自动售货机

**一、设计要求：**

设计一个自动售货机控制系统，该系统能完成对商品信息的存储，购买，硬币处理，余额计算等功能，自己设计具体的流程和系统。

**二、代码实现：**

①功能模块

//Copyright @DENGJIXIANG

module shop(reset,clk,set,display,set\_quantity,set\_price,money,buy,close,goods,balance,priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD);

input [3:0] money,buy,set,display,set\_quantity,set\_price;//投币，想要买的商品，确认设置，显示信息，设置数量，设置价格

input reset,clk,close;//复位，时钟，购买完成退出

output [3:0] goods,balance,priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD;//出货的商品，余额，四种商品的价格和数量

reg [7:0] ram[0:3];//储存数组，高四位为数量，低四位为价格

reg [3:0] counter,//硬币储存器

balance,priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD;

reg [3:0] goods;

integer i;

always @\*

begin

if(reset)//初始化

begin

counter=0;

priceA=0;priceB=0;priceC=0;priceD=0;quantityA=0;quantityB=0;quantityC=0;quantityD=0;

for(i=0;i<4;i=i+1)

ram[i]=0;//将储存数组全部赋值0

end

else if(close)//购买结束，退出

begin

counter=0;

end

if(ram[buy-1][7:4]>=1)//验证商品是否有库存

begin

if(ram[buy-1][3:0]<=counter)//验证投币金额是否足够买商品

begin

balance=counter-ram[buy-1][3:0];//将购买后的余额实时输出

ram[buy-1][7:4]=ram[buy-1][7:4]-1;//购买后商品的数量实时减一

goods=buy;//输出已经购买的商品

end

else

$display("余额不足！");

end

else

$display("已售空！");

end

always @set//设置商品的价格和数量

begin

ram[set-1][3:0]=set\_price;//将需要设置的价格存入储存数组对应set的位置，这里我在ram[set-1]中设为set-1，目的在于方便操作，例如我想设置商品1（即商品A），则set=1的商品信息存于ram[0]即储存数组的第一位

ram[set-1][7:4]=set\_quantity;//设数量，原理同上

end

always @money//投入的硬币，可投任意数量任意金额的钱

counter=counter+money;

always @display//将设置好的或者有变化的商品信息显示出来，这里我设置了四个商品，可以任意增加，但要对应ram[]储存数组的存储量

begin

priceA=ram[0][3:0];

priceB=ram[1][3:0];

priceC=ram[2][3:0];

priceD=ram[3][3:0];

quantityA=ram[0][7:4];

quantityB=ram[1][7:4];

quantityC=ram[2][7:4];

quantityD=ram[3][7:4];

end

endmodule

//Copyright @DENGJIXIANG

②测试模块

//Copyright @DENGJIXIANG

module stimulus;

reg [3:0] money,buy,set,display,set\_quantity,set\_price;

reg reset,clk,close;

wire [3:0] goods,balance;

wire [3:0] priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD;

shop test(reset,clk,set,display,set\_quantity,set\_price,money,buy,close,goods,balance,priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD);

initial

begin

clk=1'b0;

reset=1;

set=0;

display=0;

#5 reset=0;

end

initial

forever #10 clk=~clk;

initial

begin

//开始设置，可任意设置商品价格和数量

#10 set=1;display=0;set\_quantity=4'd10;set\_price=4'd5;

#10 set=0;

#10 set=2;set\_quantity=4'd10;set\_price=4'd4;

#10 set=0;

#10 set=3;set\_quantity=4'd10;set\_price=4'd3;

#10 set=0;

#10 set=4;set\_quantity=4'd10;set\_price=4'd10;

#10 set=0;display=4'd1;

//第一次购买

#10 money=4'd10; //投币十元

#5 buy=4'd2;//购买商品2（即商品B）

#10 close=1;display=0;//购买结束，退出时显示余额和商品剩余数量

#5 close=0;display=4'd1;buy=4'd0;//若不将close，display，buy信号复位，则结果出错

//第二次购买，已经设置的商品

#10 money=4'd8;//投币八元

#5 buy=4'd1;//购买商品1（即商品A）

#10 close=1;display=0;//购买结束

#5 close=0;display=4'd1;buy=4'd0;

#5 buy=4'd1;//购买商品1（即商品A）

#10 close=1;display=0;//购买结束

#5 close=0;display=4'd1;buy=4'd0;

end

initial

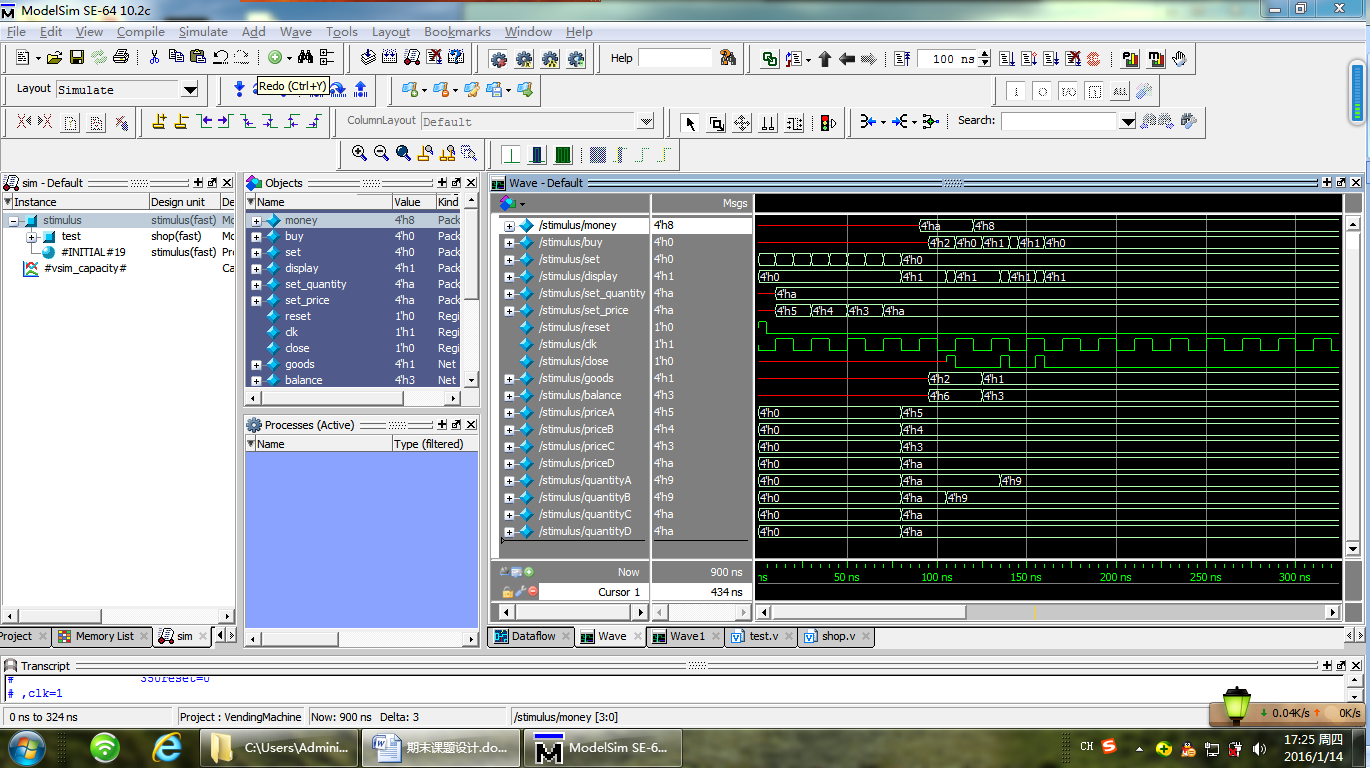
$monitor($time, "reset=%d\n,clk=%d\n,set=%d\n,display=%d\n,set\_quantity=%d\n,set\_price=%d\n,money=%d\n,buy=%d\n,close=%d\n,goods=%d\n,balance=%d\n,priceA=%d\n,priceB=%d\n,priceC=%d\n,priceD=%d\n,quantityA=%d\n,quantityB=%d\n,quantityC=%d\n,quantityD=%d\n",

reset,clk,set,display,set\_quantity,set\_price,money,buy,close,goods,balance,priceA,priceB,priceC,priceD,quantityA,quantityB,quantityC,quantityD);

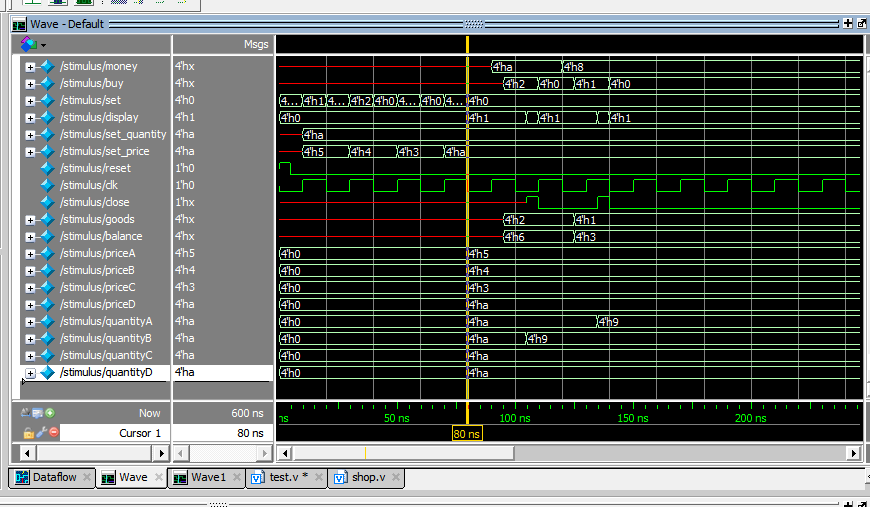
endmodule

//Copyright @DENGJIXIANG

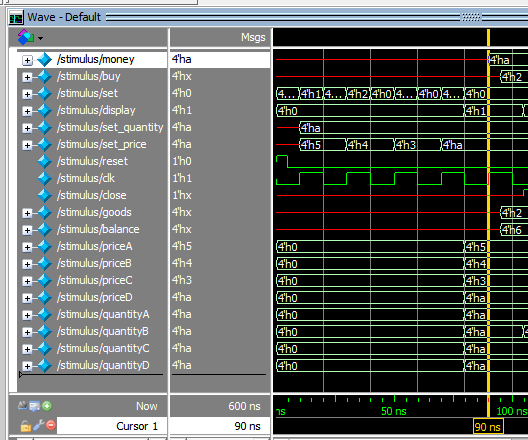
**三、仿真测试：**



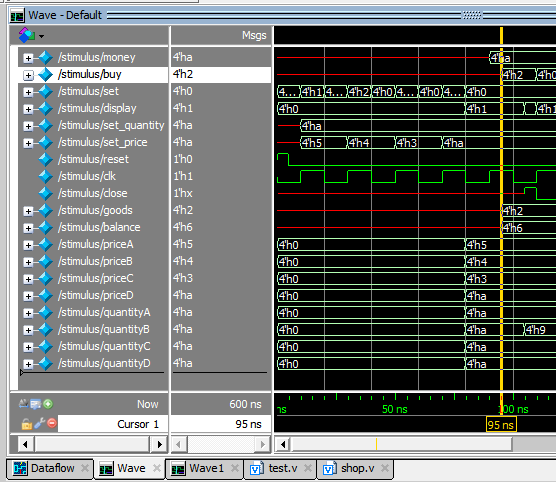
1.80ns时设置商品



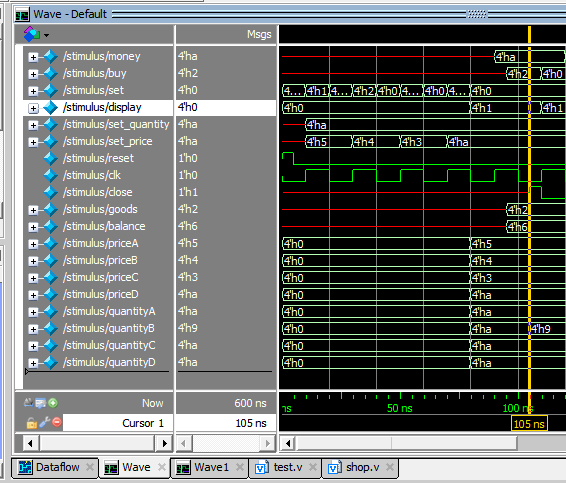
2.90ns时第一次投币



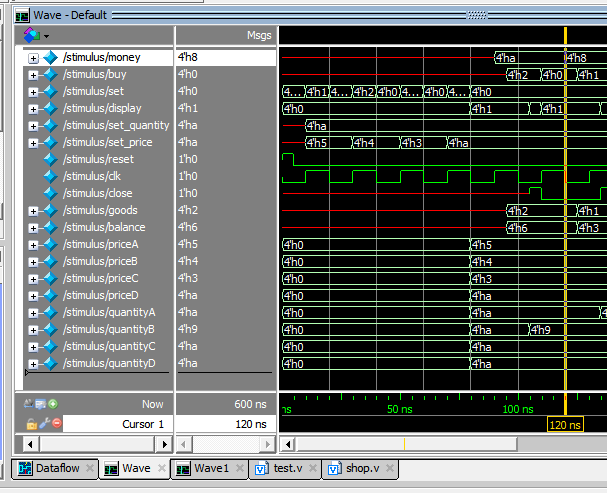
3.95ns时输入购买的商品，第一次为购买商品2，价格为4元，出货goods=2，余额balance=10-4=6



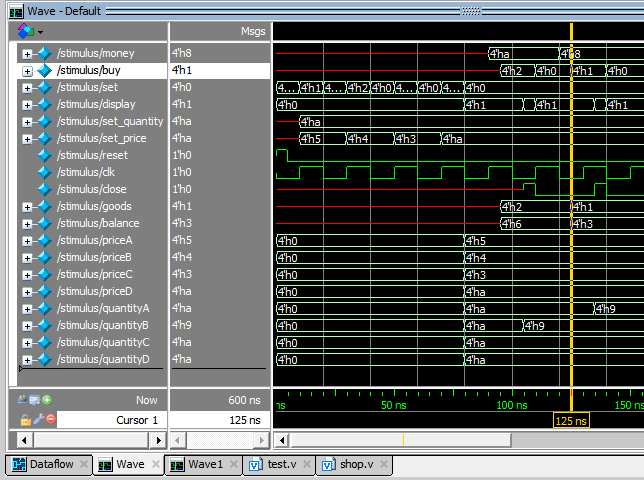
4.105ns时输入display，更新商品信息，商品2（即商品B）数量减一，其他不变



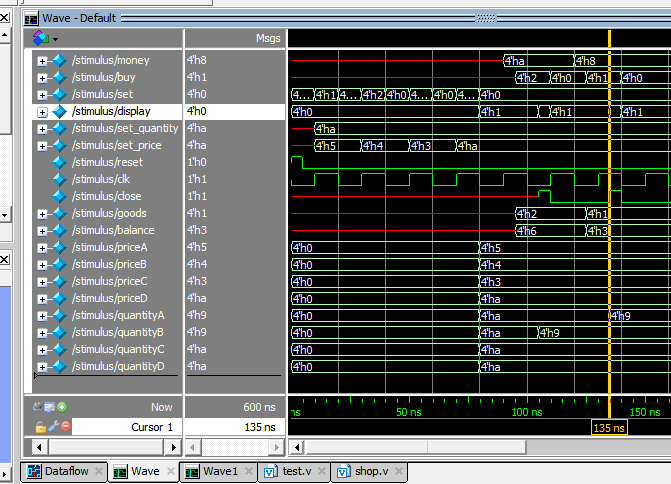
5.120ns时，第二次投币，投币八元



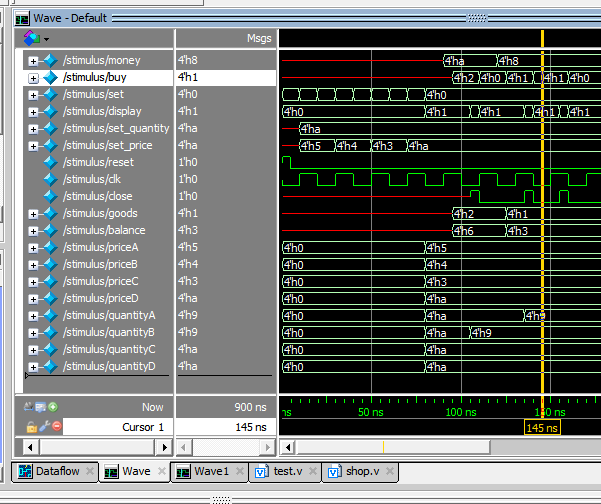
6.125ns时，输入购买的商品1（即商品A），价格为5元，出货goods=1，余额balance=8-5=3



7.135ns时，更新商品信息，商品1（即商品A）数量减一，其他不变



8.145ns时，想再次购买商品1，但不能购买，因为余额不足



9.可进行多次购买，只要余额和商品库存足够