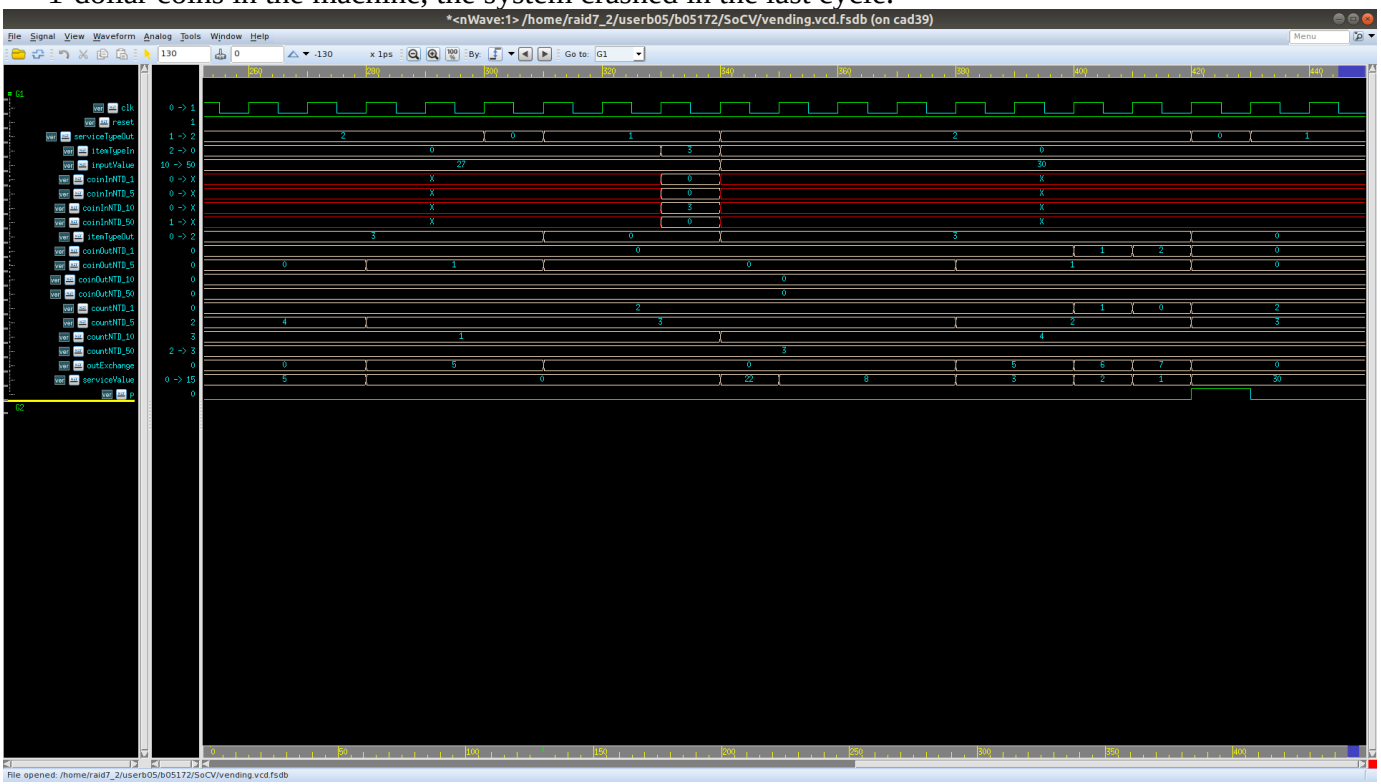
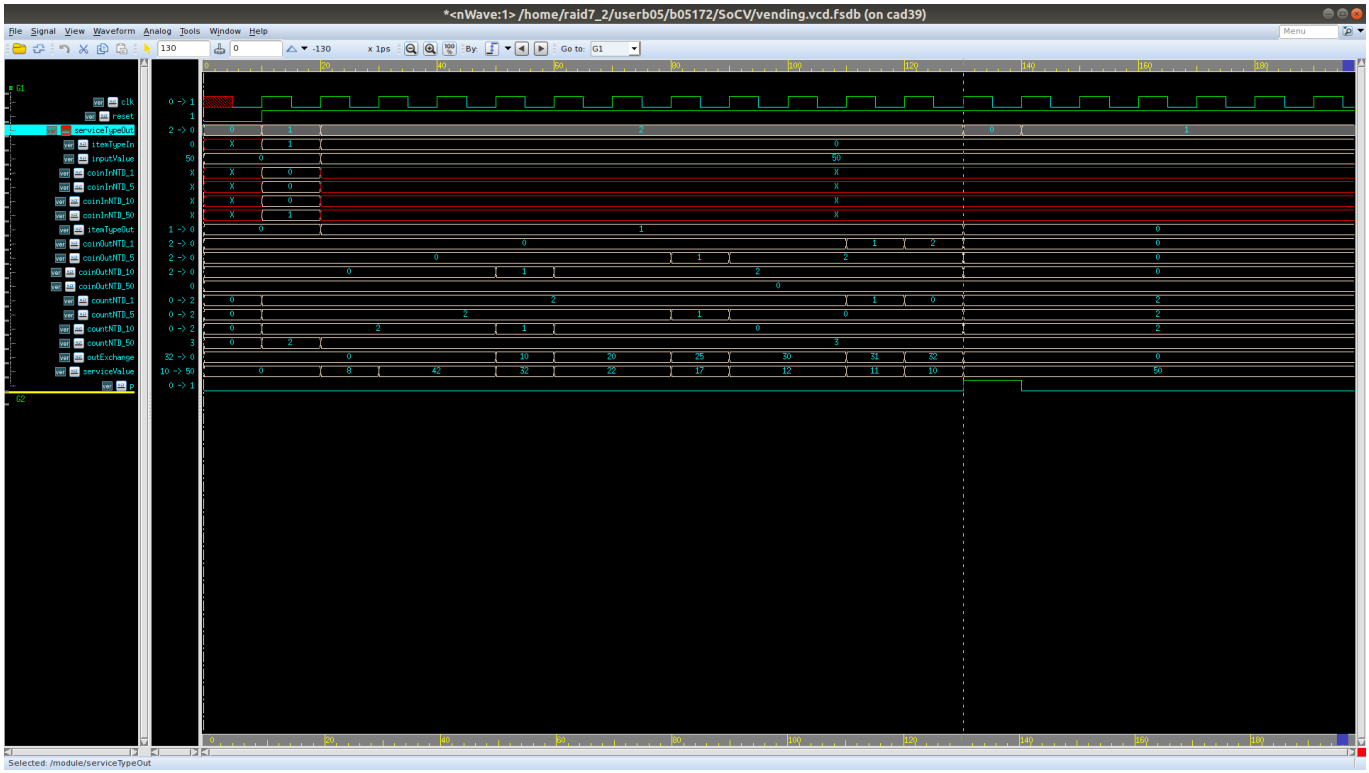
I used verdi to help verify the design, monitoring when would signal “p” rise. By observe the wave form and trace source code, I understood the verilog design architecture.
Then I tested some corner cases and found that in the original input.pattern given, when I append the pattern of requesting type 3 item ($22) with 3 10-dollar coins inserted, signal “p” would rise. Furthermore, it arise after searching for change of 1-dollar coins, finding that it is not enough coins in the machine.

Below see an example. In the last request, the vending machine should give back 8 dollars of exchange. It searched for coins of changes from 50-dollars coins, 10, 5, to 1. It was normal until searching for the 1-dollar coins. When it should find 3 1-dollar coins of change but there are only 2 1-dollar coins in the machine, the system crashed in the last cycle.



See next page.

To confirm the assumption, I simulate another pattern of requesting type 1 item ($8) with 1 50-dollars inserted, finding a similar result.



So I trace the verilog code, finding that the code discribing such condition is in vending.v: 224. The right behavior should be that it found that there are not enough changes to give, so it gave back corresponding amount of money user had inserted, searching for the refunding from 50-dollar to 1-dollar coin again. That is to say, set “serviceCoinType” to “NTD_50”. Under such circumstances, “itemTypeOut” should be “ITEM_NONE” representing that there are no item given. The above feature had already been done in line 227 and 226. Therefore, there is only one bug in the code, line 236 should be SERVECE_BUSY instead of SERVICE_OFF, meaning that it is busy finding the refund rather than finishing finding changes.

```
222             `NTD_1  : begin
223               if (serviceValue >= `VALUE_NTD_1) begin
224                 if (countNTD_1 == 3'd0) begin
225                   serviceValue_w    = inputValue;
226                   itemTypeOut_w     = `ITEM_NONE;
227                   serviceCoinType_w = `NTD_50;
228                   countNTD_50_w     = countNTD_50 + coinOutNTD_50;
229                   countNTD_10_w     = countNTD_10 + coinOutNTD_10;
230                   countNTD_5_w      = countNTD_5 + coinOutNTD_5;
231                   countNTD_1_w      = countNTD_1 + coinOutNTD_1;
232                   coinOutNTD_50_w   = 3'd0;
233                   coinOutNTD_10_w   = 3'd0;
234                   coinOutNTD_5_w    = 3'd0;
235                   coinOutNTD_1_w    = 3'd0;
236                   serviceTypeOut_w  = `SERVICE_OFF;
```