



宁波大学
NINGBO UNIVERSITY

2021-2022 学年第 2 学期

数字系统设计工程实践

(课号: 103D47B01)

实 验 报 告

实验名称: 含“1”统计电路设计

学 院 信息科学与工程学院

班 级 阳明 2 班

姓 名 熊 康

学 号 206001232

指导教师 张 跃 军

完成时间 2022.03.10

目 录

一、实验任务与要求	3
二、系统原理与方案设计	3
2.1 程序 ASM 图	3
2.2 程序状态机图	4
2.3 程序设计	4
2.3.1 脉冲传递	4
2.3.2 “0”“1”单独输入	6
2.3.3 （没有使用状态机）	7
三、系统实现与实验结果分析	8
四、结论	9
五、附录	9
5.1 脉冲传递	9
5.2 “0”“1”分开输入	11
5.3 无状态机设计	13

一、实验任务与要求

输入一串二进制脉冲序列，设计一个含 1 个数统计电路对二进制序列中脉冲为 1 的信号的总数进行计数，并显示结果。

- (1) 使用状态机实现“控制器-受控器”模型，状态机中的状态数须大于或等于 3 个；
- (2) 要求输入的串行二进制数据位数为 15 位，且使用按键输入；
- (3) 系统时钟、启动信号等由按键手动输入；
- (4) “1”的个数由数码管显示，显示进制为十六进制；
- (5) 设计时需给出系统仿真结果。

二、系统原理与方案设计

2.1 程序 ASM 图

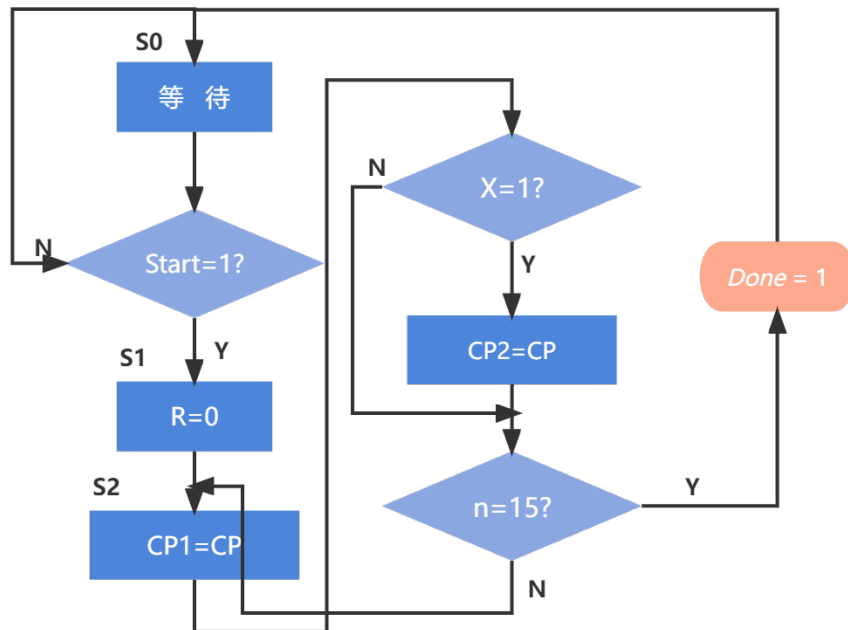


图 1 含“1”统计电路控制器 ASM 图

采用状态机分为 3 个状态，先进入等待状态，当 Start=1 时，程序运行。再当 R=1 时，程序开始进行计数，直到计数到 15 时，程序结束，结束状态灯亮起，然后返回到等待（初始）状态。

2.2 程序状态机图

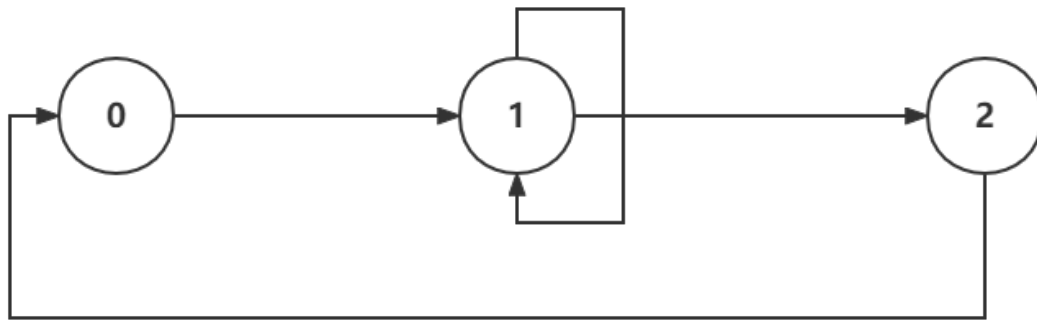


图 2 状态机图

程序分为 3 个状态：

- 1) 0 表示初始状态，所有变量初始为初始值；
- 2) 1 表示状态机反复在状态 1 工作，直到 $n=15$ 的时候进入状态 2；
- 3) 2 表示系统已经完成工作，结束状态灯亮起。

2.3 程序设计

在改变实现条件的情况下设计了三种计数“1”方案

2.3.1 脉冲传递

1.程序设计图

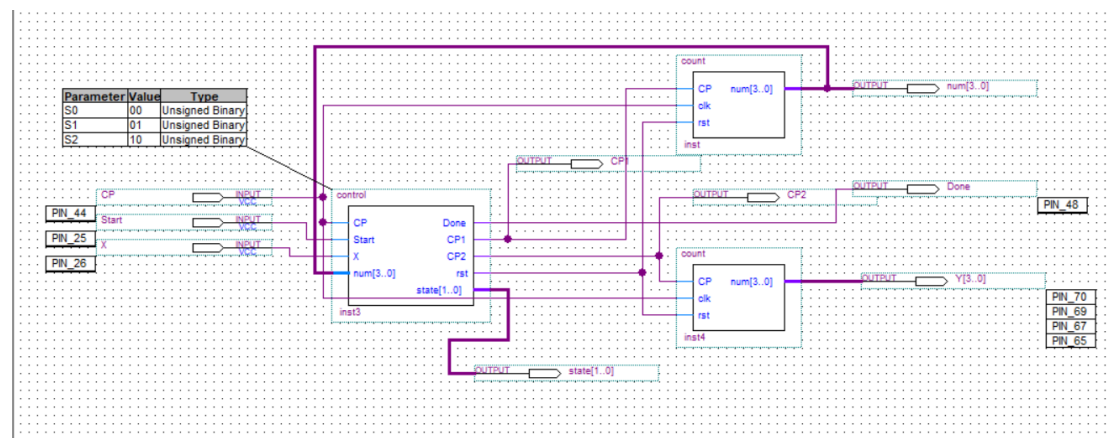


图 2 脉冲传递设计 block 图

CP: 表示输入脉冲，即每次按动按钮。

Start: 1 表示程序运行，0 表示程序关闭以及重置。

X: 配合 CP 使用。1 表示下一个输出波形为 1，0 表示下一个输出波形为 0。

Done: 1 表示程序结束，0 表示程序正在运行或程序关闭。

Y: 输出当前统计的“1”的个数。

2.程序模块介绍

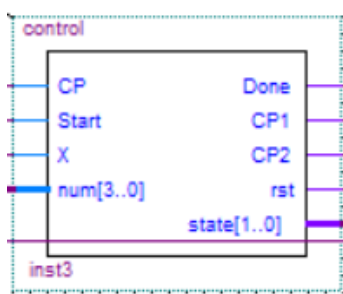


图 2 程序模块图控制器

控制器模块，状态机主体在控制器中运行。实现输入输出，以及控制受控器的运行。

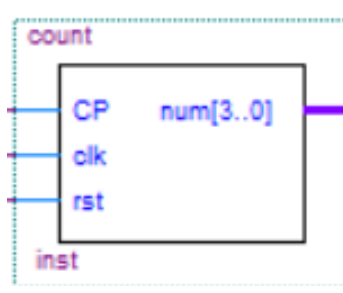
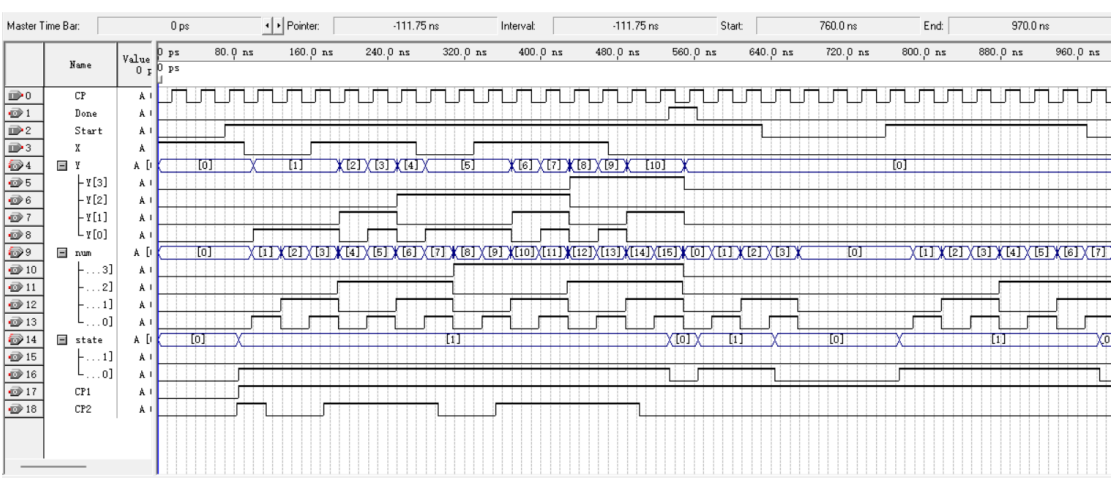


图 3 程序模块图受控器

受控器模块，这里主要功能是计数器，统计“1”的个数，以及输入的脉冲个数。

3.程序仿真



2.3.2 “0” “1” 单独输入

1.程序设计图

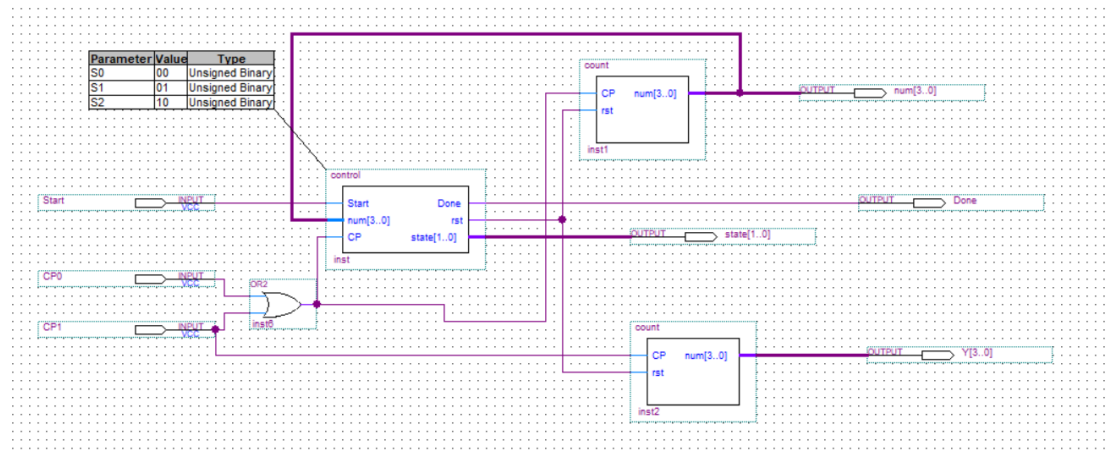


图 5 “0” “1” 单独输入设计 block 图

与方案 1 不同的是，这里采用两个脉冲输入 CP0 和 CP1，如果输入 0，则 CP0 经过一个脉冲；如果输入 1，则 CP1 经过一个脉冲，程序其他部分基本一致。

2.程序模块介绍

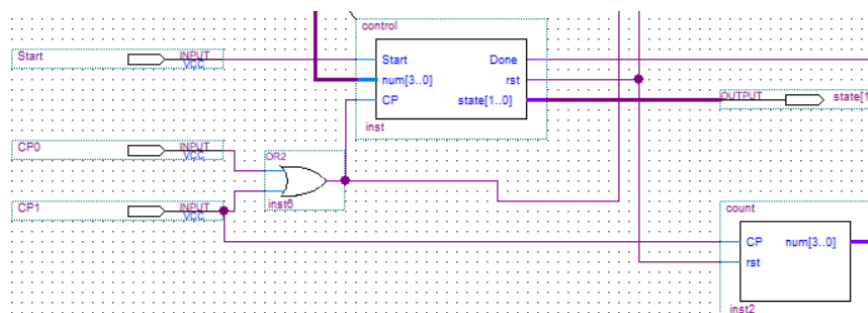


图 6 模块介绍图

与方案 1 相比，CP 为 CP0 和 CP1 “与” 连接，“1” 计数器直接连接 CP1。

3.程序仿真

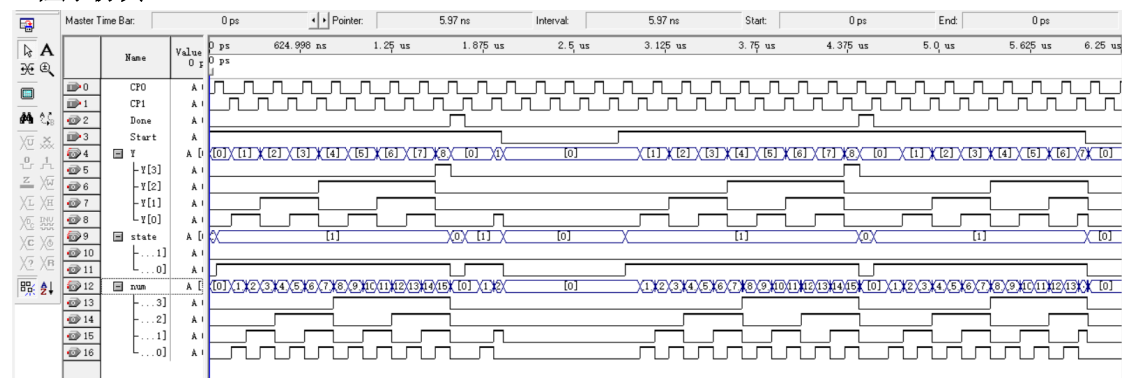


图7 “0”“1”单独输入程序仿真图

当 CP0 或 CP1 输入一个脉冲，num 加一；当 CP1 输入一个脉冲，Y 加一。当 Start=0 时，程序重置，所有清零。

2.3.3 （没有使用状态机）

1.程序设计图

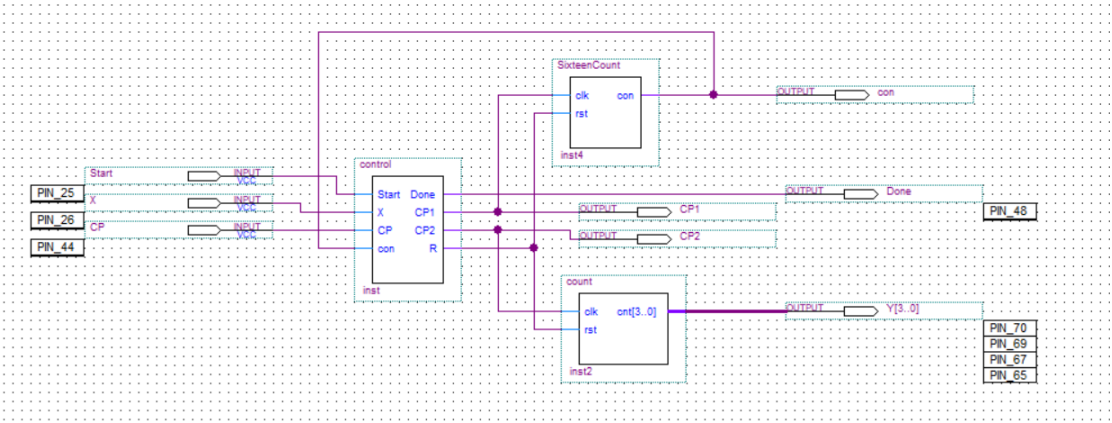


图 8 不使用状态机设计 block 图

2.程序模块介绍

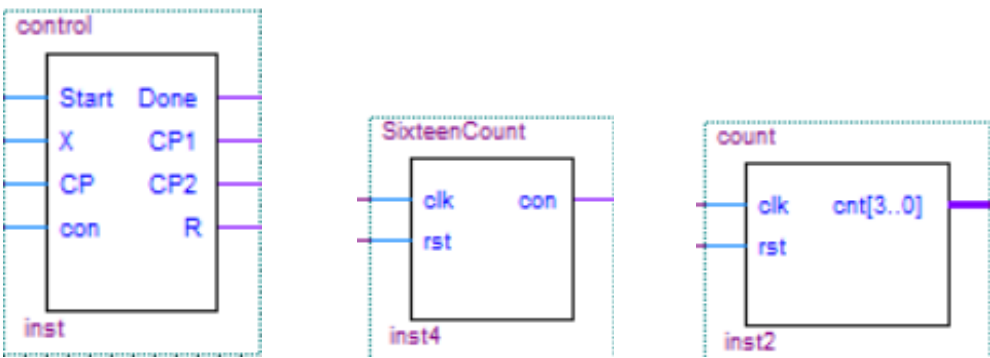


图 9 模块介绍图

控制器内没有使用状态机结构。另外计数对于输入“1”和普通输入分开进行计数。

3.程序仿真

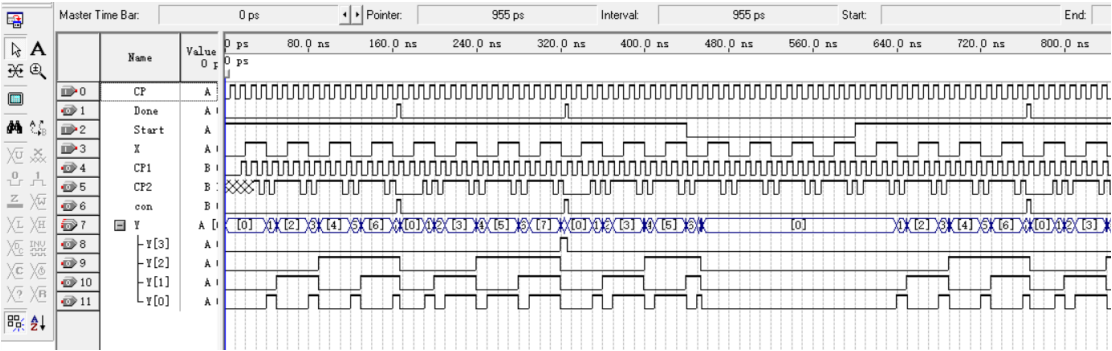


图 10 不使用状态机程序仿真图

通过分析波形图可知，该方案存在缺点：Done=1 时间过短，跳变速度快。也就是说，在电路板上程序结束灯亮起后就被立刻置零。

三、系统实现与实验结果分析

实现结果展示：图 11 程序按下 Start 进入状态 1；图 12 按下 X 并输入脉冲表示输入一个“1”，系统内统计“1”为 1；图 13 在图 12 基础上继续按下 4 次脉冲键，系统内统计“1”为 5；图 14 在图 13 基础上按下 X，关闭 X 后按下 5 次脉冲，此时系统内统计“1”为 5，且已经输入 10 个数。图 15 直接按下 Start，关闭 Start 后程序关闭，状态重置为 0；图 16 连续按动 15 次后，系统结束灯亮，表示已经输入 15 个数。

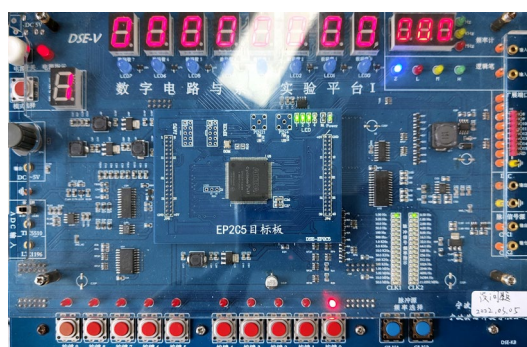


图 11 按键 0 亮 (Start=1)

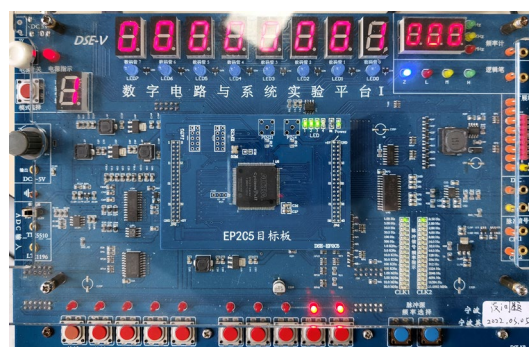


图 12 “1” 状态灯亮 (X=1) 后触发 1 次

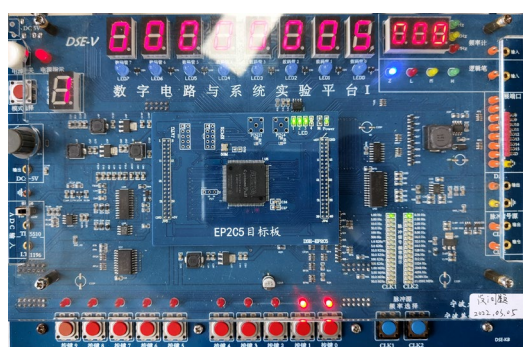


图 13 “1” 状态灯亮 (X=1) 后继续触发 4 次

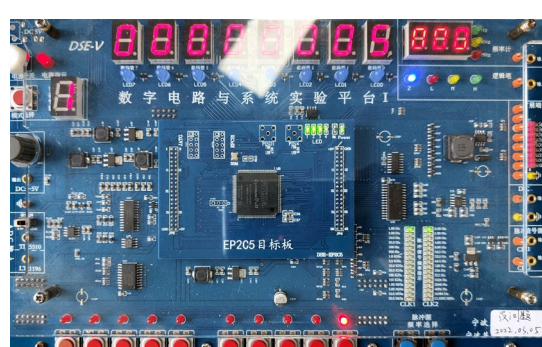


图 14 “1” 状态灯暗 (X=0) 后触发 5 次

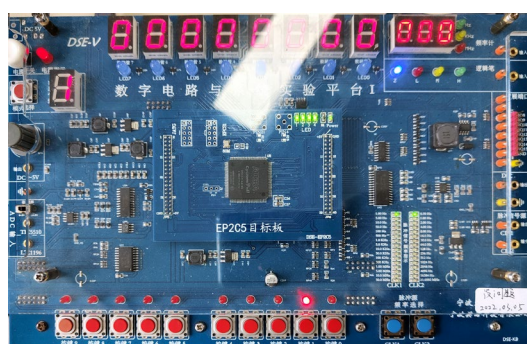


图 15 清零 (Start=0)

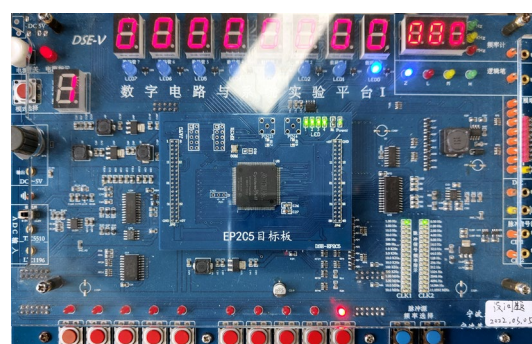


图 16 连续按动 16 次后灯亮(Done=1)

四、结论

通过本次实验熟练掌握数字系统设计中控制器-受控器模型有限状态机（FSM）的设计与应用。

在实验中发现状态机能够解决较为复杂逻辑的电路问题，通过状态的次序简化电路的运行，也能避免逻辑冲突。而如果没有用状态机，如以上第三种设计和图 14，输出 Done=1 维持时间过短，即出现逻辑冲突，程序容易出错。

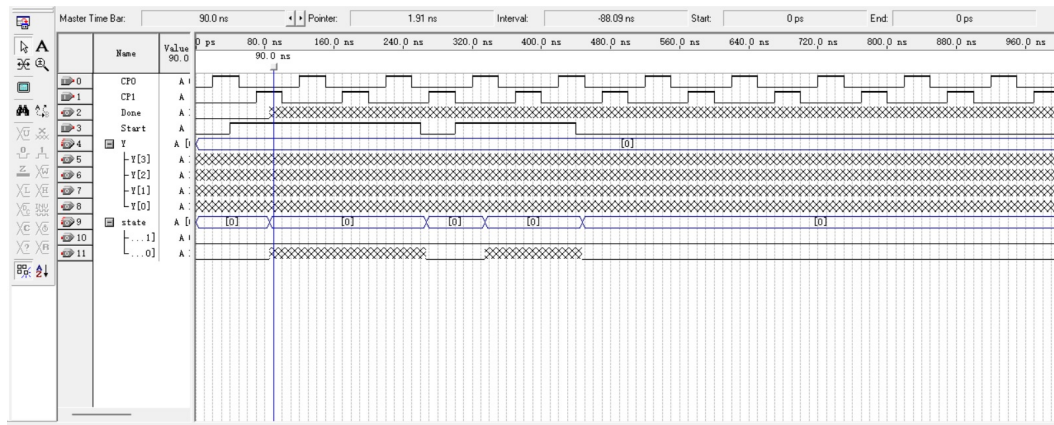


图 17 第 3 种设计过程中出现的错误

五、附录

5.1 脉冲传递

```
1. /*****
2. Author: 熊康
3. E-mail: 1390000666@qq.com
4. Device: DSE-EP2C5
5. Tool: Quartus 9.0
6. Function: 控制器（记录 1 的个数）
7. Version: 2022-3-3 v1.0
8. *****/
9. module control(CP,Start,X,Done,num,CP1,CP2,rst,state);
10. input CP,Start,X;
11. input [3:0] num;
12. output reg Done,CP1=0,CP2=0,rst;
13. parameter S0=2'b00,S1=2'b01,S2=2'b10;
14. output reg [1:0] state=S0;
15.
16. always @(posedge CP or negedge Start)
17. begin
18.     if(!Start)
19.         begin
```

```
20.         state=S0;
21.         rst=1;
22.     end
23. else
24.     begin
25.         case(state)
26.     S0:
27.         begin
28.             Done=0;
29.             state=S1;
30.             rst=0;
31.         end
32.     endcase
33.     case(state)
34.     S1:
35.         begin
36.             if(X==1)
37.                 begin
38.                     CP2<=1;
39.                     CP1<=1;
40.                 end
41.             else
42.                 begin
43.                     CP2<=0;
44.                     CP1<=1;
45.                 end
46.             if(num==4'b1111) state=S2;
47.         end
48.     endcase
49.     case(state)
50.     S2:
51.         begin
52.             Done=1;
53.             state=S0;
54.             rst=1;
55.         end
56.     endcase
57.     end
58. end
59. endmodule
```

```

1. /*****
2. Author: 熊康
3. E-mail: 1390000666@qq.com
4. Device: DSE-EP2C5
5. Tool: Quartus 9.0
6. Function:计数器（记录1的个数）
7. Version: 2022-3-3 v1.0
8. *****/
9. module count(CP,clk,num,rst);
10. input clk,rst,CP;
11. output reg [3:0] num;
12.
13. always @(negedge clk)
14.     begin
15.         if(!rst)
16.             begin
17.                 if(CP==1) num=num+4'b0001;
18.             end
19.         else
20.             num=4'b0000;
21.         end
22.
23. endmodule

```

5.2 “0”“1”分开输入

```

1. /*****
2. Author: 熊康
3. E-mail: 1390000666@qq.com
4. Device: DSE-EP2C5
5. Tool: Quartus 9.0
6. Function:计数器（记录1的个数）
7. Version: 2022-3-10 v1.0
8. *****/
9. module count(CP,rst,num);
10. input CP,rst;
11. output reg [3:0] num;
12.
13. always @(posedge CP or posedge rst)
14.     begin
15.         if(rst==1)
16.             num = 4'b0000;
17.         else
18.             num = num + 4'b0001;

```

```

19.
20.     end
21.
22. endmodule

```

```

1. /*****
2. Author: 熊康
3. E-mail: 1390000666@qq.com
4. Device: DSE-EP2C5
5. Tool: Quartus 9.0
6. Function:控制器（记录1的个数）
7. Version: 2022-3-10 v1.0
8. *****/
9. module control(Start,num,Done,rst,state,CP);
10. input Start;
11. input [3:0] num;
12. input CP;
13. output reg Done,rst=0;
14. parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10;
15. output reg [1:0] state=S0;
16.
17. always @(posedge CP or negedge Start) begin
18.     if(!Start) begin
19.         state = S0;
20.         rst=1;
21.     end
22.     else begin
23.         case(state)
24.             S0: begin
25.                 Done = 0;
26.                 state = S1;
27.                 rst=0;
28.             end
29.             endcase
30.
31.         case(state)
32.             S1: begin
33.                 if(num==4'b1111) state = S2;
34.             end
35.             endcase
36.
37.         case(state)

```

```

38.         S2: begin
39.             Done = 1;
40.             state = S0;
41.             rst = 1;
42.         end
43.     endcase
44. end
45. end
46. endmodule

```

5.3 无状态机设计

```

1.  /*****
2.  Author: 熊康
3.  E-mail: 1390000666@qq.com
4.  Device: DSE-EP2C5
5.  Tool: Quartus 9.0
6.  Function: 控制器（记录1的个数）
7.  Version: 2022-2-24 v1.0
8.  *****/
9.  module control(Start,X,CP,con,Done,CP1,CP2,R);
10. input Start,X;
11. input CP;
12. input con;
13. output reg Done=0;
14. output reg CP1,CP2,R;
15.
16. always @(*)
17. begin
18.     CP1=CP;
19.     if(X==1 && con==0) CP2 = CP;
20.     if(!con) begin
21.         R=Start;
22.         Done<=0;
23.     end
24.     else begin
25.         Done=1;
26.         R=0;
27.     end
28.
29. end
30. endmodule

```

```

1.  /*****
2.  Author: 熊康
3.  E-mail: 1390000666@qq.com
4.  Device: DSE-EP2C5
5.  Tool:   Quartus 9.0
6.  Function:计数器（记录1的个数）
7.  Version: 2022-2-24 v1.0
8.  *****/
9.  module count(clk,rst,cnt);
10. input clk;
11. input rst;
12. output reg [3:0] cnt;
13.
14. always @(posedge clk or negedge rst)
15.     begin
16.         if(!rst)
17.             cnt<=0;
18.         else begin
19.             cnt <= cnt + 4'b0001;
20.         end
21.     end
22.
23. endmodule

```

```

1.  /*****
2.  Author: 熊康
3.  E-mail: 1390000666@qq.com
4.  Device: DSE-EP2C5
5.  Tool:   Quartus 9.0
6.  Function:计数器至16（记录1的个数）
7.  Version: 2022-2-24 v1.0
8.  *****/
9.  module SixteenCount(clk,rst,con);
10. input clk;
11. input rst;
12. reg [3:0] cnt;
13. output reg con=0;
14.
15. always @(posedge clk or negedge rst)
16.     begin
17.         if(!rst) begin
18.             cnt<=0;

```



```
19.         con=0;
20.     end
21.     else begin
22.         if (cnt == 4'b1111) con<=1;
23.         cnt <= cnt + 4'b0001;
24.     end
25. end
26.
27. endmodule
```