|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学部/院 | 智能与计算学部 | 年级 | 2020级 | 班级 | 网络空间安全1班 |
| 姓名 | 石子跃 | 学号 | 3020244294 | 实验日期 | 2022/5/24 |

实验项目名称 自动贩售机的设计与实现

**一. 实验目的**

1. 掌握有限状态机的设计方法。；

2. 能够使用 SystemVerilog 进行三段式状态机的建模。

**二. 实验内容**

采用有限状态机，基于 SystemVerilog HDL 设计并实现一个报纸自动贩售机。

整个工程的顶层模块如图 4-3 所示，输入/输出端口如表 4-1 所示。使用 4 个七

段数码管实时显示已付款和找零情况。其中，两个数码管对应“已付款”，另两个

数码管对应“找零”，单位为分。通过 1 个拨动开关对数字钟进行复位控制。使用

两个按键模拟投币，其中一个按键对应 5 分，另一个按键对应 1 角。使用 1 个

LED 灯标识出售是否成功，灯亮表示出售成功，否则表示已付款不够，出售失败。

假设报纸价格为 15 分，合法的投币组合包括：

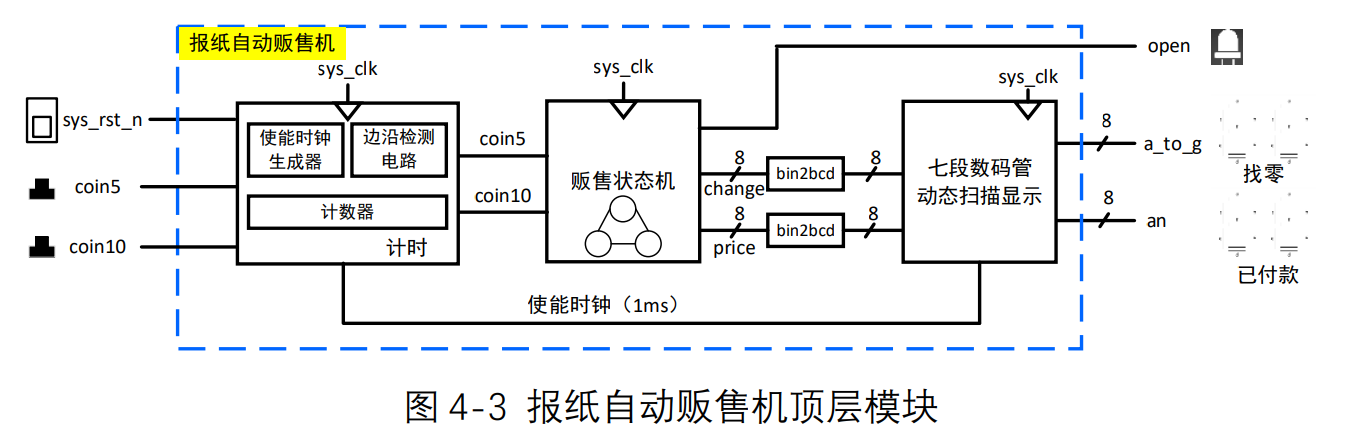
⚫ 1 个 5 分的硬币和一个 1 角的硬币，不找零

⚫ 3 个五分的硬币，不找零

⚫ 1 个 1 角的硬币和一个 5 分的硬币，不找零

⚫ 两个 1 角的硬币是合法的，找零 5 分。

当投入硬币的组合为上面 4 种之一时，则购买成功，LED 灯亮。购买成功后，LED

灯持续亮 10 秒，然后自动熄灭，同时 4 个数码管也恢复为 0。

报纸自动贩售机由 4 部分构成。

⚫ 第一部分是计时器模块，该模块又由 3 个子模块构成，分别是计数器电

路、使能时钟生成电路和边沿检测电路。

⚫ 第二部分是整个自动贩售机电路的核心——贩售机状态机。状态机根据

投币情况产生“已付款”和“找零”输出。此外，如果已付款超过 15 分，则

将 LED 灯点亮，表示出售成功。

⚫ 第三部分是两个 8 位二进制转 BCD 模块，分别将二进制的“已付款”和

“找零”值转化为 BCD 编码，即 10 进制数。本实验中，该模块不需要实

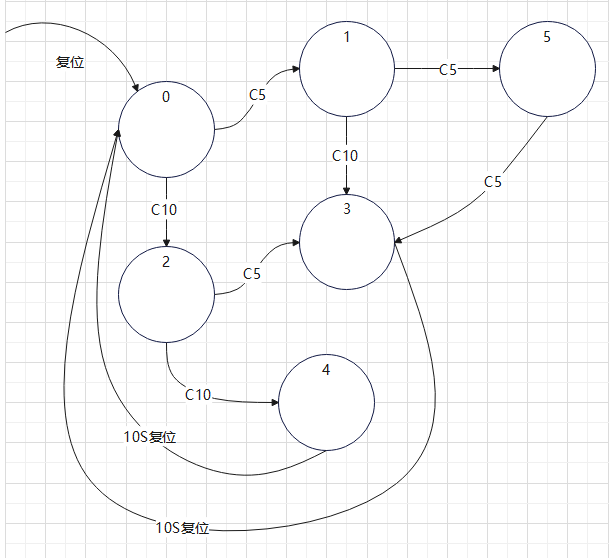
现，由教师直接提供 IP 使用。

⚫ 第四部分是 7 段数码管动态扫描显示模块，它实现“已付款”和“找零”值

的最终显示。

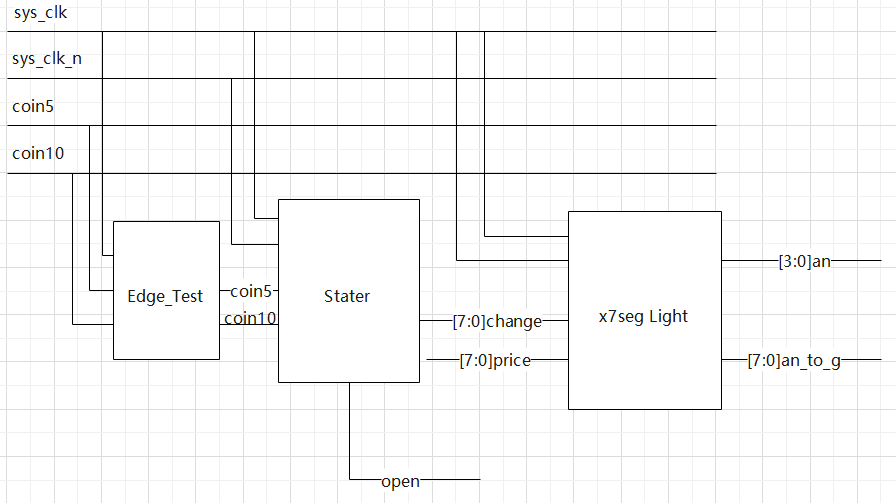
**三．实验原理与步骤（注：步骤不用写工具的操作步骤，而是设计步骤）**

**1. 画出自动贩售机的状态转换图。**



**2. 画出自动贩售机电路的原理图（模块级别即可，如使能时钟模块、边沿**

**检测模块等）。**



**3. 报纸自动贩售机的 SystemVerilog 代码。**

**一．贩卖机主函数**

**module vend(**

**input sys\_clk, sys\_rst\_n,**

**input coin5, coin10,**

**output [3 : 0] an,**

**output [7 : 0] a\_to\_g,**

**output open**

**);**

**logic [7:0]change,price;**

**Stater Machine(sys\_clk,sys\_rst\_n,coin5,coin10,change,price,open);**

**x7seg Light(sys\_clk,sys\_rst\_n,change,price,an,a\_to\_g);**

**endmodule**

**二．状态机**

**module Stater(**

**input sys\_clk,**

**input sys\_rst\_n,**

**input coin5,coin10,**

**output [7:0]change,price,**

**output open**

**);**

**logic C5,C10;**

**Edge\_Test Coin5(sys\_clk,coin5,C5);**

**Edge\_Test Coin10(sys\_clk,coin10,C10);**

**logic [2:0]cstate,nstate;**

**integer m;**

**Countdown Se10(sys\_clk,sys\_rst\_n,open,m);**

**always\_ff @(posedge sys\_clk)begin**

**if(~sys\_rst\_n||m==250000000)cstate <= 0;**

**else cstate <= nstate;**

**end**

**always\_comb begin**

**case(cstate)**

**3'd0:begin**

**if(C5) nstate = 3'd1;**

**else if(C10) nstate = 3'd2;**

**else nstate=cstate;end**

**3'd1:begin**

**if(C5) nstate = 3'd5;**

**else if(C10) nstate = 3'd3;**

**else nstate=cstate;end**

**3'd2:begin**

**if(C5) nstate = 3'd3;**

**else if(C10) nstate = 3'd4;**

**else nstate=cstate;end**

**3'd5:begin**

**if(C5) nstate = 3'd3;**

**else nstate=cstate;end**

**default:**

**nstate=cstate;**

**endcase**

**end**

**logic [7:0]binchange,binprice;**

**logic open;**

**always\_ff@(posedge sys\_clk)begin**

**case(nstate)**

**3'd0:begin binchange<=8'd0;binprice<=8'd0;open<=0;end**

**3'd1:begin binchange<=8'd0;binprice<=8'd5;open<=0;end**

**3'd2:begin binchange<=8'd0;binprice<=8'd10;open<=0;end**

**3'd3:begin binchange<=8'd0;binprice<=8'd15;open<=1;end**

**3'd4:begin binchange<=8'd5;binprice<=8'd20;open<=1;end**

**3'd5:begin binchange<=8'd0;binprice<=8'd10;open<=0;end**

**endcase**

**end**

**bin2bcd\_0 bcdchange(binchange,change);**

**bin2bcd\_0 bcdprice(binprice,price);**

**endmodule**

**三．7段数码管**

**module x7seg(**

**input sys\_clk,**

**input sys\_rst\_n,**

**input [7:0]change,**

**input [7:0]price,**

**output [3:0]an,**

**output [7:0]an\_to\_g**

**);**

**logic [1:0]clock;**

**logic [3:0]temp\_an;**

**logic [3:0]temp;**

**logic [7:0]temp\_to\_g;**

**integer m;**

**Enableclock enclock(sys\_clk,sys\_rst\_n,clock);**

**always\_comb begin**

**case(clock)**

**2'b00: begin temp\_an=4'b0001; temp=price[3:0];end**

**2'b01: begin temp\_an=4'b0010; temp=price[7:4];end**

**2'b11: begin temp\_an=4'b1000; temp=change[7:4];end**

**2'b10: begin temp\_an=4'b0100; temp=change[3:0];end**

**endcase**

**end**

**always\_comb begin**

**case(temp)**

**4'd0:temp\_to\_g=8'b01000000;**

**4'd1:temp\_to\_g=8'b01111001;**

**4'd2:temp\_to\_g=8'b00100100;**

**4'd3:temp\_to\_g=8'b00110000;**

**4'd4:temp\_to\_g=8'b00011001;**

**4'd5:temp\_to\_g=8'b00010010;**

**4'd6:temp\_to\_g=8'b00000010;**

**4'd7:temp\_to\_g=8'b01111000;**

**4'd8:temp\_to\_g=8'b00000000;**

**4'd9:temp\_to\_g=8'b00011000;**

**default:temp\_to\_g=8'b00000000;**

**endcase**

**end**

**assign an=temp\_an;**

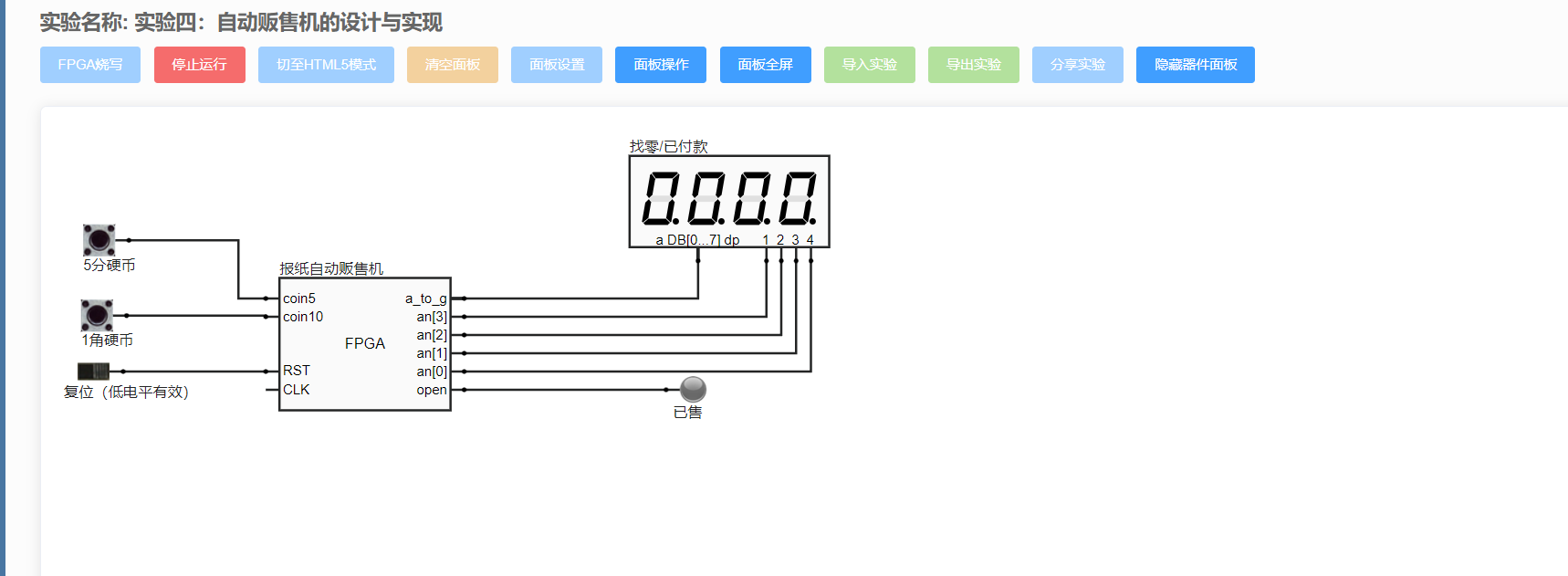
**assign an\_to\_g=temp\_to\_g;**

**endmodule**

**4. 给出仿真的波形图和远程 FPGA 平台验证的截图。**

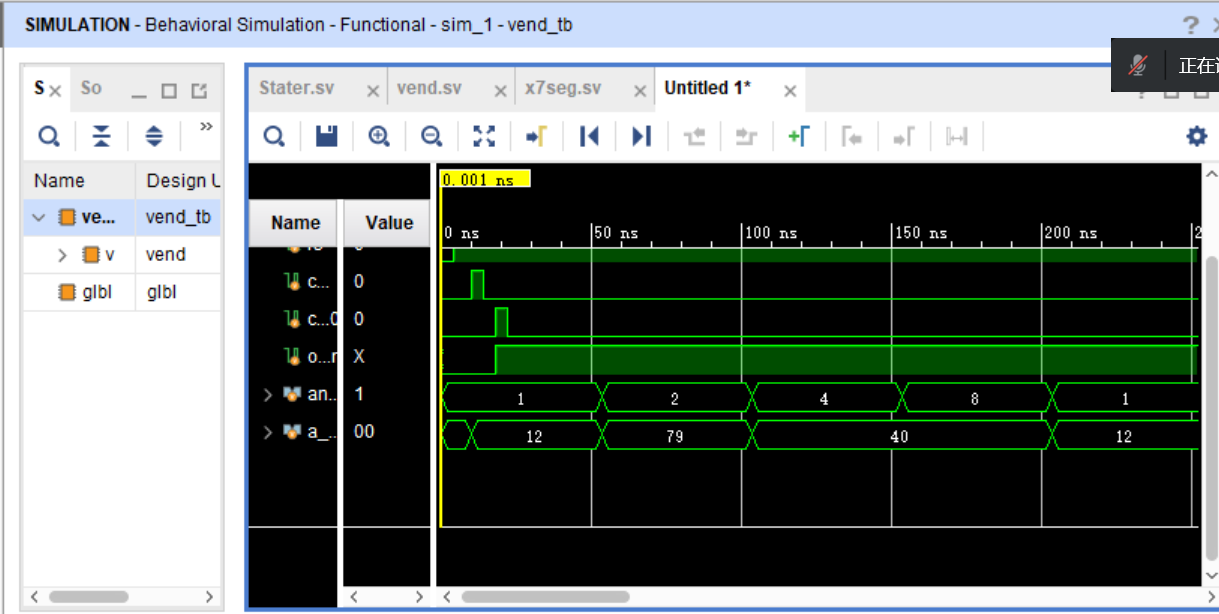
**四．仿真与实验结果（注：仿真需要给出波形图截图，截图要清晰，如果波形过长，可以分段截取；实验结果为远程FPGA硬件云平台的截图）**

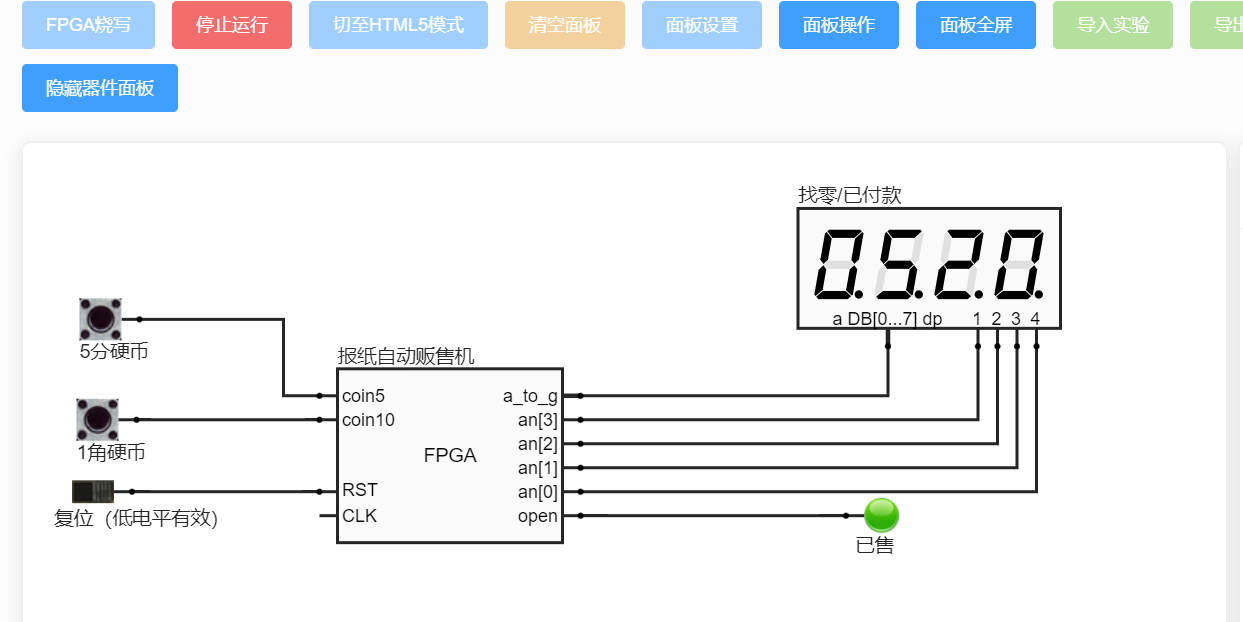
注：远程FPGA硬件云平台截图只需要一个测试激励即可





等待十秒后open也灭掉了





**五．实验中遇到的问题和解决办法**

1.工程框架总体较为繁琐

刚开始就着急进行编程，走了一些弯路。

多阅读实验报告，了解工程结构后，在进行相应编程。

2.行为仿真时速度太慢

对使能时钟频率和计时频率进行适当放大，提高仿真速度

3.暂停时，使能时钟也暂停了

**教师签字：**

**年 月 日**