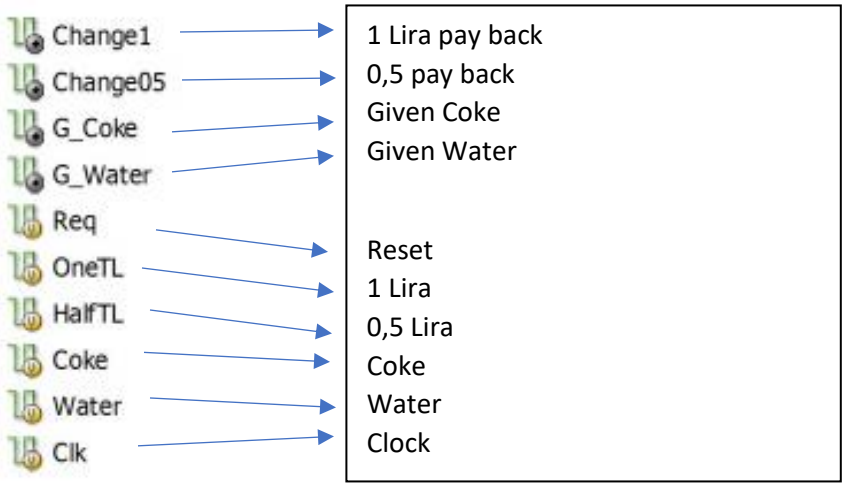
Emre Hancı
21604552

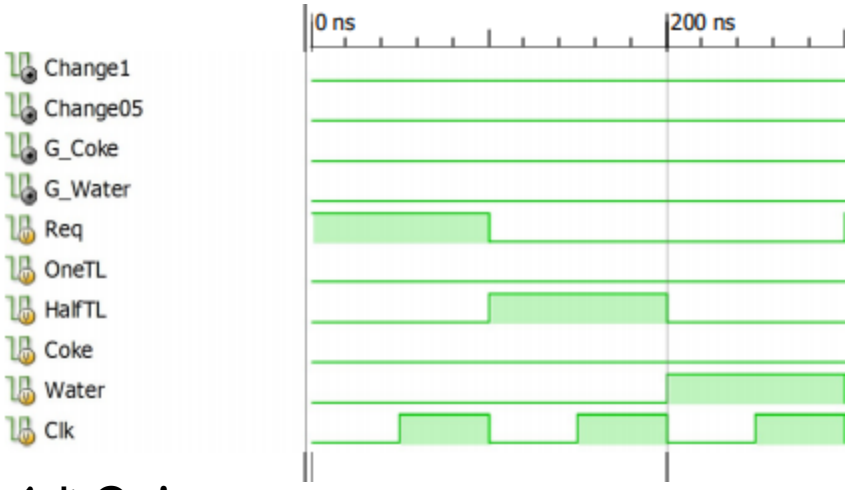
Lab. Project2

Vending Machine

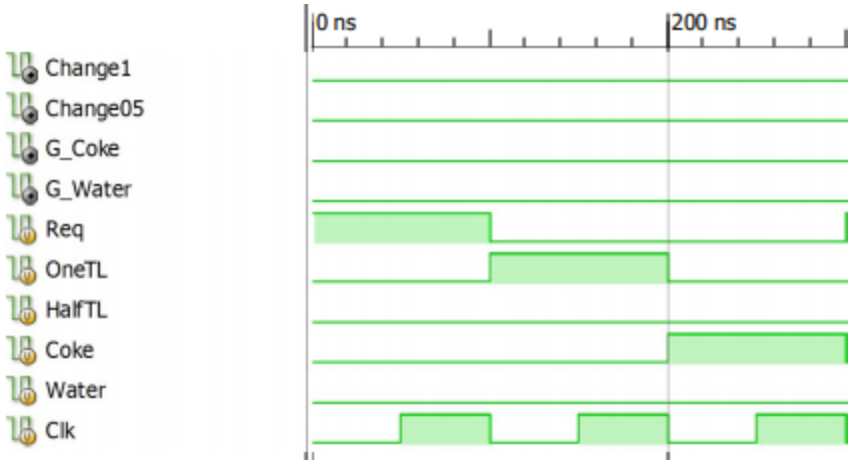
My design works until 3 Turkish Liras because if a person wants a coke and if that person has 4 of 1 Turkish Lira 3 of them is enough for a coke. Therefore If a person wants a buy a thing from that machine mak 3 Liras is enough. If a person gives 3 Liras to machine and wants water, machine will give 1 and a half liras back.



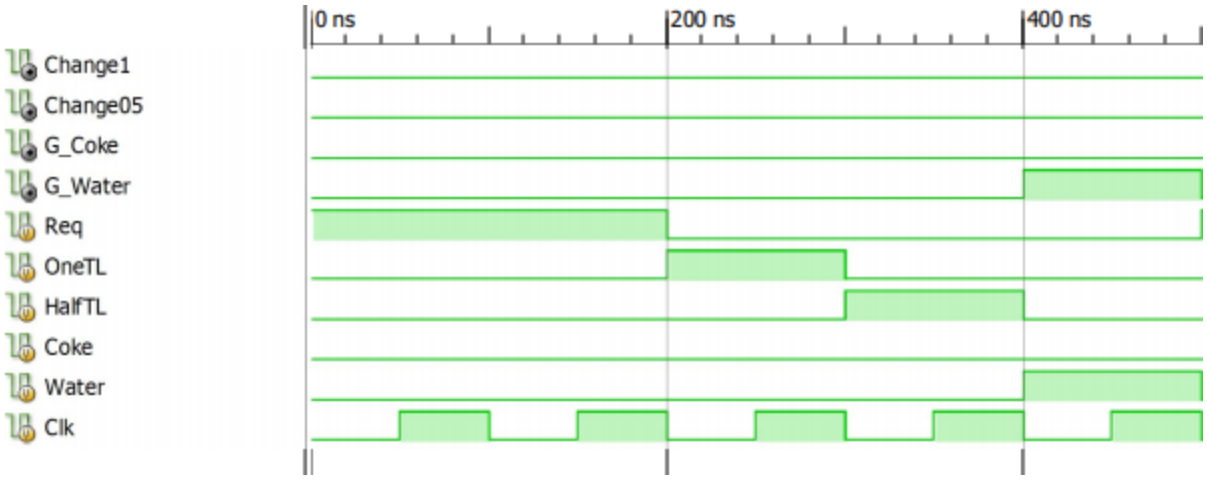
-0,5 ₺ Water



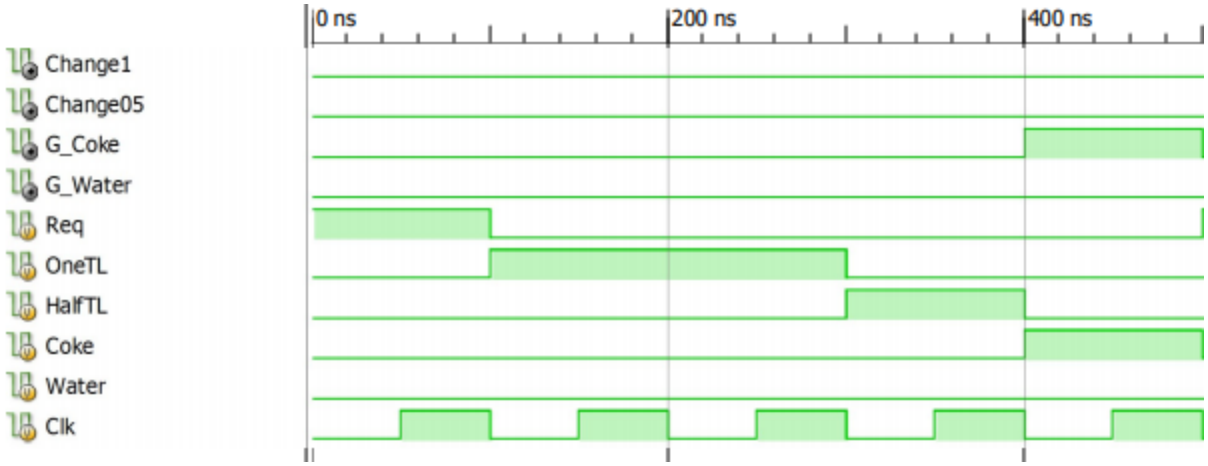
-1 ₺ Coke



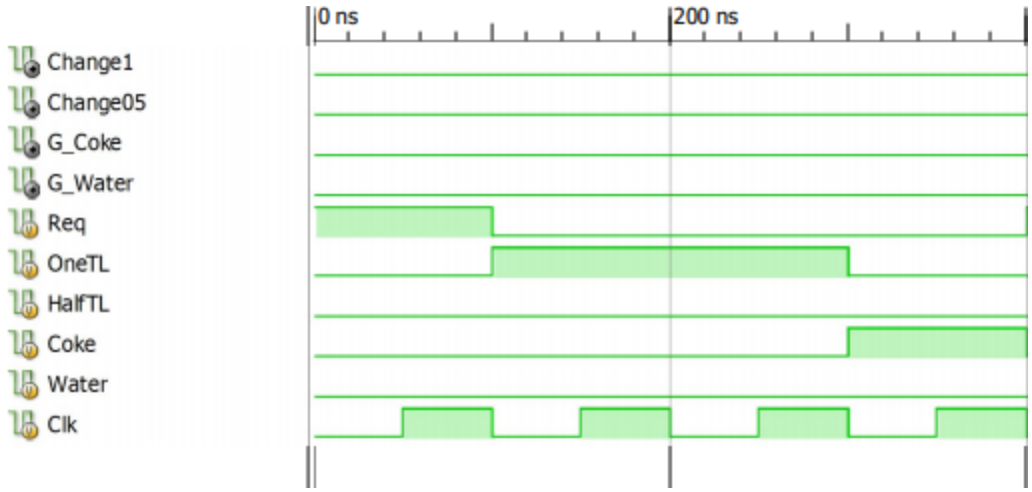
-1,5 ₺ Water



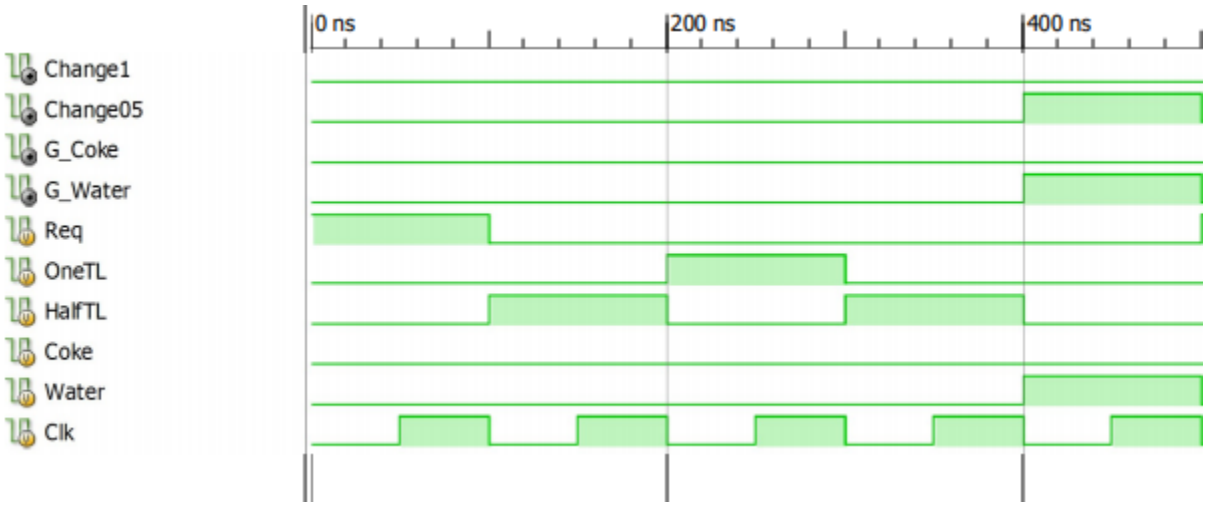
-2,5 ₺ Coke



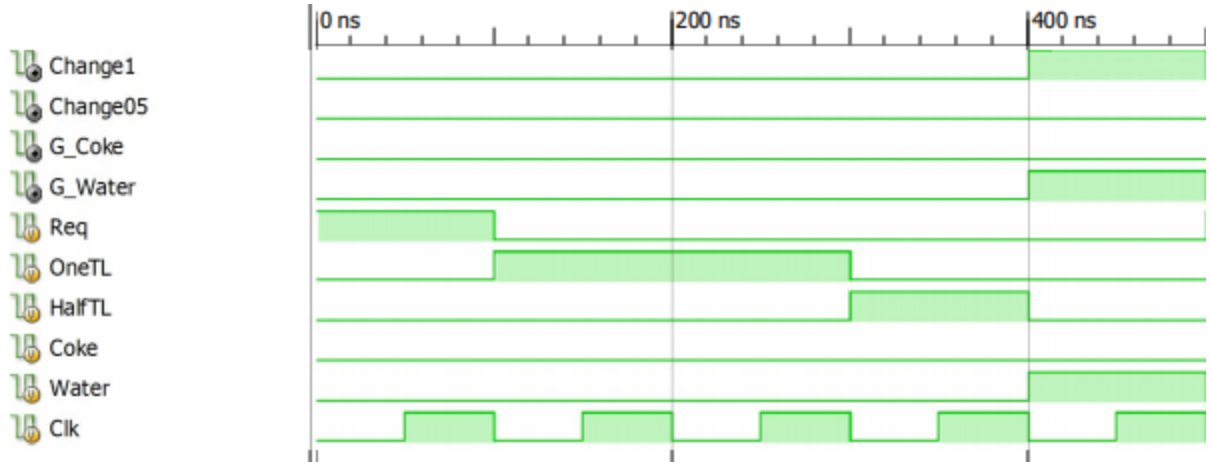
-2 ₺ Coke



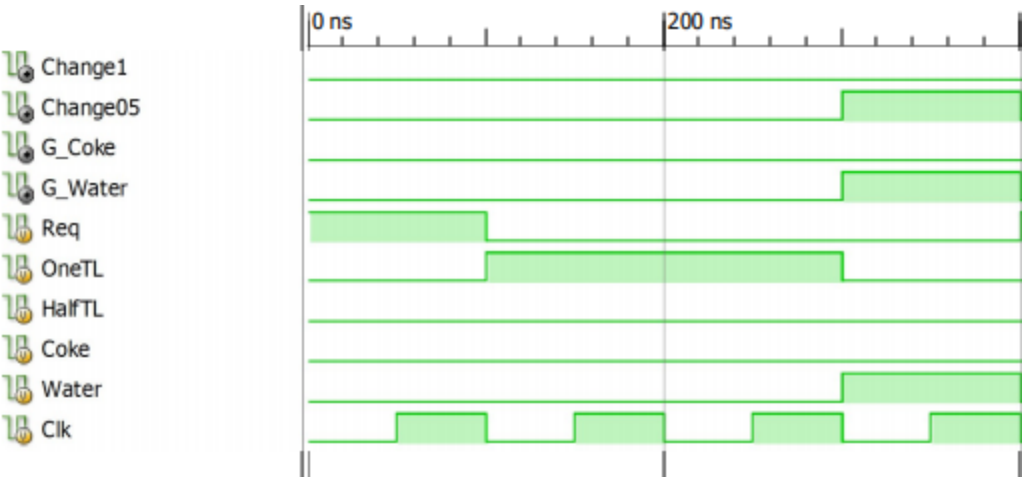
-2 ₺ Water



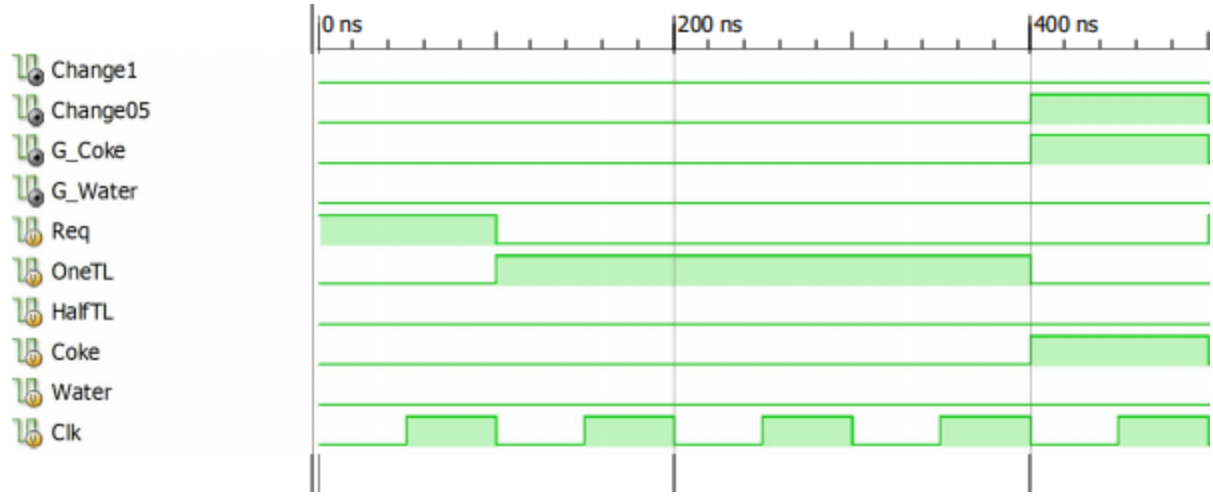
-2,5 ₺ Water



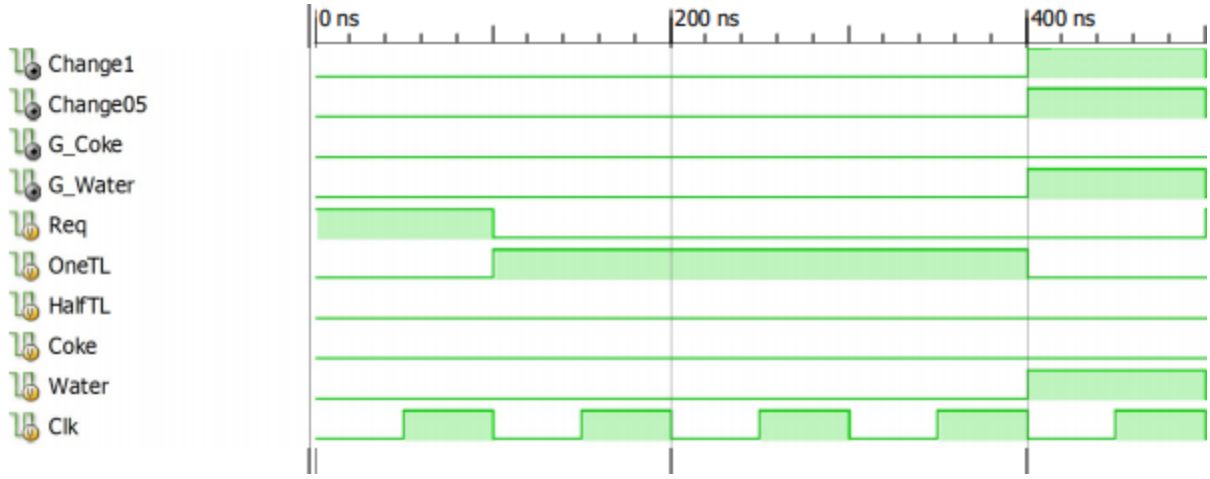
-2 ₺ Water



-3 ₺ Coke

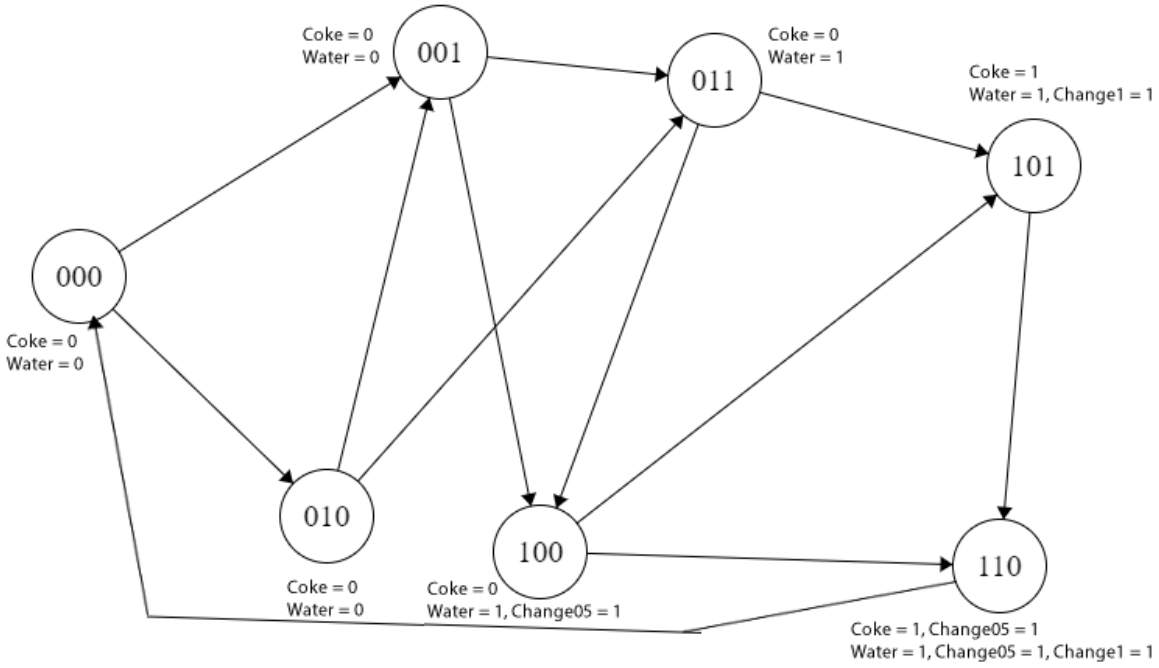


-3 ₺ Water



-State Diagram

000 = 0, 001 = 1, 010 = 0.5, 011 = 1.5, 100 = 2, 101 = 2.5, 110 = 3



-Verilog Codes

```
module VendingMachine(Req,OneTL,HalfTL,Change1,Change05,Coke,Water,Clk,G_Coke,G_Water);
input Req,OneTL,HalfTL,Coke,Water,Clk;
output G_Coke,G_Water,Change1,Change05;
reg [2:0] Curr;
reg [2:0] Next;
reg G_Coke,G_Water,Change1,Change05;
always @(posedge Clk,posedge Req)
	if( Req ) begin
		Curr <= 3'b000;
		Change05 <= 0;
		Change1 <= 0;
		G_Coke <= 0;
		G_Water <= 0;
	end else begin
		Curr <= Next;
	end
always @(OneTL or HalfTL or Coke or Water or Curr)begin
	Next <= 3'b000;
	case (Curr)
	3'b000:begin
		if(OneTL)begin
			Next <= 3'b001;
		end
		else if (HalfTL)begin
			Next <= 3'b010;
		end
	end
	3'b001:begin
		if(OneTL)begin
			Next <= 3'b100;
		end
```

```
            else if (HalfTL)begin
                        Next <= 3'b011;
            end
end
3'b010:begin
            if(OneTL)begin
                        Next <= 3'b011;
            end
            else if (HalfTL)begin
                        Next <= 3'b001;
            end
end
3'b011:begin
            if(OneTL)begin
                        Next <= 3'b110;
            end
            else if (HalfTL)begin
                        Next <= 3'b100;
            end
            else if (Water)begin
                        Next <= 3'b000;
                        G_Water <= 1;
            end
end
3'b100:begin
            if(OneTL)begin
                        Next <= 3'b110;
            end
            else if (HalfTL)begin
                        Next <= 3'b101;
            end
            else if (Water)begin
```

```
                Next <= 3'b000;
                G_Water <= 1;
                Change05 <= 1;
        end
end
3'b101:begin
        if (HalfTL)begin
                Next <= 3'b110;
        end
        else if(Water)begin
                Next <= 3'b000;
                G_Water <= 1;
                Change1 <= 1;
        end
        else if(Coke)begin
                Next <= 3'b000;
                G_Coke <= 1;
        end
end
3'b110:begin
        if(Water)begin
                Next <= 3'b000;
                G_Water <= 1;
                Change1 <= 1;
                Change05 <= 1;
        end
        else if(Coke)begin
                Next <= 3'b000;
                G_Coke <= 1;
                Change05 <= 1;
        end
end
```

```
        default:begin
                Next <= Next;
        end
        endcase
end
endmodule
```

-Testbench

```
module Test;

	// Inputs
	reg Req;
	reg OneTL;
	reg HalfTL;
	reg Coke;
	reg Water;
	reg Clk;

	// Outputs
	wire Change1;
	wire Change05;
	wire G_Coke;
	wire G_Water;

	// Instantiate the Unit Under Test (UUT)
	VendingMachine uut (
		.Req(Req),
		.OneTL(OneTL),
		.HalfTL(HalfTL),
		.Change1(Change1),
		.Change05(Change05),
		.Coke(Coke),
		.Water(Water),
		.Clk(Clk),
		.G_Coke(G_Coke),
		.G_Water(G_Water)
	);

	initial begin
```

```
// 0,5 Lira Water.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
Clk = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 1;
#10;
// 1 Lira Coke.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
```

```
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 1;
Water = 0;
#10;
// 2 Lira Water.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 1;
#10;
```

```
// 2 Lira Water.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 1;
#10;
// 1,5 Lira Water.
```

```
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 1;
#10;
// 2 Lira Coke.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
```

```
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 1;
Water = 0;
#10;
// 2,5 Lira Coke.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
```

```
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 1;
Water = 0;
#10;
// 2,5 Lira Water.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
```

```
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 1;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 1;
#10;
// 3 Lira Coke.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
```

```
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 0;
HalfTL = 0;
Coke = 1;
Water = 0;
#10;
// 3 Lira Water.
Req = 1;
OneTL = 0;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
#10;
Req = 0;
OneTL = 1;
HalfTL = 0;
Coke = 0;
Water = 0;
```

```
                #10;
                Req = 0;
                OneTL = 1;
                HalfTL = 0;
                Coke = 0;
                Water = 0;
                #10;
                Req = 0;
                OneTL = 0;
                HalfTL = 0;
                Coke = 0;
                Water = 1;
                #10;
                //Stop
                Req = 1;
                OneTL = 0;
                HalfTL = 0;
                Coke = 0;
                Water = 0;
                #10;
        end
  always #5 Clk=!Clk;

endmodule
```