Here we add 3 more monitor a, b, c to see whether our vending machine make change correctly when Item A, B, C sold respectively. Below is the definition of monitor.

```
assign a = initialized && (serviceTypeOut == `SERVICE_OFF) &&
(itemTypeOut == `ITEM_A) && (outExchange != (inputValue - `COST_A));//
check the exchange for item A
assign b = initialized && (serviceTypeOut == `SERVICE_OFF) &&
(itemTypeOut == `ITEM_B) && (outExchange != (inputValue - `COST_B));//
check the exchange for item B
assign c = initialized && (serviceTypeOut == `SERVICE_OFF) &&
(itemTypeOut == `ITEM_C) && (outExchange != (inputValue - `COST_C));//
check the exchange for item C
```

As long as the machine does not drop wrong item, we can make sure the vending machine would not have problem in most case.

For the test pattern, we run an 100000 cycles simulation with input randomly generated by python. Amount of these, around 6900 input are accepted, and none of them reveal a bug. From this result we can trust our design have no bug to some extend.