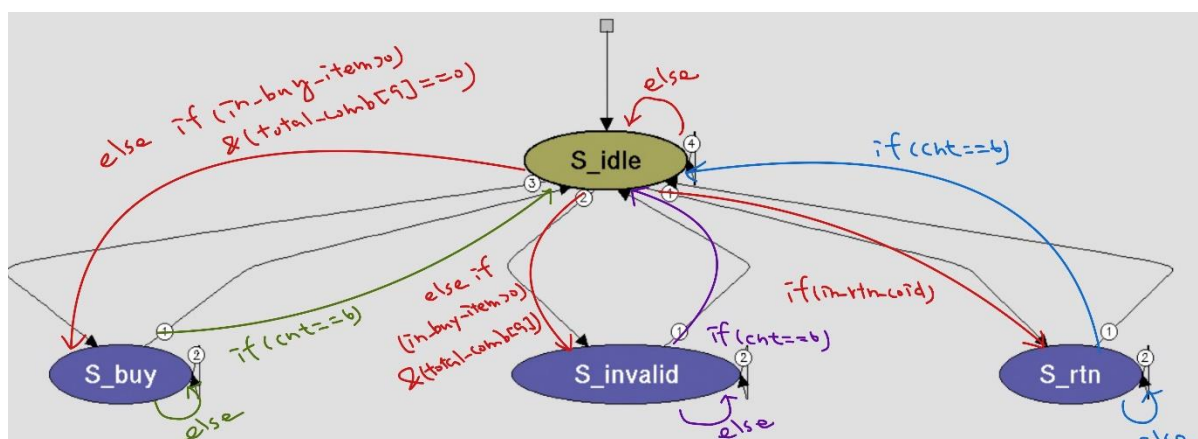


109511207 電機 13 蔡宗儒

109511207 電機 13 蔡宗儒

我做了一個 FSM，設計了四個 State，分別為 S\_idle(等待買商品 or 退幣)、S\_buy(商品購買成功)、S\_rtn(退幣)、S\_invalid(商品購買失敗)來判斷各種條件。其中我用了 item\_price\_reg 來存商品的價格，sell\_num\_reg 存商品販售的數量，並在 out\_valid 為 High 時傳給 out\_sell\_num。最後還做了一個計數器來將 out\_valid 來高 6 個 Cycle，累加器來累積 out\_monitor 的值，找零則做了一個 total\_reg 來得知 out\_monitor 扣掉商品價格為多少，並將值輸出計算零錢的分布。

### (三)FSM 設計圖



### (四) 遇到的困難與解決方法

#### 1. 如何計算找零

最一開始想到的方法是用 `total_reg` 將 `out_monitor` 扣掉商品價格的值存下來，再用 `combinational logic` 去計算  $/50$  後的商數(即為 `coin_50` 的值，我直接用一個除法器)、50 的餘數( $\text{total\_reg} - 50 * \text{coin\_50}$ )，有了 50 的餘數再去計算 20 的商數、20 的餘數，以此類推。但這麼做很浪費面積，因為需要一個除法器與許多加法器，後來想到其實可以在 `in_coin` 進來時每超過 50 就扣 50 並讓 `coin_50 + 1`，然後再一個一個 Cycle 慢慢算值，理論上這樣可以把一個除法器與多個減法器變成沒有除法器，一個減法器(因為 `coin_50`, `coin_20` ... 都在不同 Cycle 輸出，所以可以寫成 `assign a = b - c`，再寫兩個大的 MUX 去判定每個當下的 `b` 值和 `c` 值，就可以達成共用減法器)。我認為經過 `tradeoff` 之後應該是會比較小的，但實際做完卻是面積增加 1000。我認為可能的原因有兩個，第一個可能就是我不夠好，因為一開始架構訂好之後，後面再變動需要多加判斷條件和一些計算，但因為花了太多時間修架構，面積也壓不下去，於是只好繼續用原來的方法；第二原因可能是 `b` 跟 `c` 的 MUX 真的很大，比原本用除法器與多個減法器還大，但我個人認為這個機率較低。

#### 2. 為什麼 02 會 Setup Time Violation?

這個原因可能有很多，例如 `Input`、`Output` 都沒檔 DFF，中間的計算量太大導致 `Combinational Logic` 太多太長等等。而我自己一開始是遇到後者，看 `Timing Report` 發現 `Critical Path` 是從 `in_buy_item` 到 `coin_50_reg`，這中間經過了算 `out_monitor`-商品價格的運算以及  $/50$ 、算 50 餘數的計算，所以應該是  $/50$  和算 50 的餘數導致這個計算太長來不及輸出，於是我後來改成了 `totoal_reg` 將 `out_monitor`-商品價格存下來，改完之後 `Critical Path` 就變成了 `in_buy_item` 到 `total_reg`，少了  $/50$  和取 50 餘數的計算，也就不會 `Setup Time Violation` 了。然後為了 `Performerce` 跟 `Area`，我沒有將 `Input` 訊號檔一層 DFF，但我知道這並不是好的設計。

### 3. 怎麼處理不能買的狀況?

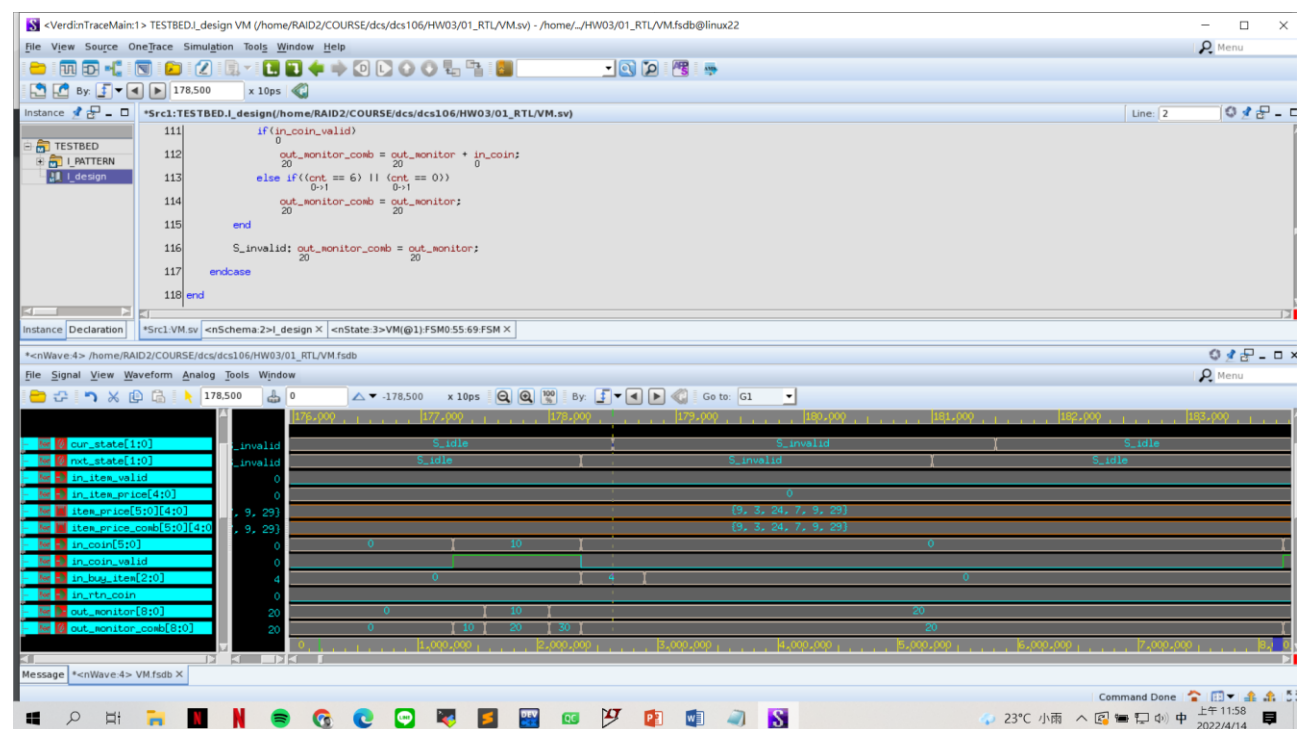
我將 total\_reg 多設了 1 個 Bit 做為 Signed Bit，這樣如果不能買時，1 就會存在 Signed Bit 之中，可以只用這個 Signed Bit 方便判斷我設計的 FSM 的 State 以及其他條件，且這樣也能不用做  $\geq$  之類的比較，也就能省下幾個比較器的面積。如下圖，total\_comb 的第 9 個 Bit 即為 Signed Bit。

```
((total_comb[9]) ? S_invalid : S_buy) : S_idle;
```

#### (五) 心得

這次的作業花了我大概 30~40 個小時去做，真的是做了非常久，甚至還是在悲慘的期中周。但其實做只花了大概 10 小時，剩下的時間都在想辦法優化並壓面積。我認為應該是有較好的演算法(遇到的困難第 1 項有提到)去處理找零錢的問題，但是我實在是太晚想到了，處理很久之後面積反而增加，於是乎就只能放棄。下次應該先好好想架構跟演算法在開始動工，不然像這次一樣使用了 FSM 後會因為架構跟狀態都寫好了，要修改會需要考慮很多條件，並有可能多做許多運算。不過最後有慢慢壓下來還是稍微有成就感的。

#### (六) Verdi(這個真的爆幹好拿來除錯)



查看第 20 筆測值 out\_monitor 是否有繼續維持 20

#### (七) Area

```
Combinational area:      7683.984113
Buf/Inv area:           765.072024
Noncombinational area:  6153.840118
Macro/Black Box area:   0.000000
Net Interconnect area:   undefined (No wire load specified)

Total cell area:         13837.824231
Total area:              undefined
```