NYCU-ECE DCS-2024

HW03

Design: DCSTCO supermarket

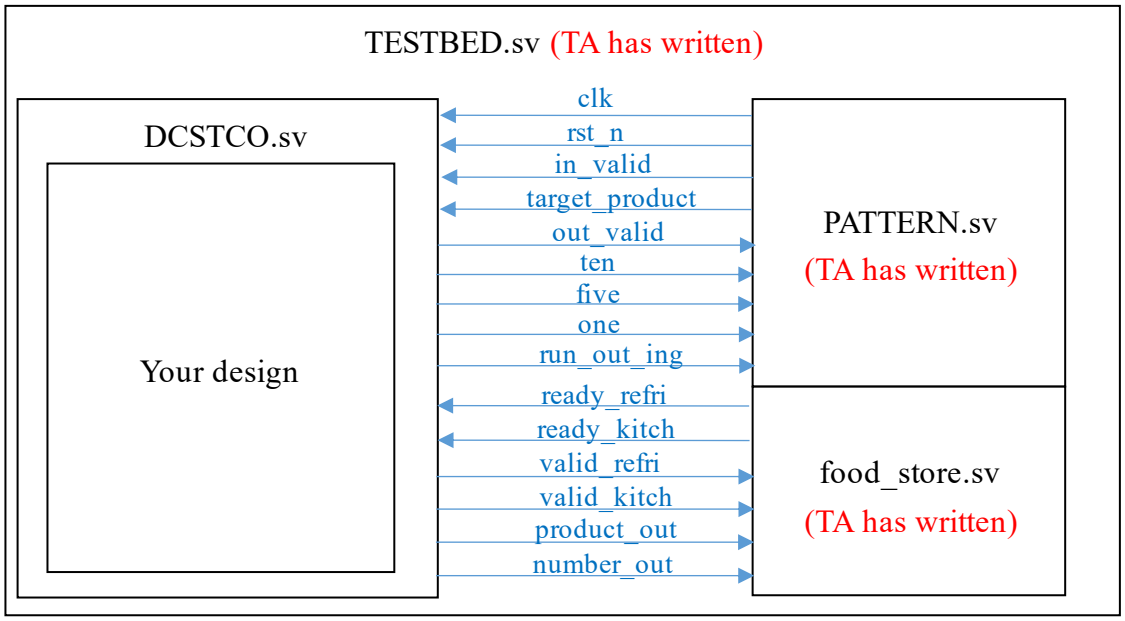
資料準備

1. 從 TA 目錄資料夾解壓縮:

   % tar -xvf ~dcsTA01/HW03.tar

2. 解壓縮資料夾 HW03 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

本次作業目標是在模擬市場販賣商品的過程。顧客 (PATTERN) 會提供自己想要購買的清單 (target_product)，接著會出現可能發生的兩種狀況。第一種狀況為店內商品 (product_register) 充足，此時需計算出顧客所購買東西的總價格向顧客收錢。第二種狀況為店內商品 (product_register) 不足， 此時需要根據熟食或冷食分別向 kitchen 或 refrigerator 去補給對應的食物到店內。補給完成之後要通知顧客食材不夠，無法購買 (run_out_ing)。以下為本次作業之介紹。

- **Input**

  輸入會給予顧客所要購買的食物清單 (12bits)，本次作業共有4樣食物可以購買，每樣食物以 3bits 去描述該向食物所要購買的數量。從 MSB 到 LSB 依序對應的食物為 nugget、fried rice、apple、peach。

  EX: Input series => 011101110001 (nugget: 3 / fried rice: 5 / apple: 6 / peach: 1)

- 第一種狀況

  店內商品 (product register) 數量全足以賣給顧客 ：

  1. 計算出顧客所要購買物品的總價 (各項價格請參考 Table.1)
  2. 根據總價格換算成 (10元、5元、1元) 的形式輸出

Table 1: Cost of products

Product	Nugget	Fried Rice	Apple	Peach
Cost	3	5	2	4

- 第二種狀況 **(只要有一樣商品不夠，就屬於此狀況)**

  店內商品 (product register) 數量不足以賣給顧客 ：

  1. 把所有不足的店內商品數量補回到50 (若食材數量剩5，就補45回去)
  2. 透過 AHB Interconnect 去對儲存在 refrigerator 和 kitchen 的食材數量進行扣掉的動作 (物品儲存的位置請參考 Table.2)
  3. 最後把 run_out_ing 拉成 1，不需要計算購買食物的總額

Table 2: Position of products

Product	Nugget	Fried Rice	Apple	Peach
Store Place	Kitchen	Kitchen	Refrigerator	Refrigerator
Address	1	0	1	0

Inputs: From Pattern.sv

Signal name	Number of bit	Description
clk	1-bit	Clock
rst_n	1-bit	Asynchronous active-low reset
in_valid	1-bit	拉起時代表 target_product is valid
target_product	12-bits	顧客所要購買的食物 (食物對應 bit 數請參閱上方的設計描述)

Outputs: To Pattern.sv

Signal name	Number of bit	Description
out_valid	1-bit	拉起時輸出結果，持續 1 cycle。
ten	4-bits	顧客總消費的 10 元部分
five	1-bit	顧客總消費的 5 元部分
one	3-bits	顧客總消費的 1 元部分
run_out_ing	1-bit	商品不足，所有商品補充完畢後拉起

Inputs: From food_store.sv

Signal name	Number of bit	Description
ready_refri	1-bit	拉起時代表 refri 端可以接收資料
ready_kitch	1-bit	拉起時代表 kitch 端可以接收資料

Outputs: To food_store.sv

Signal name	Number of bit	Description
valid_refri	1-bit	拉起時表示當前的 product_out 和 number_out 為 refrigerator 的資訊
valid_kitch	1-bit	拉起時表示當前的 product_out 和 number_out 為 kitchen 的資訊
product_out	1-bit	要補充商品的 address (Address in Table.2)
number_out	6-bits	要補充商品的數量

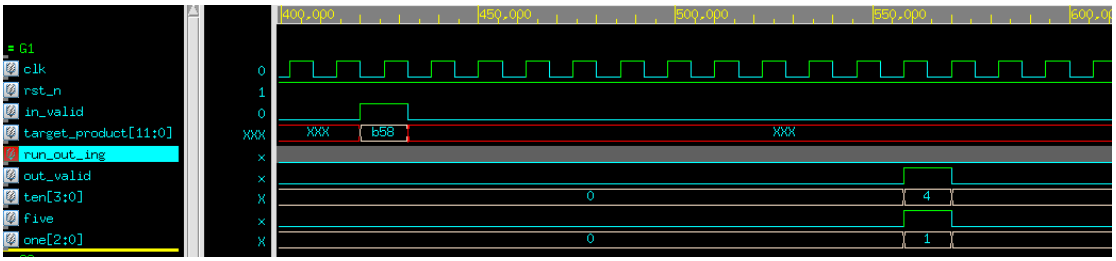
Specifications

1. Top module name: **DCSTCO** (File name: **DCSTCO.sv**)
2. Please use SystemVerilog to finish your homework
3. After asynchronous reset，all output signals should be reset to 0
   (only all output signals to pattern.sv)
4. Your latency should be less than 100 cycles for each pattern
5. The next input pattern will come in 1~4 cycles after out_valid of this pattern is pulled down
6. Ready signal in Food_store.sv will have random 5~10 cycles interval
7. Output can only be high for 1 cycle and your output answers must be correct
8. in_valid and out_valid cannot be overlap
9. All output signals have to be reset to 0 when out_valid is low
10. Input delay = 0.5 * clock period; Output delay = 0.5 * clock period
11. 02_SYN result cannot include any errors、latches and slack must be MET and non-negative
12. 03_GATE cannot include any timing violations
13. 03_GATE Latency must be the same with 01_RTL
14. After asynchronous reset, you must reset your product register to zero
    (These registers are used to record the rest of products in shop)

```
//---------------------------------------------------------------------
//   DON'T MODIFIED THE REGISTER'S NAME (PRODUCT REGISTER)
//---------------------------------------------------------------------
logic [6:0] nugget_in_shop, fried_rice_in_shop ;
logic [6:0] apple_in_shop , peach_in_shop ;
//---------------------------------------------------------------------
```

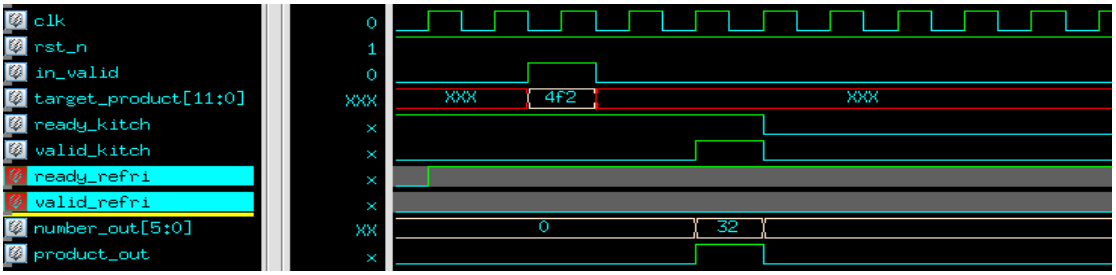
Waveform Example

1. 店內物品充足時，請輸出購買物品的總價格 (將價格以10, 5, 1單位輸出)

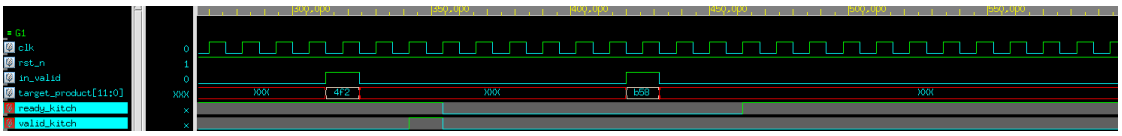


Waveform Example

2. 店內物品不足時 (nugget數量不足，valid_kitch和ready_kitch handshake 後 pull low)



3. Ready signal 間隔 5~10 cycles 才會再拉成1



Upload File

1. Code 請使用 09_UPLOAD/01_upload 上傳
2. Demo1 請在 4/18 23:59 之前上傳，Demo2 請在 4/25 23:59 之前上傳
3. Report_dcsXX.pdf, XX is your server account. 上傳至 NewE3
4. 請在 4/25 23:59 之前上傳

Grading Policy

1. Pass the RTL & Synthesis & Gate simulation. 70%
2. Performance (Total Latency * Area) 30%
3. Report (必繳) 0%

Note

Template folders and reference commands:

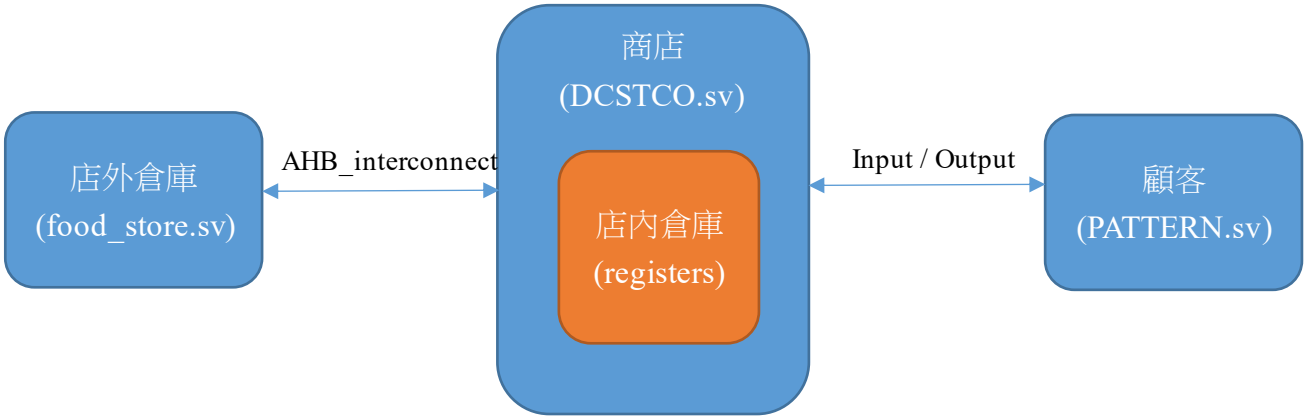
1. 01_RTL/ (RTL simulation) → **./01_run**
2. 02_SYN/ (Synthesis) → **./01_run_dc**
3. 03_GATE/ (GATE simulation) → **./01_run**
4. 09_UPLOAD/ (upload) → **./01_upload**

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。

Appendix

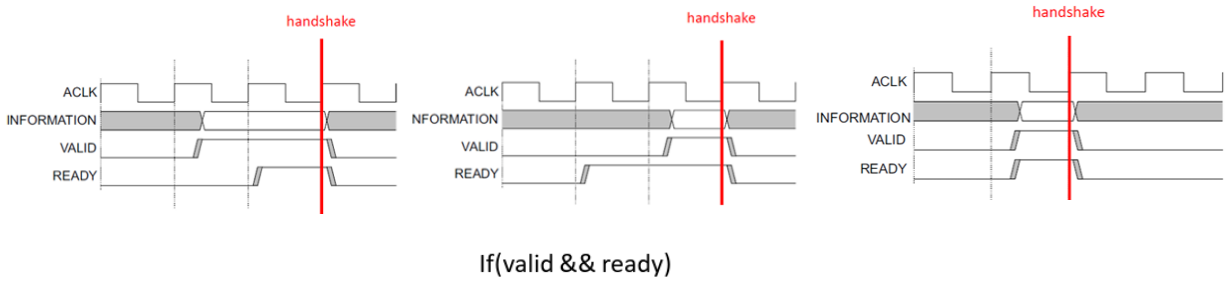
- 怕大家無法理解上面 TA 所提供的 block diagram，這邊提供一個具體的系統圖 (詳細腳位請參考上方第3頁的IO說明)



- TA 會在每10筆 pattern 過後(包含第0筆)檢查是否有確實透過 AHB 去扣除倉庫內物品的補充數量

```
if (pat_pointer % 10 === 0) begin
    if (golden_kitch[0] !== kitchen.mem[0] || golden_kitch[1] !== kitchen.mem[1] ||
        golden_refri[0] !== refrigerator.mem[0] || golden_refri[1] !== refrigerator.mem[1]) begin
        fail;
```

- AHB_interconnect補充: 須等 valid & ready 同時變為1，才算 handshake 成功 (關於AHB更詳細的說明，請參閱Lab5投影片)



If(valid && ready)