In addition to monitor “p”, I added two monitors “eat” and “balance” to guard the potential bugs. Here’s the description of the monitors:

1. eat: This monitor check if the vending machine eats customers’ money. After the vending machine finish calculation (BUSY state), it’ll enter OFF state. When the machine is in OFF state, check if the output changes plus the value of output item equals to the initial input coin value. For example, for item A (8 NTD), if I initially thrown 10 NTD into the vending machine, it should output item A and 2 NTD eventually, which means “10 = 8 + 2”. If the output changes plus the value of output item doesn’t equal to the initial input coin value, the monitor should detect and report the bug.

2. balance: This monitor check if the changes stored in the vending machine works properly. For example, if the vending machine initially has 132 NTD, after selling one item A (8 NTD), it should have 140 NTD stored in the vending machine. This monitor check if “140 = 132 + 8”. If the vending machine fails to manage the changes inside itself, the monitor should detect and report the bug. In addition, since the stored coin numbers inside vending machine is limited to 7, the machine might definitely lose coins when reaching the storing limit. Therefore, I add a 1-bit limit detection in this monitor to ensure if the machine lose coins due to storing limit, this monitor would not detect this scenario.

The test pattern is provided in “./hw/hw1/verify/input-cex.pattern”. I’ve tested almost 1000 test pattern and none of them trigger the monitor.