

初賽時間為 4 月 1 日(三) **8:30~20:30**，參賽者請於 **10 點半前**完成下列步驟進行初賽登錄，主辦單位將依完成此步驟之隊伍數決定各組最後得獎名額，請務必完成登錄動作，以免影響你的權益。

- 請將你的隊伍參賽資料 E-mail 至 ic2007@vlsilab.ee.ncku.edu.tw，

信件內容格式如下:

參賽組別：類比電路設計

參賽編號：(例：9840001)

參賽姓名：張三、李四

- 信件標題請標示為「**IC 設計競賽初賽資料登錄**」

2009 IC Design Contest Preliminary

標準元件數位電路設計 - 研究所組

NAND Flash Memory Controller

1. 問題描述

請完成一快閃記憶體控制(NAND Flash Memory Control)電路設計。此控制電路，可依主控端(Host)之操控指令來對快閃記憶體(NAND Flash Memory)進行讀寫，讀取時是將快閃記憶體的資料讀出並存放至內部記憶體(Internal Memory)；寫入時是將內部記憶體的資料寫至快閃記憶體之中。

本控制電路有 4 只信號輸入(clk, rst, cmd, F_RB)、7 只信號輸出(done, M_RW, M_A, F_CLE, F_ALE, F_REN, F_WEN)及 2 只記憶體雙向輸出入信號(M_D, F_IO)，關於各輸入/輸出信號的功能說明，請參考表一。

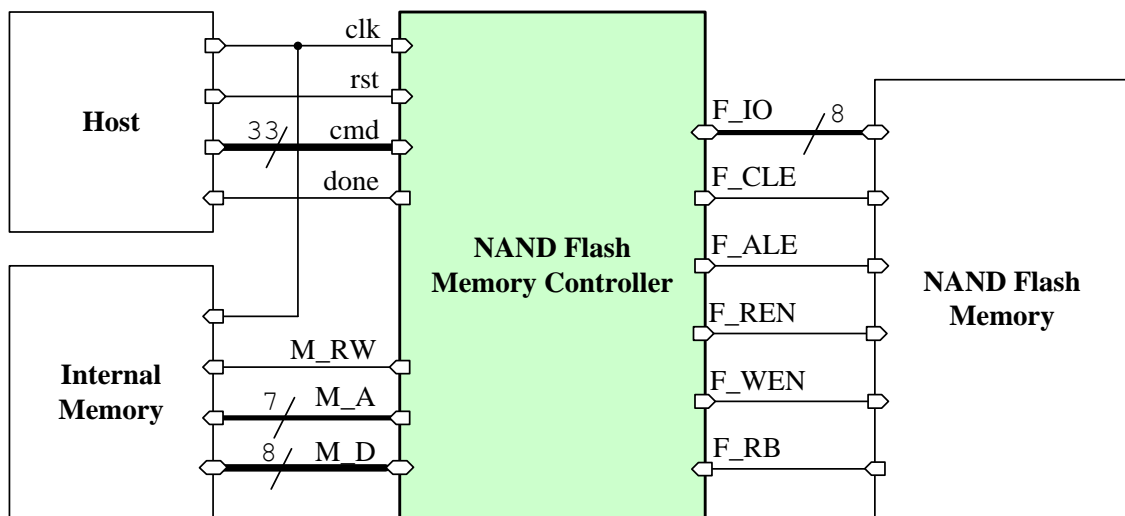
本試題有使用到記憶體模型(memory simulation model)，其中內含時序檢查，若要避免 RTL 模擬時所產生的時序檢查錯誤(setup or hold violation)，可參考附錄 B 的第 4 點來進行模擬。

每個參賽隊伍必須根據下一節所給的設計規格完成設計。參賽隊伍可藉由 CIC 所提供的輸入指令及正確結果檔來檢查設計是否有達到要求，詳情請參考附錄 B。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 08:30。當 IC 設計競賽結束後，CIC 會根據第三節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄 D 中所列的要求，附上評分所需要的檔案。

2. 設計規格

2.1 系統方塊圖



圖一、系統方塊圖

2.2 輸入/輸出介面

表一、輸入/輸出信號

信號名稱	輸入/輸出	位元寬度	說明
<i>clk</i>	input	1	時脈信號。 說明：主控端與內部記憶體於時脈正緣時動作。
<i>rst</i>	input	1	高位準非同步 (active high asynchronous)之系統重置信號。 說明：此信號於系統啟動時送出。
<i>cmd</i>	input	33	指令輸入信號。 說明：本控制器需根據此指令來對快閃記憶體進行讀寫，詳細定義請參考 2.3.1。
<i>done</i>	output	1	完成指令輸出信號。 說明：當此信號為 low 時表示控制電路正在對快閃記憶體進行讀寫；此信號為 high 時表示控制電路完成讀寫指令，主控端也會發出下一筆指令。
<i>M_RW</i>	output	1	內部記憶體讀寫控制信號。 說明：當此信號為 high 時是對內部記憶體進行讀取；low 則是進行對內部記憶體寫入。
<i>M_A</i>	output	7	內部記憶體七位元位址信號。
<i>M_D</i>	inout	8	內部記憶體八位元雙向資料輸出入埠。
<i>F_IO</i>	inout	8	快閃記憶體八位元雙向輸出入埠。 說明：此輸出入埠可被視為輸入指令、輸入位址及輸出入資料。
<i>F_CLE</i>	output	1	快閃記憶體指令致能信號。 說明：當此信號為 high 時，F_IO 會被視為「指令」，且快閃記憶體於 F_WEN 為正緣時讀取指令。
<i>F_ALE</i>	output	1	快閃記憶體位址致能信號。 說明：當此信號為 high 時，F_IO 會被視為「位址」，且快閃記憶體於 F_WEN 為正緣時讀取位址。
<i>F_REN</i>	output	1	快閃記憶體連續讀取致能信號。 說明：在進行資料的連續讀取時，資料會在此信號 負緣 後經過 t_{REA} 的時間出現於 F_IO，同時將快閃記憶體內的位址加 1。若 F_IO 為輸出時，此信號須維持在 high。
<i>F_WEN</i>	output	1	快閃記憶體寫入致能信號。 說明：當 F_IO 為指令、位址及資料輸入時，由此信號的 正緣 來對快閃記憶體進行寫入。若 F_IO 為輸入時，此信號須維持在 high。
<i>F_RB</i>	input	1	快閃記憶體完成/忙碌信號。 說明：當此信號為 low 表示快閃記憶體正在忙碌，要等待此信號為 high 時才能進行動作(重置指令除外，請參考表三)。

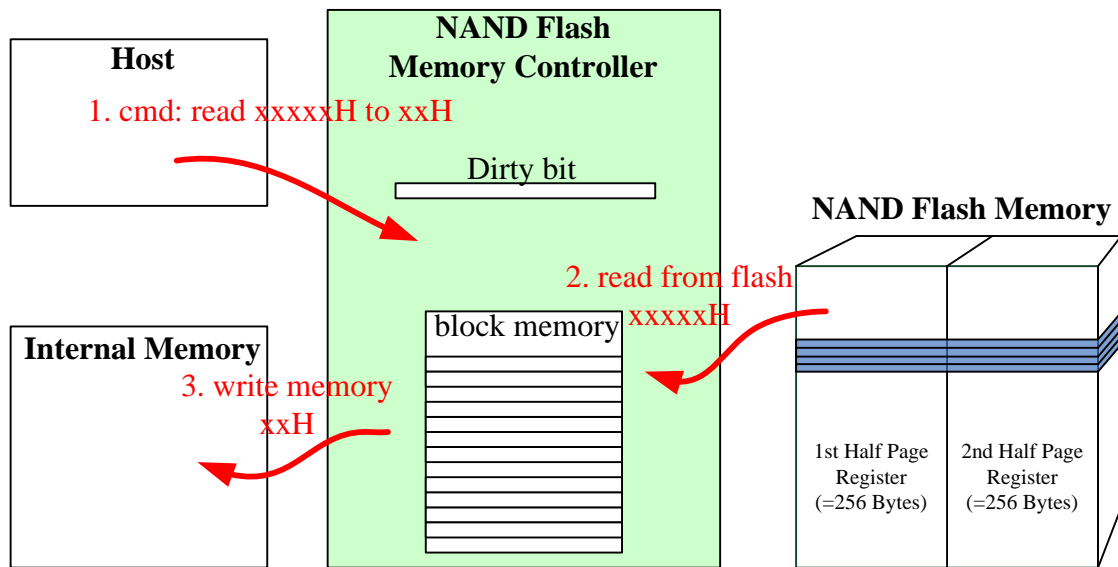
2.3 系統功能描述

快閃記憶體控制電路依主控端的指令(詳細指令說明請先參考 2.3.1)來對快閃記憶體進行讀寫。

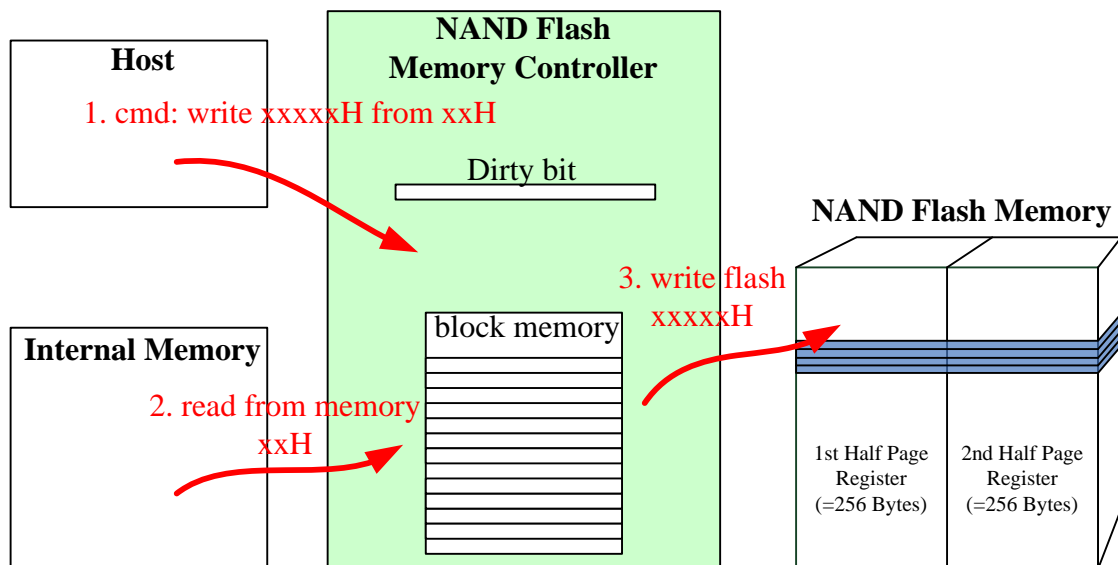
對快閃記憶體讀取時，是從指令中的快閃記憶體起始位址開始進行連續讀取(快閃記憶體之功能請參考 2.5)，並依據指令讀取規定的長度(資料筆數)存放至內部記憶體之中，存放的位址也必須依指令中所給定的內部記憶體起始位址來進行寫入(內部記憶體之功能請參考 2.5)，如圖二所示。

對快閃記憶體寫入時，則必須根據指令中內部記憶體起始位址來讀取指定長度(資料筆數)的資料，並將其寫入快閃記憶體之中，此時是由快閃記憶體起始位址開始進行連續寫入，如圖三所示。

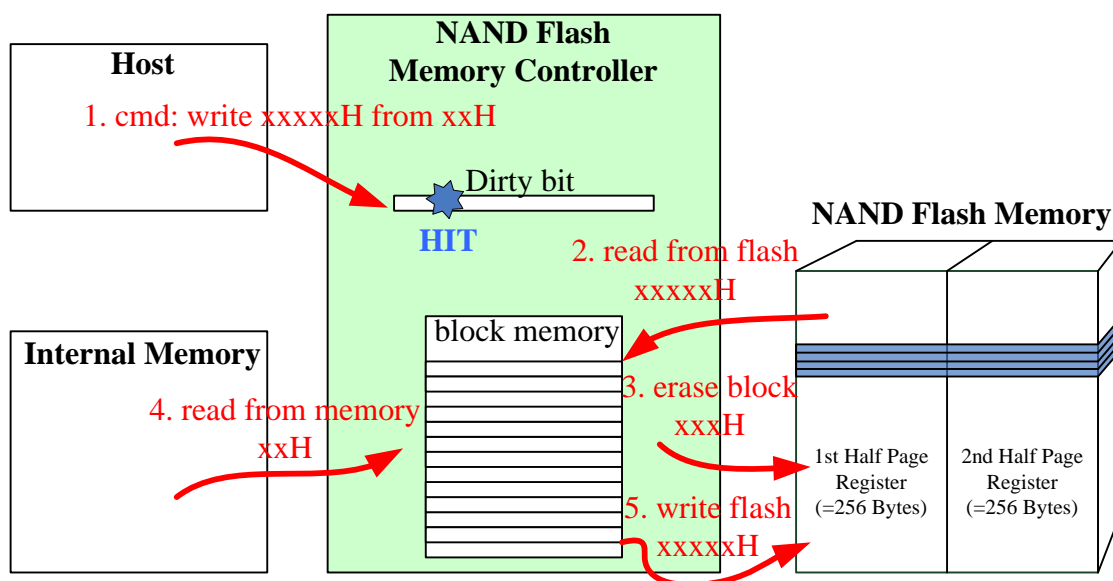
在快閃記憶體寫入須特別注意同一頁是否有被重覆寫入，若要進行二次寫入需先將資料讀取出來後，對此區塊進行抹除後，才能再次寫入。如圖四所示，當指令為寫入，且快閃記憶體起始位址之頁面曾被寫入時，就必須先將快閃記憶體的相關區塊讀取出來，再進行抹除後，才能將內部記憶體讀出，連同快閃記憶體之原始資料一併寫入快閃記憶體之中。



圖二、快閃記憶體讀取指令之範例



圖三、快閃記憶體寫入指令之範例



圖四、快閃記憶體寫入指令之範例(同一頁進行二次寫入)

為了方便處理同一頁重覆寫入的情況，可參考下面兩點建議：

1. 內建區塊記憶體(block memory)，用以儲存快閃記憶體之區塊。
2. 利用 dirty bit 來記錄快閃記憶體每個頁(page)是否曾被寫入。

2.3.1 主控端(Host)指令格式

主控端會在每個時脈正緣時，檢查快閃記憶體控制電路的 done 信號，若 done 為 high，主控端會立即發出指令(於 cmd 埠)，且維持至下次 done 為 high(見圖七)。主控端之指令格式如表二所示，位元 0 至位元 6 表示此次讀寫的長度，可表示長度為 0 ~ 127；位元 7 至位元 13 表示內部記憶體的讀取或寫入的起始位址，可表示之位址為 0 ~ 127；位元 14 至位元 31 表示快閃記憶體的讀取或寫入的起始位址，可表示之位址為 0 ~ 262143；位元 32 則表示此指令要對快閃記憶體進行的是讀取或寫入，此位元為 1 表示讀取，0 表示寫入。

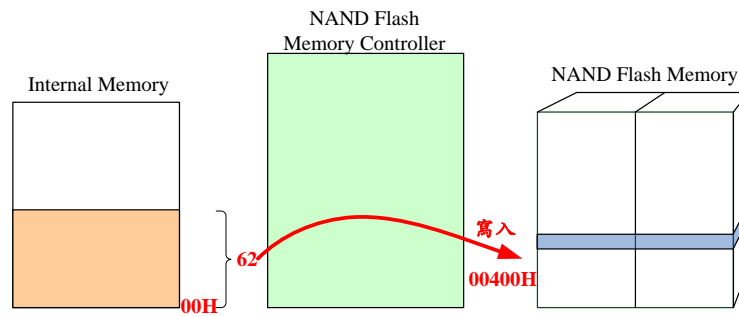
表二、主控端之指令格式

32	31	14	13	7	6	0
讀/寫	快閃記憶體起始位址					
	內部記憶體起始位址					
	讀寫長度					

快閃記憶體控制電路須先判斷指令為讀取或是寫入。以下，我們使用兩組指令來做說明：

指令一、0_00000000100000000000_00000000_0111110

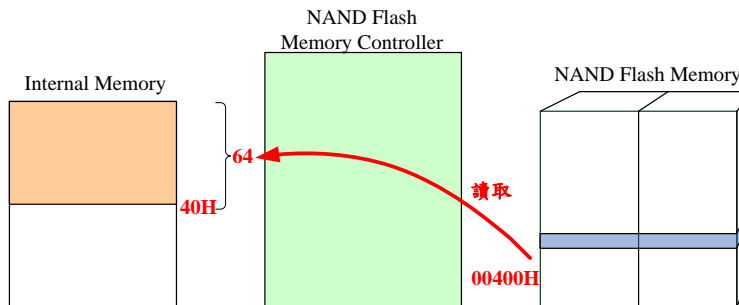
如圖五所示，指令一是要對內部記憶體 00H 的位址讀取 62 筆資料，並將這些資料連續寫入至快閃記憶體裡，且起始位址為 00400H。



圖五、指令一範例說明

指令二、1_00000000100000000000_1000000_1000000

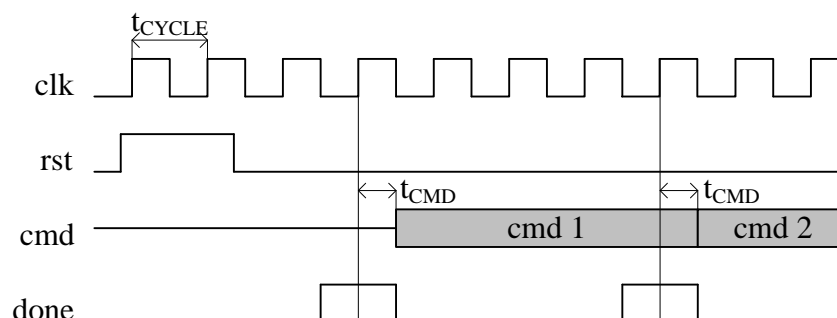
如圖六所示，指令二是要將資料由快閃記憶體中起始位址為 00400H 的位址開始讀取，接著將這些資料寫入至內部記憶體的 40H 位址之中，且寫入長度為 64 筆。



圖六、指令二範例說明

2.4 時序規格圖

快閃記憶體控制電路與主控端之間的時序波形如圖七所示，其中的時脈週期 t_{CYCLE} 預設為 10ns， t_{CMD} 則定義為半個時脈週期，也就是 $t_{\text{CYCLE}}/2$ 。主控端會在時脈正緣時檢查 done 信號，當 done 為 high 時便立即送出指令(cmd)信號，在經由 t_{CMD} 的延遲時間後會到達控制電路。控制電路在接受完新指令，應立即將 done 信號為 low，直到完成此讀寫指令後，再將 done 為 high 以接收下一筆指令。



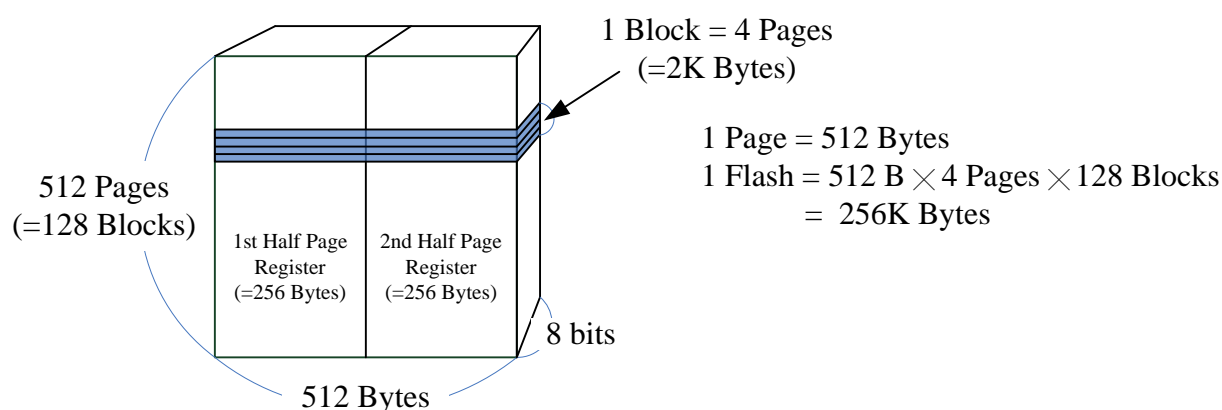
圖七、系統時序圖

2.5 快閃記憶體(NAND Flash Memory)功能描述

本題目中所使用的快閃記憶體容量為 256K 位元組(byte)，其具有 512 個列(row)或頁(page)，且每頁有 512 個位元組。快閃記憶體陣列如圖八所示，此快閃記憶體共分為 128 區塊(block)，每個區塊內有 4 個頁，且每頁又可分為前半頁(1st Half Page Register)及後半頁(2nd Half Page Register)。快閃記憶體之特性有下面幾點：

1. 進行讀取或寫入時是以頁為單位，只要給予起始位址，便可開始進行連續的讀取或寫入，最多可讀寫一個頁。換言之，只要在同一頁(512 位元組)內就可被連續地讀寫，無法連續至下一頁。如圖十四、十五所示，為連續讀寫十筆之範例。
2. 若連續寫入長度跨越兩個頁，則跨頁時須重新給予讀寫指令。
3. 快閃記憶體無法對同一頁進行二次寫入(即使是同一頁裡的不同位址)，若要對同一頁進行第二次寫入，則須先進行抹除。
4. 進行抹除(Erase)時，則是以區塊(block)為單位，一次抹除一個區塊。圖十七為區塊抹除之波形示意圖。

此快閃記憶體的位址是由 8 位元的 F_IO 來多次傳輸，這種方式不僅可減少快閃記憶體的腳位數且無論容量多寡也不須增減腳位數。此記憶體共有 256K 位元組需要 18 條位址，因此使用 F_IO 需要 3 個週期來分別輸入欄位址(Column address)與列位址(Row address)。在一般的讀寫時，需要使用到 3 個週期的位址且置於指令之後，但若是在進行抹除時，則只需要 2 個週期的列位址。指令、位址及資料要經由 F_IO 來輸入時，F_WEN 必須為”Low”，而資料將會在 F_WEN 為正緣時被讀取。當 F_CLE 為 High 時，F_IO 會被視為指令輸入；當 F_ALE 為 High 時，F_IO 會被視為位址輸入。



	F_IO[7]	F_IO[6]	F_IO[5]	F_IO[4]	F_IO[3]	F_IO[2]	F_IO[1]	F_IO[0]	
1 st cycle	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	欄位址
2 nd cycle	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	列位址(頁位址)
3 rd cycle	L	L	L	L	L	L	L	A ₁₇	

註：1. A₈ 是經由指令 00H(A₈ = 0)或 01H(A₈ = 1)來指定為”Low”或”High”。

指令 00h (Read)：定義起始位址在前半頁(1st Half Page Register)

指令 01h (Read)：定義起始位址在後半頁(2nd Half Page Register)

2. “L”是指要設定為”Low”。

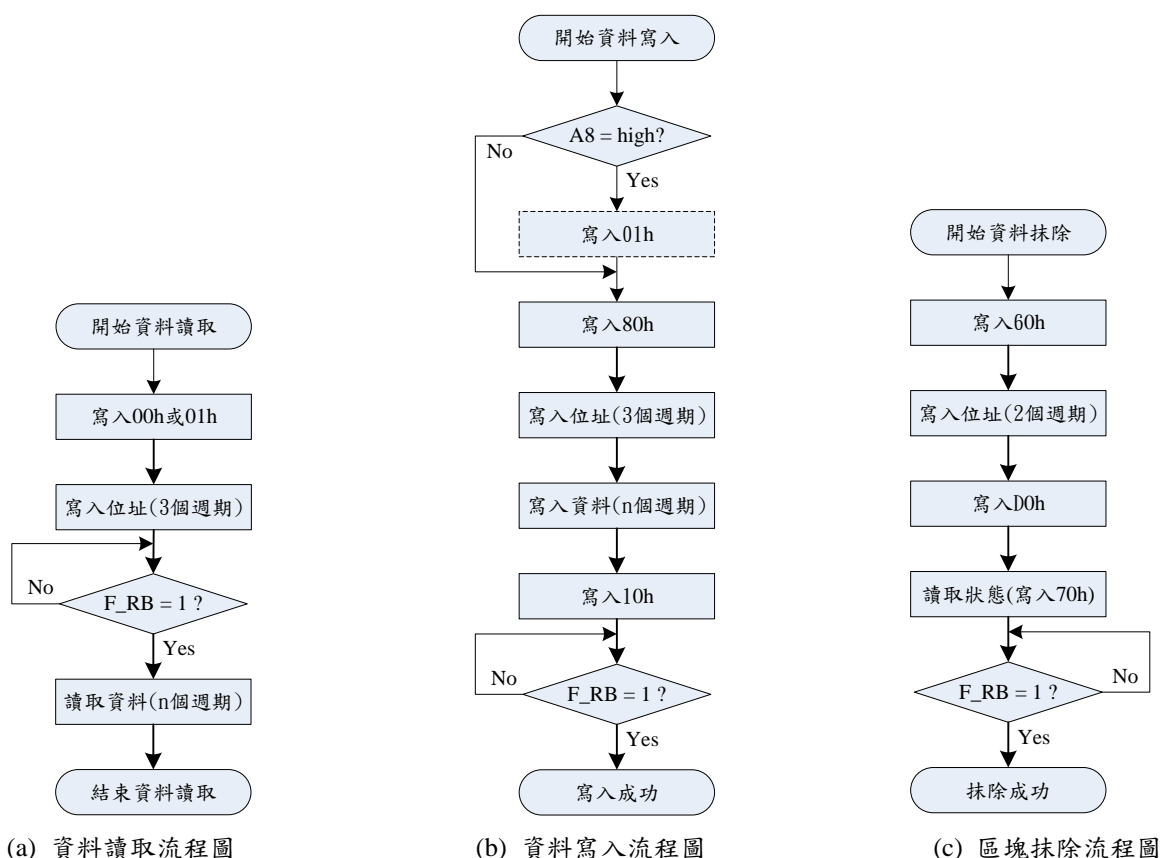
3. 當輸入位址的週期數比所需位址的還多時，快閃記憶體會忽略多餘的位址。

圖八、快閃記憶體之陣列結構

快閃記憶體的動作是經由將指令寫入其內部的暫存器來進行之，我們將此快閃記憶體之指令整理如下表所示：

表三、快閃記憶體之指令集

功能	指令 1	指令 2	忙碌時可使用
重置(Reset)	FFh		○
讀取(Read)	00h/01h		
寫入(Write)	80h	10h	
區塊抹除(Block Erase)	60h	D0h	

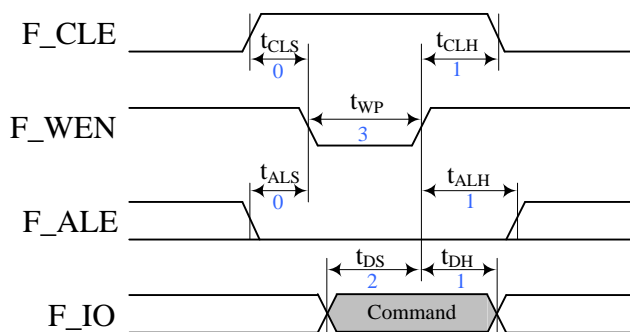


圖九、快閃記憶體動作流程圖

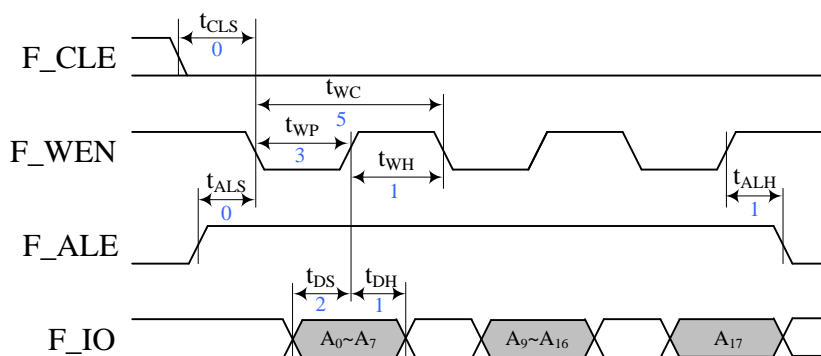
此快閃記憶體的初始狀態為「忙碌」，因此一開始必須對快閃記憶體進行重置的動作，也就是要對快閃記憶寫入 FFh 的指令，接著便可開始對快閃記憶體進行讀寫及抹除動作。讀寫及抹除動作如上方的流程圖所示，須配合表三的指令再加上指定位址來進行之。資料讀取的動作較為簡單，只要判斷 A8 來決定指令是要使用 00h 或是 01h，再加上欲讀取之起始位址後，便可進行資料的連續讀取。資料寫入的指令一般是以 80h 為起始，但若是想直接對 A8 為 high 的位址做寫入，則可在 80h 前加入 01h 的指令，經由一連串資料寫入，在寫入 10h 的指令後，資料才會由暫存器寫入快閃記憶體之中，隨即便會進入忙碌狀態，等待 F_RB 為 high 時表示完成寫入動作。區塊抹除的動作與資料寫入相似，除指令不同外，較特別的就是其位址輸入只需要 2 個週期的列位址，便可將指定的區塊進行抹除。

2.5.1 快閃記憶體之時序規格

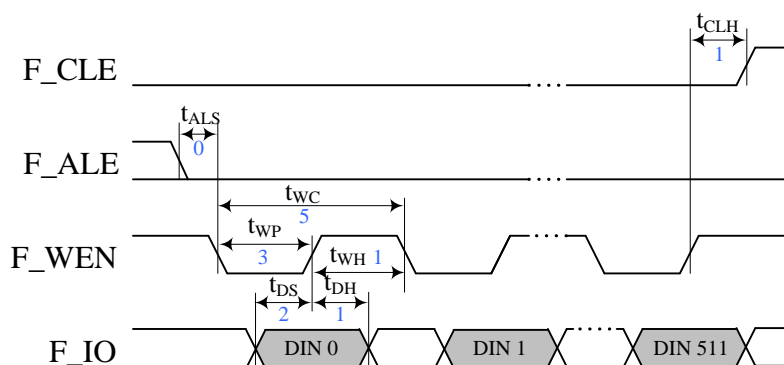
以下將快閃記憶體先細分成四個週期來表示時序，分別是圖十至圖十三的指令週期、位址週期、資料輸入週期及資料連續讀取週期。接著，圖十四至圖十六則是以波形來說明資料讀取、資料寫入及區塊抹除等動作。最後，時序的數值整理於表四。



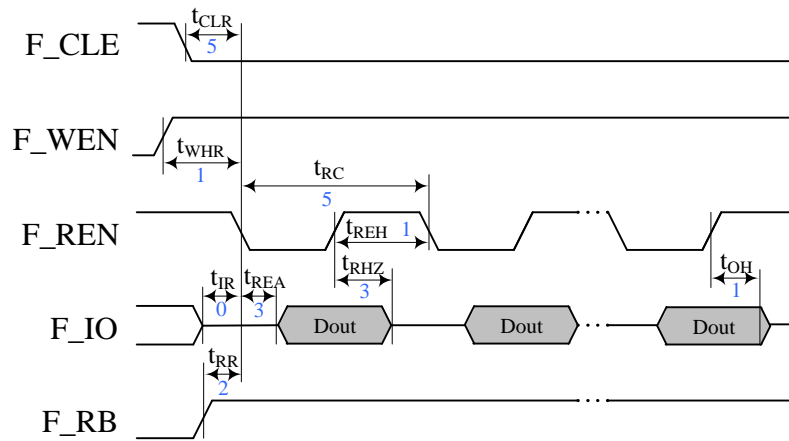
圖十、指令週期



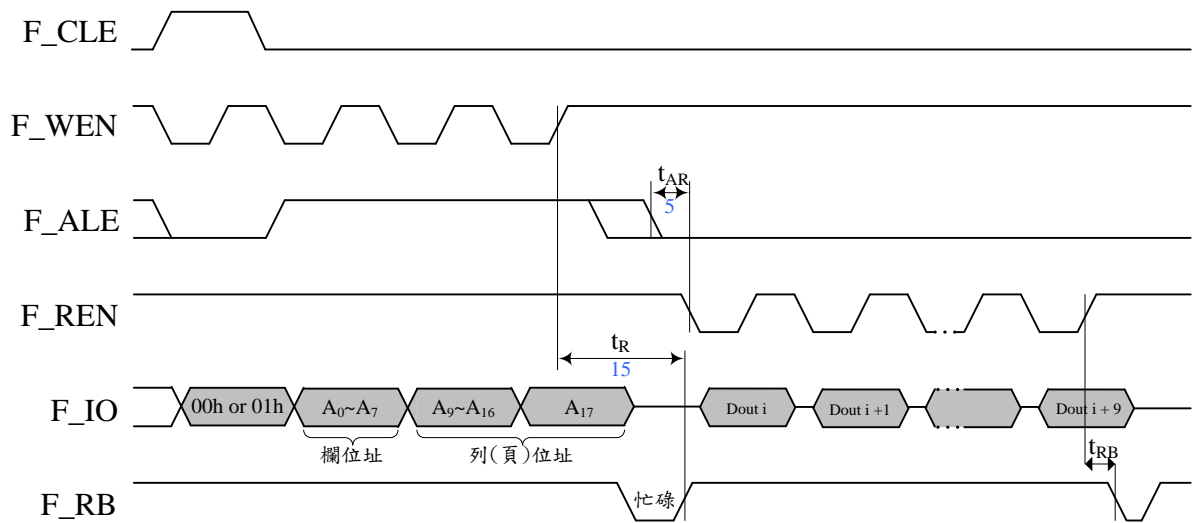
圖十一、位址週期



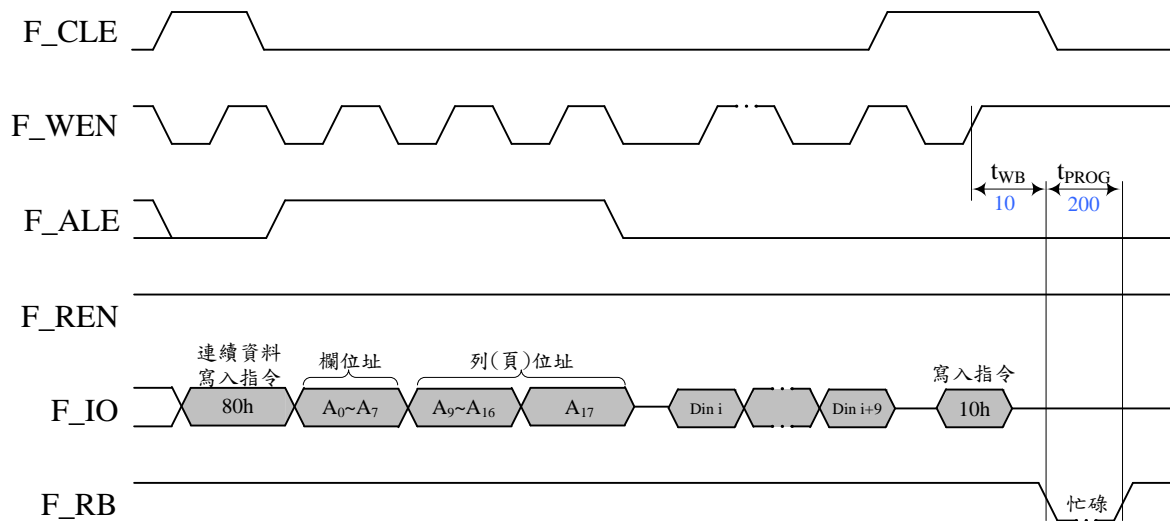
圖十二、資料輸入週期



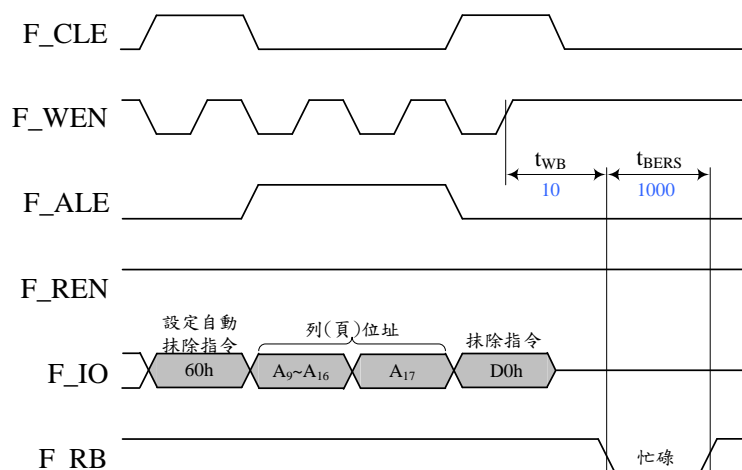
圖十三、資料連續讀取週期



圖十四、資料連續讀取十筆之波形



圖十五、資料連續寫入十筆之波形



圖十六、區塊抹除波形

表四、快閃記憶體時序參數表

Symbol	Description	Min.	Max.	Unit
t _{PROG}	Program Time	-	200	ns
t _{BERS}	Block Erase Time	-	1	μs
t _{CLS}	F_CLE Setup Time	0	-	ns
t _{CLH}	F_CLE Hold Time	1	-	ns
t _{WP}	F_WEN Pulse Width	3	-	ns
t _{ALS}	F_ALE Setup Time	0	-	ns
t _{ALH}	F_ALE Hold Time	1	-	ns
t _{DS}	Data Setup Time	2	-	ns
t _{DH}	Data Hold Time	1	-	ns
t _{WC}	Write Cycle Time	5	-	ns
t _{WH}	F_WEN High Hold Time	1	-	ns
t _R	Data Transfer from Cell to Register	-	15	ns
t _{AR}	F_ALE to F_REN Delay	5	-	ns
t _{CLR}	F_CLE to F_REN Delay	5	-	ns
t _{RR}	Ready(F_RB=1) to F_REN Low	2	-	ns
t _{WB}	F_WEN High to Busy(F_RB=0)	-	10	ns
t _{RC}	Read Cycle Time	5	-	ns
t _{REA}	F_REN Access Time	-	3	ns
t _{RHZ}	F_REN High to Output Hi-Z	-	3	ns
t _{OH}	F_REN High to Output Hold Time	1	-	ns
t _{REH}	F_REN High Hold Time	1	-	ns
t _{IR}	Output Hi-Z to F_REN Low	0	-	ns
t _{RB}	Last F_REN High to Busy	-	10	ns
t _{WHR}	F_WEN High to F_REN Low	6	-	ns

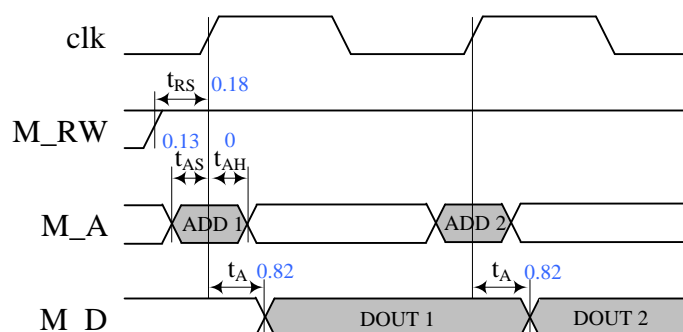
2.6 內部記憶體功能描述

內部記憶體是以 Register File 型式來實現的靜態記憶體，其容量為 128 位元組(byte)，每個位元組為 8 位元。所以，內部記憶體的位址信號(M_A)為 7 個位元；資料信號(M_D)為 8 個位元，並且具有同步正緣觸發的時脈信號(clk)與記憶體讀寫信號(M_RW)。內部記憶體須要兩個時脈週期來完成讀取動作與一個時脈週期來完成寫入動作。內部記憶體的 8 位元資料信號為雙向埠，因此在 verilog coding 時可能會需要使用到下列的語法(快閃記憶體之 F_IO 也可參考使用)：

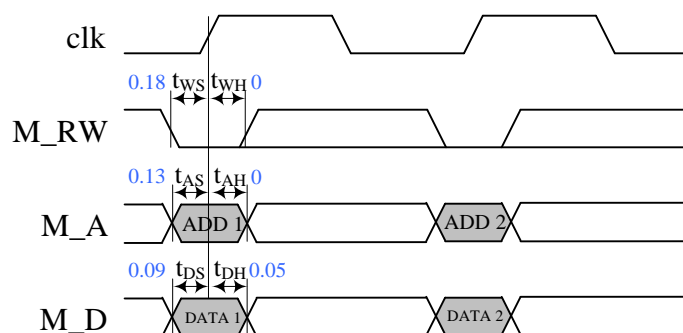
```
assign output = (enable) ? input : 'bz;
```

2.6.1 內部記憶體之時序規格

內部記憶體的讀取與寫入動作主要是以 M_RW 這個信號來做控制，其資料讀取與寫入之波形時序圖，如下面圖十七、圖十八所示。**特別注意若沒有要進行寫入時，請將 M_RW 保持為 high。**最後，時序的數值整理於表五。



圖十七、內部記憶體資料讀取波形



圖十八、內部記憶體資料寫入波形

表五、內部記憶體時序參數表

Symbol	Description	Min.	Unit
t_A	Access Time	0.82	ns
t_{RS}	Read Enable Setup Time	0.18	ns
t_{WS}	Write Enable Setup Time	0.18	ns
t_{WH}	Write Enable Hold Time	0	ns
t_{AS}	Address Setup Time	0.13	ns
t_{AH}	Address Hold Time	0	ns
t_{DS}	Data Setup Time	0.09	ns
t_{DH}	Data Hold Time	0.05	ns

3. 評分標準

主辦單位的評分人員將依照參賽者提供之系統時脈進行 RTL simulation 與 gate-level simulation，以驗證設計正確性，並且依據設計檔上傳至 CIC FTP 檔案伺服器(請參閱附錄 E)的時間來進行排名。各參賽隊伍應於參賽者定義的系統時脈下，確保輸出結果無設置與保持時間(setup/hold time)的問題，並完全符合 CIC 所提供的標準設計結果為準。為避免參賽者使用一個以上的內建記憶體(block memory)，本題限制合成面積(cell area)應小於 $3,000,000\mu\text{m}^2$ ，超過此面積視為未完成。

CIC 將本試題區分為下面四個等級來作為功能完成度之評分，若為同一等級則以檔案上傳時間來評分，且對每個等級分別進行 RTL 與 gate-level simulation：

1. A 等級：完成測試樣本一至四 (tb1 & tb2 & tb3 & tb4)
2. B 等級：完成測試樣本一至三 (tb1 & tb2 & tb3)
3. C 等級：完成測試樣本一至二 (tb1 & tb2)
4. D 等級：完成測試樣本一 (tb1)

請注意，我們將以各參賽隊伍的設計結果正確為前提，並以最後上傳檔案的時間為依據。一旦設計經評審驗證後，完成同一等級者，上傳時間越早，其所得到的分數就越高。但倘若參賽者於競賽時間內無法完成設計，也請務必將設計檔案內容上傳，主辦單位亦將根據設計內容的完成度給予記分。審查成績將另擇期通知。

附錄

在附錄 A 中說明本次競賽之軟體環境；附錄 B 為主辦單位所提供各參賽者的設計檔說明；附錄 C 為測試樣本詳細資料；附錄 D 為評分用檔案，亦即參賽者必須回傳至 CIC 的檔案資料；附錄 E 則為設計檔上傳步驟說明。

附錄 A 軟體環境

競賽所提供的設計軟體與版本如下表六。驗證評分時，係以所列軟體及版本作為驗證依據。

表六、設計軟體版本

Functionality	Corresponding EDA tools
Logic Simulator	nc-verilog v6.1 modelsim v6.3a
Logic Synthesizer	design-compiler v2008.09-sp2

附錄 B 設計檔案說明

1. 下表七為主辦單位所提供各參賽者的設計檔案

表七、設計檔

檔名	說明
testfixture.v	測試樣本檔。此測試樣本檔定義了時脈週期與測試樣本之輸入信號
NFC.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
flash.v	快閃記憶體模擬檔
t13rf128x8.v	內部記憶體模擬檔
synopsys_dc.setup	Design Compiler 初始設定範例檔案
NFC.sdc	Design Compiler 電路合成規範檔
p1_cmd.dat	測試樣本一之指令
p1_mem.dat	測試樣本一之記憶體比對檔
p2_cmd.dat	測試樣本二之指令
p2_mem.dat	測試樣本二之記憶體比對檔
p3_cmd.dat	測試樣本三之指令
p3_mem.dat	測試樣本三之記憶體比對檔
p4_cmd.dat	測試樣本四之指令
p4_mem.dat	測試樣本四之記憶體比對檔
report.000	結果報告範本

2. 請使用 *NFC.v(vhd)*，進行快閃記憶體控制電路之設計。其模組名稱、輸出/入埠宣告如下所示：

```

module NFC(clk, rst, cmd, done, M_RW, M_A, M_D, F_IO, F_CLE, F_ALE, F_REN, F_WEN, F_RB);

    input clk;
    input rst;
    input [32:0] cmd;
    output done;
    output M_RW;
    output [6:0] M_A;
    inout  [7:0] M_D;
    inout  [7:0] F_IO;
    output F_CLE;
    output F_ALE;
    output F_REN;
    output F_WEN;
    input  F_RB;

endmodule

```

3. 快閃記憶體為非同步信號，參賽者可自行修改電路合成規範檔(*NFC.sdc*)以符合自己的電路設計。以下為快閃記憶體時序設定之範例，參賽者可參考使用：

```
set_input_delay 4 -clock clk [get_ports F_IO[*]]

set_min_delay 2 -from clk -to [get_ports F_IO[*]]
set_min_delay 2 -from clk -to F_CLE -fall
set_min_delay 2 -from clk -to F_ALE -rise
set_min_delay 0.5 -from clk -to F_REN
set_min_delay 0.5 -from clk -to F_WEN

set_max_delay 0.5 -from clk -to F_CLE -rise
set_max_delay 0.5 -from clk -to F_ALE -fall
set_max_delay 1 -from clk -to F_REN
set_max_delay 1 -from clk -to F_WEN
set_max_delay [expr $cycle - 2] -from clk -to [get_ports F_IO[*]]
set_min_delay 2 -from F_RB
```

4. 比賽共提供四組測試樣本，**參賽者可依下面範來進行模擬：**

- ncverilog 指令範例如下：
ncverilog testfixture.v NFC.v +define+tb1
- 若使用 modelsim，則是在 compiler verilog 時，使用下面指令：
vlog testfixture.v +define+tb1
- 上述指令中**+define+tb1**指的是使用第一組測試樣本模擬，若須使用其它測試樣本請自行修改此參數。以第二組測試樣本為例：**+define+tb2**。
- 關於模擬時使用的一些記憶體（t13rf128x8.v、flash.v），因已經以 include 方式加在 testfixture.v 裏，所以不需加在模擬指令裏。
- **若 RTL 模擬時，要避免時序檢查以減少錯誤訊息，可於模擬指令中加入+notimingchecks**

5. 因波形檔很大，請以 fsdb dump 取代 vcd dump，dump fsdb 須使用指令如下：

- ncverilog 指令範例如下：
ncverilog testfixture.v NFC.v +define+tb1+FSDB +access+r
- modelsim 使用者，請直接使用內建波形來進行除錯。

附錄 C 測試樣本

比賽共提供四組測試樣本，為方便設計者除錯之用，將測試樣本之輸入指令詳列如下：

◆ 測試樣本一

- ⇒ 說明：測試基本讀寫
- ⇒ 檔案：p1_cmd.dat
- ⇒ 指令：

指令編號	讀/寫	快閃記憶體起始位址	內部記憶體起始位址	讀寫長度
1	0 (寫入)	00400 H	00 H	62
2	1 (讀取)	00400 H	40 H	64

◆ 測試樣本二

- ⇒ 說明：測試讀寫與 A8 為 high 時的寫入
- ⇒ 檔案：p2_cmd.dat
- ⇒ 指令：

指令編號	讀/寫	快閃記憶體起始位址	內部記憶體起始位址	讀寫長度
1	0 (寫入)	00e01 H	01 H	105
2	1 (讀取)	00e00 H	03 H	107
3	0 (寫入)	1d904 H	42 H	60
4	1 (讀取)	1d903 H	00 H	61

◆ 測試樣本三

- ⇒ 說明：測試跨頁的寫入與讀取
- ⇒ 檔案：p3_cmd.dat
- ⇒ 指令：

指令編號	讀/寫	快閃記憶體起始位址	內部記憶體起始位址	讀寫長度
1	0 (寫入)	00064 H	00 H	127
2	0 (寫入)	33d88 H	00 H	127
3	0 (寫入)	0c3cd H	00 H	127
4	0 (寫入)	1b207 H	00 H	127
5	0 (寫入)	011e1 H	00 H	127
6	1 (讀取)	00064 H	01 H	127
7	1 (讀取)	0c3cd H	01 H	127
8	1 (讀取)	011e1 H	01 H	127
9	1 (讀取)	33d88 H	01 H	127
10	1 (讀取)	1b207 H	01 H	127

◆ 測試樣本四

⇒ 說明：測試同一頁重覆寫入

⇒ 檔案：p4_cmd.dat

⇒ 指令：

指令編號	讀/寫	快閃記憶體起始位址	內部記憶體起始位址	讀寫長度
1	0 (寫入)	1e78c H	00 H	127
2	0 (寫入)	1e1e1 H	00 H	127
3	1 (讀取)	1e1e1 H	01 H	127
4	1 (讀取)	1e78c H	01 H	127
5	0 (寫入)	1e306 H	02 H	125
6	0 (寫入)	03039 H	02 H	125
7	1 (讀取)	1e306 H	03 H	125
8	1 (讀取)	03039 H	03 H	125
9	1 (讀取)	1e78c H	01 H	127
10	1 (讀取)	1e1e1 H	01 H	127

附錄 D 評分用檔案

評分所需檔案可分為三部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 RTL code，若設計採模組化而有多個設計檔，請務必將合成所要用的各 module 檔放進來，以免評審進行評分時，無法進行編譯；(2)gate-level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report.000 檔，以方便主辦單位進行評分，report.000 的格式如圖十九所示。(report 檔以後三碼序號表示版本，若繳交檔案更新版本，則新版的 report 檔檔名為 report.001，依此類推)

表八、評分用檔案

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	report.xxx	design report
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	*_syn.vg	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	SDF timing information generated by Synopsys Design Compiler
	*_syn.ddc	design database generated by Synopsys Design Compiler

隊號(Team number): 999999

完成模擬之測試樣本等級(level of test pattern) : A

--- RTL category---

使用之 HDL 模擬器名稱(HDL simulator): nc-verilog

RTL 檔案名稱(RTL filename): NFC.v, sub1.v, sub2.v

--- Pre-layout gate-level ---

gate-level 檔案名稱(gate_level filename): NFC_syn.vg

gate-level sdf filename: NFC_syn.sdf

design compiler 合成資料庫(dc library): NFC_syn.ddc

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫)

圖十九、report.000 的範本

附錄 E 檔案上傳

所有包含於如附錄 D 中表格所示的檔案，均需要提交至 CIC。並且，提交的設計檔案，需要經過壓縮於同一個資料夾下，步驟如下：

1. 建立一個 result_xxx 資料夾。其中“xxx”表示繳交版本。例如“000”表示為第一次上傳；“001”表示為第二度上傳；002 表示為第三度上傳，以此類推…。
2. 參考附錄 D 評分用檔案，將所有繳交檔案複製到 result_xxx 資料夾
3. 執行 tar 指令將 result_xxx 資料夾包裝起來，tar 的指令範例如下：

```
tar cvf result_xxx.tar result_xxx
```

其中 xxx 表示繳交版本

執行完後應該會得到 result_xxx.tar 的檔案

4. 使用 ftp 將 result_xxx.tar 及 report.xxx 一併上傳至 CIC 提供的 ftp server，result_xxx.tar 與 report.xxx 之“xxx”編號需一致，評審將以**最後上傳的設計檔及報告檔編號進行評分作業**。

ftp 的帳號和密碼在 4 天前已用 email 寄給各參賽者。若有任何問題，請聯絡 CIC

FTP site1 (台灣大學)：iccftp.ee.ntu.edu.tw (140.112.20.92)

FTP site2 (晶片中心)：iccftp.cic.org.tw (140.126.24.18)

FTP site3 (成功大學)：iccftp.ee.ncku.edu.tw (140.116.156.55)

5. 若你需要繳交更新版本，請重覆以上步驟，並記得修改 report 檔及 tar 檔的版本編號，因為你無法修改或刪除或覆蓋之前上傳的資料。