

NCTU-EE DCS-2019

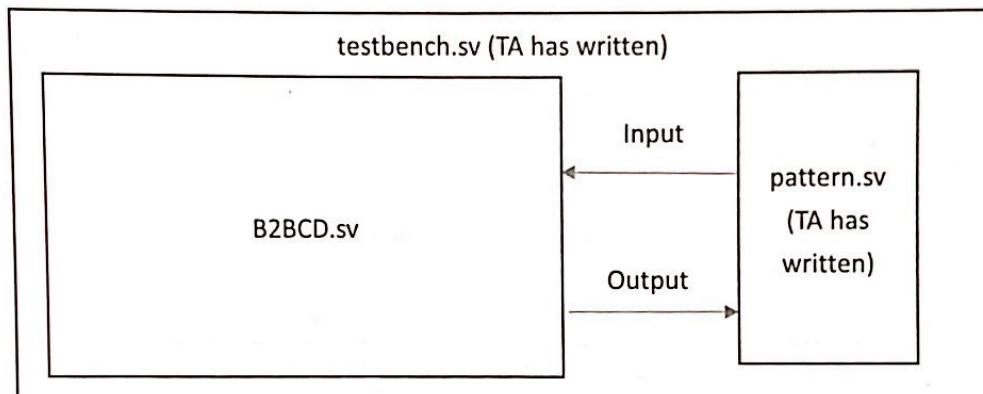
OT

Design: B2BCD

資料準備

1. 從 TA 目錄資料夾解壓縮
% tar -xvf ~dcsta01/OT.tar
2. 解壓縮資料夾 OT 包含以下：
 - A. 00_TESTBED/
 - B. 01_RTL/
 - C. 02_SYN/

Block Diagram



設計描述

OT 主要分成 sorting、B2BCD、7 段顯示器三大部分，下面依序講解。

Sorting:

一開始 in_data(4bit)會連續給出 4 個個位數字(0~9)，需要經過 sorting 後，以 A, B, C, D 表示(A>B>C>D)，並且進行以下的合併再進行兩位數乘法。D 與 C 放在十位數，B 與 A 放在個位數，如下所示。

$$DB \times CA$$

Example: in_data 連續吐 1、5、5、1，sorting 後變為，5, 5, 1, 1(ABCD)，則為 $15 \times 15 = 225$ 。

B2BCD:

這是一個將二進位轉成 BCD(二進碼十進數)的過程，在這種編碼下我們會用 4 位元表示一個十進位的數字。例如：25₍₁₀₎在 BCD 中會表示成 0010_0101_(BCD)。上一步驟乘法出來後為 8bit(不會 overflow)，並且可以用以下過程將 8bit 的 2 進

位，轉成用 4bit+4bit+4bit 共 12bit 二進位的方式表示十進位結果。

Exaple: 參考下圖

假設計算完後的結果為 255，可以用 1111_1111(8bit)來表示二進位。
將其經過 BCD 轉換後會變成 0010_0101_0101(12bit)來表示其 BCD 形式，分別就代表 0010(2)、0101(5)、0101(5)，三個十進位。

以下介紹轉換過程，可參考下圖。

Step1: 將 data 往左 shift 1bit。

Step2: 檢查 Hundreds、Tens、Units 各 4bit 數值是否大於 4，若大於則加上 3。(在下圖中都是以綠色框框代表需要加 3)

Step3.4.5.6... 重複 step1、step2，直到 shift 八次後則得到結果。

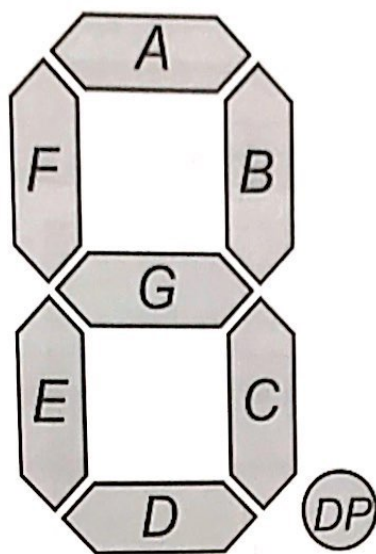
最後的結果會得到 4bit 的 Hundreds 0010、4bit 的 Tens 0101、4bit 的 Units 0101。

| Operation | Hundreds | Tens | Units | Binary | |
|-----------|----------|---------|---------|---------|---------|
| HEX | | | | F | F |
| Start | | | | 1 1 1 1 | 1 1 1 1 |
| Shift 1 | | | 1 | 1 1 1 1 | 1 1 1 |
| Shift 2 | | | 1 1 | 1 1 1 1 | 1 1 |
| Shift 3 | | | 1 1 1 | 1 1 1 1 | 1 |
| Add 3 | | | 1 0 1 0 | 1 1 1 1 | 1 |
| Shift 4 | | 1 | 0 1 0 1 | 1 1 1 1 | |
| Add 3 | | 1 | 1 0 0 0 | 1 1 1 1 | |
| Shift 5 | | 1 1 | 0 0 0 1 | 1 1 1 | |
| Shift 6 | | 1 1 0 | 0 0 1 1 | 1 1 | |
| Add 3 | | 1 0 0 1 | 0 0 1 1 | 1 1 | |
| Shift 7 | 1 | 0 0 1 0 | 0 1 1 1 | 1 | |
| Add 3 | 1 | 0 0 1 0 | 1 0 1 0 | 1 | |
| Shift 8 | 1 0 | 0 1 0 1 | 0 1 0 1 | | |
| BCD | 2 | 5 | 5 | | |

7 段顯示器:

將上一步的結果用三個七段顯示器表示出來，百位數用 seg_100，十位數用 seg_10，個位數用 seg_1。

下面附上七段顯示器的 spec。



| Digit | Display |
|-------|---------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |

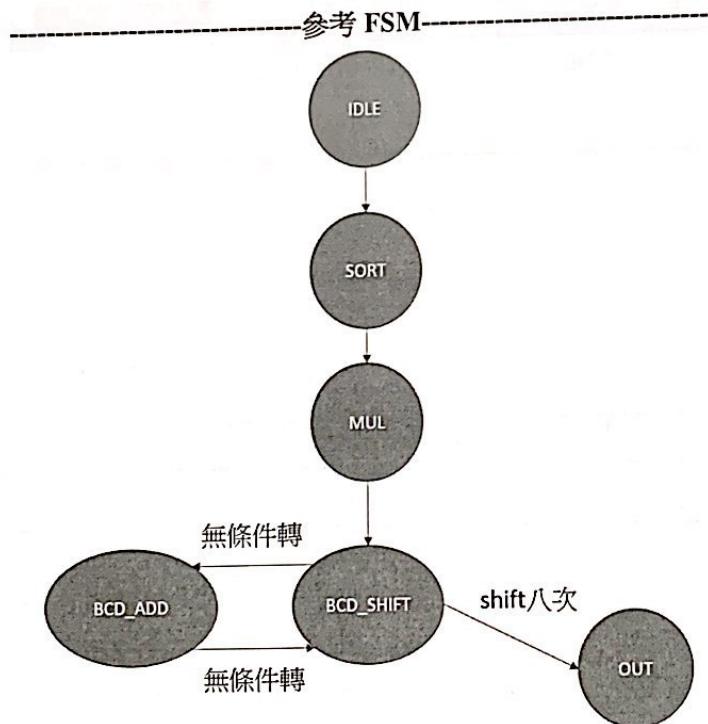
MSB(Most Significant Bit) is A and LSB(Least Significant Bit) is G. DP will be ignored.

Example: 上一步驟出來的 12bit 10 進位為 255，因此為

Seg_100 = 7'b 1101101

Seg_10 = 7'b 1011011

Seg_1 = 7'b 1011011



Input

| Signal name | Number of bit | Description |
|-------------|---------------|--|
| clk | 1 bit | Clock 5ns |
| rst_n | 1 bit | Asynchronous active-low reset |
| in_valid | 1 bit | 為 1 時代表給 in_data 資料，連續給滿 4 cycle |
| in_data | 4 bit | 計算用資料，連續吐 4cycle，收完後必須先經過 sorting。(unsigned) |

Output

| Signal name | Number of | Description |
|-------------|-----------|---------------------------------------|
| out_valid | 1 bit | 為 1 時代表 output seg_100 等值，維持一個 cycle。 |
| seg_100 | 7 bit | 顯示百位數的七段顯示器。 |
| seg_10 | 7 bit | 顯示十位數的七段顯示器。 |
| seg_1 | 7 bit | 顯示個位數的七段顯示器。 |

這次提供 5 組測資，請利用讀檔的方式下去讀取檔案。

Input.txt 每四個數字為一組，依序給值

Output.txt 每一個數字為乘法後的結果

```

1 1
2 0
3 4
4 1
5 5
6 1
7 0
8 2
9 0
10 5
11 0
12 4
13 1
14 1
15 1
16 1
17 8
18 0
19 0
20 1

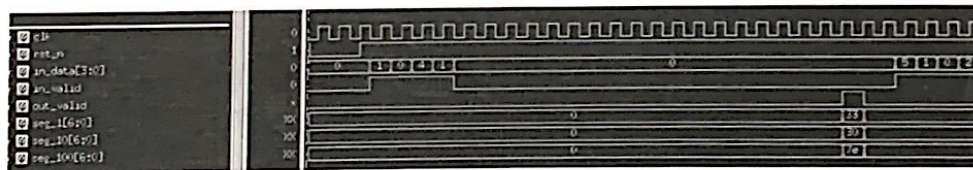
```

```

1 14
2 30
3 20
4 121
5 8

```

Example Waveform



Specification

1. Top module name : B2BCD(File name: B2BCD.sv)
2. 所有 output 必須為 0，在非同步負準位 reset。
3. 02_SYN result 不行有 error 且不能有 latches。
4. Clock period 5ns。
5. Input delay = 0.5 * clock period; output delay = 0.5* clock period;
6. 一定要使用以上 B2BCD 的方式

上傳檔案

1. Code 使用 09_upload 上傳。
2. 請 6/3 15:30 之前上傳(上傳時讓助教去協助)

Grading Policy

1. Pass the RTL & Synthesis simulation. 100%

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) ./01_run
2. 02_SYN/ (Synthesis) ./01_run_dc