

# 硬件描述语言及 FPGA 设计

## 平日作业

姓名: \_\_\_\_\_

学号: \_\_\_\_\_

专业: \_\_\_\_\_

一、单选题:

1. SignaltapII 的采样时钟的频率必须: ( )。
  - A. 小于被测信号频率;
  - B. 大于被测信号频率;
  - C. SignaltapII 和 Modelsim 一样, 都是软仿真工具;
  - D. 以上都不对。
2. 片上可编程器件的英文缩写是( )。
  - A、SOP
  - B、SOPC
  - C、SoC
  - D、SPI
3. 下面的软件中, 专门用于 FPGA 仿真的是: ( )
  - A. QuartusII
  - B. Modelsim
  - C. protues
  - D. Symplify
4. 下列关于同步和异步复位描述正确的是 ( )。
  - A、同步复位是不受时钟影响;
  - B、使用 FPGA 设计时芯片的异步复位和同步复位可随意替换使用;
  - C、同步复位需要在时钟沿来临的时候才会对整个系统进行复位;
  - D、同步复位最大的优点是, 数据通路可以不依赖于时钟而复位可用。
5. 下面说法正确的是: ( )
  - A. Modelssim 不能进行后仿真
  - B. 不可综合语句是没有用的
  - C. Verilog HDL 没有包含大量不可综合的语句
  - D. QuartussII9.0 支持仿真
6. 下列设计流程中, 哪一步不能在 QarutusII 上实现: ( )
  - A. RTL 级描述
  - B. 逻辑综合
  - C. 门级网表
  - D. 芯片级版图
7. 在下列程序中, always 状态将描述一个带同步 Nset 输入端的上升沿触发器, 则空括号内应填入 ( )。

```
always @ ( )
    if (! Nset)
        Q<=1;
    else
        Q<=D;
```

  - A、posedge Clock
  - B、negedge Nset
  - C、negedge Nset or posedge Clock
  - D、negedge Nset or negedge Clock
8. INTEL FPGA 芯片不包括 ( )。
  - A、Cyclone V
  - B、Kintex-7

C、Arria D、Stratix

9. 下列哪些语句不可以被综合成电路 ( )。

A、initial

B、always

C、assign

D、for

## 二、填空题:

1. 目前世界上符合 IEEE 标准的硬件描述语言有: \_\_\_\_\_。

2. initial 块执行\_\_\_\_\_次。

3. 行为级建模中的过程赋值语句用于对\_\_\_\_\_类型的变量赋值。

4. 若条件分支比较多, 使用\_\_\_\_\_语句比较方便。

5. 顺序块使用关键字\_\_\_\_\_来表示。

6. Verilog HDL 程序是由\_\_\_\_\_构成。

7. Verilog HDL 模块中的语句位于关键字\_\_\_\_\_之中。

8. 函数语句使用关键字\_\_\_\_\_来定义。

9. always 块可以表示\_\_\_\_\_逻辑或\_\_\_\_\_逻辑。

10. 状态机分为: \_\_\_\_\_和\_\_\_\_\_。

11. NiosII 的内核分为三个工作模式: \_\_\_\_\_、\_\_\_\_\_和经济型。

12. 状态机的置位和复位分为: \_\_\_\_\_和\_\_\_\_\_。

13. 按照可综合模块设计原则, 用 always 块对组合逻辑建模时, 用\_\_\_\_\_。

14. Verilog 模型的行为描述包括: \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

15. 同一模块中多个 always 语句执行顺序是: \_\_\_\_\_。

16. SignalTapII 的测试接口是 JTAG, 其设计文件后缀名是\_\_\_\_\_。

17. 可编程逻辑单元 LE 主要包括 3 个部分: \_\_\_\_\_、\_\_\_\_\_、  
输出寄存器逻辑。

18. Quartus 的输入方法中, 采用原理图输入法建立的文件后缀为\_\_\_\_\_。

19. 目前 Altera FPGA 主要: \_\_\_\_\_、\_\_\_\_\_、Cyclone。

20. 目前最具代表性的软核嵌入式系统处理器包括: \_\_\_\_\_、\_\_\_\_\_。

21. 目前 Altera FPGA 的配置数据主要下载方式包括: \_\_\_\_\_、\_\_\_\_\_。

22. 根据仿真逻辑意义的不同, 仿真可分为三类: \_\_\_\_\_、\_\_\_\_\_、RTL 级仿真。

23. 结构化的过程块包括: \_\_\_\_\_、\_\_\_\_\_。

24. always 块执行次数: \_\_\_\_\_。

25. assign 语句用于对\_\_\_\_\_类型的变量赋值。

26. 行为级条件语句使用关键词\_\_\_\_\_ 来表示。
27. 块语句按照语句执行时序可分为：\_\_\_\_\_、\_\_\_\_\_。
28. 并行块使用关键字\_\_\_\_\_来表示。
29. Verilog HDL 最基本的 4 种数据类型包括：\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、和 integer。
30. 任务使用关键字\_\_\_\_\_来定义。
31. 带有 posedge 或 negedge 关键字的事件表达式表示\_\_\_\_\_的时序逻辑。
32. Verilog 模型的描述包括：行为描述、\_\_\_\_\_ 和 \_\_\_\_\_ 级\_\_\_\_\_。
33. 同一模块中多个 initial 语句执行顺序是：\_\_\_\_\_。
34. 在 quartusII 中，图形文件的扩展名是\_\_\_\_\_。
35. 目前主要的 FPGA 公司：\_\_\_\_\_、\_\_\_\_\_、Microsemi (Actel)。
36. Altera FPGA 内部的 CPU 分为：\_\_\_\_\_、\_\_\_\_\_、固核。
37. 目前世界上符合 IEEE 标准的硬件描述语言有：\_\_\_\_\_、\_\_\_\_\_。

三、简答题：

1、 基于数字系统设计流程包括哪些步骤？各步骤的功能？

答：

2、 请描述使用 modelsim 进行仿真的基本流程。

答

3、FPGA 与 ARM、DSP 从资源配置来看的区别有哪些。

答

4、FPGA 与 ARM、DSP 开发语言本身的区别有哪些。

答：

5、FPGA 系统结构和资源包括哪些？画出分布图。

答：

6、NIOS II 硬件开发流程包括哪些？

答：

#### 四、名词解释：

1. SoC

答：

2. LUT

答：

3. JTAG

答：

4. ISP

答：

5. 下载

答：

6. FPGA

答：

7. wire 型

答：

8. reg 型

答：

9. 布局布线

答：

10. 逻辑综合

答：

11. RTL 设
12. 时序约束
13. Verilog HDL
14. 时序分析
15. 综合优化
16. CPLD
17. IP 核
18. ISP 下载
19. EDA
20. ASIC
21. FIFO
22. 时序逻辑
23. 约束文件
24. 行为级建模
25. 综合优化
26. 时序路径
27. 综合约束
28. Testbench
29. 组合逻辑
30. 硬件仿真

## 五、电路设计题

1. 设计实现下表 1 功能的电路程序(模块名称 dff)

表 1 功能表

clk	clr	reset	d	q	qn
↑	0	x	x	0	1
↑	1	0	x	1	0
↑	1	1	0	0	1
↑	1	1	1	1	0



2. 设计实现下表 2 功能的电路程序(模块名称 decode)

表 2 功能表

EN	D2D1D0 (d)	Q[7:0] (b)
0	x	11111111
1	0	11111110
1	1	11111101
1	2	11111011
1	3	11110111
1	4	11101111
1	5	11011111
1	6	10111111
1	7	01111111

3.设计实现下表 3 功能的电路程序(模块名称 CODE-83)

表 3 功能表

clk	reset	d
↑	0	0
↑	1	1
↑	1	2
↑	1	3
↑	1	4
↑	1	5
↑	1	6
↑	1	7

4. 设计实现下表 4 功能的电路程序(模块名称 MUX4\_1)

表 4 功能表

S[1:0] (b)	Y
00	D0
01	D1
10	D2
11	D3

5.设计实现下表 5 功能的电路程序(模块名称 counter)

表 5 功能表

<u>clk</u>	reset	d
↑	0	0
↑	1	1
↑	1	2
↑	1	3
↑	1	4
↑	1	5
↑	1	6
↑	1	7

6. 设计实现下表 6 功能的电路程序(模块名称 DECODER\_3\_8)

表 6 功能表

EN	D[2:0]	Q[7:0]
0	X	11111111
1	000	11111110
1	001	11111101
1	010	11111011
1	011	11110111
1	100	11101111
1	101	11011111
1	110	10111111
1	111	01111111

7. 设计下图 1 电路图的程序(模块名 FF)。

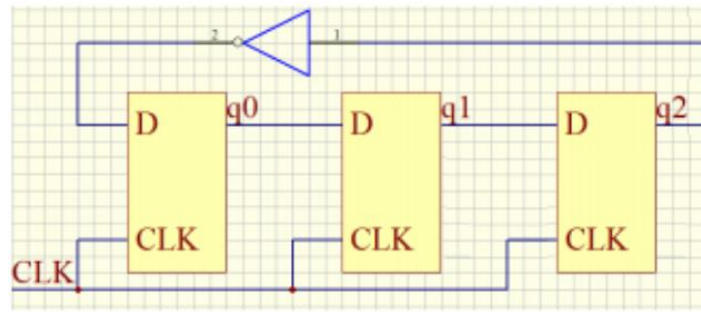


图 1 电路图

8. 用门级结构描述下图 2 的 D 触发器。

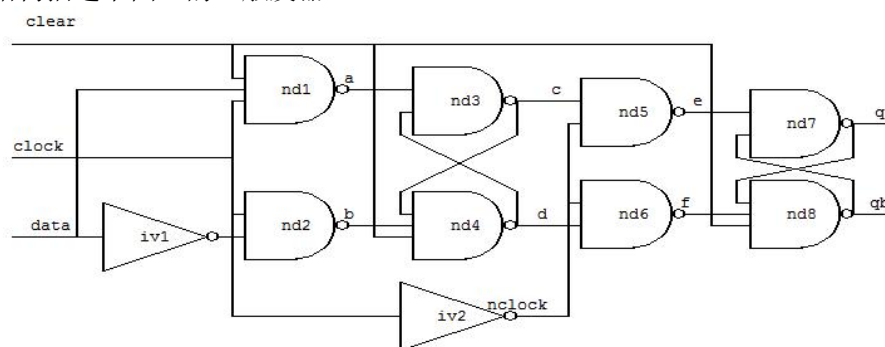


图 2 门级 D 触发器

9.用 Verilog 描述下图 3 的门级 SR 锁存器电路(模块名 SR\_LATCH)。

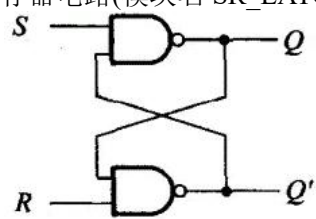


图 3 电路图



10. 用 Verilog 描述下图 4 的门级 JK 触发器电路(模块名 JK\_FF)。

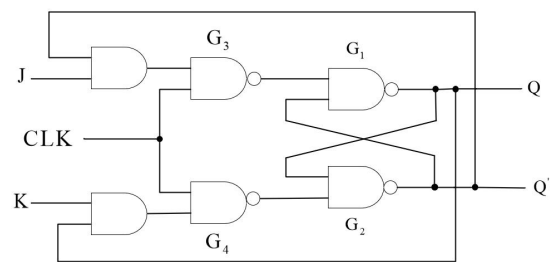


图 4 电路图

11.设计下图 5 电路图的程序(模块名 muxtwo)。

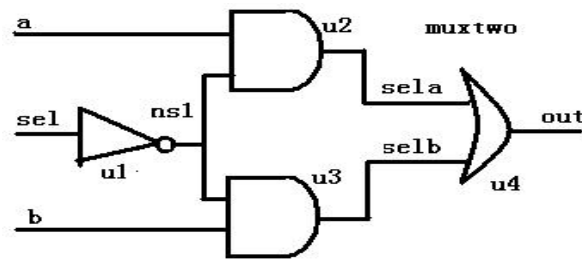


图 5 电路图

12.用 Verilog 描述下图 6 的门级 T 触发器电路(模块名 T\_FF)。

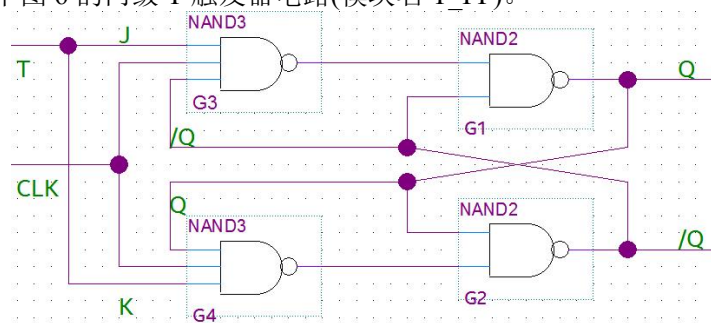


图 6 T 触发器电路图

13.设计一个如图 7 的电机控制（停止、低速、高速）模块（模块名 MotorControl）。

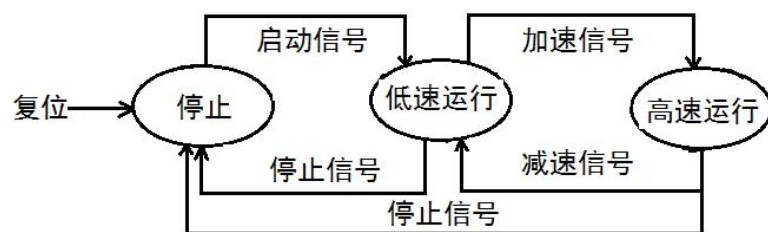


图 7 电机控制图

14.设计一个带使能功能的模 60 计数器、输出为 BCD 的电路模块（模块名 MOD60）。

15.设计下图 8 电路图的程序(模块名 TrafficLightController)



图 8 序列检测电路图

16.设计下图 9 电路图的程序(模块名 XLJC)。

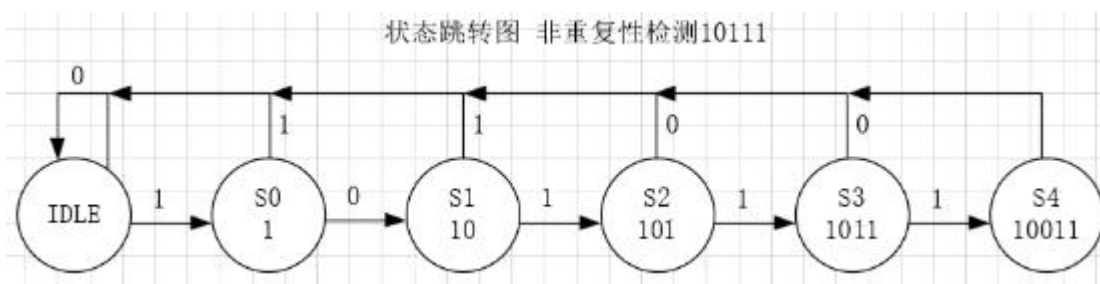


图 9 序列检测电路

17.设计下图 10 所示的电路程序(模块名 FSM)。

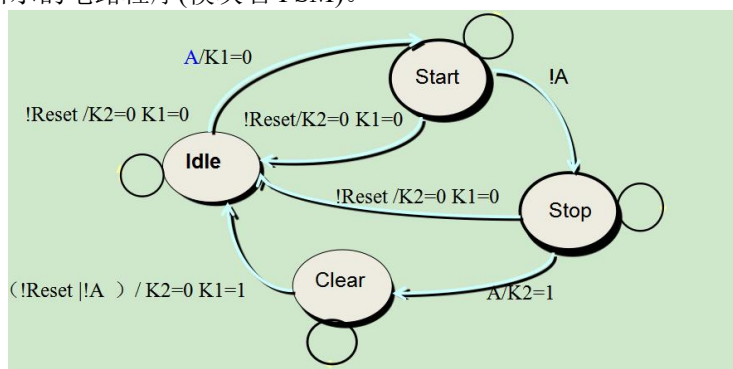


图 10 状态转移电路图



18.设计下图 11 电路图的程序(模块名 XLJC)。

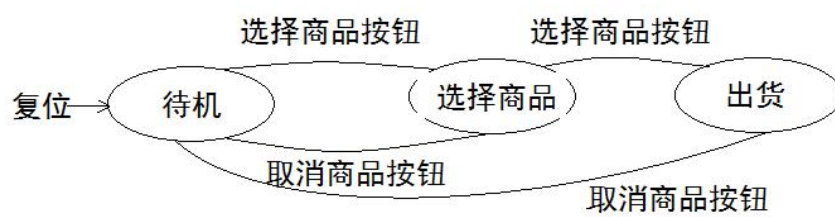


图 11 自动售货机电路

## 六、综合题

1. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

```
//对 CPA 四分频，且占空比为 1/3
module div4(CPA, CPB)
output CPB;
input CPA;
Reg CPB;
always @ (posedge CPA)
    if (i==3)
        begin i=0; CPB<=b' 1; end
    else
        begin i=i+1; CPB<=0; end
    end
endmodule
```

2. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

//带异步清零的 JK 触发器逻辑功能描述

```
module JKCFQ(Q, J, K, CLK, RD)
output Q;
input J, K, CLK, RD;
always @ (posedge CLK)
begin
    if(!RD)    Q<=0;
    else
    case(J, K)
        2' b00 : Q<=Q; 2' b01 : Q<=0;
        2' b10 : Q<=1; 2' b11 : Q<=~Q;
        default : Q<=1' bx;
    endcase
end
Endmodule
```

3. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

```
//异步清零的4位计数器
module Hardreg(d,clk,clrb,q)
input clk,clrb;
input [3:0]d;
output [3:0]q;
Reg [3:0]q;
always @ (posedge clk)
    if (clrb)
        begin q<=b' 1; end
    else
        begin q<=d+1;end
    end
endmodule
```

4. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

```
module Example(clk, data_in, data_out);  
    input clk;  
    input [3:0] data_in;  
    output data_out;  
    reg temp;  
    always @(posedge clk) begin  
        temp = data_in[0];  
    end  
    always @(data_in) begin  
        case(data_in)  
            4'b0001: data_out = 1'b1;  
            4'b0010: data_out = 1'b0;  
        endcase  
    end  
    assign data_out = temp;
```

5. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

```
module Counter(clk, rst, cnt)
input clk, rst;
output cnt;
always @(posedge clk) begin
    if (rst) cnt = 0;
    else cnt <= cnt + 1;
always @(*) begin
    $display("Current count: %d", cnt);
end
initial begin
    cnt <= 0;
end
endmodule
```

6. 试分析下面程序，找出程序中存在的语法错误，写出正确的完整程序。

```
module Mux(input [1:0] sel, input [3:0] D, output Y);
  reg Y;
  always @(sel or D) begin
    casez(sel)
      2'b00: Y = D[0];
      2'b01: Y = D[1];
      2'b10: Y = D[2];
    endcase
  end
  assign D = {D[2:0], 1'b0};
  initial begin
    #10 Y <= 0;
  end
end
```

7. 分析电路描述，画出产生的波形。

```
module CLK_tb(clk);  
    output clk;  
    reg clk;  
    initial  
        begin  
            #3  clk = 1;  
            #4  clk = 0;  
            #6  clk = 1;  
            #2  clk = 0;  
            #5  clk = 1;  
            #12 clk = 0;  
        end  
endmodule
```



8. 分析电路描述，画出产生的波形。

```
module Pulse (clk);  
    output clk;  
    reg clk;  
    initial  
        fork  
            #2 clk = 1; #7 clk = 0;  
            #11 clk = 1; #15 clk = 0;  
            #16 clk = 1; #21 clk = 0;  
        join  
    Endmodule
```

9. 分析电路描述，画出产生的波形。

```
module Clk_tb (clk);  
  output clk;  
  reg clk;  
  initial clk = 1;  
  initial  
    fork  
      #2 clk = ~clk; #5 clk = ~clk;  
      #10 clk = ~clk;    #15 clk = ~clk;  
      #20 clk = ~clk;   #25 clk = ~clk;  
    join  
Endmodule
```

10. 分析电路描述，画出产生的波形。

```
module WAVE_tb;  
  reg clk;  
  initial begin  
    clk = 0;  
    forever #5 clk = ~clk;  
  end  
endmodule
```

11. 分析下图 12 中的波形，设计一个测试模块，产生同样的波形（`timescale 1us/1us`）。

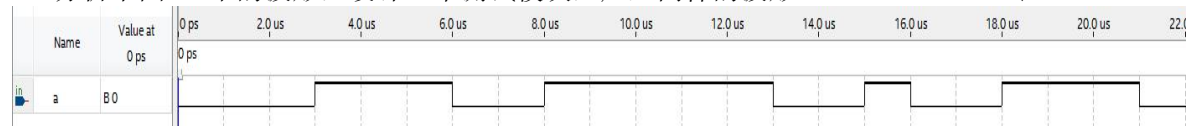


图 12 波形图

12. 分析下图 13 中的波形，设计一个测试模块（LED\_tb），产生同样的波形（`timescale 1us/1us）。

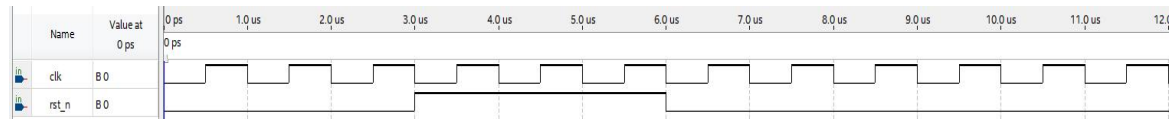


图 13 波形图

13. 根据下图 14 所示的层次化设计模块图，设计相关电路程序，同时设计测试模块（top\_tb）。

$$Y = (A + B) * (C + D)$$

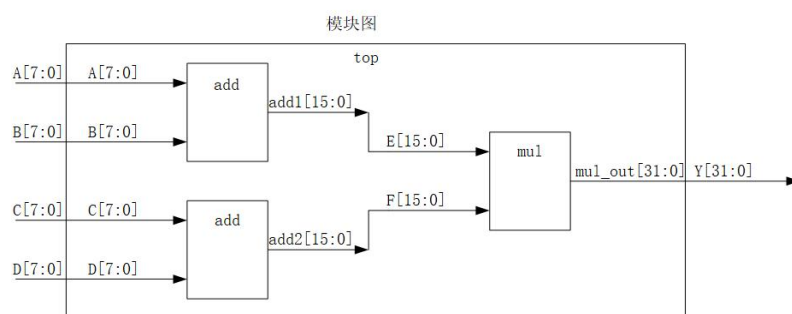


图 14 层次化设计模块图

14. 根据下图 15 所示的层次化设计模块图，设计相关电路程序，同时设计测试模块（div\_8\_tb）。

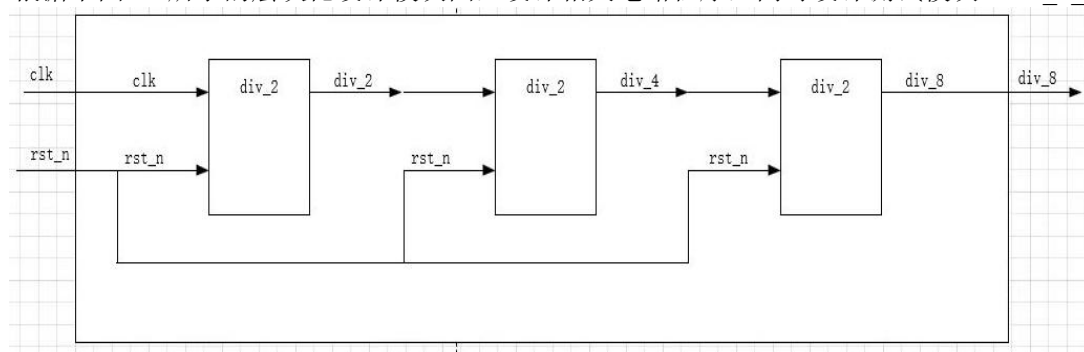


图 15 层次化设计模块图

15.根据下图 16 所示的层次化设计模块图，设计相关电路程序，同时设计测试模块（d 4 tb）。

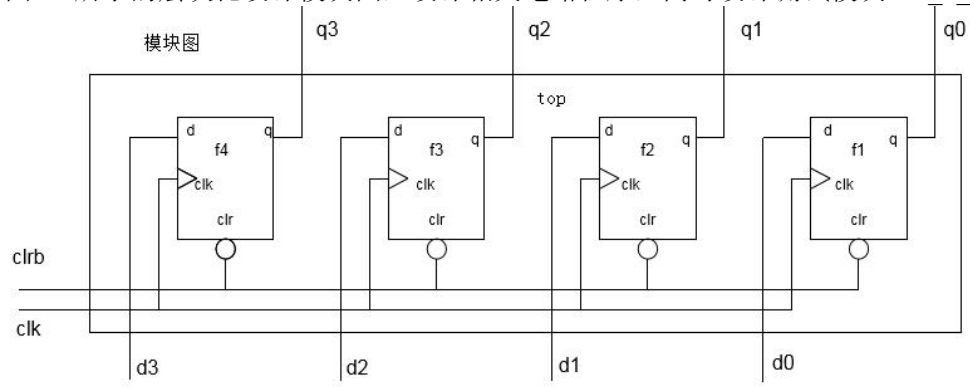


图 16 层次化设计模块图



16.根据下图 17 所示的层次化设计模块图，包含一个分频器模块和一个计数器模块，分频输出驱动计数器，设计相关电路模块，同时设计测试模块（DIV\_16\_tb）。

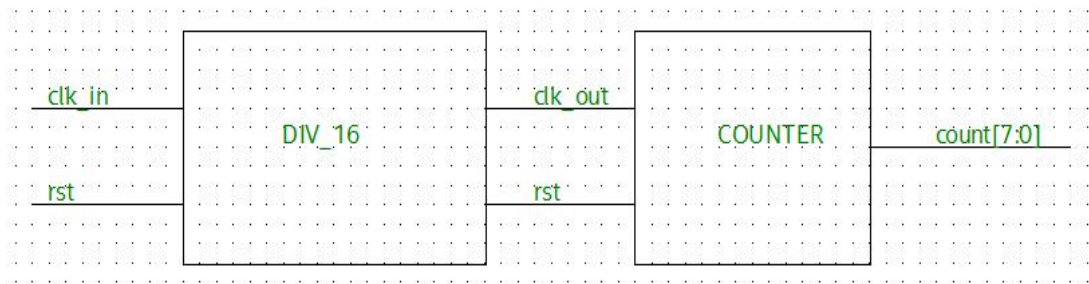


图 17 层次化设计模块图

17.根据下图 18 层次化模块图设计电路模块及测试模块（模块名 ADDER\_8BIT，测试模块 ADDER\_8BIT\_tb）。

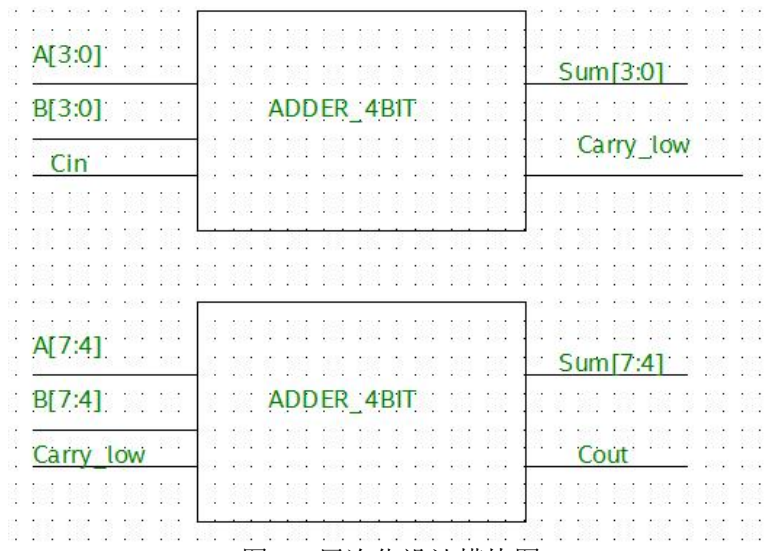


图 18 层次化设计模块图

18.根据层次化模块图 19 设计电路程序及测试模块（模块名 MULTIPLEXER，测试模块 mux\_tb）。

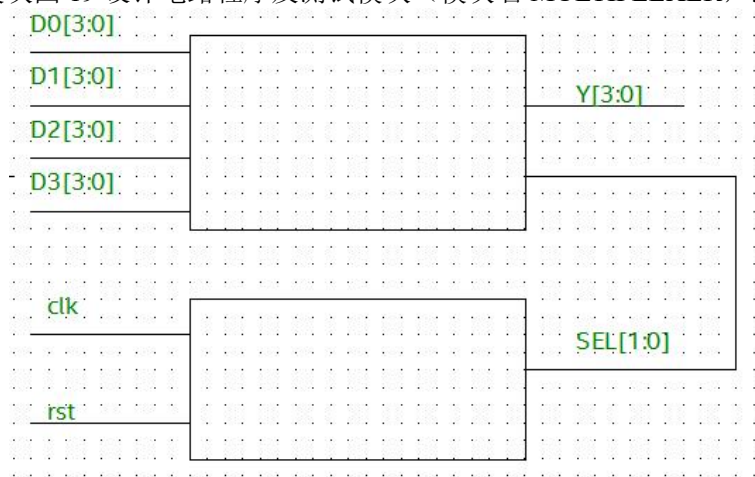


图 19 层次化设计模块图