选择题

1. 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 Altera 公司生产的器件中， MAX7000 系列属 FPGA 结构。

2. 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　　C. 双向电路　　D. 三态控制电路

3. 综合是 EDA 设计流程的关键步骤，在下面对综合的描述中，（ D ）是错误的。

A.综合就是把抽象设计层次中的一种表示转化成另一种表示的过程；

B.综合就是将电路的高级语言转化成低级的，可与 FPGA / CPLD 的基本结构相映射的网表文件；

C.为实现系统的速度、面积、性能的要求，需要对综合加以约束，称为综合约束；

D.综合可理解为一种映射过程，并且这种映射关系是唯一的，即综合结果是唯一的。

4. 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 Altera 公司生产的器件中，MAX7000 系列属 FPGA 结构。

5. 以下关于状态机的描述中正确的是（ B ）

A．Moore 型状态机其输出是当前状态和所有输入的函数

B．与 Moore 型状态机相比，Mealy 型的输出变化要领先一个时钟周期

C．Mealy 型状态机其输出是当前状态的函数

D．以上都不对

6. 目前应用最广泛的硬件描述语言是（ B ）。

A. VHDL

B. Verilog HDL

C. 汇编语言

D. C 语言

7. 一模块的 I/O 端口说明： “input [7:0] a;”，则关于该端口说法正确的是（ A ）。

A. 输入端口，位宽为 8

B. 输出端口，位宽为 8

C. 输入端口，位宽为 7

D. 输出端口，位宽为 7

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(B)。

①功能仿真 ②时序仿真 ③逻辑综合 ④配置 ⑤分配管脚

A．③① B．①⑤ C．④⑤ D．④②

9. 下列标识符中， (A)是不合法的标识符。

A． 9moon B． State0 C． Not_Ack_0 D． signall

10. 下列语句中，不属于并行语句的是：(D)

A．过程语句 B． assign 语句 C．元件例化语句 D． case 语句

11. 已知 “a =1’ b1; b=3'b001;”那么 {a,b} ＝（ C ）

(A) 4'b0011 (B) 3'b001 (C) 4'b1001 (D) 3'b101

12. 在 verilog 中，下列语句哪个不是分支语句？（ D ）

(A) if-else (B) case (C) casez (D) repeat

13. 在 verilog 语言中整型数据在默认情况与（ C ）位寄存器数据在实际意义上是相同的。

(A) 8 (B) 16 (C) 32 (D) 64

14. 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 Altera 公司生产的器件中， MAX7000 系列属 FPGA 结构。

15. 请根据以下两条语句的执行，最后变量 A 中的值是 （ A ）

reg [7:0] A;

A=2'hFF;

A. 8'b0000_0011 B. 8'h03 C. 8'b1111_1111 D. 8'b11111111

16. 下列描述中采用时钟正沿触发且 reset 异步下降沿复位的代码描述是 （ C ）

A、 always @(posedge clk, negedge reset)

if(reset)

B、 always@(posedge clk, reset)

if (!reset)

C、 always @(posedge clk, negedge reset)

if(!reset)

D、 always @(negedge clk, posedge reset)

if (reset)

17. 关于过程块以及过程赋值描述中，下列正确的是（ A ）

A、在过程赋值语句中表达式左边的信号一定是寄存器类型；

B、过程块中的语句一定是可综合的；

C、在过程块中，使用过程赋值语句给 wire 赋值不会产生错误；

D、过程块中时序控制的种类有简单延迟、边沿敏感和电平敏感。

18. Verilog 语言与 C 语言的区别，不正确的描述是（ C ）

A 、 Verilog 语言可实现并行计算， C 语言只是串行计算；

B、 Verilog 语言可以描述电路结构， C 语言仅仅描述算法；

C、 Verilog 语言源于 C 语言，包括它的逻辑和延迟；

D、 Verilog 语言可以编写测试向量进行仿真和测试。

19. 11. 下列模块的例化正确的是（ C ）。

A. Mydesign design(sin(sin), sout(sout));

B. Mydesign design(.sin(sin), .sout(sout));

C. Mydesign design(.sin(sin), .sout(sout););

D. Mydesign design(.sin(sin); .sout(sout));

20. 下列关于 Verilog HDL 语言中模块的例化说法错误的是（ B ）。

A. 在引用模块时， 有些信号要被输入到引用模块中， 有些信号要从引用模块中输出

B. 在引用模块时，必须严格按照模块定义的端口顺序来连接

C. 在引用模块时可以用“ .”符号，表明原模块是定义时规定的端口名，用端口名和被引用模块的端口相对应，提高程序的可读性和可移植性

D. 在语句“ Mydesign design(.port1(port1), .port2 (port2)); ”中，被引用的模块为 Mydesign 模块

21. 下列 Verilog HDL 语言中寄存器类型数据定义与注释矛盾的是（ D ）。
A. reg [3:0] sat //sat 为 4 位寄存器
B. reg cnt //cnt 为 1 位寄存器
C. reg [0:3] mymem [0:63] //mymem 为 64 个 4 位寄存器的数组
D. reg [1:5] dig //dig 为 4 位寄存器

22. 下列关于非阻塞赋值运算方式（如 b<=a;）说法错误的是（ B ）。
A. 块结束后才完成赋值操作
B. b 的值立刻改变
C. 在编写可综合模块时是一种比较常用的赋值方式
D. 非阻塞赋值符“ <=”与小于等于符“ <=”意义完全不同，前者用于赋值操作，后者是关系运算符，用于比较大小。

23. 下列关于阻塞赋值运算方式（如 b=a;）说法错误的是（ A ）。
A. 赋值语句执行完后，块才结束
B. b 的值在赋值语句执行完后立刻就改变的
C. 在沿触发的 always 块中使用时，综合后可能会产生意想不到的结果
D. 在“always”模块中的 reg 型信号都采用此赋值方式

24. 在下列 Verilog HDL 运算符中，属于三目运算符的是（ C ）。
A. &&
B. ！ ==
C. ？：
D. ===

25. 当 a <0 时， s 的值是（ C ）。
assign s= (a >=2) ? 1 : (a < 0) ? 2: 0;
A. 0
B. 1
C. 2
D. 其他

26. 在 Verilog HDL 语言中的位拼接运算符是（ A ）。
A. { }　　B. < >　　C. ( )　　D. ' '

27. 下面语句中，信号 a 会被综合成（ B ）。
```
reg [5:0] a;
always @(posedge clk)
if (ss>10)
a <= 20;
else if (ss > 15)    a <= 30;
```
A. 寄存器
B. 触发器

C. 连线资源
D. 其他

28. 下列程序段中无锁存器的是（ C ）。

A. always @ (al or d)
begin
if(al) q<= d;
end

B. always @ (al or d)
begin
if(al) q<=d;if(!al) q<=!d;
end

C. always @ (al or d)
begin
if(al)
q<=d;
else
q<=0;
end

D. always @ (sel[1:0]
or a or b)
case(sel[1:0])
2' b00: q<=a;
2' b11; q<=b;
Endcase

29. 程序段如下 ：

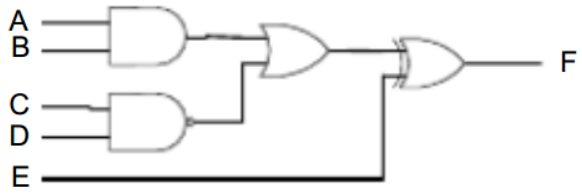
```
begin:
reg[7:0] tem;
count = 0;
tem = rega;
while(tem)
begin
if(tem[0]) count = count +1;
tem = tem >>1;
end
end
```

如果 rega 的值为 8 ' b10101011，则程序结束后， count 的值是（ ）。
A. 4
B. 5
C. 6
D. 7

30. 多路选择器简称多路器，它的输入输出端口情况是（ ）。
A. 多输入，多输出
B. 多输入，单输出
C. 单输入，多输出
D. 单输入，单输出

填空题

1. 用 EDA 技术进行电子系统设计的目标是最终完成 ASIC 的设计与实现。
2. 可编程器件分为 FPGA 和 CPLD 。
3. 随着 EDA 技术的不断完善与成熟， 自顶向下的设计方法更多的被应用于 Verilog HDL
4. 设计当中。
5. 目前国际上较大的 PLD 器件制造公司有 Altera 和 Xilinx 公司。
6. 完整的条件语句将产生 组合 电路，不完整的条件语句将产生时序电路。
7. 阻塞性赋值符号为 =， 非阻塞性赋值符号为<= 。
8. 有限状态机分为 Moore 和 Mealy 两种类型。
9. EDA 缩写的含义为 电子设计自动化 (Electronic Design Automation)
10. 状态机常用状态编码有 二进制、格雷码 和独热码 。
11. Verilog HDL 中任务可以调用 其他任务 和函数。
12. 系统函数和任务函数的首字符标志为 $ ，预编译指令首字符标志为 # 。
13. 可编程逻辑器件的优化过程主要是对 速度 和资源 的处理过程。
14. 大型数字逻辑电路设计采用的 IP 核有 软 IP、 固 IP 和硬 IP。
15. IEEE 标准的硬件描述语言是 verilog HDL 和 VHDL 。
16. Verilog 语言规定的两种主要的数据类型分别是 wire(或 net) 和 reg 。程序模块中输入，输出信号的缺省类型为 wire(或 net) 。
17. Verilog 语言规定了逻辑电路中信号的 4 种状态，分别是 0， 1， X 和 Z。其中 0 表示低电平状态， 1 表示高电平状态， X 表示 不定态（或未知状态） ， Z 表示 高阻态 。
18. 块语句有两种， 一种是 begin-end 语句， 通常用来标志 顺序 执行的语句； 一种是 fork-join 语句，通常用来标志 并行 执行的语句。
19. 写出表达式以实现对应电路的逻辑功能_____



assign F= E ^ ((A&B) | (!(C&D)))

20. 下面两段代码中信号 in， q1， q2 和 q3 的初值分别为 0， 1， 2 和 3，那么经过 1 个时钟周期后，左侧程序中 q3 的值变成 0 ，右侧程序中 q3 的值变成 2 。

```
always @(posedge clk)
begin
    q1 = in;
    q2 = q1;
    q3 = q2;
end
```

```
always @(posedge clk)
begin
    q1 <= in;
    q2 <= q1;
    q3 <= q2;
end
```

名词解释

1. EDA
2. ASIC 专用集成电路
3. RTL 寄存器传输级
4. FPGA 现场可编程门阵列
5. SOPC 可编程片上系统
6. CPLD 复杂可编程逻辑器件
7. LPM 参数可定制宏模块库
8. EDA 电子设计自动化
9. IEEE 电子电气工程师协会
10. IP 知识产权核
11. ISP 在系统可编程
12. LUT：查找表
13. HDL：硬件描述语言
14. RTL：寄存器传输逻辑

简答题

1. 简要说明仿真时阻塞赋值与非阻塞赋值的区别。

非阻塞（non-blocking）赋值方式（b<= a）：

b 的值被赋成新值 a 的操作，并不是立刻完成的，而是在块结束时才完成；

块内的多条赋值语句在块结束时同时赋值；

硬件有对应的电路。

阻塞（blocking）赋值方式（b = a）：

b 的值立刻被赋成新值 a；

完成该赋值语句后才能执行下一句的操作；

硬件没有对应的电路，因而综合结果未知。

2. 简述有限状态机 FSM 分为哪两类？有何区别？有限状态机的状态编码风格主要有哪三种？ FSM 的三段式描述风格中，三段分别描述什么？

根据内部结构不同可分为摩尔型 Moore 状态机和米里型 Mearly 状态机两种。

摩尔型状态机的输出只由当前状态决定，而次态由输入和现态共同决定；米里型状态机的输出由输入和现态共同决定，而次态也由输入和现态决定。状态编码主要有三种：连续二进制编码、格雷码和独热码。

3. Verilog HDL 语言进行电路设计方法有哪几种
   ①自上而下的设计方法（ Top-Down ）
   ②自下而上的设计方法（ Bottom-Up ）
   ③综合设计的方法

4. 简述 moore 状态机和 mealy 状态机的区别

答：从输出的时序上看，Mealy 机的输出是当前状态和所有输入信号的函数，它的输出是在输入变化后立即发生的。Moore 机的输出则仅为当前状态的函数，在输入发生变化时还必须等待时钟的到来，时钟使状态发生变化时才导致输出的变化。

Moore 型状态机：次态=f(现状，输入)，输出=f（现状）；

Mealy 型状态机：次态=f(现状，输入)，输出=f（现状，输入）；

5. 简述 FPGA 内部主要结构及其功能

答：FPGA 由 6 部分组成，分别为可编程输入/输出单元、基本可编程逻辑单元、嵌入式块 RAM、丰富的布线资源、底层嵌入功能单元和内嵌专用硬核等。

大多数 FPGA 的 I/O 单元被设计为可编程模式，即通过软件的灵活配置，可适应不同的电器标准与 I/O 物理特性;可以调整匹配阻抗特性，上下拉电阻;可以调整输出驱动电流的大小等。查找表完成纯组合逻辑功能。查找表完成纯组合逻辑功能。嵌入式块 RAM 可以配置为单端口 RAM、双端口 RAM、伪双端口 RAM、CAM、FIFO 等存储结构。布线资源连通 FPGA 内部所有单元。

6. 简述基于数字系统设计流程包括哪些步骤 ？

包括五个步骤：

⑴、 设计输入： 将设计的结构和功能通过原理图或硬件描述语言进行设计或编程， 进行
语法或逻辑检查，通过表示输入完成，否则反复检查直到无任何错误。

⑵、逻辑综合：将较高层的设计描述自动转化为较低层次描述的过程，包括行为综合，
逻辑综合和版图综合或结构综合，最后生成电路逻辑网表的过程。

⑶、布局布线：将综合生成的电路网表映射到具体的目标器件中，并产生最终可下载文件的过程。

⑷、仿真：就是按照逻辑功能的算法和仿真库对设计进行模拟，以验证设计并排除错误
的过程，包括功能仿真和时序仿真。

⑸、编程配置：将适配后生成的编程文件装入到 PLD 器件的过程，根据不同器件实现编程或配置。

7. 简述 CPLD 和 FPGA 主要区别。

答：CPLD：主要逻辑阵列块、宏单元、扩展乘积项和可编程连线阵列构成。程序掉电不丢失。

FPGA：FPGA 中多使用 4 输入的 LUT，所以每一个 LUT 可以看成一个有 4 位地址线的 16x1 的 RAM，程序掉电丢失，需重新加载。

8. EDA 设计的基本设计过程。

答：图形输入 HDL;文本输入； 综合;适配;时序仿真与功能仿真;编程下载;硬件测试 .

9. 解释什么是功能仿真？综合

答：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。仿真过程可不涉及任何具体器件的硬件特性。

10. 简述 Verilog 描述的状态机的一般结构包含哪几个部分。

答：说明部分、主控时序进程、主控组合进程、辅助进程。

（1） 说明部分：使用 parameter 定义系统状态: 用于描述状态 。

（2） 主控时序进程：负责状态机运转和在时钟驱动下负责状态转换的进程。

（3） 主控组合进程：根据外部输入的控制信号和当前状态的状态值确定下一状态（next_state）的取向。

（4） 辅助进程：用于配合状态机工作的组合进程或时序进程

11. 简述 Verilog HDL 编程语言中函数与任务运用有什么特点？

函数和任务都能独立完成相应电路功能，通过在同一模块中的调用实现相应逻辑电路功能。但它们又有以下不同：

⑴、函数中不能包含时序控制语句，对函数的调用，必须在同一仿真时刻返回。而任务可以包含时序控制语句，任务的返回时间和调用时间可以不同。

⑵、在函数中不能调用任务，而任务中可以调用其它任务和函数。但在函数中可以调用其它函数或函数自身。

⑶、函数必须包含至少一个端口，且在函数中只能定义 input 端口。任务可以包含 0 个或任何多个端口，且可以定义 input、 output 和 inout 端口。

⑷、函数必须返回一个值，而任务不能返回值，只能通过 output 或 inout 端口来传递执行结果。

12. 简述 FPGA 与 CPLD 两种器件应用特点。

CPLD 与 FPGA 都是通用可编程逻辑器件， 均可在 EDA 仿真平台上进行数字逻辑电路设计，它们不同体现在以下几方面：

⑴ FPGA 集成度和复杂度高于 CPLD ， 所以 FPGA 可实现复杂逻辑电路设计，而 CPLD 适合简单和低成本的逻辑电路设计。

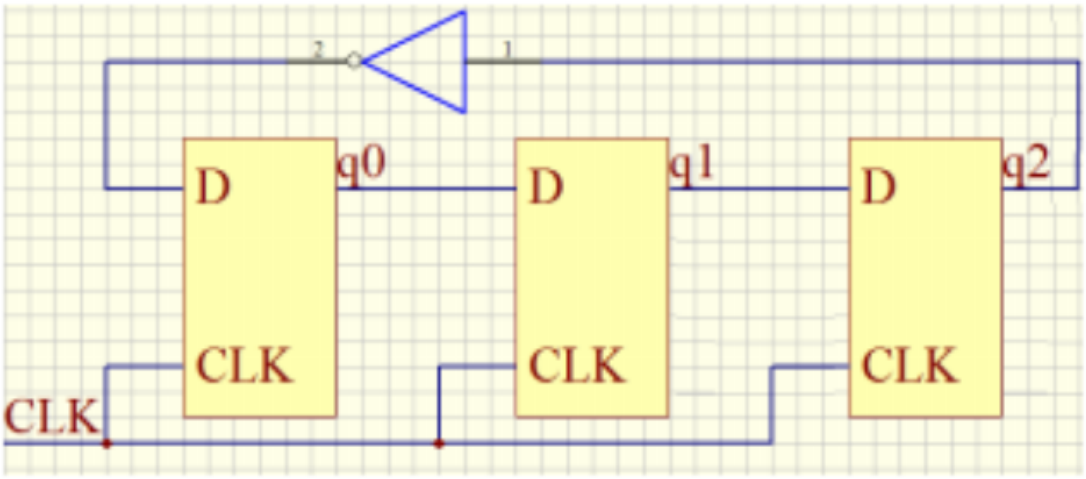
⑵、 FPGA 内主要由 LUT 和寄存器组成，倾向实现复杂时序逻辑电路设计，而 CPLD 内主要由乘积项逻辑组成，倾向实现组合逻辑电路设计。

⑶、 FPGA 工艺多为 SRAM 、 flash 等工艺，掉电后内信息消失，所以该类型需外配存储器，而 CPLD 工艺多为 EEPROM 等工艺，掉电后信息不消失，所

以不用外配存储器。
⑷、 FPGA 相对 CPLD 成本高，但都可以在内都镶嵌硬核和软核，实现片上系统功能。

13. 画出下面程序综合出来的电路图。

```
always@(posedge clk)
begin
q0<=~q2;
q1<= q0;
q2<= q1;
end
```



程序填空

1. 下面是 case 语句编写的 3-8 译码器电路，将横线上的语句补上，使程序形成完整功能

```
module decoder38 (sel,csout);
_____①_____
_____②_____
reg[7:0] csout;

always@(____③____)
    begin
        case(____④____)
            3'b000:csout=8'b11111110;
            3'b001:csout=8'b11111101;
            3'b010:csout=8'b11111011;
            3'b011:csout=8'b11110111;
            3'b100:csout=8'b11101111;
```

```
                    3'b101:csout=8'b11011111;
                    3'b110:csout=8'b10111111;
                    3'b111:csout=8'b01111111;
                   default:csout=8'b11111110;
                endcase
        end
endmodule
```

①input[2:0] sel; ② output[7:0] csout;③ sel,csout ④sel

2. 下面是通过 case 语句实现四选一电路部分程序，将横线上的语句补上，使程序形成完整功能

```
module mux41a(a,b,c,d,s1,s0,y);
    input a,b,c,d;
    _____①_____
    output y;
    _____②_____
    always@ (a,b,c,d,s1,s0)
    begin:MUX41
        case(____③____)
            2'b00:y=a;
            2'b01:y=b;
            2'b10:y=c;
            2'b11:y=d;
            ___④___:y=a;
        endcase
    end
endmodule
```

① input s1,s0; ② reg y; ③ {s1,s0} ④ default

3. 下面是通过循环语句实现程序，用以统计一个 8 位二进制数中含 1 的数量，将横线上的语句补上，使程序形成完整功能。

```
        module num_1_e3_16(x,num);
            _____①_____
            output [3:0]num;
            reg [3:0]num;
            _____②_____
```

```
        always@(    ③    )
        begin
            num=0;
            for(i=0;i<=7;i=i+1)
                    ④
        end
endmodule
```

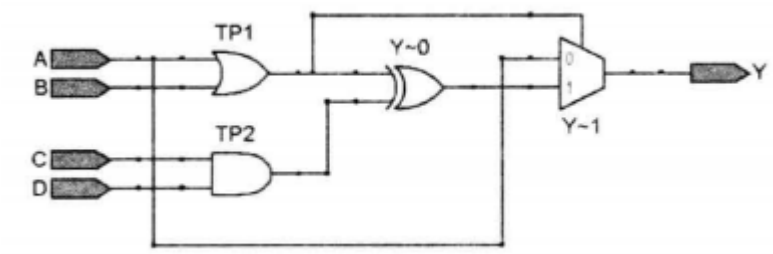
① input [7:0]x; ② integer i; ③ x ④ if(x[i]) num=num+1;

4. 下面是程序功能是一个具有同步置 1，异步清零 0 的 D 触发器。端口说明 RST：异步清零 CLK：时钟输入,SET：同步置 1,EN：同步使能， D：数据输入,Q：数据输出

```
module e5_3(RST,CLK,SET,EN,D,Q);
    input RST,CLK,SET,EN,D;
        ①
    reg Q;
    always@(          ②          )
        begin
            if(    ③    ) Q<=0;
        else if(EN)
            begin
            if(    ④    )      Q<=1;
             else      Q<=D;
            end
        end
endmodule
```

① output Q; ② posedge CLK or negedge RST ③ !RST ④ SET

对应 RTL 图完成 Verilog 程序。




```
module e5_6(set,D,clk,en,reset,Q);
    input set,D,clk,en,reset;
```

```
      ①
reg Q;
always@(          ②          )
    begin
        if(reset)   ③
        else if(set)   ④
        else if(en)   ⑤
        end
endmodule
```

① output Q; ② posedge clk or posedge reset or posedge set ③ Q<=0; ④ Q<=1; ⑤ Q<=D;

编程题

1. 试用 Verilog HDL 描述一个带进位输入、输出的 8 位全加器。
端口： A、 B 为加数， CI 为进位输入， S 为和， CO 为进位输出

2. 设计一个 4 位 4 输入最大数值检测电路，其中，a、b、c 和 d 为 4 位二进制数，输出为 max。

```
module max_in4(a,b,c,d,max);
  input   [3:0]a,b,c,d;
  output [3:0]max;

  wire [3:0]max1=(a>=b)?a:b;
  wire [3:0]max2=(c>=d)?c:d;
  assign max=(max1>max2)?max1:max2;
endmodule
```

3. 编写一个带异步清零、异步置位的 D 触发器。
端口： CLK 为时钟， D 为输入， CLK 为清零输入端， SET 为置位输入端； Q 输出端。

```
module DFF1(q,qn,d,clk,set,reset);
  output q,qn;
  input  d,clk,set,reset;
  reg    q,qn;
  always @(posedge clk or negedge set or negedge reset)
    begin
      if(!reset)     begin
                     q=0;qn=1;
                     end
      else if(!set)  begin
                     q=1;qn=0;
                     end
      else           begin
                     q=d;qn=~d;
                     end
    end
endmodule
```

4. 设计一个带有异步复位控制端和时钟使能控制端的 10 进制计数器。
端口设定如下：
输入端口： CLK ：时钟， RST：复位端， EN：时钟使能端， LOAD ：置位控制端，DIN ：置位数据端；输出端口： COUT：进位输出端， DOUT ：计数输出端

```
 mdule CNT10 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
input CLK    ;
input EN    ;
input RST   ;
input LOAD   ;
input [3:0] DATA   ;
output [3:0] DOUT   ;
output COUT   ;
reg [3:0] Q1 ;
reg COUT ;
assign DOUT = Q1;
always @(posedge CLK or negedge RST) begin
          if (!RST)     Q1 <= 0;
           else  if (EN)
            begin
              if (!LOAD)      Q1 <= DATA;
              else if (Q1<9)   Q1 <= Q1+1;
                   else       Q1 <= 4'b0000;
            end
    end
always @(Q1)
  if (Q1==4'h9)   COUT = 1'b1;
  else            COUT = 1'b0;
endmodule
```

5. 编写一个 4 位加法计数器 VHDL 程序的进程（不必写整个结构框架），要求复
位信号 reset **低电平时计数器清零，变高后，在上升沿开始工作；输入时钟信号为 clk，输出为**

6. 填写完成一个 8-3 线编码器的真值表（5 分），并写出其 verilog 程序

8 -3 线编码器真值表

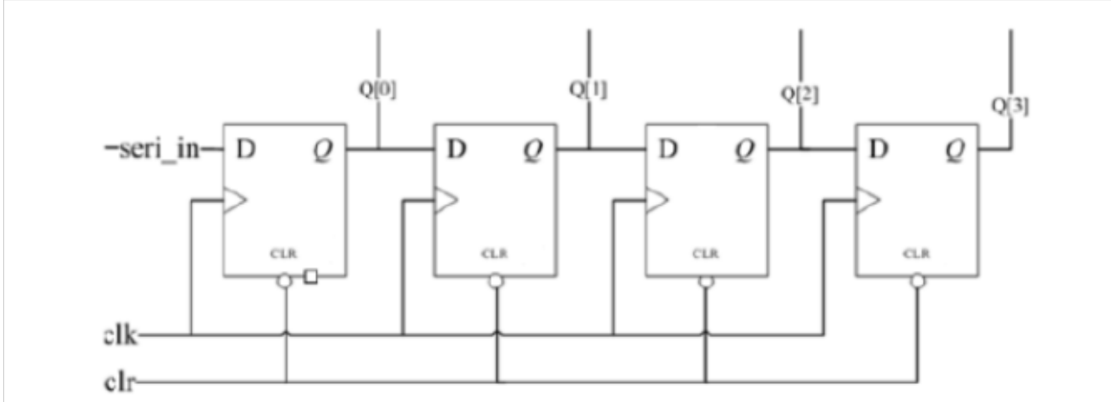
en	b	y0y1y2
1	00000000	000
1	00000010	001
1	00000100	010
1	00001000	011
1	00010000	100
1	00100000	101
1	01000000	110
1	10000000	111
0	xxxxxxxx	高阻态

7. 用 IF 语句编写一个四选一电路，要求输入 d0～d3, s 为选择端，输出 y。

8. 现有输入信号是一个占空比位 50%的方波，用 VHDL 设计一个时钟的 5 分频电路，输出为占空比位 20%，写出 VHDL 代码。

9. 现有输入信号是一个占空比位 50%的方波，用 VHDL 设计一个时钟的 5 分频电路，输出为占空比位 50%，写出 VHDL 代码。

10. 试用 verilog 语言描述：图示为一个 4 位移位寄存器，是由四个 D 触发器
（分别设为 U1，U2， U3， U4）构成的。其中 seri_in 是这个移位寄存器的串行输入； clk 为移位时脉冲输入；clr 为清零控制信号输入； Q[1]~Q[3] 则为移位寄存器的并行输出。



11. 设计一个带复位端且对输入时钟 clk 进行二分频模块，并画出仿真波形。

```
module m2(out,clk,reset);
input reset,clk;
output out;
reg out;
always @(negedge clk)
begin
if(reset)
out<=0;
else
out<=~out;
end
endmodule
```

12. 设计一带异步复位端、异步置数段（低电平有效）的四位加法计数器，时钟 clk 上升沿有效），复位信号 clr，置数信号 load、输入数据 data、输出 qout 。并画出仿真波形。

```
module adder_4(qout,clr,clk,load,data);
output[3:0] qout;
input[3:0] data;
input load,clr,clk;
reg[3:0] qout;
always @(posedge clk or negedge load or negedge clk)
begin
if(!load)
qout<=data;
else if(!clr)
qout<=0;
else qout<=qout+1;
end
endmodule
```

13. 试设计一个 3/8 译码器，规定模块定义为 module Decoder(Out,In,En)，其中 Out 为译码器输出， In 为译码器输入， En 为译码使能输入。要求：写出 3/8 译码器 Verilog HDL 设计程序并注释；

```
module decoder(Out,In,En); (2 分)
output [7:0] out;
input [2:0] in;
input en; //IO 定义 (3 分)
reg [7:0] out;
```

```
always @ (In or En)
begin
if(En == 0) //若 En 为低电平， 3 输出无效电平 （2 分）
Out = 8 ’ b0;
else // 若 En 为高电平， 3/8 译码 （3 分）
case(in)
3 ’ b000: Out = 8 ’ b00000001; //0
3 ’ b001: Out = 8 ’ b00000010; //1
3 ’ b010: Out = 8 ’ b00000100; //2
3 ’ b011: Out = 8 ’ b00001000; //3
3 ’ b100: Out = 8 ’ b00010000; //4
3 ’ b101: Out = 8 ’ b00100000; //5
3 ’ b110: Out = 8 ’ b01000000; //6
3 ’ b111: Out = 8 ’ b10000000; //7
endcase
end
```

设计一个