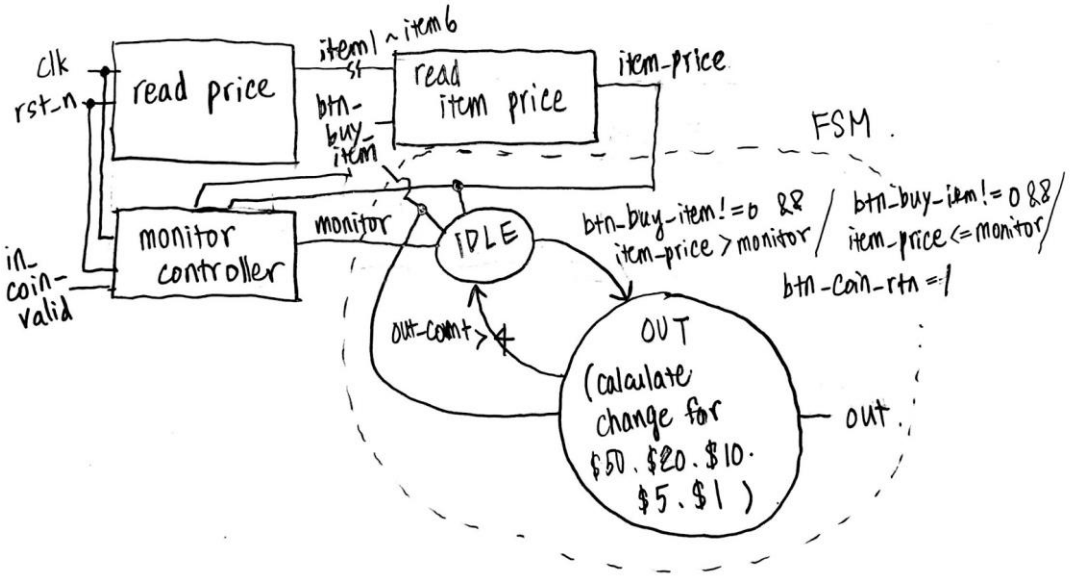
Block Diagram (with FSM implementation)

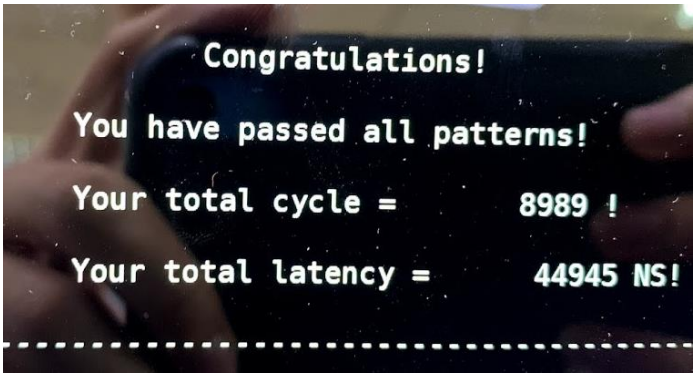


設計優化過程

版本 1:

最初的版本在計算找零錢方面幾乎全引用以前 lab05 的 code，只有另外外加讀價、判斷購買商品價格、顯示器控制等三個功能。但助教本來就有在 lab 的時候提到，那個找零錢 state 的設計是最基礎好懂的，但就會花很長的時間完成計算，而且，也十分耗面積。

時間方面看下圖就知道 latency 長的驚人，因為單單找一種零錢就要跳好多次 state，所以從 btn_buy_item 到 out_valid 就會有好幾個 cycle 的 delay，這個問題在下面的版本 2 有進行解決。



把每種零錢都存入一個 register 也花費了很多空間，$50, $20, $10, $5, $1 都要有一個 register 去存，十分地耗面積，這個問題我也有在版本 2 改良。這個設計版本的總面積如下：

```
544
545   Number of ports:                           33
546   Number of nets:                           591
547   Number of cells:                          566
548   Number of combinational cells:            499
549   Number of sequential cells:                67
550   Number of macros/black boxes:               0
551   Number of buf/inv:                        101
552   Number of references:                      53
553
554   Combinational area:               7624.108944
555   Buf/Inv area:                     1017.878436
556   Noncombinational area:            4680.244831
557   Macro/Black Box area:                0.000000
558   Net Interconnect area:      undefined  (No wire load specified)
559
560   Total cell area:                 12304.353775
```

版本 2:

這次我把運算過程用到的好幾個 state 都省略掉，只留下 IDLE 和 OUT，而計算就都在 OUT 裡進行，所以 latency 就能降為 0，因為馬上算，馬上輸出，就不會有那麼多的 delay 了!

再者，這樣即算即輸出的設計也省了很多面積，因為就不用另外的 register 去存每一個零錢找多少 (雖然這樣的設計會用到除法器，而除法器相較占空間，這個問題將再版本 3 再進行解決)，但整體而言面積有被縮小。

```
403   Number of ports:                           33
404   Number of nets:                           473
405   Number of cells:                          448
406   Number of combinational cells:            366
407   Number of sequential cells:                82
408   Number of macros/black boxes:               0
409   Number of buf/inv:                         78
410   Number of references:                      52
411
412   Combinational area:               5894.380906
413   Buf/Inv area:                      778.377628
414   Noncombinational area:            5355.504051
415   Macro/Black Box area:                0.000000
416   Net Interconnect area:      undefined  (No wire load specified)
417
418   Total cell area:                 11249.884957
419   Total area:                 undefined
420   1
```

```
--------------------------------------------

              Congratulations!

       You have passed all patterns!

       Your total cycle =          0 !

       Your total latency =          0 NS!

--------------------------------------------
```

版本 3:

在此版本，latency 一樣為 0，主要是針對面積的部分進行改良，而改良就是拿掉幾個先前提到的除法器。如下圖，就舉找$20 為例，版本 2 的零錢數目是用除法器去算要找多少個$20，但前面已經找完了$50，也就是說剩下最多$49，能給的$20 最多兩個，最少零個，共三總可能，在版本 3 用 if-else 去實現就不用用到除法器了，省下了許多面積!

```
out <= (change/20);
change <= change - 20 * (change/20);
```

版本 2

```
if(change >= 40) begin
  out <= 'd2;
  change <= change - 40;
end
else if(change >= 20) begin
  out <= 'd1;
  change <= change - 20;
end
else begin
  out <= 'd0;
  change <= change;
end
```

版本 3

```
401
402   Number of ports:                         33
403   Number of nets:                         478
404   Number of cells:                        454
405   Number of combinational cells:          387
406   Number of sequential cells:              67
407   Number of macros/black boxes:             0
408   Number of buf/inv:                       77
409   Number of references:                    52
410
411   Combinational area:               5954.256120
412   Buf/Inv area:                      768.398428
413   Noncombinational area:            4713.508823
414   Macro/Black Box area:                0.000000
415   Net Interconnect area:      undefined  (No wire load specified)
416
417   Total cell area:                 10667.764943
```