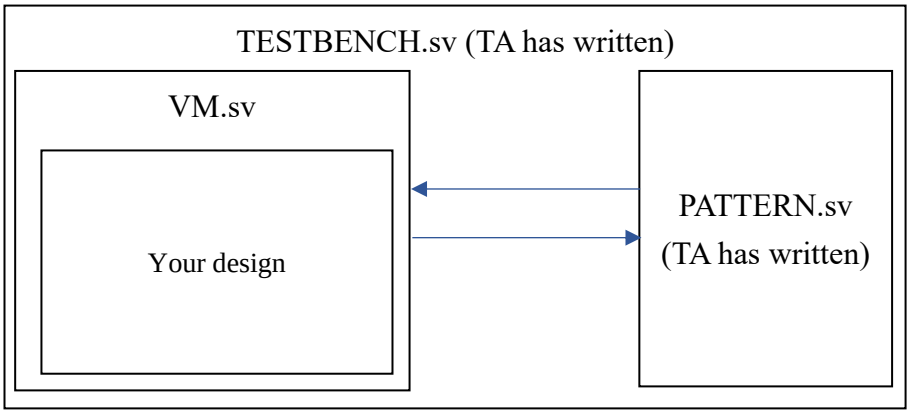
NYCU-ECE DCS-2022

HW03 Design: Vending Machine (FSM)

資料準備

1. 從 TA 目錄資料夾解壓縮:
   % tar -xvf ~dcsta01/ HW03.tar
2. 解壓縮資料夾 hw01 包含以下:
   a. 00_TESTBED/
   b. 01_RTL/
   c. 02_SYN/
   d. 03_GATE/
   e. 09_UPLOAD/

Block Diagram



設計描述

這次作業要求設計一台販賣機。

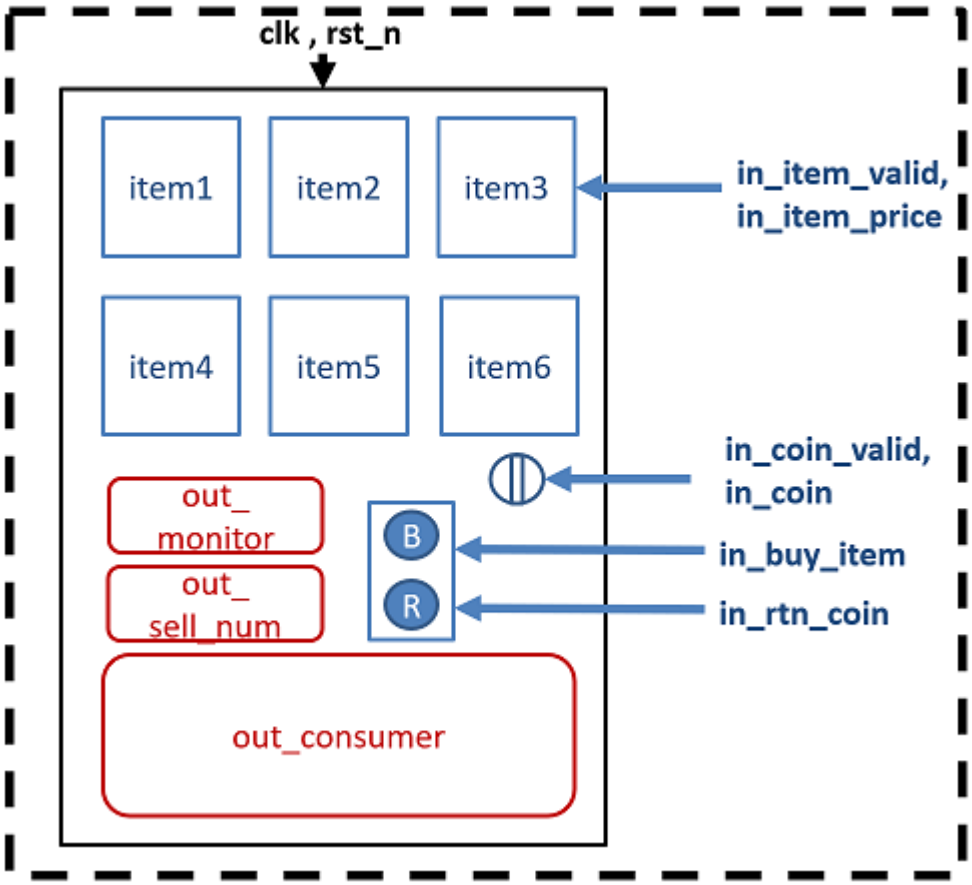
這台販賣機販售6種不同商品，Pattern在in_item_valid 為 1時，依序輸入in_item_price將商品1-6定價，in_item_valid 降下時，即代表定價完成。

定價之後，Pattern在in_coin_valid為1時，依序輸入in_coin代表投幣，並且會在out_monitor上顯示目前金額。in_coin_valid降下時，即代表投幣完成。

投幣後，Pattern會給予欲購買的物品編號(in_buy_item)或選擇退幣(in_rtn_coin)，最後商品與退幣由out_consumer表示，依次序輸出商品1-6的代表號、50元的數量、20元的數量、10元的數量、5元的數量、1元的數量

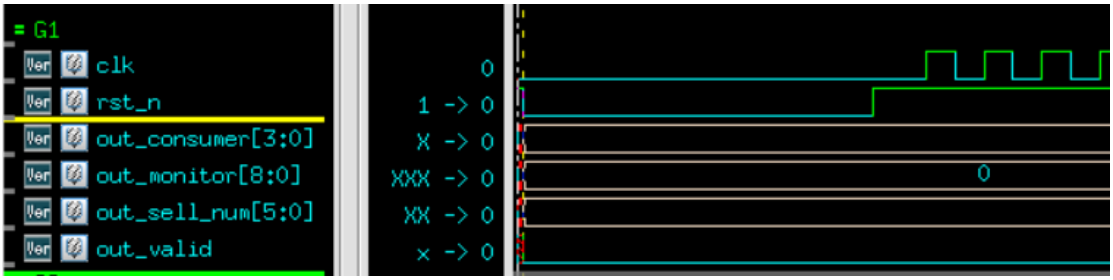
而out_sell_num則表示販賣機商品1-6累積賣出的數量(在in_item_valid為1時會歸零重新計算，因為是新的一批商品)

示意圖如下，藍色箭頭為 pattern 給予 design 的 input、紅色為 design 的 output。

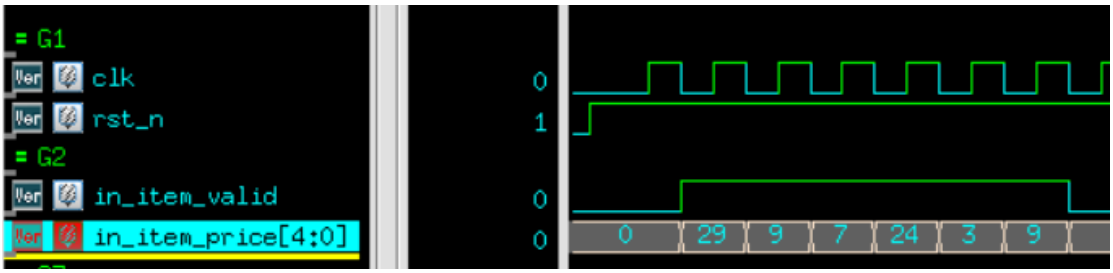


本次販賣機有以下規則：

1. 在 rst_n 之後，所有 output signal 必須為 0。

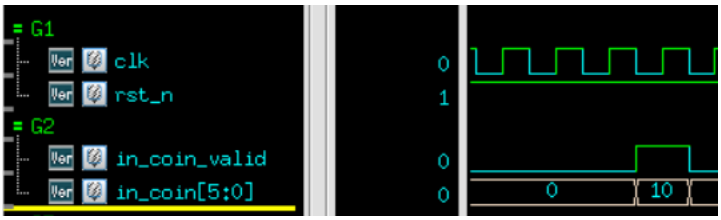


2. 在 in_item_valid 為 1 時，Pattern 依序輸入 in_item_price，代表為商品 1-6 定價。(共 6 個 cycle)

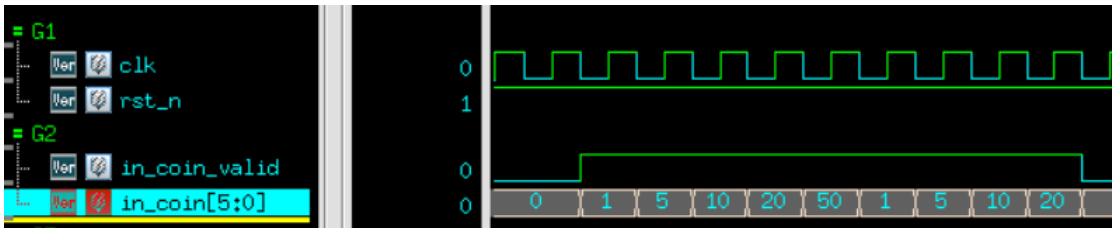


3. 在 in_coin_valid 為 1 時，Pattern 輸入 in_coin，代表投幣，in_coin_valid 為 1-9 cycle，代表投幣最少 1 次，最多 9 次。

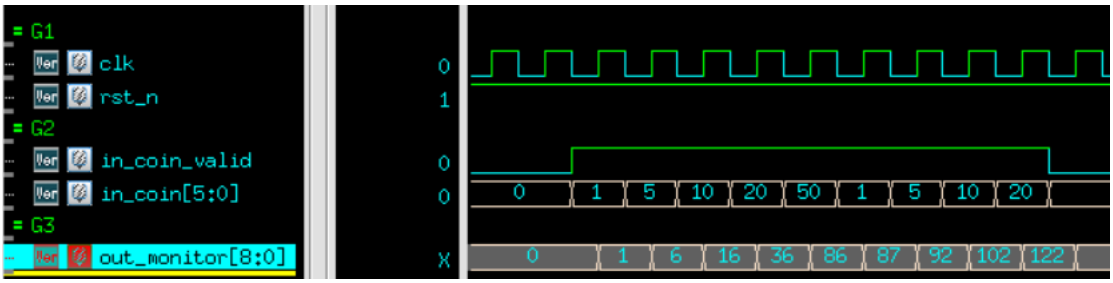
(1)投幣1次



(2)投幣9次

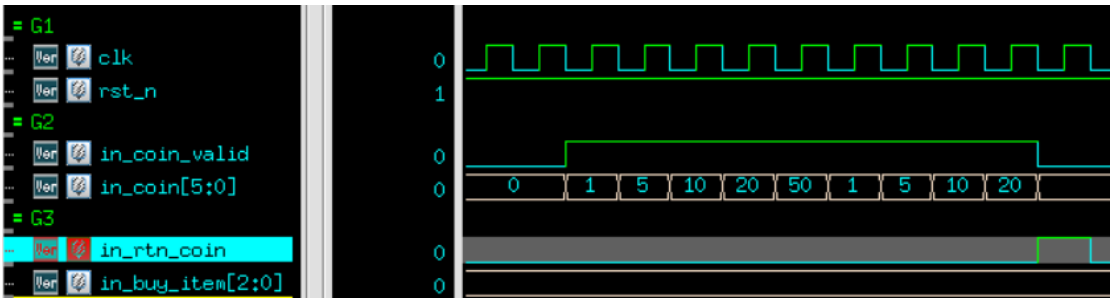


4. 每次**投幣後，out_monitor 立即顯示累加金額。 (每個 cycle 檢查且 pattern 不會使其 overflow)**

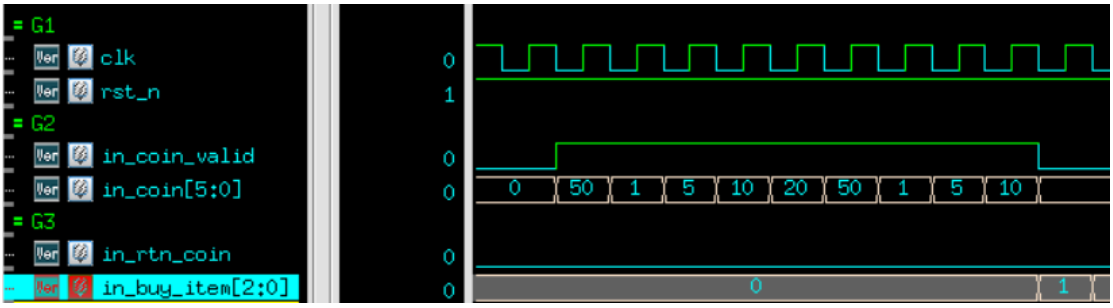


5. 在in_coin_valid落下後，pattern會**立即**給予買商品(in_buy_item) 或 退幣(in_rtn_coin)。

(1) 退幣(in_rtn_coin)

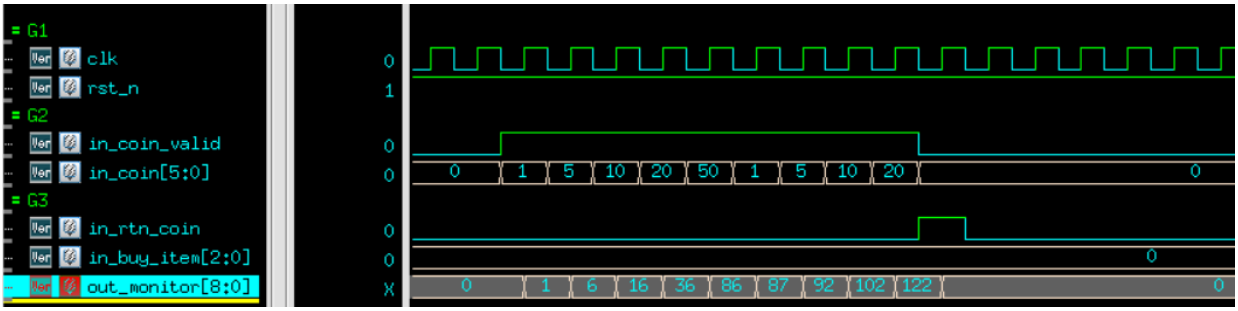


(2) 買商品(in_buy_item)

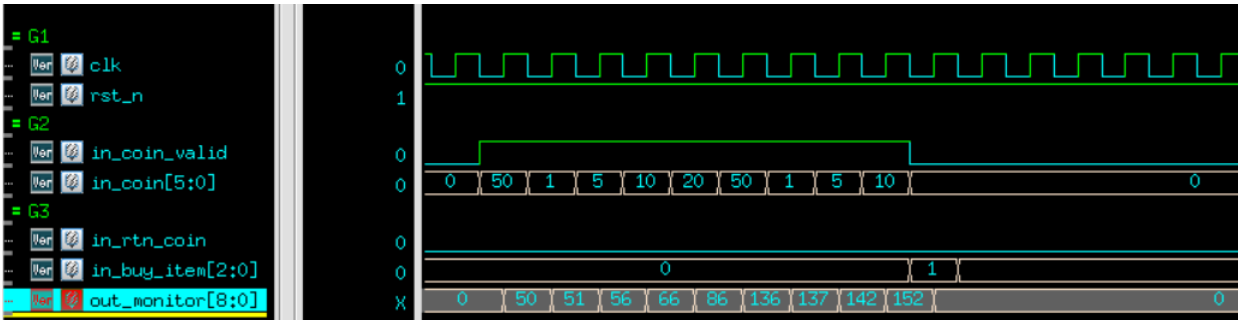


6. 選擇買商品或退幣後，out_monitor 會歸零。

(1) 退幣(in_rtn_coin)



(2) 買商品(in_buy_item)



7. 當按下的商品(in_buy_item)，且金額足夠，輸出選擇的商品號碼，退回剩下錢幣。當按下退幣(in_rtn_coin)，輸出商品號碼為 0，退回所有錢幣。

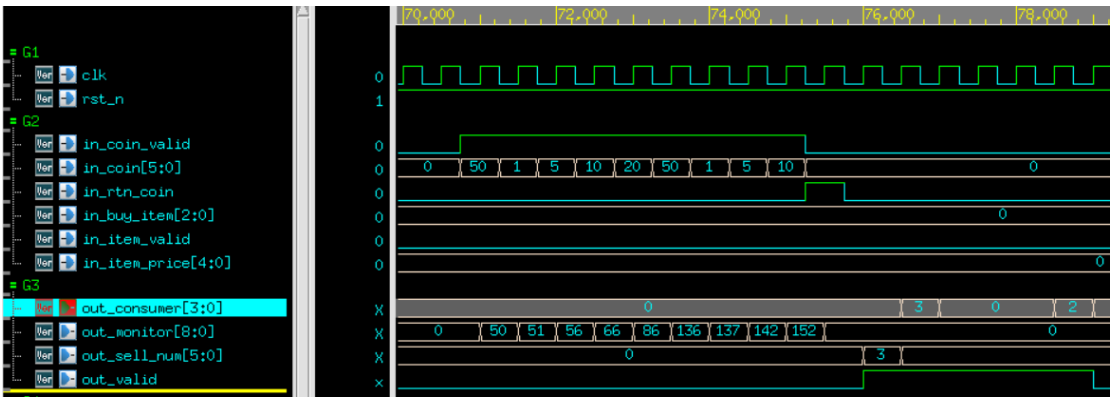
8. 機器必須out_valid拉起輸出給pattern檢查輸出

(1)out_consumer，依序第一個為商品號碼(1-6，參照備註)、第二個為50元的數量、第三個為20元的數量、第四個為10元的數量、第五個為5元的數量、第六個為1元的數量。

(i) 退幣(in_rtn_coin)

範例為共投入152元，in_rtn_coin被拉起，沒有購買商品輸出0(參照備注)，共需要退回152元，所以找3個50元、0個20元、0個10元、0個5元、2個1元，

輸出為[0,3,0,0,0,2]

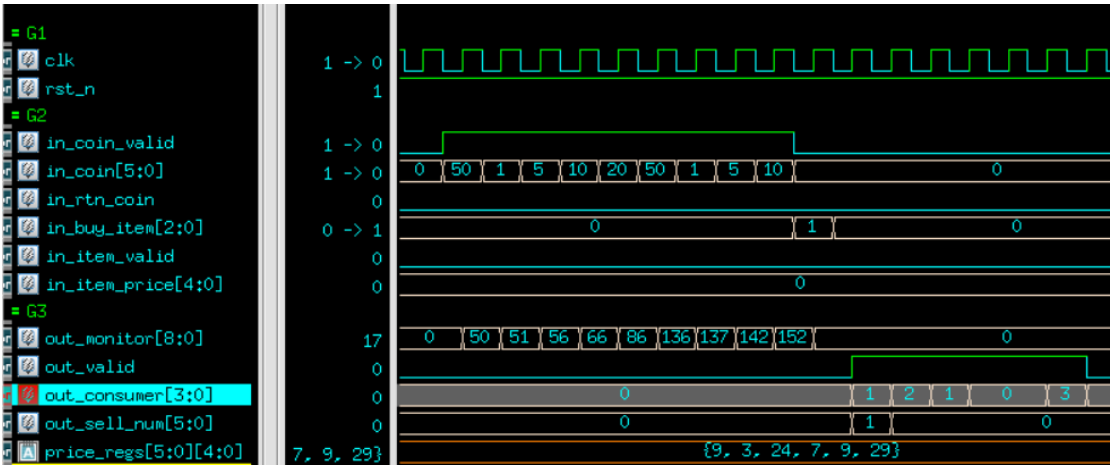


(ii) 買商品(in_buy_item)

範例為共投入152元，購買商品1需要29元，可以購買商品1，共需要找123元，

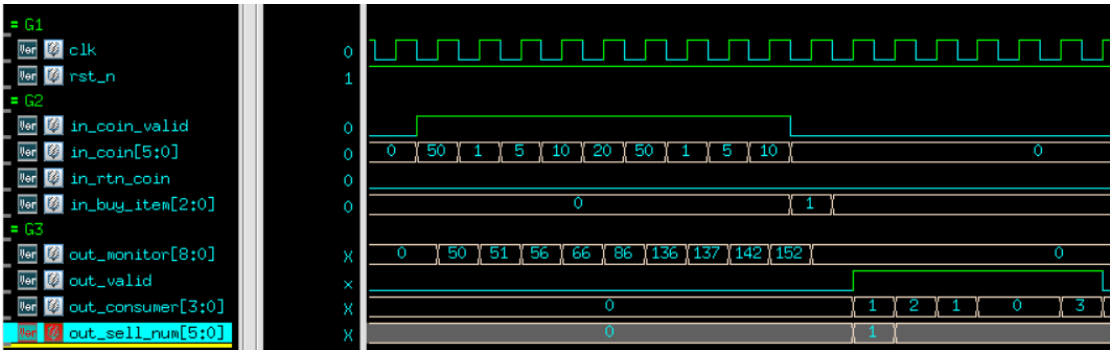
所以找2個50元、1個20元、0個10元、0個5元、3個1元，

輸出為[1,2,1,0,0,3]

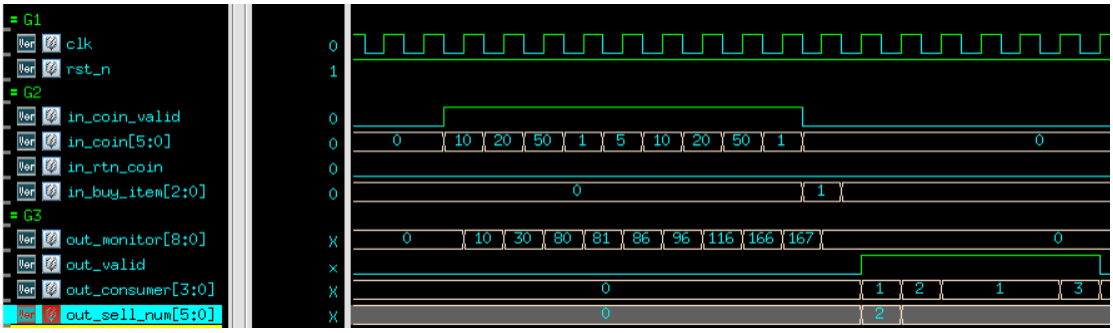


(2) out_sell_num，依序輸出商品1~6累積賣出的數量，每次in_item_valid拉起會重新歸零計算，因為是新的一批商品

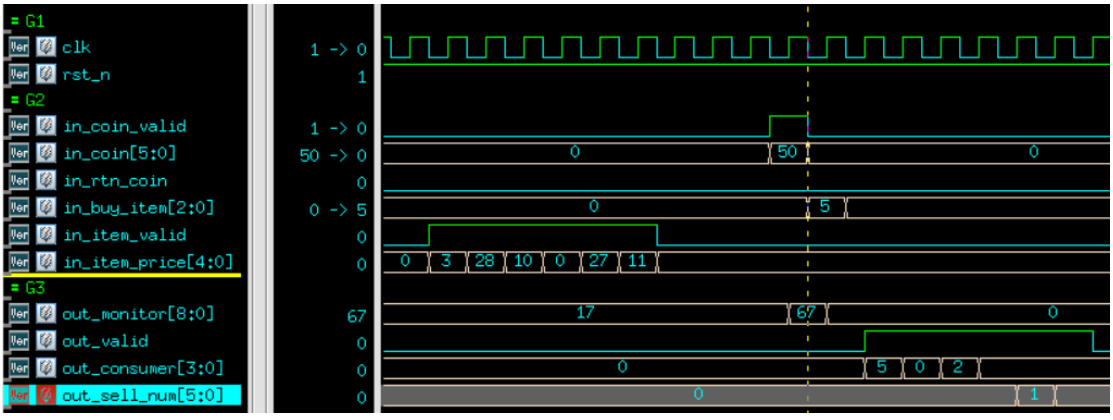
(i)第一次賣出商品1，商品賣出數量為[1,0,0,0,0,0]



(ii)第二次賣出商品1，商品賣出數量為[2,0,0,0,0,0]

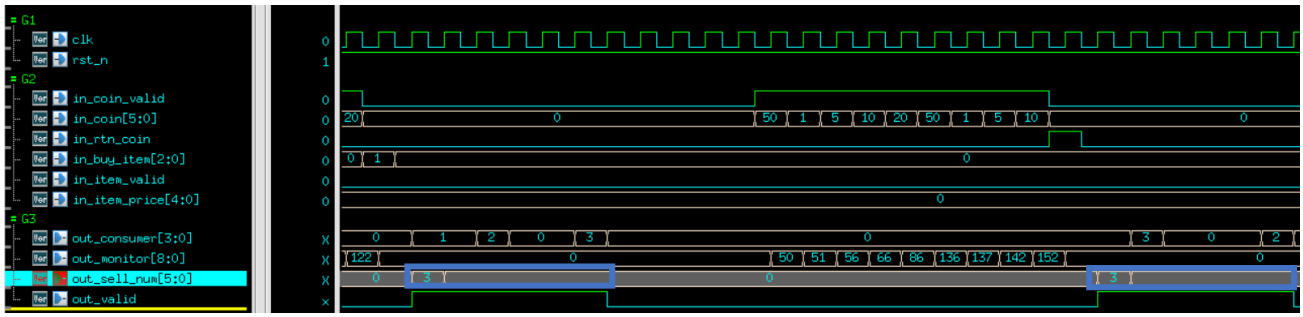


(iii) in_item_valid拉起會重新歸零計算，範例為新的一批商品，賣出商品5，所以新的一批商品賣出數量為[0,0,0,0,1,0]



(iv) 退幣(in_rtn_coin)

out_sell_num輸出這一批商品1~6累積賣出數量，因為這次沒有賣出任何商品，故輸出跟上次相同，輸出均為[3,0,0,0,0,0,](圖中藍色框框)



9. 特殊狀況：投幣結束後，out_monitor 顯示金額低於選擇商品(in_buy_item)的價格，代表無法購買，故當 out_valid 為 1 時 out_consumer 輸出為六個 0，
out_sell_num 輸出為這一批商品累積賣的數量。
out_monitor 顯示會維持到下一次投幣 in_coin_valid 為 1 時的 in_coin，且會繼續累加其金額。

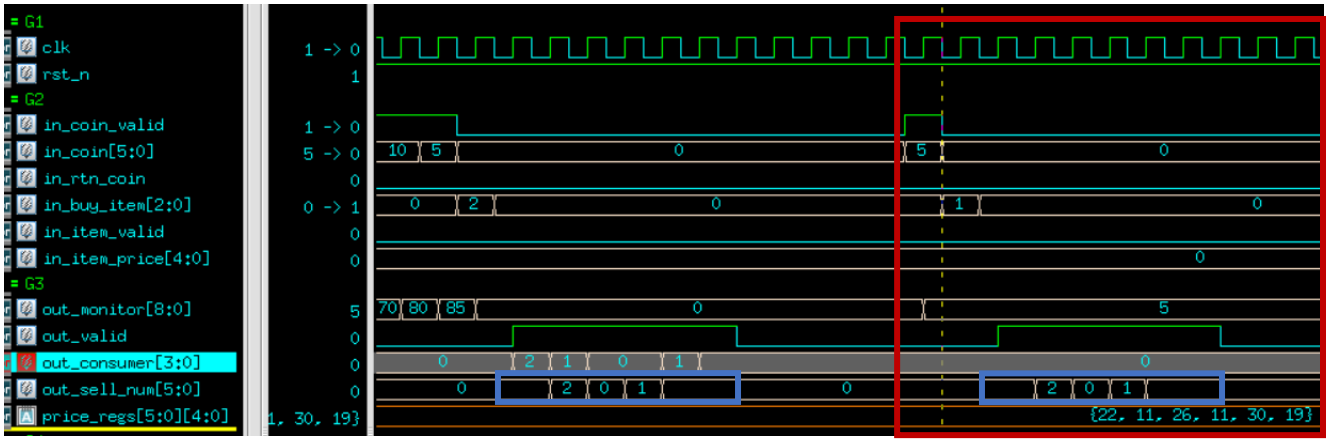
(1)無法購買

範例為僅投入5元，想要購買商品1，因為商品1需要19元，無法購買

out_monitor保持5元，等待下次投錢繼續累加

out_consumer輸出6個cycle的0

out_sell_num輸出這一批商品1~6累積賣出數量，因為這次沒有賣出任何商品，故輸出跟上次相同，輸出均為[0,2,0,1,0,0,](圖中藍色框框)



10. 備註：

Btn_buy_item	定義
0	沒有買到
1	商品 1
2	商品 2
3	商品 3
4	商品 4
5	商品 5
6	商品 6

Coin 只有 5 種 50 元、20 元、10 元、5 元、1 元。

Inputs

Signal name	Number of bit	Description
clk	1	clock
rst_n	1	Asynchronous active-low reset
in_coin_valid	1	When getting high means state in_coin
in_coin	6	投幣訊號(1-9 cycle)
in_item_valid	1	When getting high means state in_item_price
in_item_price	5	定價訊號，共 6cycle， 依序為商品 1~6 定價。
in_rtn_coin	1	退幣訊號
in_buy_item	3	購買商品訊號，1-6 代表商品 1~6 ，參照備註

Outputs

Signal name	Number of bit	Description
out_monitor	9	顯示金額，每次 in_coin 後累加， 購買商品或退幣之後顯示 0， 但如果低於選擇商品價格，則保持訊號至 下一次 in_coin 繼續累加。 ***※每個 cycle 都會檢查***
out_valid	1	必須在 in_coin_valid 落下後 100cycle 內拉 起，out_valid 持續 6 個 cycle。
out_consumer	4	依序輸出商品號碼(參照附錄)、50 元數量、 20 元數量、10 元數量、5 元數量、1 元數 量，共 6 個 cycle，6 筆資料。
out_sell_num	6	依序輸出這一批商品 1~6 累積賣出的數量 每次 in_item_valid 拉起會重新歸零計算， 因為是新的一批商品， ，共 6 個 cycle，6 筆資料。

Specifications

1. Top module name: **VM**(File name : **VM.sv**)
2. 在非同步負準位 reset 後，所有的 output 訊號必須全部歸零。
3. Output 要在 Input 結束後的 100 cycles 內輸出。
4. Output 要輸出連續 6 cycles，不能多不能少。
5. 02_SYN result 不行有 error 且不能有任何 latch。
6. Clock period 5 ns。
7. Input delay = 0.5 * clock period; Output delay = 0.5 * clock period
8. Separate your combinational and sequential blocks!

Example waveform

參考上面說明

上傳檔案

1. Code使用09_upload上傳。
2. report_dcsxxx.pdf, xxx is your server account. 上傳至new E3。
3. 1demo請在 4/14 (四)15:30 am 上課之前上傳
2demo請在4/21 (四)15:30 am 上課之前上傳

Grading policy

1. Pass the RTL& Synthesis & Gate-level simulation. 70%
2. Performance 20%
   Ranking formula: (total latency + number of patterns) * area
3. Report 10%
4. Combinational、Sequential Logic 沒有分開寫 -10%

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) → **./01_run**
2. 02_SYN/ (synthesis) → **./01_run_dc**
3. 03_GATE/ (gate-level simulation) → **./01_run**
4. 09_UPLOAD/ (upload) → **./09_upload**

報告請簡單且重點撰寫，不超過兩頁A4，並包括以下內容

1. 描述你的設計方法，包含但不限於如何加速(減少critical path)或降低面積。
2. 基於以上，畫出你的架構圖(Block diagram)
3. 心得報告，不侷限於此次作業，對於作業或上課內容都可以寫下。
4. 遇到的困難與如何解決。