

B. 数字部分（共 50 分）

计分栏

一 (10 分)	二 (15 分)	三 (15 分)	四 (10 分)	合计

注意：答案直接写在试卷的相应位置上。

一. 填空题（共 10 分，每空 1 分）

1. 变量是在程序运行过程中其值可以改变的量。在 Verilog 中变量分为两种，一种是线网类型，一般表示硬件电路的物理连线，另一种是_____，对应的是具有状态保持作用的电路元件。
2.

```
always begin #5 clk=0;  
#10 clk=~clk; end
```

产生的波形的占空比为_____。
3. reg 类型的数组通常用于描述存储器，语句

```
reg[31:0] memory[1:1024];
```

所定义的存储器的位宽为_____。
4. 在对变量进行赋值时，可以采用不确定值 x，高阻态 z 进行赋值操作。定义 a 为 8 位的线网类型变量，语句

```
a = 8'b1;
```

 和

```
a = 8'b1X;
```

 执行完后 a 的值分别为_____和_____。
5. 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_____。
6. 利用位拼接运算符可以实现多个信号的某些位的重新组合，{1,0} 经过位拼接之后的值为_____。
7. 一个大型的组合电路总延时为 100ns，采用流水线将它分成两个较小的组合电路，不考虑寄存器建立时间延迟，理论上该流水线电路最高工作频率可以为_____MHz。
8. 有表达式：

```
wire [7:0] x;  
_____ /*请补充变量 y 的定义*/
```

```
assign x=8'hEB;
```

```
assign y= &x;
```

请问 y 的值为：_____。

二. 电路分析与设计（共 15 分，每小题 5 分）

（异步时序电路设计的方法与同步时序电路设计的方法基本一致，可以采用多个 always 块语句实现时序逻辑，区别在于异步时序电路中的多个 always 块的时钟信号可以为不同的时钟触发信号。）观察图 1 所示的异步时序电路，在 B_part 部分进行信号生成，在 C_part 部分完成逻辑控制，B_part 部分信号生成波形如 A_part 部分所示。

请解答如下问题：（注意：本题共包含 3 道小题！并注意模块的完整性。）

- (1) 针对图 1 中 C_part 部分，利用 Verilog 语言，采用行为描述风格设计该电路，电路接口参看 C_part 部分电路，注意：不允许使用“xor”、“D_FF”等元件，模块名为 amod。

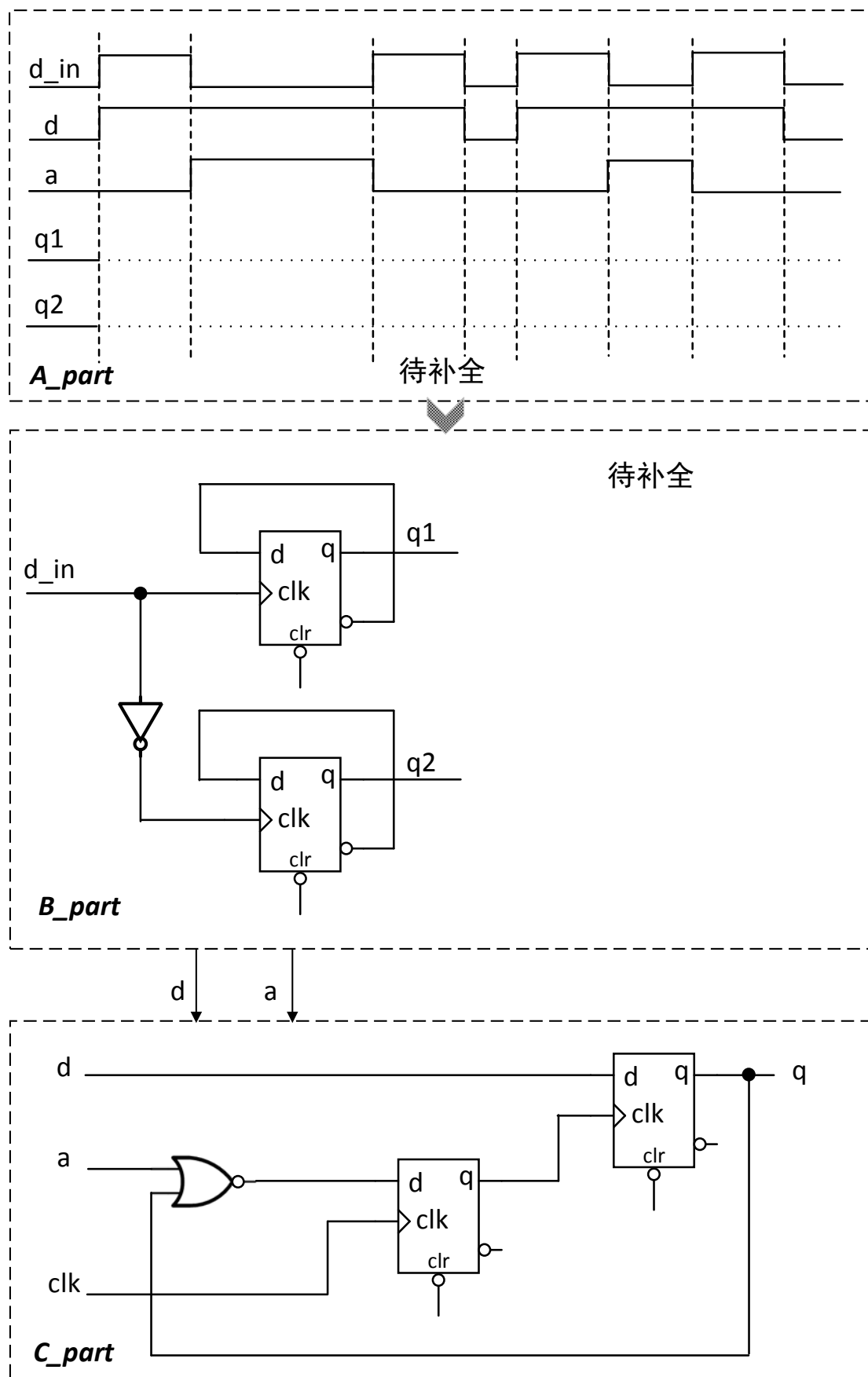


图 1

- (2) 在图 1 中的 B_part 部分完成信号匹配和生成, (依据于输入的 d_in 信号, 产生对应的输出信号 d 和 a, 并将生成的 d 和 a 最终接入到 C_part 部分。) 输入信号 d_in 与其匹配生成的输出信号 d 和 a 的时序关系如 A_part 所示。(为了实现该模块, 可以在 D 触发器 (上升沿触发) 的基础上完成设计。) 结合 A_part 部分的部分电路, 首先给出中间信号 q1 和 q2 的波形; 然后在 B_part 部分补全该电路原理图。
- (3) 在第(2)问的基础上, 根据补全的 B_part 部分的电路原理图, 利用 Verilog 语言, 采用门级建模的方法设计该电路, 模块名为 signal。电路接口参看 B_part 部分电路。(设 Verilog 语言支持多输入与门 (and)、或门 (or)、与非门 (nand)、或非门 (nor), 以及反相器 (not) 等基本门电路, 其模块接口输出参数在前, 输入参数在后。) 对于 D 触发器, 其 Verilog 模块如下, 可以直接实例化 D_FF 以实现 D 触发器的调用。

```
// module D_FF ( Delay Flip-Flop )  
module D_FF ( clk, clr, D, Q, Q_n ) ;  
    input clk, clr, D ;  
    output Q, Q_n ;  
    reg Q ;  
    assign Q_n = ~Q ;  
    always @ ( posedge clk or negedge clr )  
        if ( !clr ) Q <= 0 ;  
        else Q <= D ;  
endmodule
```

三. 电路分析与测试（共 15 分，第 1 小题 8 分，第 2 小题 7 分）

针对一个对 8 位并行输入的数据字中各个比特位为“1”的个数进行计数输出的模块，在全局时钟节拍下实现“1”个数的计数，如图 2 所示。同时保证计数个数从 0 依次增长到最大值并保持（如图 2 中第一个数据字 8'h07，其各个比特位为“1”的个数为 3，在第 5 个采样时刻，计数个数保持 3 不变）。当输入的 8 位数据字的采样周期中“1”的计数没能增长到最大值就更新到新的数据字（如图 2 中第二个数据字 8'h4F，其各个比特位为“1”的个数为 5，需要 6 个采样时刻才能完全计满，但在第 5 个采样时刻数据字变化成 8'h08），则计数器对更新后的数据字中“1”的个数重新进行计数。

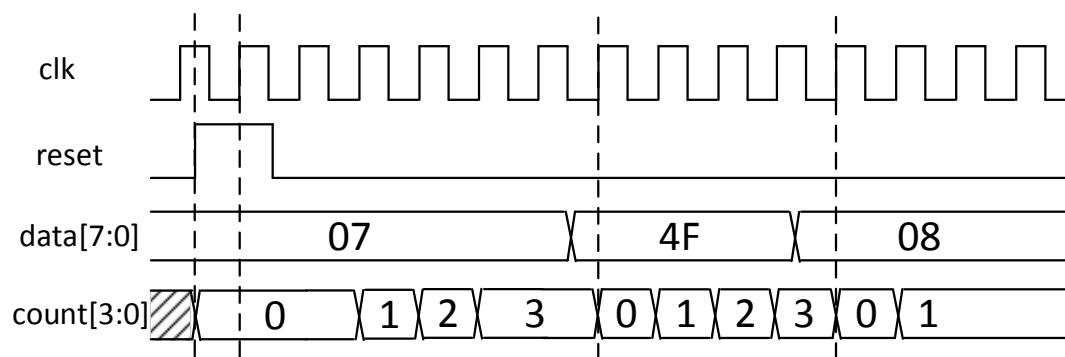


图 2

该模块顶层结构框图如图 3 所示。



图 3

请解答如下问题：（注意：本题共包含 2 道小题！并注意逻辑的正确性。）

- (1) 根据计数功能，利用 Verilog 语言，采用可综合风格设计计数器 bitcounter 模块，接口参照图 3。

```
module bitcounter (clk, reset, count, data);
    input clk, reset;
    input [7:0] data;
    output [3:0] count;

endmodule
```

(2) (为了验证模块设计的正确性, 需要编写测试程序(test bench)。) 测试程序在全局时钟控制下生成如图 4 所示的激励信号。测试流中共有 255 个测试数据, 在 reset 信号之后, 从 8'h01 依次增长到 8'hFF, (即测试数据内容与数据序号相同; 图 4 中第一个数据 8'h00 为 reset 信号后初始化值) 测试数据时间长度为时钟周期的随机整倍数 (随机数的系统函数为 \$random), 随机范围为 1~8 倍, 请编写相应的测试模块, 并注意测试时钟周期。

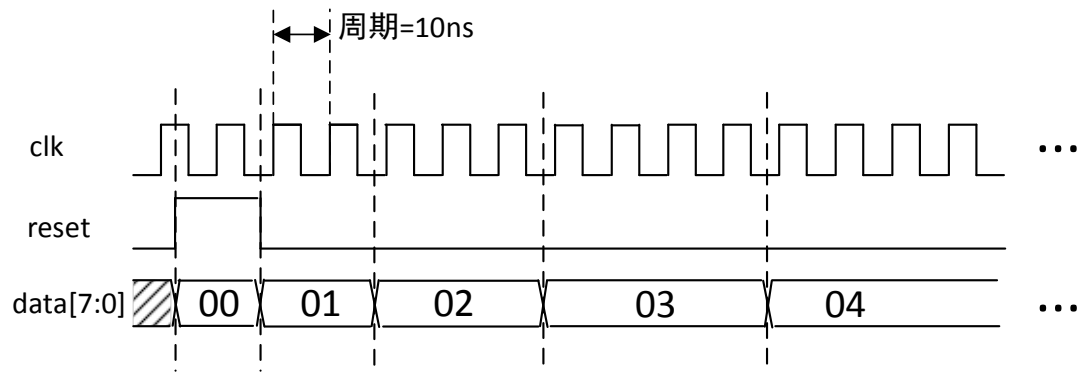


图 4

```
`timescale 1ns/1ns
module tb_bitcounter ;
    reg clk, reset ;
    reg [7:0] data;
    wire [3:0] count;

    reg [8:0] i;    //用于 255 个数据字循环输出

endmodule
```

四. 有限状态机设计（共 10 分）

请采用“二段式”的风格，实现如图 5 所示的有限状态机，注意逻辑的正确性。

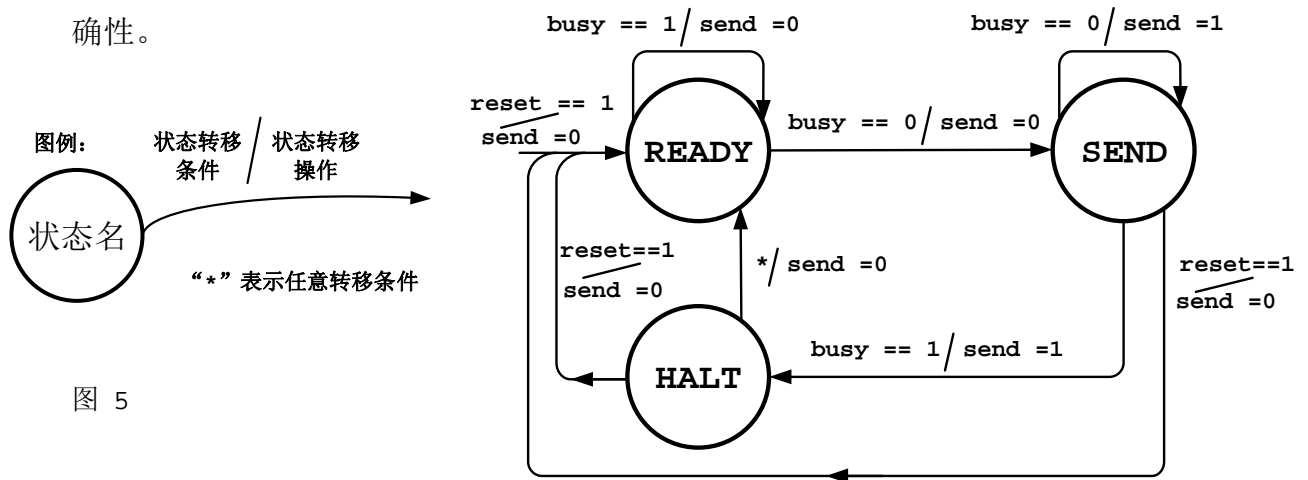


图 5

```
module tx_FSM (clock, reset, busy, send);
    parameter [1:0] READY = 2'b00, SEND = 2'b11, HALT = 2'b10 ;
    input clock ;
    input reset, busy ;
    output send ;

    reg [1:0] state, next_state ;
    always @( posedge clock or posedge reset ) // 请填写代码

    always @( busy or state ) // 请填写代码

endmodule
```