

Verilog 复习题

一、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 CPLD 和 FPGA。
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 设计当中。
4. 目前国际上较大的 PLD 器件制造公司有 Altera 和 Xilinx 公司。
5. 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。
6. 阻塞性赋值符号为 =，非阻塞性赋值符号为 <=。
7. 有限状态机分为 Moore 和 Mealy 两种类型。
8. EDA 缩写的含义为电子设计自动化(Electronic Design Automation)
9. 状态机常用状态编码有二进制、格雷码和独热码。
10. Verilog HDL 中任务可以调用其他任务和函数。
11. 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 #。
12. 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。
13. 大型数字逻辑电路设计采用的 IP 核有软 IP、固 IP和硬 IP。

二、选择题

1. 已知 “a=1b' 1; b=3b'001;” 那么{a,b}= (C)
(A) 4b'0011 (B) 3b'001 (C) 4b'1001 (D) 3b'101
2. 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
3. Verilog HDL 语言进行电路设计方法有哪几种 (8 分)
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. 在 verilog 语言中，a=4b'1011，那么 &a= (D)
(A) 4b'1011 (B) 4b'1111 (C) 1b'1 (D) 1b'0
5. 在 verilog 语言中整型数据与 (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
6. 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。
7. 子系统设计优化，主要考虑提高资源利用率减少功耗（即面积优化），以及提高运行速度（即速度优化）；指出下列哪些方法是面积优化 B。
①流水线设计
②资源共享
③逻辑优化
④串行化
⑤寄存器配平

⑥关键 路径法

A. ①③⑤ B. ②③④ C. ②⑤⑥ D. ①④⑥

8、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

9、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

10、P,Q,R 都是 4bit 的输入矢量，下面哪一种表达形式是正确的 5)

1)input P[3:0],Q,R;

2)input P,Q,R[3:0];

3)input P[3:0],Q[3:0],R[3:0];

4)input [3:0] P,[3:0]Q,[0:3]R;

5)input [3:0] P,Q,R;

11、请根据以下两条语句的执行，最后变量 A 中的值是_____①_____。

reg [7:0] A;

A=2'hFF;

① 8'b0000_0011 ② 8'h03 ③ 8'b1111_1111 ④ 8'b11111111

12. 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

三、EDA 名词解释（10 分）

ASIC	专用集成电路	RTL	寄存器传输级
FPGA	现场可编程门阵列	SOPC	可编程片上系统
CPLD	复杂可编程逻辑器件	LPM	参数可定制宏模块库
EDA	电子设计自动化	IEEE	电子电气工程师协会
IP	知识产权核	ISP	在线系统可编程

三、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。

阻塞赋值是在该语句结束是立即完成赋值操作；非阻塞赋值是在整个过程块结束是才完成赋值操作。

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？

根据内部结构不同可分为摩尔型状态机和米里型状态机两种。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、简述基于数字系统设计流程包括哪些步骤？

包括五个步骤：

(1)、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。

(2)、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。

(3)、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。

(4)、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。

(5)、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。

4、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：

(1)、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。

(2)、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。

(3)、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。

(4)、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。

5、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：

(1)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。

(2)、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。

(3)、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。

(4)、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。

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

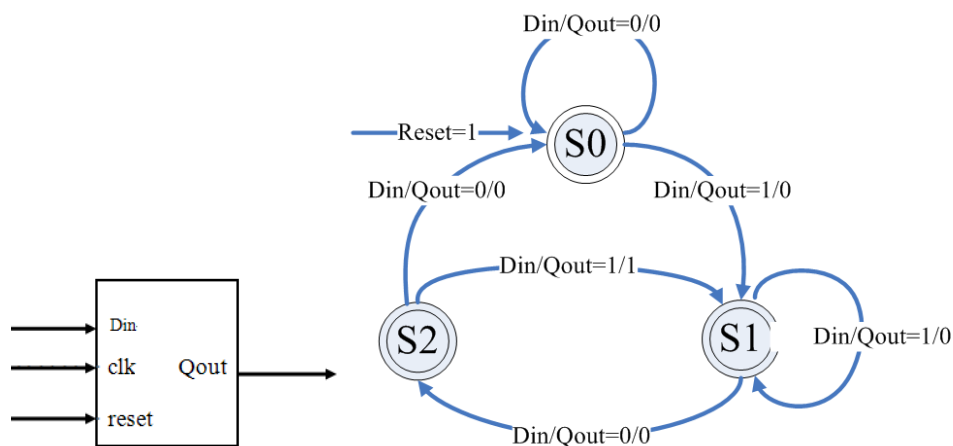
的三段式描述风格中，三段分别描述什么？（本题 6 分）

答：Mearly 型，Moore 型；前者与输入与当前状态有关，而后者只和当前状态有关；Binary，Gray，

One-Hot 编码；分别为状态保存，状态切换，输出；

四、计算题

1、 利用有限状态机，以格雷码编译方式设计一个从输出信号序列中检测出 101 信号的电路图，其方块图、状态图和状态表如图表示。



目前状态CS	下一状态NS和输出Qout	
	Din=0	Din=1
S0=00	S0, 0	S1, 0
S1=01	S2, 0	S1, 0
S2=11	S0, 0	S1, 1

```
module melay(clk,Din,reset,Qout);
input clk,reset;
input Din;
output Qout;
reg Qout;
parameter[1:0] S0=2'b00,S1=2'b01,S2=2'b11;
reg[1:0] CS;
reg[1:0] NS;
```

```
always @(posedge clk or posedge reset)
begin
    if(reset==1'b01)
        CS=S0;
    else CS=NS;
end
```

```
always @(CS or Din)
begin
    case(CS)
        S0:beign
            if(Din==1'b0)
                begin
                    NS=S0;
                    Qout=1'b0;
                end
            else
```

```

        begin
            NS=S1;
            Qout=1'b0;
        end
    end
S1:begin
    if(Din==1'b0)
        begin
            NS=S2;
            Qout=1'b0;
        end
    else
        begin
            NS=S1;
            Qout=1'b0;
        end
    end
S2:beign
    if(Din==1'b0)
        begin
            NS=S0;
            Qout=1'b0;
        end
    else
        begin
            NS=S1;
            Qout=1'b0;
        end
    end
end
endcase
end
endmodule

```

4、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 mux4_to_1 (out, i0, i1, i2, i3, s1, s0);
output out;
input i0, i1, i2, i3;
input s1, s0;
reg out;

```

```

always @(s1 or s0 or i0 or i1 or i2 or i3)
case ( _____ ) //Switch based on concatenation of control signals
    _____
    _____
    _____
    _____

default: $display("Invalid control signals");
endcase
endmodule

```

```

case({s1,s0})
2'b00:out=i0;
2'b01:out=i1;
2'b10:out=i2;
2'b11:out=i3;

```

4、根据图 3 给定的两个 2 位全加器信号关系及实现的 4 位全加器功能部分程序，在下列部分程序中的横线上填入必要语句，实现 4 位全加器的完整功能。

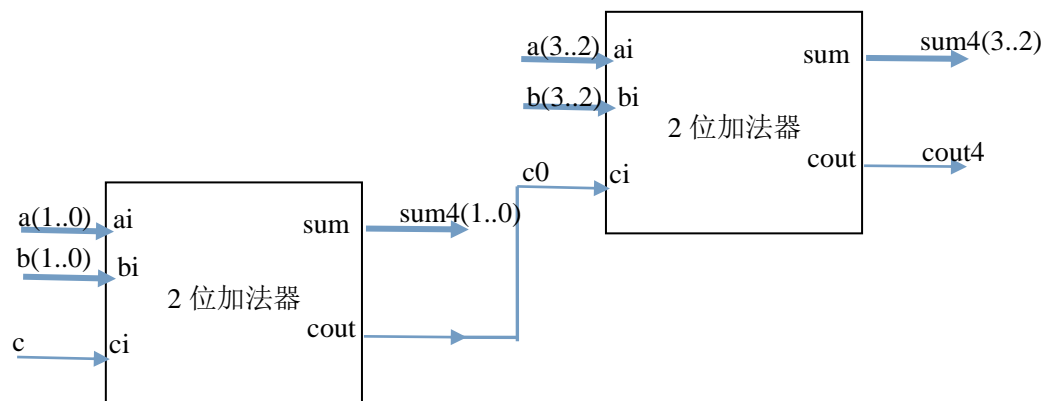


图 3

```

//底层 4 位全加器程序
module add2(ai,bi,ci,sum,cout);
input [1:0]ai,bi;input ci;
output [1:0]sum; reg [1:0]sum;
output cout;reg cout;
always @(ai,bi,ci)
    {cout,sum}=ai+bi+ci;
endmodule

```

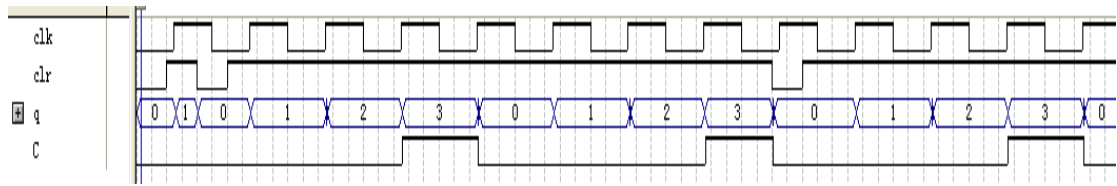
```

//顶层 8 位全加器程序
module fadd4(a,b,c,sum4,cout4);
input [3:0]a,b;input c;
output [3:0] sum4output cout4;
wire c0;
add4 U1( a[1:0],b[1:0],c,c0,sum4[1:0]);
add4 U2( a[3:0],b[3:0],c0,cout4,sum4[3:0]);
endmodule

```

5、根据下列给定的仿真输入输出波形图 2，说明完成此功能的电路是什么功能电路？并写出对应的 Verilog HDL 描述程序（图中 clk,clr 为输入，q,c 为输出）。

4 进制加法计数器



```

module counter(clk,clr,q,c)
input clk,clr;
output ret[1:0] q;
output c;
always@(posedge clk or negedge clr)
begin
    if(~clr)  q<=2'h0;
    else
    begin
        if(2'h3==q) q<=2'h0;
        else q<=q+2'h1;
    end
end
assign c=(2'h3==q)
endmodule

```

6、采用结构描述方法设计一个二进制数字半加器，输入数据 ai 与 bi，并将和输出到 so，进位输出到 co，给出详细设计过程。

输入		输出	
ai	bi	so	co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$so = aibi + \overline{aibi} = ai \oplus bi, co = aibi$$

由输入输出逻辑表达式，采用与门 and 和异或门 xor 进行结构描述的程序如下：（6 分）

```

module hadd (ai,bi,so,co);
input ai,bi;
output so,co;
xor(so,ai,bi);
and(co,ai,bi);
endmodule

```

6、采用结构描述方法设计一个二进制数字比较器，比较输入数据 a 与 b 的大小，并分别输

出到 x, y 和 z, 给出详细设计过程。

$$x = \overline{ab} + ab, y = \overline{a}b, z = a\overline{b}$$

```
not(not_a,a);
not(not_b,b);
and(ab,a,b);
and(not_ab,not_a,not_b);
or(x,ab,not_ab);
and(y,not_a,b);
and(z,a,not_b);
```

7、采用结构描述方法设计一个 3 人竞选数字电路, 输入数据[2:0]x, 要求 2 人以上为 1 表示通过, 且输出为 y 为 1, 否则输出相反, 给出详细设计过程。

```
module three1(x,y);

input [2:0] x; output y;

y=a&b+a&c+b&c=ab+ac+bc;

wire a,b,c;

and(a,x[0],x[1]); and(b,x[1],x[2]); and(c,x[1],x[0]);

or(y,a,b,c);

endmodule
```

五、程序注解 (20 分,每空 1 分)

<pre>module AAA (a,b); output a input [6:0] b reg[2:0] sum; integer i; reg a always @(b) begin sum = 0; for(i = 0;i<=6;i = i+1) if(b[i]) sum = sum+1; if(sum[2]) a = 1; else a = 0; end endmodule</pre>	<p>定义模块名为 AAA, 端口为 a, b</p> <p>定义 a 为输出端口</p> <p>定义 b 为输出端口, b 为 7 位二进制数</p> <p>sum 为 reg 型变量, 用于统计赞成的人数</p> <p>定义整型变量 i 为循环控制变量</p> <p>定义 a 为寄存器变量</p> <p>过程语句, 敏感变量为 b</p> <p>语句块</p> <p>sum 初值为 0</p> <p>for 语句, 统计 b 为 1 的个数</p> <p>条件语句</p> <p>只要有人投赞成票, 则 sum 加 1</p> <p>若超过 4 人赞成, 则表决通过</p> <p>若不到 4 人, 则不通过</p>
--	--

本程序的逻辑功能是: 7 人投票表决器

六、VerilogHDL 编程题（1、2 小题 10 分，3 小题 20 分） 要求：写清分析设计步骤和注释。

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，CIN 为进位输入，S 为和，COUT 为进位输出

```
module add4v(a,b,ci,s,co);
    input[3:0] a;
    input[3:0] b;
    input ci;
    output[3:0] s;
    output co;
    wire[3:0] carry;
    function fa_s(input a,input b,input ci); fa_s = a ^ b ^ ci; endfunction
    function fa_c(input a,input b,input ci); fa_c = a & b | a & ci | b & ci; endfunction
    assign s[0] = fa_s(a[0],b[0],ci); assign carry[0] = fa_c(a[0],b[0],ci);
    assign s[1] = fa_s(a[1],b[1],carry[0]); assign carry[1] = fa_c(a[1],b[1],carry[0]);
    assign s[2] = fa_s(a[2],b[2],carry[1]); assign carry[2] = fa_c(a[2],b[2],carry[1]);
    assign s[3] = fa_s(a[3],b[3],carry[2]); assign co = fa_c(a[3],b[3],carry[2]); endmodule
```

2. 编写一个带异步清零、异步置位的 D 触发器。

```
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
```

// 带同步清 0/同步置 1(低电平有效)的 D 触发器.

module dff_syn(q,qn,d,clk,set,reset); //定义模块为 dff_syn, 端口为 q,qn,d,clk,set,reset

input d,clk,set,reset; output q,qn

reg q,qn; //定义端口 d,clk,set,reset 为输入端口,q,qn 为输出端口

always @(posedge clk) //对 clk 信号上升沿有效

begin

if(~reset) begin q<=1'b0;qn<=1'b1;end //同步清零,低电平有效

else if(~set) begin q<=1'b1;qn<=1'b0;end //同步置位, 低电平有效

```

        else begin q<=d; qn<=~d; end //q 输出为 d, qn 输出为非 d;
    end
endmodule //模块结束

```

3.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。

```

module CNT10 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
input CLK ,EN, RST, 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

```