

2018 Digital IC Design

Homework 4: Image Display Control

1. Introduction

請完成一影像顯示控制(Image Display Control)電路設計，其架構如圖一。此控制電路，可依指定之操控指令，使顯示端的影像進行影像平均(Average)、X軸及Y軸鏡像(Mirror)與水平及垂直方向的平移(Shift)功能。本控制電路有 5 只信號輸入(cmd, cmd_valid, IROM_Q, clk, reset) 及 7 只信號輸出(IROM_EN, IROM_A, IRB_RW, IRB_D, IRB_A, busy, done)，關於各輸入輸出信號的功能說明，請參考表一。

2. Design Specifications

2.1 Block diagram

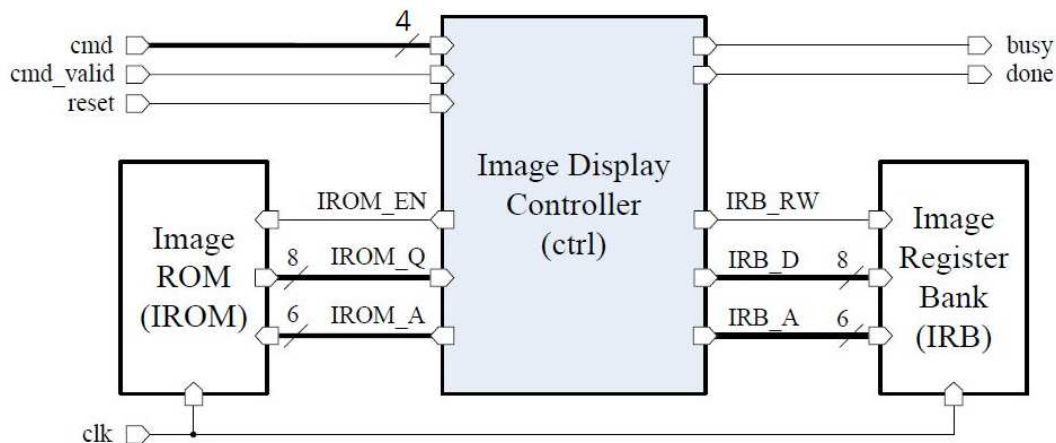


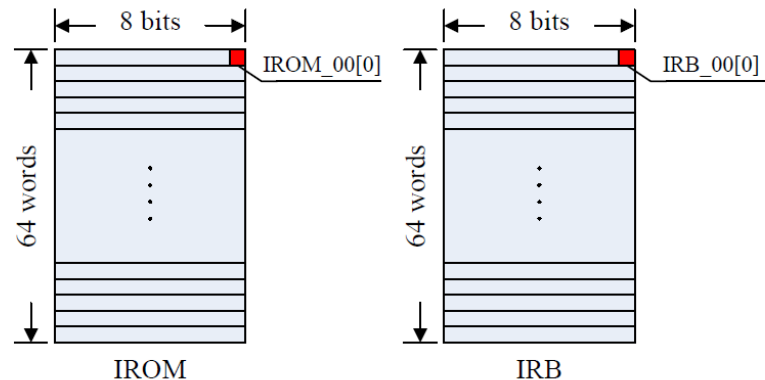
圖 一、系統方塊圖

信號名稱	輸出/入	位元寬度	說明
<i>reset</i>	input	1	高位準非同步(active high asynchronous)之系統重置信號。 說明:本信號應於系統啟動時送出。
<i>clk</i>	input	1	時脈信號。 說明:此系統為同步於時脈正緣(posedge)之同步設計。
<i>cmd</i>	input	4	指令輸入信號。 說明:本控制器共有八種指令輸入，相關指令說明請參考表二。指令輸入只有在 <i>cmd_valid</i> 為 high 及 <i>busy</i> 為 low 時為有效指令。
<i>cmd_valid</i>	input	1	讀寫控制訊號。 說明:當本信號為 high 時表示 <i>cmd</i> 指令為有效指令輸入。
<i>IROM_Q</i>	input	8	Image ROM 八位元資料輸出埠。
<i>IROM_A</i>	output	6	IROM 六位元位址信號。
<i>IROM_EN</i>	output	1	IROM 致能控制訊號。 說明:當本信號為 low 時，表示是將啟動 IROM 進行讀取；信號為 high 時，表示將關閉 IROM。
<i>busy</i>	output	1	系統忙碌訊號。 說明:當本信號為 high 時，表示此控制器正在執行現行指令(current)，而無法接受其他新的指令輸入；當本信號為 low 時，系統會開始輸入指令。reset 時，default 設定為 high。
<i>done</i>	output	1	當控制器完成寫入 IRB 時，將 <i>done</i> 設為 high 表示完成。
<i>IRB_A</i>	output	6	IRB 六位元位址信號。
<i>IRB_D</i>	output	8	IRB 八位元資料輸入埠。
<i>IRB_RW</i>	output	1	IRB 讀寫控制訊號。(本試題只有使用寫入功能) 說明:當本信號為 low 時，表示是將啟動 IRB 進行寫入。

表一、各輸入輸出信號功能

2.2 Image ROM(IROM)與Image Register Bank(IRB)規格描述

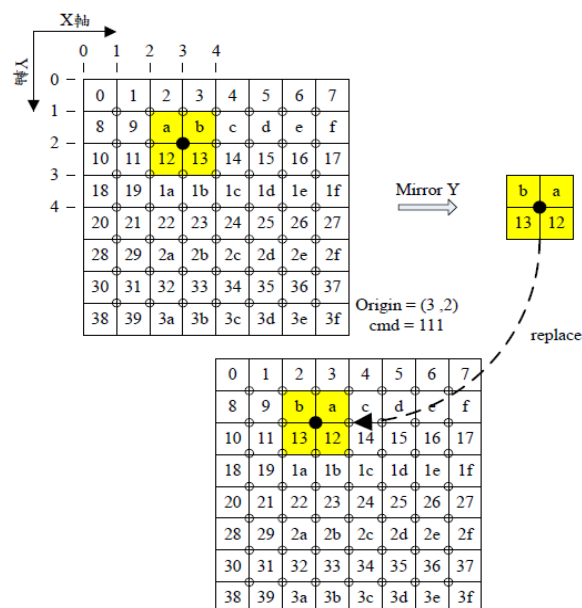
本題使用到兩個記憶體模組，一為唯讀記憶體(ROM)格式另一為單埠 Register file格式。Image ROM(IROM)模組的記憶體寬度為8位元，而記憶體深度為64個word。 Image Register Bank(IRB)模組的記憶體寬度為8位元，而記憶體深度也為64個word。如下圖二所示。



圖二、系統功能圖

2.3 系統功能描述

當 reset 結束後，影像顯示控制器之輸入端從 IROM 讀取一張 8x8 大小的影像。資料影像顯示控制器必須處理使用者輸入之指令，取得顯示相關之座標(origin)及資料參數，使得顯示端達到平均、平移以及鏡像功能，並將經過指令處理完的影像資料寫入 IRB，如圖三所示。



圖三、系統功能圖

2.3.1 輸入與輸出端之影像及參數規範

[影像輸入]

提供輸入端影像資料存至IROM。此影像為8x8共64筆測試樣本，每筆樣本為8位元資料，並且依左而右；由上而下，同學必須由IROM讀取影像資料，並且依左而右；由上而下存至影像控制電路中。(如圖三所示資料，輸入的順序為 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, ………,3d, 3e, 3f)

註:以下僅為圖例示範，詳細輸入影像值未必如下圖四所示。

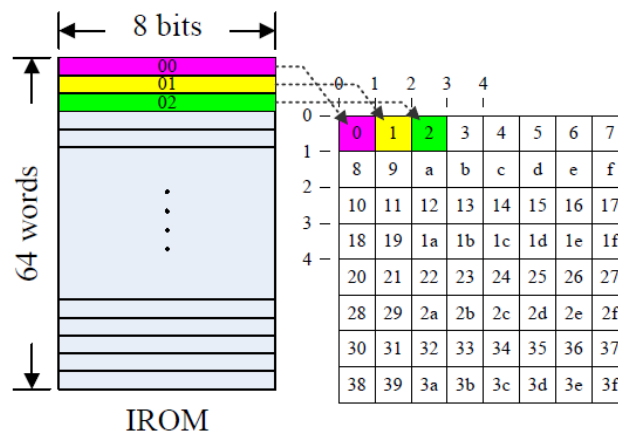


圖 四、輸入端影像

[影像輸出]

輸出端影像為8x8共64筆樣本輸出，每筆樣本為8位元資料。並且依由左而右；由上而下，且以序列(Serial)的方式循序寫入IRB內。(如圖五所示資料)。

註: 以下僅為圖例示範，詳細輸入影像值未必如下圖五所示。

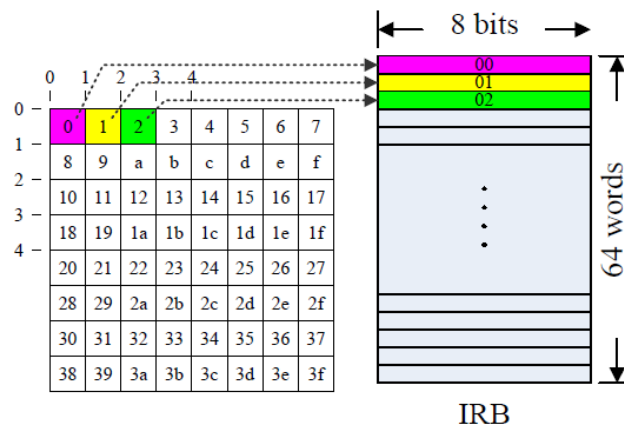
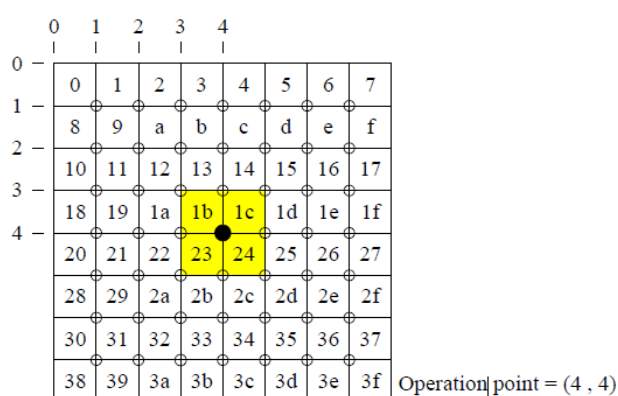


圖 五、輸出端影像資料

[操作點]

操作點(operation point)指的是在影像資料的座標點，在操作點的上下左右四格為操作影像資料，控制器將使用操作影像資料來運算。本試題已定義輸入端影像之座標軸。輸入端影像之水平方向為 X軸，垂直方向為 Y軸。此外，X軸與 Y軸座標範圍為0~+8。(如圖六所示，為確保操作影像資料不超過對應輸入影像邊界，因此限制原點之X軸與Y軸範圍最大為+1~+7)。同學須根據此座標軸，進行顯示端的畫面，進行顯示端的畫面平移(Shift)功能設計。

註：本試題規定讀入控制器後影像資料初始操作點座標為(4,4)，如下圖六所示。



圖六、輸入端影像操作點

2.3.2 影像顯示控制器功能規範

[指令定義]

影像控制器電路控制指令。輸入指令(cmd)所對應之功能如表二所示。

cmd編號	控制指令說明
0	Write
1	Shift Up
2	Shift Down
3	Shift Left
4	Shift Right
5	Average
6	Mirror X
7	Mirror Y
8	Reset_XY

9	Enhance
A	Decrease
B	Threshold
C	Inverse Threshold

表 二、輸入指令功能表

■ 寫入(Write)

- 當執行寫入(Write)指令時，控制器會依由左而右；由上而下將影像資料寫入 IRB。

■ 畫面上移(Shift Up)

- 上移顯示區塊。執行此Shift Up模式，將使操作點的Y減少1，但Y軸座標最小不可低於1。
- 當Y座標等於1時，倘若再收到上移指令，則Y軸座標將仍維持為1，操作點維持不變。

■ 畫面下移(Shift Down)

- 下移顯示區塊。執行此Shift Down模式，將使操作點的Y軸增加1，但Y軸座標最大不可大於7。
- 當Y座標等於7時，倘若再收到下移指令，則Y軸座標將仍維持為7，操作點維持不變。

■ 畫面左移(Shift Left)

- 左移顯示區塊。執行此Shift Left模式，將使操作點的X軸刪減1，但X軸座標最小不可低於1。
- 當X座標等於1時，倘若再收到左移指令，則X軸座標將仍維持為1，操作點維持不變。

■ 畫面右移(Shift Right)

- 右移顯示區塊。執行此Shift Right模式，將使操作點的X軸增加1，但X軸座標最小不可大於7。
- 當X座標等於7時，倘若再收到右移指令，則X軸座標將仍維持為7，操作點維持不變。

■ 影像資料平均(Average)

- 本題定義，當執行平均(Average)指令時，將執行目前操作點座標之影像資料取近似平均數之計算，即將目前座標所對應的4筆影像資料相加之後再除以4，當有小數點時則以無條件捨去法處理(例如 $(a+b+12+13)/4=14.5$ ，即輸出14)，輸出影像資料4筆皆輸出計算後之近似平均數，並改變原始影像資料。如圖七所示。

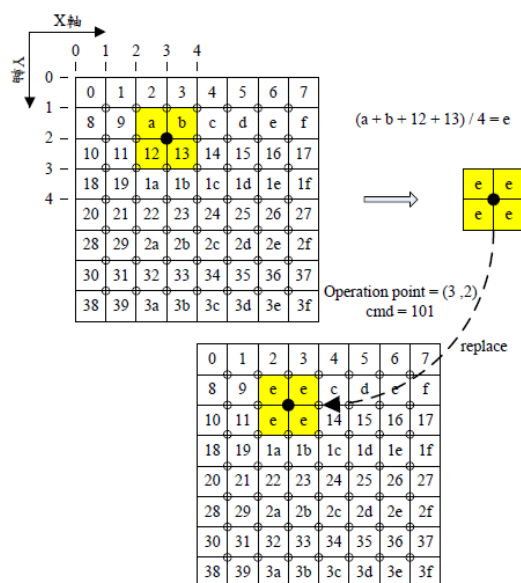
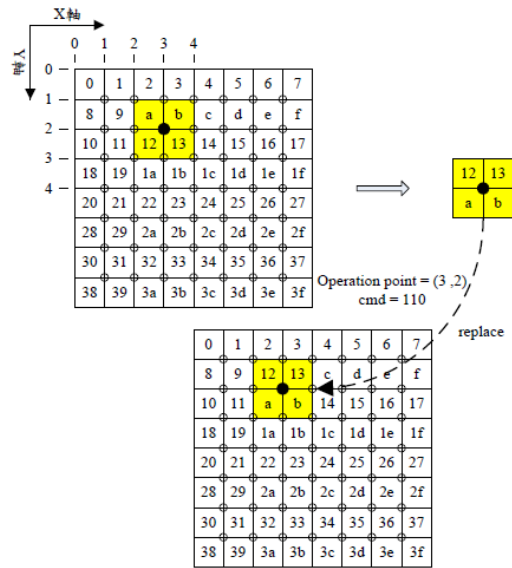


圖 七、取近似平均數影像輸出

■ 影像資料X軸鏡像(Mirror X)

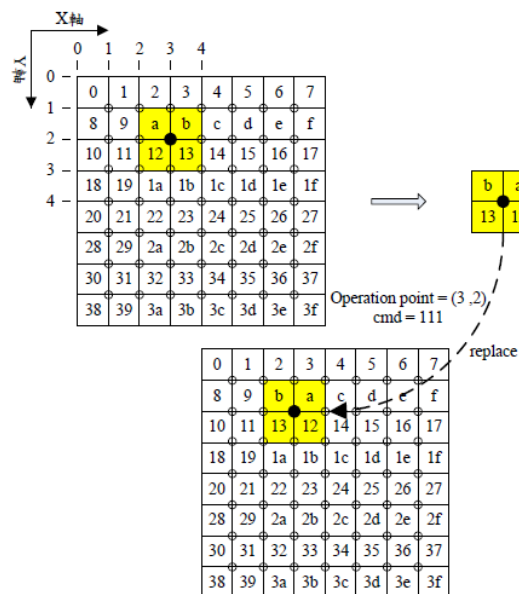
- 本題定義，當執行X軸鏡像(Mirror X)指令時，將輸出目前操作點座標之影像資料皆以目前座標對X軸翻轉，並改變原始影像資料。如圖八所示。



圖八、X 軸鏡像資料影像輸出

■ 影像資料Y軸鏡像(Mirror Y)

➤ 本試題定義，當執行Y軸鏡像(Mirror Y)指令時，將輸出目前操作點座標之影像資料皆以目前座標對Y軸翻轉，並改變原始影像資料。如圖九所示。



圖九、Y 軸映像資料影像輸出

■ 畫面回到中央(Reset_XY)

➤ 操作點的X座標、Y座標回到(4, 4)。

■ 增強(Enhance)

➤ 對操作點旁邊 4 個影像資料同時加 64_{10} 。注意，如果超過 255_{10} ，以 255_{10} 為最終結果。

50	60
70	80



114	124
134	144

■ 減少(Decrease)

➤ 對操作點旁邊 4 個影像資料同時減 64_{10} 。注意，如果低於 0_{10} ，以 0_{10} 為最終結果。

70	80
90	100



6	16
26	36

■ 二值化(Threshold)

➤ 對操作點旁邊 4 個影像資料同時做判斷如果大於 128_{10} 設為 255_{10} ，否則設 0

200	200
50	20



255	255
0	0

■ 反向二值化(Inverse Threshold)

➤ 對操作點旁邊 4 個影像資料同時做判斷如果小於 128_{10} 設為 255_{10} ，否則設 0

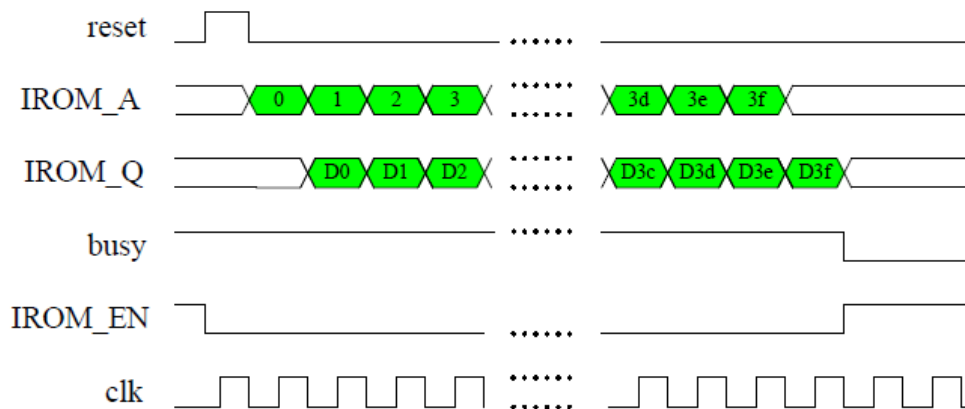
200	200
50	20



0	0
255	255

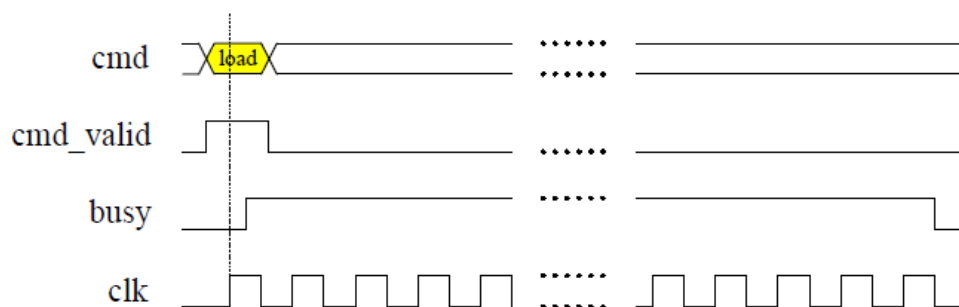
2.4 時序規格圖

- 重置(Reset)後之時序規格圖，如圖十所示。
- 在電路重置(Reset)之後，控制器將會由IROM讀取64筆影像資料。
- 當IROM_EN為low時，表示啟動IROM，即可輸入位址信號讀取IROM內的影像資料。
- 在整個處理過程中，busy皆維持為high。並在讀取完成後，將busy設回low以接受新指令輸入。



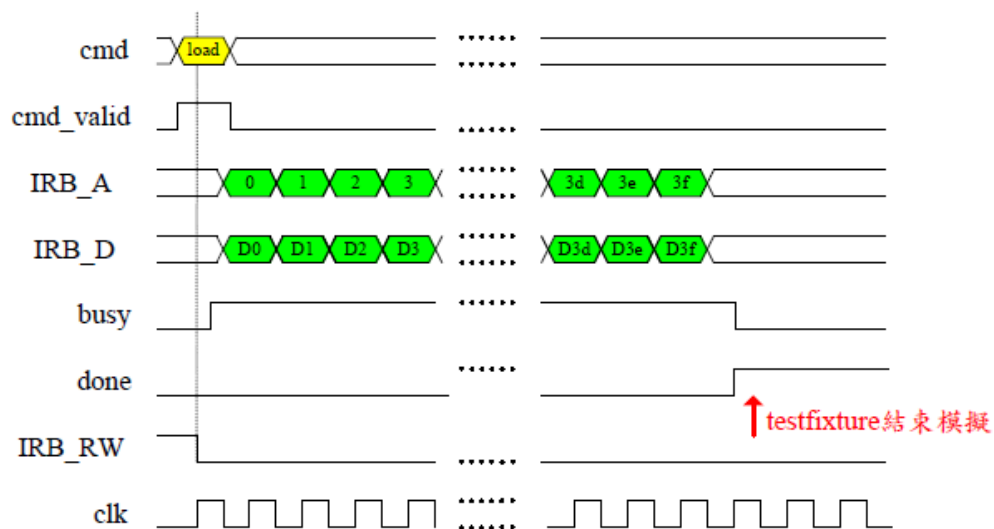
圖十、資料讀取之時序規格圖

- 其他控制指令(average、shift up、shift down、shift left、shift right)之時序規格圖，如下圖十一所示。
- 在整個處理過程中，busy皆維持為high。並在輸出完成後，將busy設回low以接受新指令輸入。



圖十一、其他控制指令之時序規格圖

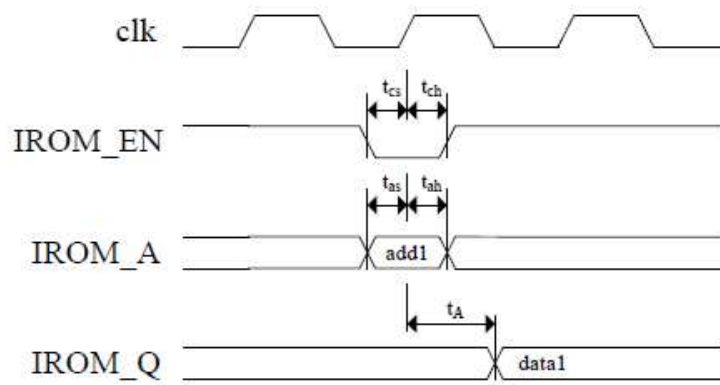
- 寫入指令(write)之时序規格圖，如下圖十二所示。
- 執行寫入指令時控制器會將處理完的影像資料寫入IRB。
- 當IRB_RW為Low時，表示對IRB寫入，即可輸入位址信號將影像資料寫入IRB。
- 在整個處理過程中，busy皆維持為high。並在輸出完成後，將busy設回low以接受新指令輸入。
- 寫入完成後，並將done信號設為high，表示寫入完成，此時testfixture會拿寫入 IRB 的資料與golden pattern比對。



圖十二、寫入指令之时序規格圖

2.5 IROM與IRB之时序規格

IROM 讀取動作主要是以IROM_EN 啟動IROM 後，輸入address 讀取資料，其資料讀取之波形时序圖，如下圖十三所示。特別注意若沒有要進行讀取時，請將IROM_EN 保持為high。最後，时序數值整理於表三。

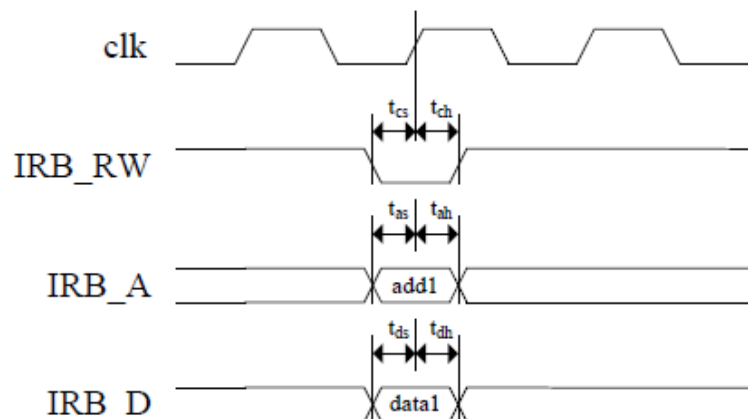


圖十三、IROM 讀取之時序規格圖

表三、IROM 時序參數表

Symbol	Description	Min	Unit
t_A	Access time	1.16	ns
t_{cs}	Chip Enable setup time	0.34	ns
t_{ch}	Chip Enable hold time	0	ns
t_{as}	Address setup time	0.29	ns
t_{ah}	Address hold time	0	ns

IRB 讀取動作主要是以IRB_RW 這個信號來控制，其資料讀取之波形時序圖，如下圖十四所示。特別注意若沒有要進行寫入時，請將IROM_RW 保持為low。最後，時序數值整理於表四。



圖十四、寫入指令之時序規格圖

表四、IRB 時序參數表

Symbol	Description	Min	Unit
t _{ws}	Write Enable setup time	0.24	ns
t _{wh}	Write Enable hold time	0	ns
t _{as}	Address setup time	0.19	ns
t _{ah}	Address hold time	0.05	ns
t _{ds}	Data setup time	0.11	ns
t _{dh}	Data hold time	0	ns

設計檔案說明

1. 以下表六為提供的設計檔案

檔名	說明
testfixture.v	測試樣本檔(testbench)。此測試樣本檔定義了時脈週期與測試樣本之輸入信號，module名稱稱為test
lcd_ctrl.v (lcd_ctrl.vhd)	參賽者影像顯示控制器lcd_ctrl的設計檔範本，已包含系統輸/出入埠之宣告 注意!!!設計者繳交之檔案，包含檔名、top module name、port name皆不能更改，在評分時因為檔名或是top module name及port name不同之問題而無法模擬，將視為設計錯誤
ROM.v	ROM模擬檔 (simulation model)
IRB.v	Register Bank模擬檔 (simulation model)
cmd1.dat	第一組測試樣本之指令
cmd2.dat	第二組測試樣本之指令
cmd3.dat	第三組測試樣本之指令
image1.dat	第一組測試樣本輸入檔案
tb1_goal.dat	第一組測試樣本之IRB比對檔
image2.dat	第二組測試樣本輸入檔案
tb2_goal.dat	第二組測試樣本之IRB比對檔
image3.dat	第三組測試樣本輸入檔案
tb3_goal.dat	第三組測試樣本之IRB比對檔

測試樣本

共提供三組測試樣本，為方便同學除錯之用，將測試樣本之影像資料及指令輸入詳列如下：

- 測試樣本一(tb1)

- 相關資料：image1.dat, cmd1.dat, tb1_goal.dat

- 影像資料

0	1	2	3	4	5	6	7
8	9	a	b	c	d	e	f
10	11	12	13	14	15	16	17
18	19	1a	1b	1c	1d	1e	1f
20	21	22	23	24	25	26	27
28	29	2a	2b	2c	2d	2e	2f
30	31	32	33	34	35	36	37
38	39	3a	3b	3c	3d	3e	3f

- 指令輸入順序：

shift up, average, shift left, mirror x, shift down, shift right, mirror y ,write

- 測試樣本二(tb2)

- 相關資料：image2.dat, cmd2.dat, tb2_goal.dat

- 影像資料：

ff	36	e7	f0	55	32	75	42
18	20	57	30	eb	af	ec	11
61	49	93	22	67	a0	05	c5
28	44	62	66	cc	76	97	79
56	28	09	ff	40	18	80	33
e6	f0	e9	ea	87	dd	ed	95
78	d4	d3	bb	f4	77	52	c3
c4	aa	b5	92	98	ee	00	a9

- 指令輸入順序：

shift up ,shift up ,mirror y ,shift up ,mirror x ,shift up ,shift left ,shift down ,shift

left ,average ,shift left ,mirror y ,shift down ,shift down ,mirror x ,shift
 right ,average ,shift down ,mirror y ,shift down ,mirror x ,shift right ,average ,shift
 right ,mirror y ,shift right ,mirror x ,shift down ,average ,shift right ,mirror x ,shift
 right ,mirror y ,shift up ,average ,shift up ,shift up ,shift left ,mirror x ,shift
 up ,average ,mirror y ,shift right ,mirror y ,write

● 測試樣本三(tb3)

■ 相關資料：image3.dat, cmd3.dat, tb3_goal.dat

■ 影像資料

96	32	0	0	0	0	96	32
C8	40	0	0	0	0	C8	40
0	0	37	37	0	0	0	0
0	0	37	37	96	0	0	0
0	0	0	A0	C8	C8	0	0
0	0	0	0	0	0	0F	0
0	0	0	0	0	64	C8	0
0	0	0	0	0	0	0	0

■ 指令輸入順序：

reset_xy ,shift right ,shift down ,enhance ,shift right ,shift down ,decrease ,reset ,shift
 right ,shift right ,shift right ,shift up ,shift up ,shift up ,threshold , reset_xy , shift left ,
 shift left ,shift left , shift up ,shift up , shift up ,inverse threshold ,write

Scoring

3.1 Functional Simulation (pre-sim) [60%]

All of the result should be generated correctly using **three test patterns** (modify the testfixture), and you will get the pass message in ModelSim simulation. You can turn off the timing check in pre-sim only.

3.2 Gate-Level Simulation (post-sim) [20%]

3.2.1 Synthesis

Your code should be synthesizable. After synthesizing in Quartus, the file named *LBP.vo* and *LBP.sdo* will be obtained.

Device : Cyclone II EP2C70F896C8

3.2.2 Simulation

All of the result should be generated correctly using **two test patterns** by *LBP.vo* and *LBP.sdo*, and you will get the pass message in ModelSim simulation. (There should be no setup or hold time violations.)

3.3 Performance [20%]

The performance is scored by the logic elements you used and the simulation time in post-sim. The scoring equation is $(Total\ logic\ elements + total\ memory\ bit + 9 * embedded\ multiplier\ 9-bit\ element) \times (longest\ gate-level\ simulation\ time\ in\ ns)$. (The smaller the better).

Submission

4.1 Submitted files

You should classify your files into three directories and compressed to .zip format. The naming rule is **studentID_version.zip**. The version is v1 for the first submission, and v2, v3... for the revisions.

<i>RTL category</i>	
<i>*.v</i>	All of your verilog RTL code
<i>Gate-Level category</i>	
<i>*.vo</i>	Gate-Level netlist generated by Quartus
<i>*.sdo</i>	SDF timing information generated by Quartus
<i>Documentary category</i>	
<i>*.pdf</i>	The report file of your design (in pdf).

4.2 Report file

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible.

SPEC: 至少含合成截圖(Quartus)、Pre-sim波型圖、Pos-sim波型圖、Pass截圖

4.3 Please submit your .zip file to folder HW4 in the ftp site.

Deadline: 107/12/21 23:55

ftp : 140.116.245.92

Username : ic_design

Password : ic_design

TA

陳威廷 weiting84610@gmail.com

蕭佑哲 e94021301@gmail.com