

```

if (serviceValue >= `VALUE_NTD_1) begin
  if (countNTD_1 == 3'd0) begin
    serviceValue_w = inputValue;
    itemTypeOut_w = `ITEM_NONE;
    serviceCoinType_w = `NTD_50;
    countNTD_50_w = countNTD_50 + coinOutNTD_50;
    countNTD_10_w = countNTD_10 + coinOutNTD_10;
    countNTD_5_w = countNTD_5 + coinOutNTD_5;
    countNTD_1_w = countNTD_1 + coinOutNTD_1;
    coinOutNTD_50_w = 3'd0;
    coinOutNTD_10_w = 3'd0;
    coinOutNTD_5_w = 3'd0;
    coinOutNTD_1_w = 3'd0;
    serviceTypeOut_w = `SERVICE_BUSY;
  end else begin
    coinOutNTD_1_w = coinOutNTD_1 + 3'd1;
    countNTD_1_w = countNTD_1 - 3'd1;
    serviceValue_w = serviceValue - `VALUE_NTD_1;
  end
end else begin // exchangeable
  serviceTypeOut_w = `SERVICE_OFF;
end
end

```

當販賣機的零錢不足以至於交易失敗時，程式會執行到 `if (countNTD_1 == 3'd0) begin` 這個迴圈。理論上應該要讓 `coinOutNTD_w(50,10,5,1)` 全部歸零，接著把原先的 `inputValue` 當成 `serviceValue`，然後重新計算要退多少錢。但是原先迴圈最後卻是跳到 `SERVICE_OFF` State，這樣變成直接輸出結果，當然會產生 bug。如果要重新計算的話，迴圈最後要跳到 `SERVICE_BUSY` State 才對，這樣才會重新計算找零。因此調整迴圈內的 `serviceTypeOut_w` 為 `SERVICE_BUSY` 後即可解決此 bug。