

2024 ICD HW2 - Verilog

IoT Data Filtering

問題描述

請完成IoT Data Filtering的電路設計。本電路可將智慧裝置或感測器收集到的巨量IoT資料依據指定應用功能作即時分析處理，最後將處理後的數值以iot_out傳輸出去。考量到模擬時間問題，本作業IoT資料量固定為60筆，每筆資料長度為128bits，指定的應用功能共計2種。功能描述分別如下：

1. Gray2Bin:

Gray code是一種二進位編碼方式，其特點是相鄰的兩個數字之間只有一位元的差異。這種編碼方式常被用於減少二進位數字轉換時的錯誤，尤其在數位通訊和工業控制等領域有廣泛的應用。

decimal	gray	binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

其編碼方式如下，其中B[k-1:0]為binary code，G[k-1:0]為Gray code，兩者長度皆為k

$$G[n] = B[n+1] \text{ xor } B[n], \text{ for } 0 \leq n \leq k-2$$

$$B[n] = B[n+1] \text{ xor } G[n], \text{ for } 0 \leq n \leq k-2$$

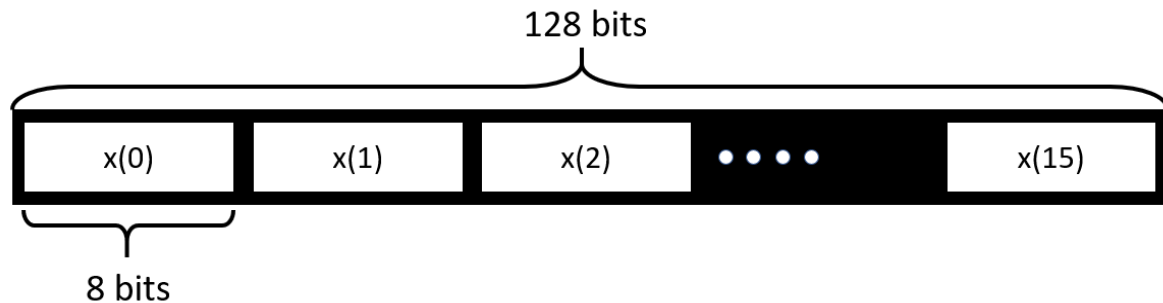
$$G[k-1] = B[k-1]$$

此次任務為將輸入的128bit資料G(n)做為Gray code，做 Gray code → Binary code的轉換，將會得到128bit的binary code結果B(n)作為輸出結果。**注意Gray2Bin是從高位開始轉換。**

2. Simple FIR Filter:

在數位訊號的處理中，FIR濾波器常用於濾波、降噪和訊號調整等方面，優點為設計簡單、運作穩定...等。

本次任務為把每筆128bits的資料從MSB到LSB的方向以8bits為單位拆成16筆8bits的 unsigned data point $\rightarrow x(0)\sim x(15)$ ，如下圖所示



而我們這次要實作的FIR濾波器公式如下

$$y(n) = 0.3125 * x(n) + 0.5625 * x(n-1) + 0.125 * x(n-2), \text{ for } n=0\sim 15$$

若計算過程中需要用到 $x(-1)$ ， $x(-2)$ 的部分則一律用0來代替，例如：

$$\begin{aligned} y(0) &= 0.3125 * x(0) + 0.5625 * x(-1) + 0.125 * x(-2) \\ \Rightarrow y(0) &= 0.3125 * x(0) + 0.5625 * 0 + 0.125 * 0 \end{aligned}$$

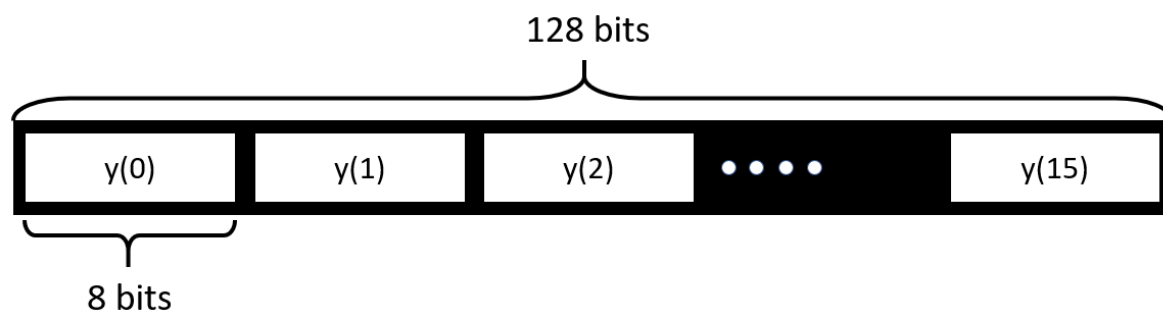
由此可得出 $y(0)\sim y(15)$ ，共16筆含小數的結果，將每筆結果取精準度到8bits整數，在此使用rounding to nearest方法，也就是若小數部分大於等於0.5便進位，反之不進位，如下圖所示

$$2.4999 \rightarrow 2$$

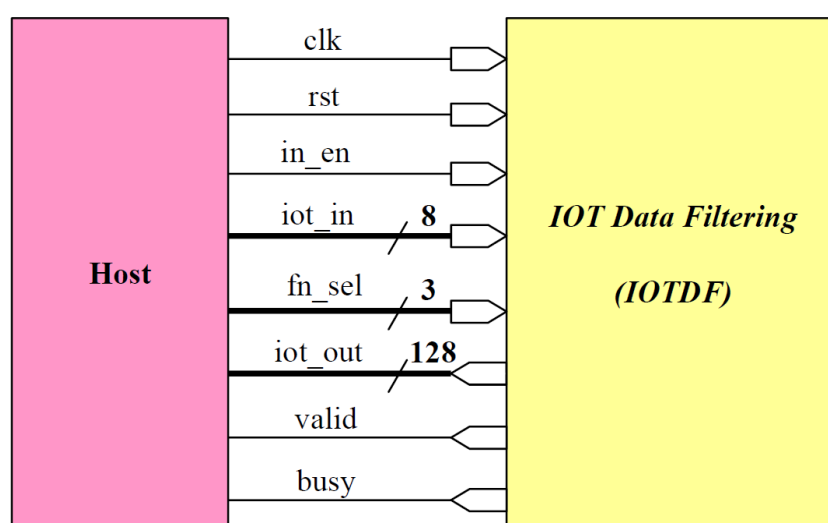
$$2.5000 \rightarrow 3$$

$$2.5001 \rightarrow 3$$

由此得出 $y(0)\sim y(15)$ 共16個8bits整數結果之後，相接後得出128bits來作為輸出結果，如下圖所示



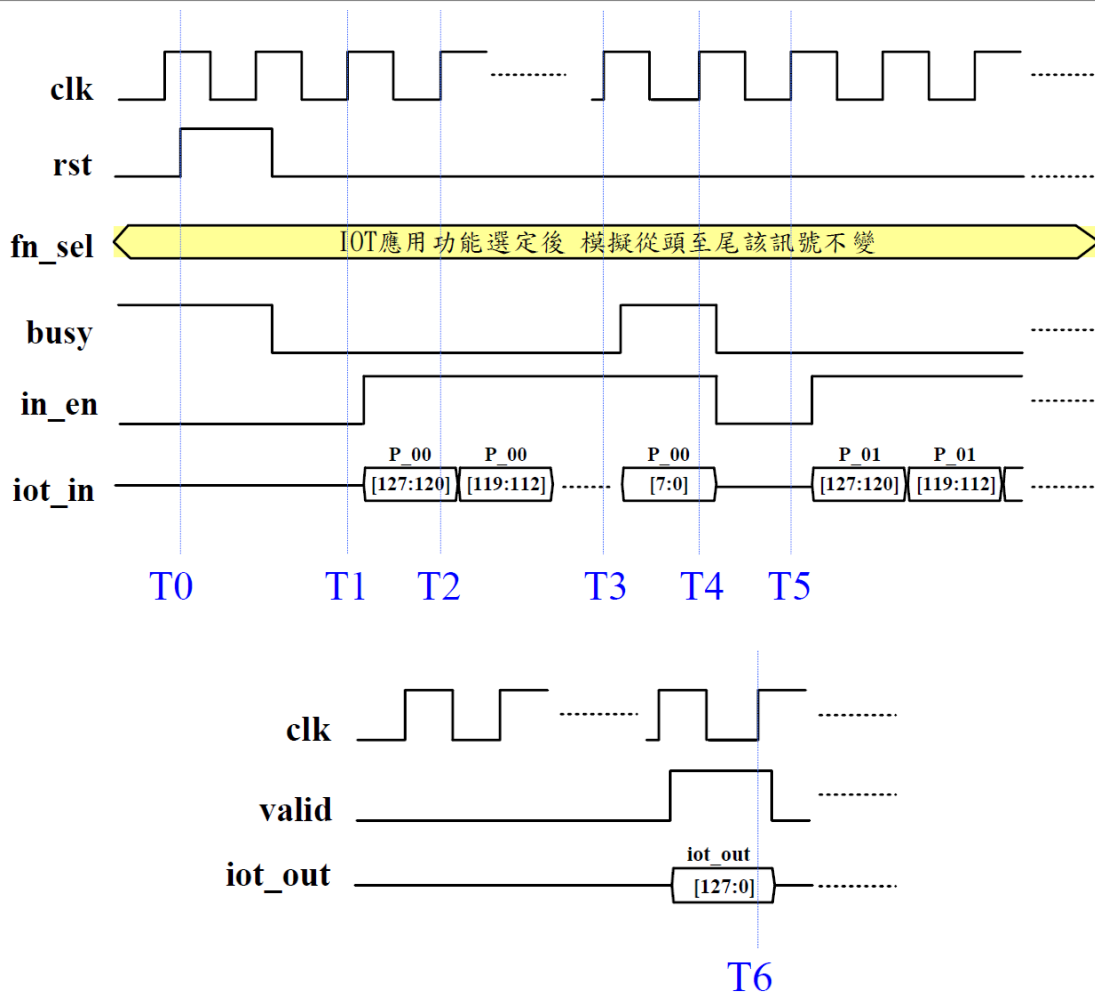
系統方塊圖



輸入/輸出介面

信號名稱	輸入/出	位元寬度	說明
clk	I	1	時脈信號，本系統為同步於時脈正緣設計。
rst	I	1	高位準非同步(active high asynchronous)之系統重製訊號。
in_en	I	1	當busy訊號為low, in_en訊號會被置為high以獲取新數據。反之，當busy訊號為high, in_en訊號會被置為low。當所有資料輸入完畢後，該訊號到模擬結束前將永遠被維持在low。
iot_in	I	8	Host端會透過此匯流排將每筆128bits資料，每次以8bits作輸入，共計16個Cycles後即可送完一筆資料。本次作業共有60筆資料。
fn_sel	I	3	本次作業有二種應用功能，透過此訊號可做為二種功能的切換，每次模擬只會固定選擇某一種功能，模擬過程中不會任意切換另一種功能。
iot_out	O	128	IOTDF電路會透過此匯流排將運算結果輸出至Host端，每次以128bits直接輸出(僅需花1個Cycle時間)。
busy	O	1	其動作已描述於in_en說明處。
valid	O	1	當valid訊號為high時，表示目前輸出的iot_out資料為有效的輸出。反之，則輸出low。

系統功能描述



1. T0 ~ T1 時間點, IOTDF 電路初始化。
2. T1時間點, Host端判斷busy訊號為low, 隨即in_en拉為high, 同時輸入IoT Data P_00[127:120] 這區間的資料。
3. T2 時間點, Host 端判斷 busy 訊號為 low, in_en 繼續維持 high, 繼續輸入 IoT Data 下一個區間 P_00[119:112]的資料, T1~T2 這區間即為連續資料輸入的範例。
4. T3 時間點, Host 端判斷 busy 訊號依舊為 low, in_en 繼續維持 high, 繼續輸入 IoT Data 最後一個區間 P_00[7:0]的資料。
5. T4 時間點, Host 端判斷 busy 訊號為 high, 隨即 in_en 為 low, IoT Data 此時輸入為 0, 表示 暫停資料輸入。
6. T1~T4 這區間為輸入完整 1 筆 IoT Data 的範例, 共計花費 16 個 Cycles, 這過程中若有需要暫停輸入, 可以自行將 busy 訊號拉為 high。
7. T5時間點以後, 請自行重覆T1 ~ T4動作之行為, 即可讓Host端繼續輸入下一筆IoT Data P_01 的資料。
8. T6 時間點, 當 IOTDF 電路運算過程需要作輸出時, 請自行將 valid 訊號拉為 high, 同時將欲輸出值透過 iot_out[127:0]直接輸出即可。

本電路之兩個Functions是透過fn_sel訊號作選擇，其對照如下表：

	fn_sel	功能	說明
F1	3'b001	Gray2Bin(N)	將每筆資料Gray code轉換為Binary code
F2	3'b010	FIR(N)	將每筆資料分別以FIR濾波器做處理

檔案說明

資料夾	檔案	說明
00_TESTBED	testfixture.v	測試樣本檔。
00_TESTBED/pattern1_data	pattern1.dat	模擬時輸入的60筆IoT Data, 2種functions皆使用同一組測試樣本。
00_TESTBED/pattern1_data	f1.dat~f2.dat	作為模擬時, 2種functions的輸出值與標準解答作比對。
01_RTL	IOTDF.v	本作業之設計檔, 請勿更改輸入/出宣告, 同學請於此檔案內作設計。
01_RTL	rtl_01.f	本作業會使用的各種檔案路徑, 若有新增.v檔的話, 必須將路徑加入此檔案

RTL模擬指令

➤ 使用 vcs 模擬指令範例如下：

```
vcs -f rtl_01.f -full64 -R +v2k -debug_access+all +define+p1+F1
```

註：本次作業有2組測試樣本及2種functions，請自行使用+define+p1+F1、+define+p1+F2、+define+p2+F1、+define+p2+F2參數作切換

模擬結果

若模擬結果都正確的話，應該可以看到如下圖的結果。

➤ 模擬結果正確

```
P55: ** Correct!! ** , iot_out=4f767ba1adb71f94c8fb1345d137a58f
P56: ** Correct!! ** , iot_out=1a8e5ea79f4b656e55686cedcf47d37b
P57: ** Correct!! ** , iot_out=e2d7c49ad64e0a11b895fc5a1f08b7b5
P58: ** Correct!! ** , iot_out=12cad57a543ff9929f59ee6a6e7d4509
P59: ** Correct!! ** , iot_out=323ebd4b7832c2dde1202bfabf121766

-----

Congratulations! All data have been generated successfully!

Total cost time: 37590.60 ns
-----PASS-----

$finish called from file "../00_TESTBED/testfixture.v", line 302.
$finish at simulation time 3759060
V C S   S i m u l a t i o n   R e p o r t
Time: 37590600 ps
CPU Time: 3.740 seconds; Data structure size: 0.0Mb
Sun Mar 24 15:10:11 2024
CPU time: 2.901 seconds to compile + 1.265 seconds to elab + 1.86
```

➤ 有錯誤時，則會出現

```
-----

Final Simulation Result as below:

-----

Pass:      0
Error:     60

-----
```

➤ 模擬時間超過End_CYCLE

```
-----

Error!!! There is something wrong with your code ...!

-----The test result is .....FAIL -----

-----
```

作業要求

1. 通過testbench的RTL Level 模擬

2. 繳交檔案如下: HW2_b**901***.zip

請將檔案打包進HW2_b**901***資料夾再進行zip壓縮

參考指令如下:

```
zip -r HW2_b**901***.zip HW2_b**901***/
```

3. 請檢查交上來的壓縮檔內HW2_b**901***資料夾中含以下檔案，檔案格式、資料夾階層或命名方式與規定不同將本次作業成績扣10分。

資料夾	檔案	描述
01_RTL	IOTDF.v	RTL Verilog Code

繳交期限

[HW2] 4/16 14:20 以前上傳至 NTU cool,不接受遲交

同學如果有任何問題,請先盡量透過 email 詢問助教。剛開始學習大家遇到的問題都會蠻

像的,如果要寄 email,請同時寄給兩位助教,記得在信件前加 [積體電路設計] 避免漏信。

助教 林克帆 r11943131@ntu.edu.tw

助教 謝承恩 r11943015@ntu.edu.tw

Appendix: Environment

- 若你在231工作站無法開啟VCS或nWave, 請先完成以下指令:

```
source ~/.chsrc_icd
```

- 開啟nWave(看waveform用):

```
nWave &
```