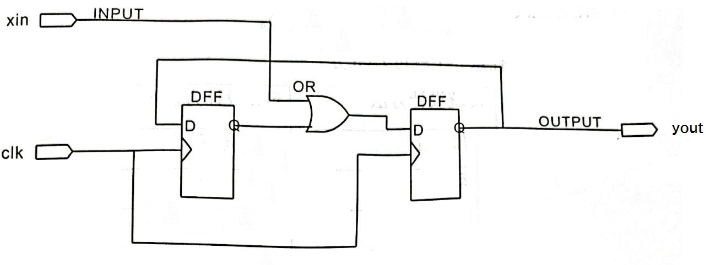
clk -----输入时钟

reset----同步复位信号，当reset=1时，系统输出复位为0，当reset=0，系统正常工作。

输出端口：

clk\_out----输出时钟，clk\_out，clk的频率关系是：

30．根据下图所示原理图写出相应的Verilog HDL程序。



题30图

31．设计一个有限状态机系统，在时钟clk上升沿的控制下，实现检测与clk同步输入的串行数据是否为“3＇b110”。（10分）

要求：

＊当串行数据是“3＇b110”时，fflag\_out=1‘b1，否则flag\_out=1‘’b0，

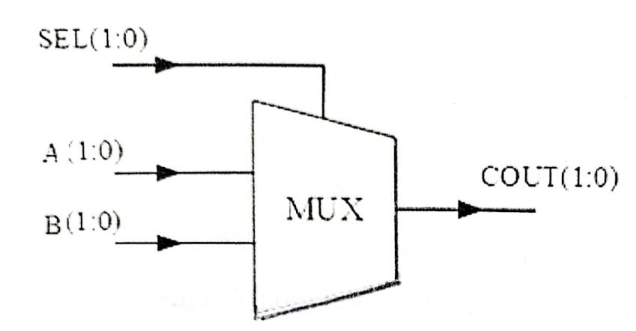
＊系统可以实现同步复位。

\*输入输出信号自行定义，并在程序注释中加以说明。

用Verilog HDL 编程实现该系统的模块程序。

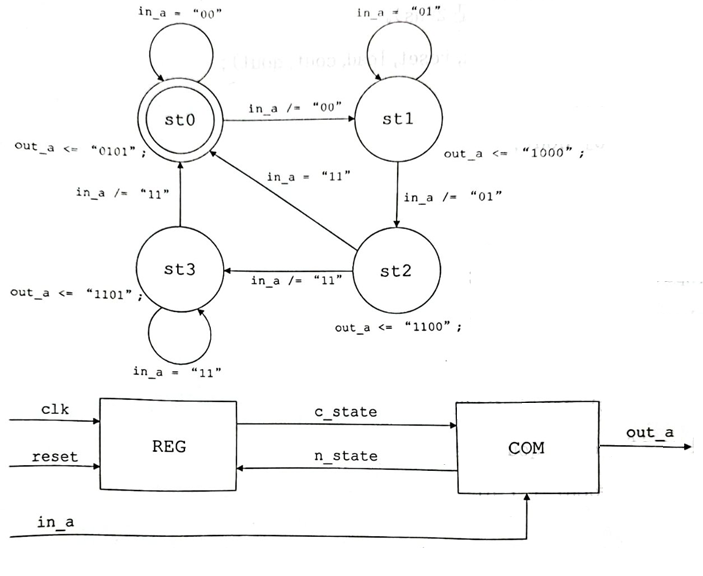
30．使用Verilog HDL描述语言设计一数据选择器MUX，其系统模块图和功能表如题30图所示

|  |  |
| --- | --- |
| SEL | COUT |
| 00 | A or B |
| 01 | AxorB |
| 10 | A and B |
| 11 | A xnor B |
| OTHERS | “XX” |



题30图

31．已知状态机状态图和状态机结构图如题31图所示，完成下列各题：



29．用Verilog HDL设计一个3位二进制编码器电路，其真值表如题29表所示

题29表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入信号 | | | | | | | | 输出信号 | | |
| I7 | 16 | 15 | 14 | I3 | 12 | I1 | 10 | Y2 | Y1 | YO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

说明：

（1）输入信号：

I7、I6、I5、I4、I3、12、I1、I0：均是位宽为1bit的信号。

（2）输出信号：

Y2、Y1、YO：均是位宽为1bit的信号。

30．用Verilog HDL设计一个异步复位、同步使能的六十进制计数器。对时钟clk上升沿进行计数。具体要求如下：

（1）输入信号：

clk：时钟信号。

reset：异步复位信号。当reset=1时，输出信号置0，当reset=0时，系统正常计数。

en：同步使能信号，当en=0，保持当前数值不变；当en=1时，系统正常计数。

（2）输出信号：

Carry：位宽为1bit，表示进位。

Cout：位宽为5bits，表示0到59的计数值。

29．题29表为3-8译码器真值表。请根据该真值表，采用Verilog HDL编写3-8译码器电路。

题29表 3-8译码器真值表

|  |  |  |
| --- | --- | --- |
| 输入信号 | | 输出信号 |
| Enable[2:0] | Data\_in[2:0] | Q[7:0] |
| 100 | 3'b000  3'b001  3'b010  3'b011  3'b100  3'b101  3'b110  3'b111 | 8'b11111110  8'b 11111101  8'b 11111011  8'b 11110111  8'b 11101111  8'b 11011111  8'b 10111111  8'b 01111111 |
| 其它 | 其它 | 8'b 11111111 |
|  |  |  |

30．用Verilog HDL设计一个串行数据转并行数据，并在输入信号的控制下，实现并行数据向左或右移位的模块，具体要求如下：

（1）在时钟clk上升沿控制下，1位数据串行输入，8位数据并行输出。

（2）输入信号：

clk：系统时钟；

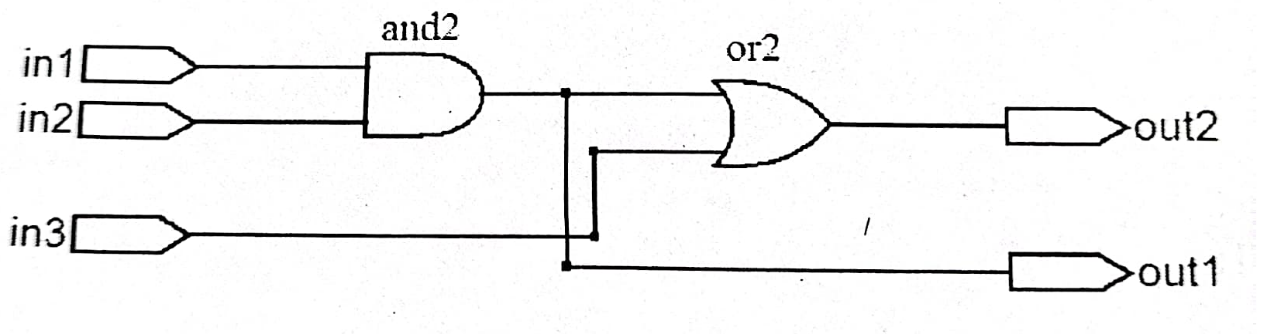
reset：同步复位信号。当resett=1’b1时，输出置0，当reseeset=1‘b0时，系统正常工作；data＿in：位宽是1的输入数据；

lorR：控制信号，控制输出数据左移或右移；当LrR＝1＇b1，输出数据由在前一时钟周期输出的并行数据右移一位，空出的位置由最新输人的数据补足而得到；当LorR＝1＇b0时，输出数据由在前一时钟周期输出的并行数据左移一位，空出的位置由最新输入的数据补足而得到。

（3）输出信号：

data＿out：位宽为8的并行数据输出

29．根据题29图所示原理图写出相应的Verilog HDL程序。



题29图

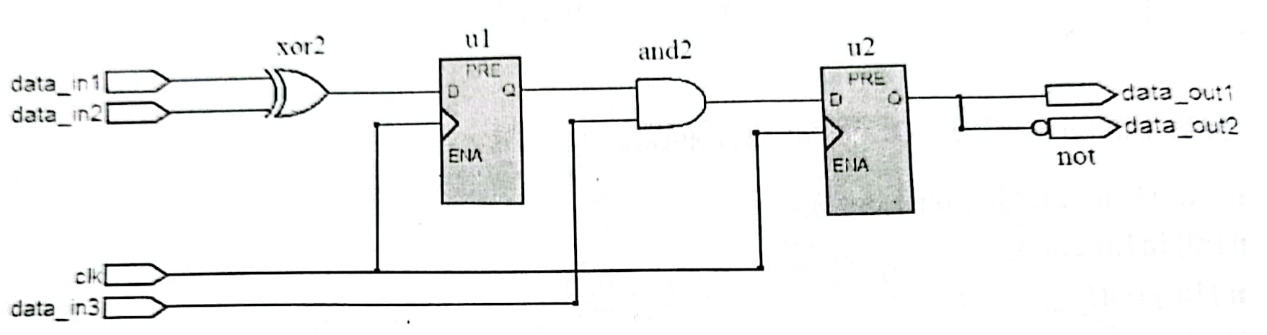
说明：

（1）模块输入端口：in1，in2，in3。

（2）模块输出端口：out1，out2。

（3）or2是二输入或门，and2是二输入与门。

30．根据题30图中所示原理图写出相应的Verilog HDL程序。



题30图

说明：

例化名为u1，u2的模块均是时钟clk上升沿触发的触发器。

xor2是二输入异或门，and2是二输入与门，not是一输入非门。

输入信号：

data＿in1，data＿in2，data＿in3：均是位宽为1bit的信号。

clk：系统时钟。

输出信号：

data＿out1，data＿out2：均是位宽为1bit的信号。

参考答案：

1. 单**项选择题（本大题共15小题，**每小题1分，共15分）

1.B 2. A 3. B 4. B 5.D 6.A 7.D 8.D 9.A 10.B 11.A 12.D 13.A 14.D 15.D

1.C 2.D 3.B 4.A 5.A 6.C 7.B 8.D 9.C 10.B 11.C 12.C 13.B 14.A 15.B

1.B 2.A 3.B 4.B 5.A 6.C 7.A 8.A 9.C

10.A 11.D 12.C 13.D 14.B 15.D

1.C  2.B 3.C  4.A  5.D  6.D 7.A 8.B 9.D

10.A  11.A  12.B  13.C  14.A  15.D

1.B 2.D 3.C 4.D 5.D 6.A 7.A 8.D 9.A 10.C 11.A 12.B 13.C 14.B 15.B

1.C  2.C  3.C  4.C  5.C 6.A  7.D 8.A 9.A 10.D 11.B 12.C  13.D  14.A  15.C

1.C 2.B 3.D 4.A 5.D 6.C 7.A 8.D 9.C 10.B 11.B 12.A 13.A 14.C 15.D

**二、名称解释题（本大题共3小题，每小题2分，共6分）**

16.可编程逻辑器件

17.现场可编程门阵列

18.查找表

16.硬件描述语言

17.寄存器传输级

18.硬核

16. ASIC 专用集成电路

17. RTL寄存器传输级

18. SOPC 可编程片上系统

16. HDL 硬件描述语言

17. PLD 可编程逻辑器件

18. SOC 片上系统

16．PLD 可编程逻辑器件

17．RTL 寄存器传输级

18．FPGA 现场可编程门阵列

16．CPLD 复杂可编程逻辑器件

17．SOPC 可编程片上系统

18．IP CORE知识产权芯核

16．EDO 电子设计最优化

17．SOPC片上可编程系统

18．CPLD 复杂可编程逻辑器件

**三、判断改错题（本大题共4小题，每小题4分，共16分）**

19.错，适配器 改为：综合器

20.对

21.对

22.错，应用软件改为软件

19.错，功能仿真 改为 时序仿真

20.对

21.对

22.错 配置,改为编程

19.(✓)

20．（x）正确：时序仿真。

21.(✓)

22．（x）正确：片内逻辑分析工具。

19．x，正确：ChipScope 是基于边界扫描测试方法的在线逻辑分析或测试工具。

20. √

21．x，正确：IP核在提供形式上可以分为硬核、软核和固核。

22. √

19．对

20．对

21．错 功能仿真改为：进行时序仿真前

22．错 FPGA改为：CPLD

19．错。 “自底向上”改为“自顶向下”。

20．对。

21．对。

22．错。 “时序仿真”改为“功能仿真”。

**四、简答题（本大题共4小题，每小题5分，共20分）**

23.什么是功能仿真，时序仿真？简述它们系统设计中的应用。

答：功能仿真是用于验证设计的逻辑功能，它是在设计输入完成之后，选择具体器件进行编译之前进行的逻辑功能验证。时序仿真是在选择了具体器件并完成布局、布线之后进行的快速时序检验，并可对设计性能作整体上的分析，这也是与实际器件工作情况基本相同的仿真。在设计中，首先进行功能仿真，待确认设计文件满足设计要求的逻辑功能后，再进行综合、布局布线和时序仿真，把握设计项目在实际器件的工作情况。

24.什么是IP核?按照IP核的功能划分可将IP核划分为那几类？

答：IP核是知识产权核心，特指可以通过知识产权贸易在各设计公司间流通的完成特定功能的电路模块。按照功能划分可将IP核划分为嵌入式IP Core ,通用IP Core。

25. 在Verilog HDL赋值语句中，过程赋值与连续赋值语句有哪些区别？

答：（1）过程赋值没有关键词，连续赋值的关键词为assign；

（2）过程赋值使用“=”和“<=”赋值，连续赋值只能用“=”；

（3）过程赋值只能出现在initial和always语句中，连续赋值不能出现在initial和always语句中；

（4）过程赋值用于驱动寄存器，连续赋值用于驱动线网。

26.简述可编程逻辑器件按器件集成度、按编程元件这两个角度分类情况。

答：按集成度分：低密度可编程器件，高密度可编程器件

按编程元件分：熔丝或反熔丝开关，浮栅编程技术，SRAM配置存储器。

23.请简要描述在EDA设计中，自顶向下设计方法的思路。

答：自顶向下首先从系统设计入手，在顶层进行功能划分和结构设计，并在系统级采用仿真手段验证设计的正确性，然后再逐级设计低层的结构，实现从设计、仿真、测试一体化。其方案的验证与设计、电路与PCB设计专用集成电路设计等都由电子系统设计师借助于EDA工具完成。

24.根据Verilog HDL语言描述的电路抽象程度不同，可以划分为哪几种层次？

答：（1）行为级或算法级描述

（2）数据流级描述

（3）门级描述

（4）开关级描述

25.按提供方式的不同，IP核是如何分类的？简述这几种提供方式在使用中的各自优缺点。

答：按提供方式分为三种：硬核，固核，软核，硬核性能稳定可靠，但是不能修改，软核可移植性好，但是性能不能最优，固核介于两者之间。

26.简述FPGA有哪五种配置方式？

答：（1）从动串行模式

（2）从动并行模式

（3）主动串行模式

（4）主动并行模式

（5）JTAG模式

23.FPGA中的嵌入微处理器一般情况下可以有3种不同的使用模式，分别是哪三种使用模式？

答：FPGA中的嵌入微处理器一般情况下可以有3种不同的使用模式，分别是状态机模式、单片机模式和定制嵌入模式。

24．答：（1）阻塞赋值：表示符号“＝”，必须是阻塞赋值完成后，才进行下一条语句的执行；赋值一旦完成，等号左边的变量值立刻发生变化。（2）非阻塞赋值：表示符号“，非阻塞赋值在赋值开始时计算表达式右边的值，到了本次仿真周期结束时才更新被赋值变量（即=>赋值不立刻生效）；非阻塞赋值允许块中其他语句的同时执行。在同一个顺序块中，非阻塞赋值表达式的书写顺序，不影响赋值的结果。

25．答：IP核是知识产权核或知识产权模块。三种形式是：硬核、固核、软核。

26．（答对任意5个就可）

|  |  |
| --- | --- |
| 传统设计方法 | EDA 设计方法 |
| 自底向上 | 自顶向下 |
| 手动设计 | 自动设计 |
| 硬软件分离原理图方式设计 | 打破硬软件屏障 |
| 原理图 | 原理图，VHDL语言等多种设计方式 |
| 系统功能固定 | 系统功能易变 |
| 不易仿真 | 易仿真 |
| 难测试修改 | 易测试修改 |
| 模块难移置共享 | 设计工作标准化，模块可移置共享 |
| 设计周期长 | 设计周期短 |

27.答：时序仿真就是接近真实器件运行特性的仿真，仿真文件中己包含了器件硬件特性参数，因而，仿真精度高。功能仿真是直接对HDL、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的过程，仿真过程不涉及任何具体器件的硬件特性。

23．答：（1）设计准备；（2）设计输入；（3）设计处理；（4）器件编程以及相应的功能仿真、时序仿真；（5）器件测试三个设计验证过程。

24．答：

（1）过程赋值没有关键词，连续赋值的关键词为assign；

（2）过程赋值使用“＝”和“＜＝”赋值，连续赋值只能用“＝”；

（3）过程赋值只能出现在initial 和always 语句中，连续赋值不能出现在initial和always语句中。

25．答：

（1）CPLD中的逻辑单元采用PAL结构，由于这样的单元功能强大，一般的逻辑在单元内均可实现，故互连关系简单，还具有很宽的输入结构，适合于实现高级的有限状态机。

（2）FPGA逻辑单元采用查找表结构，每单元只有一个或两个触发器，这样的工艺结构占用芯片面积小、速度高，每块芯片上能集成的单元数多，但逻辑单元的功能较弱，较适合实现数据型系统。

26．答：PLD器件按照编程方式不同，可以分为熔丝（Fuse）、反熔丝，浮栅编程技术、SRAM配置技术。基于SRAM配置技术的编程方式当PLD器件掉电后配置数据会丢失。

27．答：Mealy 状态机是状态机的输出是输入信号和当前状态的函数，当前状态和输入状态共同决定后续状态；Moore 状态机是状态机的输出只是当前状态的函数，由当前状态决定后续状态。

23．答：现代数字系统设计流程如下：

（1）设计准备：进行方案论证，系统设计和器件选择等设计准备工作

（2）设计输入：设计者将所设计的系统或电路按照EDA开发软件要求的某种形式表示出

来，并送入计算机的过程。

（3）设计处理：从设计输入文件到生成编程数据文件的编译过程。

（4）设计校验：使用EDA开发软件对设计进行分析，验证。

（5）器件编程：是把系统设计的下载或配置文件，通过编程电缆按一定格式装入一个或多个PLD编程存储单元，用于定义PLD内部模块的逻辑功能以及它们的相互连接关系。

24．答：IP Core 复用是指在设计新产品时采用已有的各种功能模块。

采用IP Core 复用的优点：修改非常有限，减少设计的人力和风险，缩短设计周期，确保优良品质。

25．答：功能仿真是直接对HDL、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的过程。

时序仿真是是在EDA 工具布局布线后，取得芯片的门延时、线延时信息后，并综合考虑系统各功能，对芯片真实工作情况的一种仿真。

如果功能仿真波形出错，应该修改系统方案和以HDL、原理图为代表的设计输入文件。

26．答：CPLD是基于乘积项的可编程逻辑结构，FPGA是基于查找表的可编程逻辑结构。FPGA和CPLD的基本结构由可编程逻辑单元，可编程内连阵列（快速互联通道）可编程I／O。

23．答（1）基于可编程逻辑器件PLD和EDA开发工具支撑（1分）。

（2）采用系统级、电路级、门级的逐级仿真技术，以便及早发现问题，进而修改设计方案。

（3）现代的电子应用系统正向模块化发展，或者说向软硬核组合的方向发展。对于以往成功的设计成果稍作修改、组合就能投入再利用，从而产生全新的或派生的设计模块。

（4）由于采用的是结构化开发手段，可实现多人多任务的并行工作方式，使复杂系统的设计规模和效率大幅度提高。

（5）在选择器件的类型、规模、硬件结构等方面具有更大的自由度。

24．答：逻辑综合就是使用EDA综合工具，将用HDL语言描述的寄存器传输级电路转化成门级网表。包含了三个过程：转化、优化、映射。

25．答：单独使用ModelSim有两种方式，基本方式和工程方式。基本方式的仿真步骤：首先建立仿真库，编译源代码，然后启动仿真器，执行仿真。工程方式的仿真步骤：首先建立工程，添加源文件，然后编译并导入测试文件，执行仿真。

26．答：（1）逻辑单元结构方面：系统逻辑复杂，输入变量多，但对触发器需求量较少，可用CPLD，系统所需触发器数量多，但逻辑相对简单，可用FPGA．

（2）内部互连资源与连线结构：系统要求很好的互联灵活性和很高的布线成功率，但不要求布线前预测信号传输延迟的，可以选择FPGA，如果要求较大的时间可预测性，选用CPLD

（3）配置技术：如果系统要求实现动态重构，下载信息保密性要求不高，可选用FPGA，否则，选用CPLD．

（4）逻辑规模：中小规模系统可以选CPLD，中大规模可选FPGA．

（5）FPGA和CPLD封装形式的选择：根据系统规模，保密性要求，生产成本等选择。

23．答：转化：HDL转化为各个功能单元连接的电路结构的门级网表。优化：根据设计者所施加的时序、面积等约束条件，针对实际实现的目标器件的结构将转化的门级网表按一定算法进行逻辑重组和优化。映射，从目标器件的工艺库中搜索恰当的单元来构成电路。

24．答：（1）软核比较灵活、可以改动，易于重定目标和重配置，然而预测软IP核的时序、面积与功率诸方面的性能较困难。

（2）硬核的优点是确保性能，如速度、功耗等，但硬核一般不允许更改，难以转移到新工艺或集成到新结构中，它是不可重配置的。

25．答：Simprims 用于布局布线后的仿真，Unisims 用于综合后的仿真，Xilinxcorelib用于支持IP Core的仿真。

步骤：修改 modelsim．ini 文件属性，去掉只读属性；启动Modelsim 仿真工具，创建仿真库，编译仿真库。

26．答：FPGA的分段式连线结构提供了很好的互连灵活性和很高的布线成功率，但是这种连线结构也具有明显的缺点，由于每个信号的传输途径和金属线长度各异，显然一对单元之间的互连路径可以有多种，它的信号传输延迟时间不能确定。

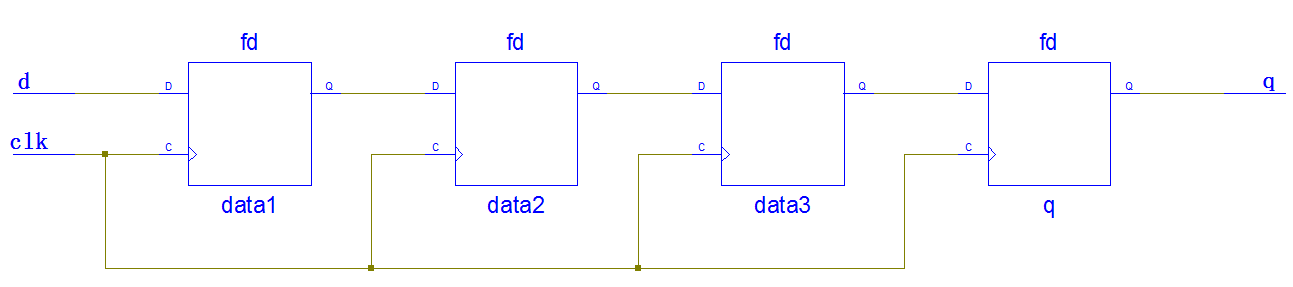
CPLD的连续式互连结构是利用具有同样长度的一些金属线实现功能单元之间的互连，即用的是集总总线，所以其总线上任意一对输入端与输出端之间的延时相等，因而有较大的时间可预测性。

**五、程序分析题（本大题共2小题，第27题12分，第28题12分，共24分）**

27.（1）input （2分） wire（2分） end（2分）

（2）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| data1 | data2 | sel | dout1 | dout2 | dout3 |
| 1’b1 | 1’b0 | 1’b1 | 1’b1 | 1’b0 | 1’b1 |

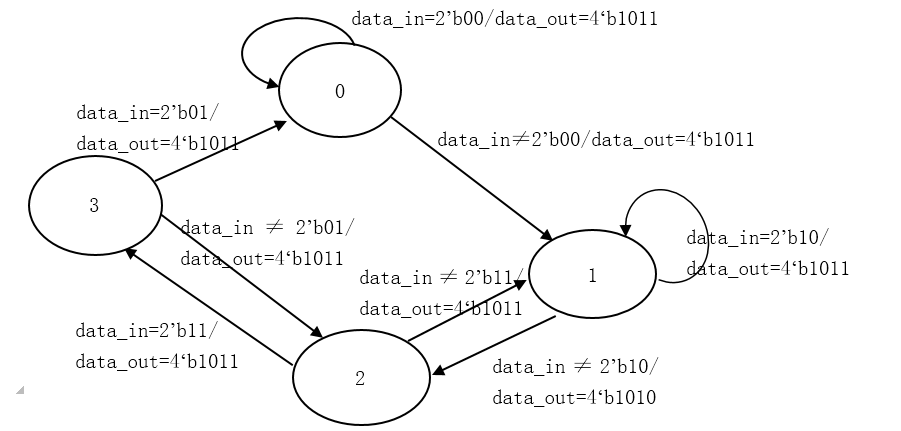
28. 

27.（1）in3 dout2 [1:0] wire else

（2）答案：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| in1 | in2 | in3 | in4 | dout1 | dout2 |
| 2’b11 | 2’b10 | 2’b01 | 2’b00 | 2’b11 | 2’b01 |

28.

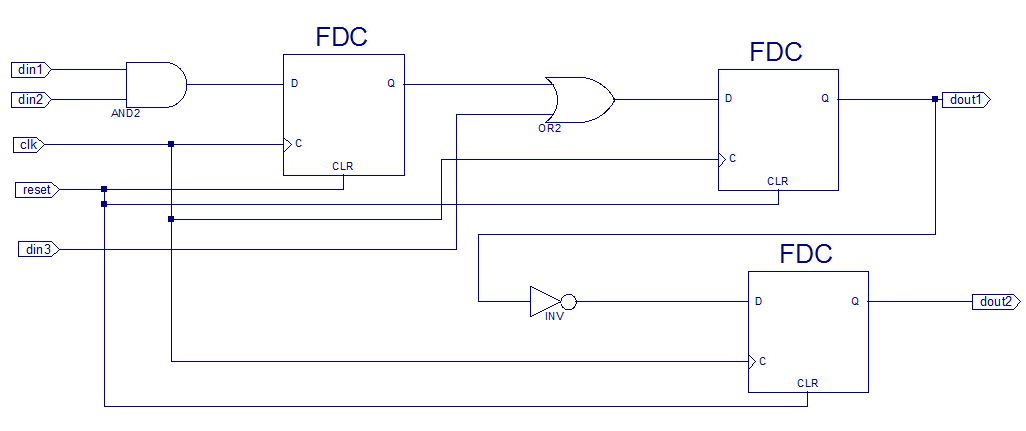


28．答案：

空（1）data＿in3 空（2）endtask 空（3）1’b0 空（4）1’b1

29．答案：

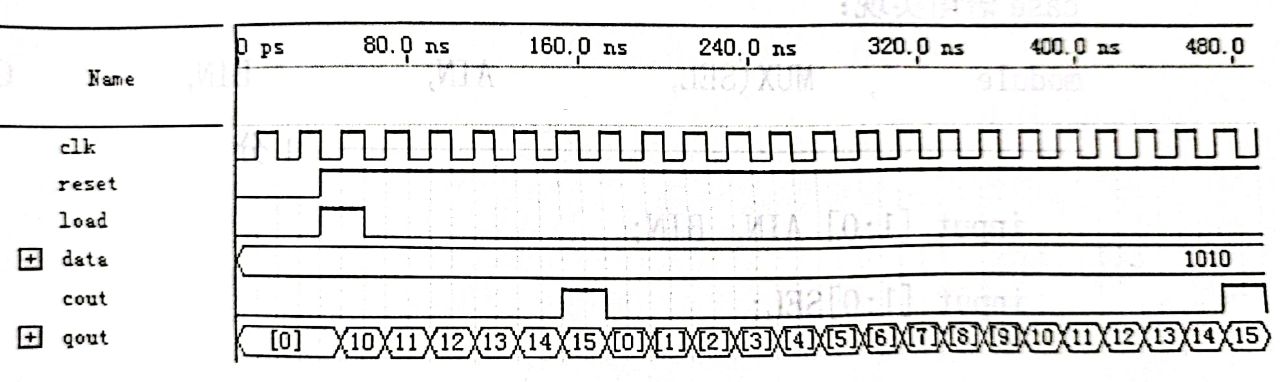
综合后输出的电路原理图如下图所示：



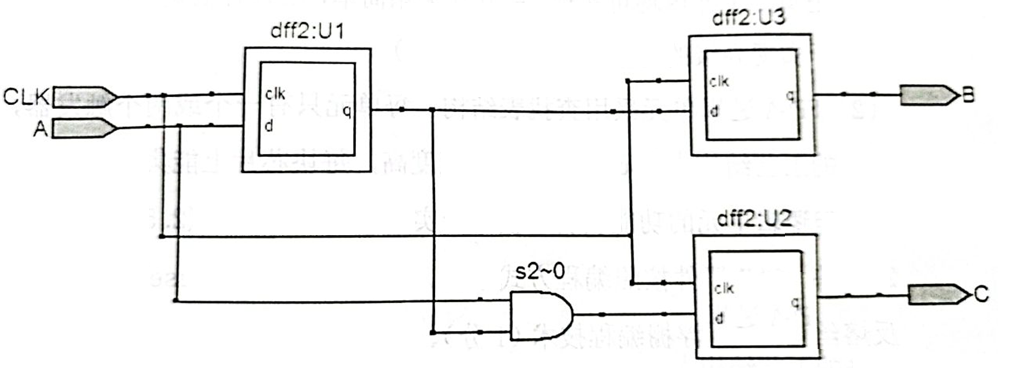
28．答案：

（1）reset在模块中起复位作用；load在模块中起数据加载作用。

（2）功能仿真波形如下：



29．答案：综合后电路：



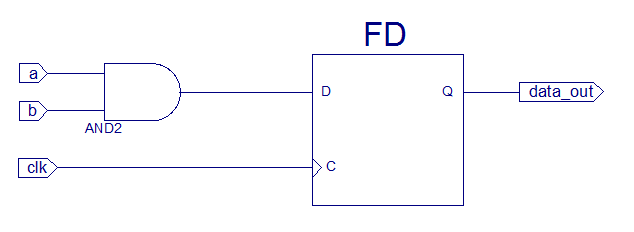
27．（1）out2 out1 ［2：0］

（2）答案：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| in1 | in2 | in3 | outl | out2 |
| 3'b001 | 3'b011 | 3'b100 | 3＇b001 | 3＇b101 |

28.

（1）reg wire data＿out

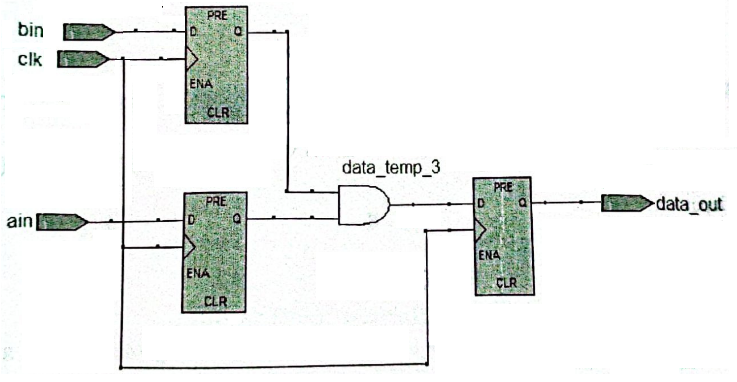
（2）

27．（1）空1 sum 空2assign

(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入信号 | | | 输出信号 | |
| a | b | cin | cout | sum |
| 3b010 | 3b011 | 1b0 | 0 | 5 |
| 3b 101 | 3b 100 | 1b1 | 1 | 2 |

28.参考答案



27．（1）out2 reg wire

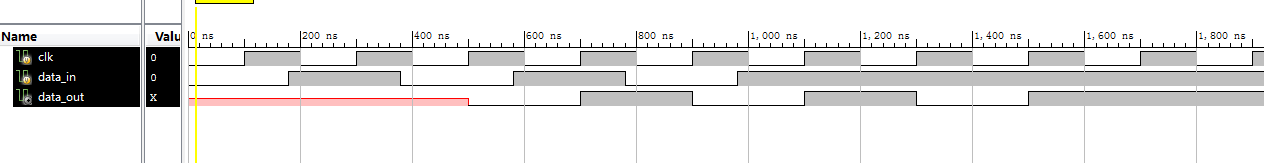
（2）答案：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| in1 | in2 | in3 | out1 | out2 |
| 2’b00 | 2’b01 | 2’b10 | 2’b00 | 2’b11 |

28.

（1）reg always begin

（2）



**六、编程题（本大题共2题，第29题9分，第30题10分，共19分）**

29、module test1(data\_in,data\_out);

input [2:0] data\_in;

output [6:0] data\_out;

reg [6:0] data\_out;

always @ (data\_in)

case(data\_in)

3'b000: data\_out =7'b1000000;

3'b001: data\_out =7'b1111001;

3'b010: data\_out =7'b0100100;

3'b011: data\_out =7'b0110000;

3'b100: data\_out =7'b0011001;

3'b101: data\_out =7′b0010010;

3'b110: data\_out =7′b0000010;

3'b111: data\_out =7′b1111000;

default : data\_out =7'b1000000;

endcase

endmodule

30.参考答案：

module test\_1(in1,in2,in3,clk,dout1,dout2 );

input in1,in2,in3;

input clk;

output dout1,dout2;

reg dout1,data\_temp;

wire dout2;

always @ (posedge clk)

begin

dout1 <= in1 & in2;

data\_temp <= in2 ^ in3;

end

assign dout2 = ~data\_temp;

endmodule

说明：程序编程结果不唯一，其他描述方式酌情给分，正确满分

29．参考答案：

module HC38(ena,din,dout);

input ena;

input [2:0] din;

output [7:0] dout;

reg [7:0] dout;

always @ (ena or din)

if(ena==1'b0)

case(din)

3'b000 : dout = 8'b01111111;

3'b001 : dout = 8'b10111111;

3'b010 : dout = 8'b11011111;

3'b011 : dout = 8'b11101111;

3'b100 : dout = 8'b11110111;

3'b101 : dout = 8'b11111011;

3'b110 : dout = 8'b11111101;

3'b111 : dout = 8'b11111110;

default : dout = 8'b11111111;

endcase

else

dout = 8'b11111111;

endmodule

30. 参考答案：

module fenpin4(clk,reset,clk\_out );

input clk,reset;

output clk\_out;

reg clk\_out;

reg [1:0] counter;

always @ (posedge clk)

if(reset)

counter <= 2'b00;

else

counter <= counter + 1'b1;

always @ (posedge clk)

if(reset)

clk\_out = 1'b0;

else

if(counter ==2'b00)

clk\_out = 1'b0;

else

if(counter ==2'b10)

clk\_out = 1'b1;

else

clk\_out <= clk\_out;

endmodule

说明：程序编程结果不唯一，其他描述方式酌情给分，正确满分

30．

module trig(xin,clk,yout);

input xin,clk;

output yout;

reg d,yout;

always @( posedge clk)

d <= yout;

always @ (posedge clk)

yout <= d | xin;

endmodule

31.

module fsm\_1(in,clk,reset,flag\_out);

input in;

input clk,reset;

output flag\_out;

parameter s0=2'b00,s1=2'b01,s2=2'b10;

reg [1:0] next\_state, cur\_state;

reg flag\_out;

always @(posedge clk)

if(reset)

cur\_state <= s0;

else

cur\_state <=next\_state;

always@(in or cur\_state)

case(cur\_state)

s0: if(in) begin flag\_out=1'b0; next\_state=s1;end

else begin flag\_out=1'b0;next\_state=s0;end

s1:if(in) begin flag\_out=1'b0;next\_state=s2; end

else begin flag\_out=1'b0;next\_state= s0; end

s2:if(in==1'b0) begin flag\_out=1'b1;next\_state=s0;end

else begin flag\_out =1'b0; next\_state= s2;end

default: begin flag\_out=1'b0;next\_state= s0; end

endcase

endmodule

说明：编程实现结果不唯一，根据实现结果酌情给分

30．答案：

case 语句实现：

module MUX(SEL,AIN,BIN,COUT);

input [1:0] AIN,BIN;

input [1:0]SEL;

output[1:0]COUT;

reg[1:0]COUT;

always@(AIN,BIN,SEL)

begin

case(SEL)

2‘b00:COUT <= AIN | BIN;

2’b01:COUT <= AIN ^ BIN;

2’b10: COUT <= AIN&BIN;

2’b11: COUT <= AIN~^BIN;

default : COUT<=2’bXX;

endcase

end

endmodule

31．参考答案：

module mooreb(clk, reset, ina, outa);

input clk,reset;

input[1:0] ina;

output [3:0] outa;

reg[1:0] c\_st, n\_st;

reg [3:0] outa;

parameter st0=2'b00,st1=2'b01,st2=2'b10,st3=2'b11;

always@(posedge clk)

if(!reset)

c\_st <= st0;

else

c\_st <= n\_st;

always@ (c\_st or ina)

case (c\_st)

st0: begin

outa<=4'b0101;

if (ina==2'b00) n\_st <= st0;

else n\_st<= st1; end

st1: begin

outa<=4'b1000;

if (ina==2'b01) n\_st<=st1;

else n\_st <= st2; end

st2: begin

outa <=4'b1100;

if(ina==2'b11) n\_st <= st0;

else n\_st <= st3; end

st3:begin

outa<=4'b1101;

if(ina==2'b11) n\_st<=st3;

else n\_st <= st0; end

default : n\_st <= st0;

endcase

endmodule

说明：编程实现结果不唯一

29．答：

module test\_b21(Y0,Y1,Y2,I7,I6,I5,I4,I3,I2,I1,I0);

input I7,I6,I5,I4,I3,I2,I1,I0;

output Y0,Y1,Y2;

reg Y0,Y1,Y2;

always @ (I7 or I6 or I5 or I4 or I3 or I2 or II or I0)

begin

case({17,16,15,14,13,12,11,10})

8'b00000001: {Y2,Y1,Y0}=3'b000;

8'b00000010: {Y2,Y1,Y0}=3'b001;

8'b00000100: {Y2,Y1,Y0}=3'b010;

8'b00001000: {Y2,Y1,Y0}=3'b011;

8'b00010000: {Y2,Y1,Y0}=3'b100;

8'b00100000: {Y2,Y1,Y0}=3'b101;

8'b01000000: {Y2,Y1,Y0}=3'b110;

8'b10000000: {Y2,Y1,Y0}=3'b111;

default:{Y2,Y1,Y0}=3'bzzz;

endcase

end

endmodule

30.

module test＿3(clk，reset，en，Carry，Cout）;

input clk,reset,en;

output Carry;

output[5:0] Cout;

reg Carry;

reg［5：0］ Cout；

always@(posedge clk or posedge reset)

if(reset)

begin

Carry<=1’b0;Cout<=6’b000000;end

else

begin

if(en)

begin

if(Cout==6’b111011)

begin Cout<=6’b000000; Carry<=1’b1;end

else

begin

Cout<=Cout+1’b1;Carry<=1’b0;

end

end

endmodule

说明：编程实现结果不唯一，根据实现结果酌情给分

29. module test7 (enable,data\_in,data\_out);

input[2:0] enable;

input [2:0] data\_in;

output[7:0] data\_out;

reg[7:0] data\_out;

always @  (enable or data\_in)

if(enable==3b100)

case(data\_in)

3b000:data\_out=8‘b11111110;

3b001:data\_out=8‘b 11111101;

3b010:data\_out= 8’b11111011;

3b011:data\_out=8’b11110111;

3b100: data\_out = 8’b11101111;

3b101: data\_out = 8’b11011111;

3b110: data\_out= 8’b10111111;

3b111:data\_out=8’b01111111;

default:data\_out=8’b11111111;

endcase

else

data\_out =8’b11111111;

endmodule

说明：设计程序答案不唯一，其他描述方式酌情给分，正确满分

30．

module shiftreg（reset，clk，data＿in，LorR，data＿out）；

input reset,clk,data\_in;

input LorR;

output[7:0] data\_out;

reg[7:0] data\_outt;

always @ (posedge clk)

if(reset)

data\_out <=8b0;

else

if(lorR)

data\_out <={data\_in,data\_out[7:1]};

else

data\_out <={data\_out[6:1],data\_in};

endmodule

说明：设计程序答案不唯一，其他描述方式酌情给分，正确满分

29.module test31(in1,in2,in3,out1,out2);

input in1,in2,in3;

output out1,out2;

reg out1;

wire out2;

always@ (in1 or in2)

out1=in1 & in2;

assign out2 = out1 | in3；

endmodule

说明：编程实现结果不唯一，根据实现情况酌情给分。

30 . module test\_ (clk,data\_in1,data\_in2,data\_in3,data\_out1,data\_out2);

input clk;

input data\_in1,data\_in2,data\_in3;

output data\_out1,data\_out2;

reg data\_out1

reg data\_temp;

always @ (posedge clk)

begin

data\_temp<=data\_in1^data\_in2;

data\_out1 <=data\_temp & data\_in3;

end

assign data\_out2=~data\_out1;

endmodule

说明：编程实现结果不唯一，根据实现情况酌情给分。