**數位電路實驗：Lab3 ─ 數位錄音機**

**第七組：洪鈺萌、劉力仁、李旻芳**

# **使用說明**

1. 按下FPGA開關鈕，將電源開啟。
2. 將SW2開關往上撥，即進入IDLE模式，此時I2C會完成初始化動作。
3. 在IDLE模式下，壓下KEY0可進入RECORD模式，即可開始錄音，再壓一次KEY0會進入RECORD\_STOP，可暫停錄音，而再壓一次KEY0又可開始錄，依此類推。
4. 在RECORD模式下，如果錄音時間超過32秒，會自動進到IDLE模式，表示SRAM儲存空間已滿；或是手動按下KEY1鈕，也可進到IDLE模式，表示結束錄音。此時在IDLE下，若重複步驟(3)，可重新錄音，原本已儲存在SRAM裡的資料會被重新寫入新的值。
5. RECORD進到IDLE下，將SW0開關往上撥，接著再按下KEY0，可進入PLAY模式，此時會將SRAM裡的音訊依序播出。此時，再壓一次KEY0會進入PLAY\_STOP，可停止播放，而再壓一次KEY0又可開始播放，依此類推。
6. 在PLAY模式下，壓下KEY2可加快播放速度，每壓一次會加快一倍，最多加快到8倍，此時LCD螢幕上會以「>>」及箭號後方的數字來表示目前播放的速度，例如：「>> 3」即表示目前是以3倍速進行播放。另外，在PLAY模式下，也可壓下KEY3來降低播放速度，每壓一次會變慢一倍，最多變慢到8倍，此時LCD螢幕上會以「<<」及箭號後方的數字來表示目前播放慢速的倍率，例如：「<< 5」即表示目前是以1/5倍速進行播放。
7. 在PLAY的「慢速」模式下，將SW1開關往上撥表示音訊是以線性內插進行處理；將開關往下撥則為零次內插。
8. 此外，在RECORD或PLAY模式下，最左邊的一組七段顯示器會顯示目前錄音或播放的秒數。當加速播放時，秒數也會隨之加快，慢速播放則相反。

# **教學說明**

此實驗目的為設計數位錄放音機，具備錄音、播放、暫停、停止播放等基本功能，取樣頻率為32 kHz，每個取樣為16 bits；並具備快速播放（2、3、4、5、6、7、8 倍速）以及慢速播放（1/2、1/3、1/4、1/5、1/6、1/7、1/8倍速），慢速播放以內插信號來實作（需支援0次內插與1次內插）。另外，可錄製的時間總長度為32秒，並搭配七段顯示器或LCD來顯示撥放的狀態和時間。

## **實驗簡介**

錄音機的核心功能是錄製聲音，並以不同方式如：重複播放、加速、慢速，或加上其他如回音等特殊效果播放聲音。大多數的錄音機利用麥克風，將自然界中聲波轉換為類比電訊號，為了記錄及處理方便，透過類比／數位轉換器（ADC）轉為數位訊號儲存。需要重播時，再透過數位／類比轉換器（DAC），將數位訊號轉回類比，並交由揚聲器（Speaker）將類比電訊號重現為聲波的形式。

本實驗所使用的系統建構在DE2-115開發平台上所提供的WM8731音效晶片，此晶片整合了前述的ADC及DAC兩個電路區塊，是數位與外界類比訊號溝通的橋梁。WM8731晶片支援最高取樣規格為立體聲24bit/96kHz，並提供Mic in、Line in兩種輸入、以及Line out輸出，Mic in與Line in的差別在於前者具有電壓增益功能。基於提供開發者足夠的使用彈性，原廠保留了若干參數上的調整空間，並且預設晶片關閉。

為了能啟動並正常使用晶片，必須以I2C（Inter-integrated Circuit）傳輸協定對晶片進行初始化，過程中晶片保持在Slave mode，意即聽命於外界指令。在完成初始化後，本音效晶片在數位端的音訊資料傳輸協定亦保有彈性，本實驗採用I2S（Inter-IC Sound）協定，以2’s complement格式serial傳送左、右聲道訊號，並且在傳輸上保持Master mode。

接著，本實驗所設計之錄音機必須記數位資料，可使用DE2-115內建的2MB SRAM。本記憶體word length為16bit，共有1024千個字元，總容量2MB。以本次所設定音訊資料規格（單聲道16bit/32kHz）而言，共可記錄長度為 1024000/32000 = 32秒的聲音訊號。另外， SRAM讀取與寫入均為16-bit parallel格式，恰巧與音訊晶片的取樣位元深度相同，因此每一筆取樣資料即為一個字元的長度。

最後，本實驗功能繁多，因此為了讓使用者能夠快速上手，建議可以在LCD顯示板上，提供淺易懂系統資訊，以LED顯示音量，或以七段顯示器顯示秒數等等。

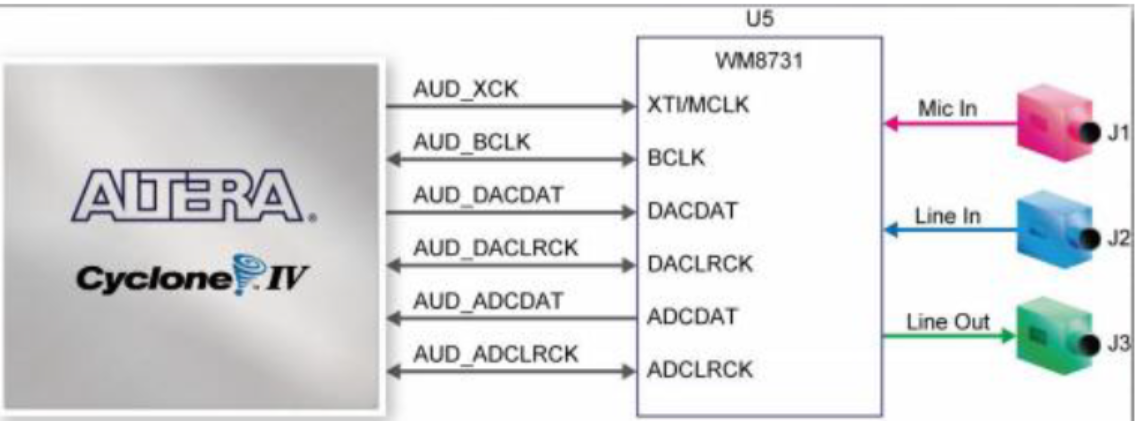
## **實驗架構**

本實驗建議至少包含以下四大部分：PLL、WM8731音訊晶片、Signal processing、Display

1. PLL：包含100kHz與12MHz的clock

多功能錄音機內包含了為數眾的模組，各別所需工作時脈不一，其中與I2C有關的功能需要100kHz的clock；其他部分則使用12MHz的clock。針對不同的時脈需求，可利用Quartus內附的ALTPLL工具來將預設的50MHz的clock轉換為100kHz和12MHz的clock。

1. WM8731音訊晶片：含I2C和I2S

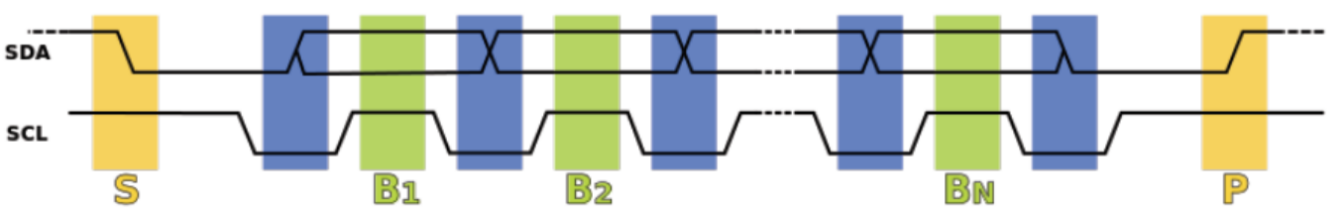


* 1. I2C

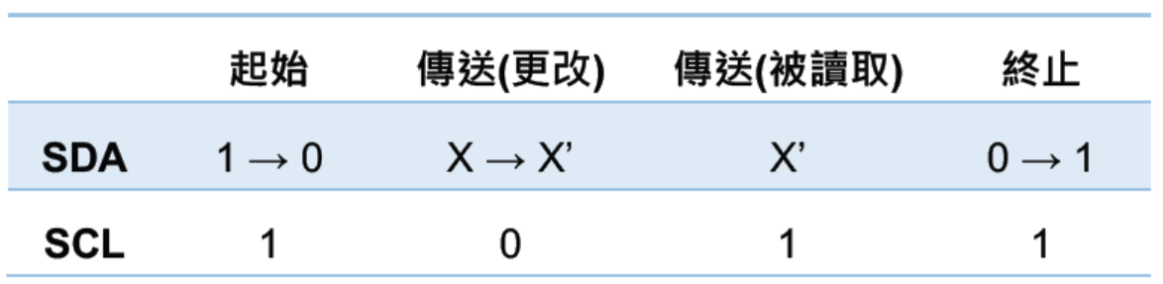
使用WM8731處理音訊前，必須先使用I2C初始化其內部registers，初始化的目的，主要在於訂定傳送格式（I2S Format）、資料精確度（16 bits）、sampling rate（32 kHz）、訊號強度加強等等。可依序寫入下表10筆24 bits的資料來進行初始化：

|  |  |
| --- | --- |
| Left Line In | 0011\_0100\_000\_0000\_0\_1001\_0111 |
| Right Line In | 0011\_0100\_000\_0001\_0\_1001\_0111 |
| Left Headphone Out | 0011\_0100\_000\_0010\_0\_0111\_1001 |
| Right Headphone Out | 0011\_0100\_000\_0011\_0\_0111\_1001 |
| Analogue Audio Path Control | 0011\_0100\_000\_0100\_0\_0001\_0101 |
| Digital Audio Path Control | 0011\_0100\_000\_0101\_0\_0000\_0000 |
| Power Down Control | 0011\_0100\_000\_0110\_0\_0000\_0000 |
| Digital Audio Interface Format | 0011\_0100\_000\_0111\_0\_0100\_0010 |
| Sampling Control | 0011\_0100\_000\_1000\_0\_0001\_1001 |
| Active Control | 0011\_0100\_000\_1001\_0\_0000\_0001 |

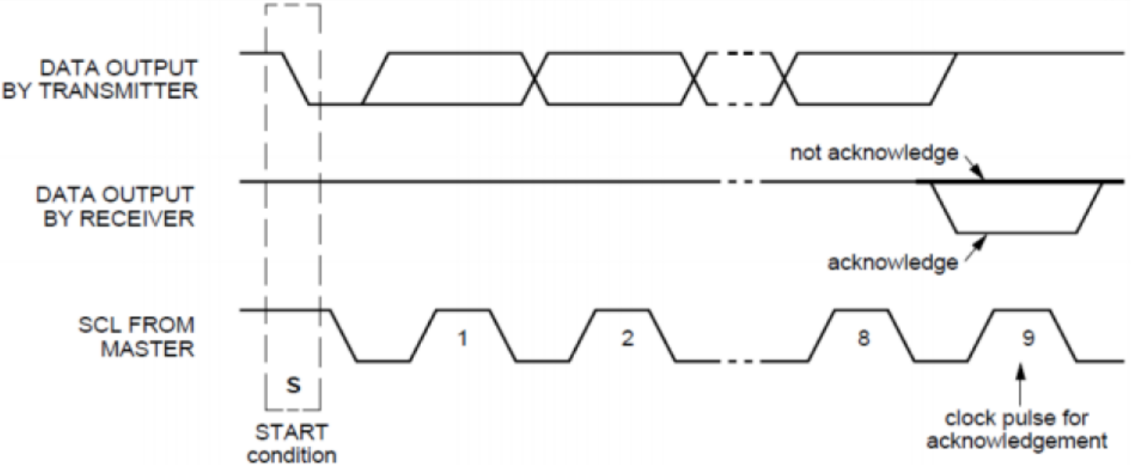
I2C的訊號傳輸仰賴兩條信號線：SDA與SCL（SCLK）。SCL掌控資料傳輸的「狀態」，而SDA則代表要傳輸的「資料」本身。發送資料端控制SCL訊號線告訴接收端SDA的資料是否已穩定可被讀取，1代表資料穩定，0則是正在改動資料。SDA除了代表被傳送的資料外，也負責標示傳送過程的開始與結束。下圖為傳輸信號線控制方式示意圖：



其中整個傳輸過程可以分成三個階段：起始（S）、傳送（B）、終止（P）。



除了上述過程外，每送出8個bit，還需要把SDA設成high impedance（1’bz），使接收端可以回傳ACT（acknowledgement bit）來告訴傳輸者是否已正確接收到訊息，如下圖所示：



而Finite State Machine設計如下：

If Start

SCL is high

Not Start

SCL is low

Not transmit

24 bits

transmit

24 bits

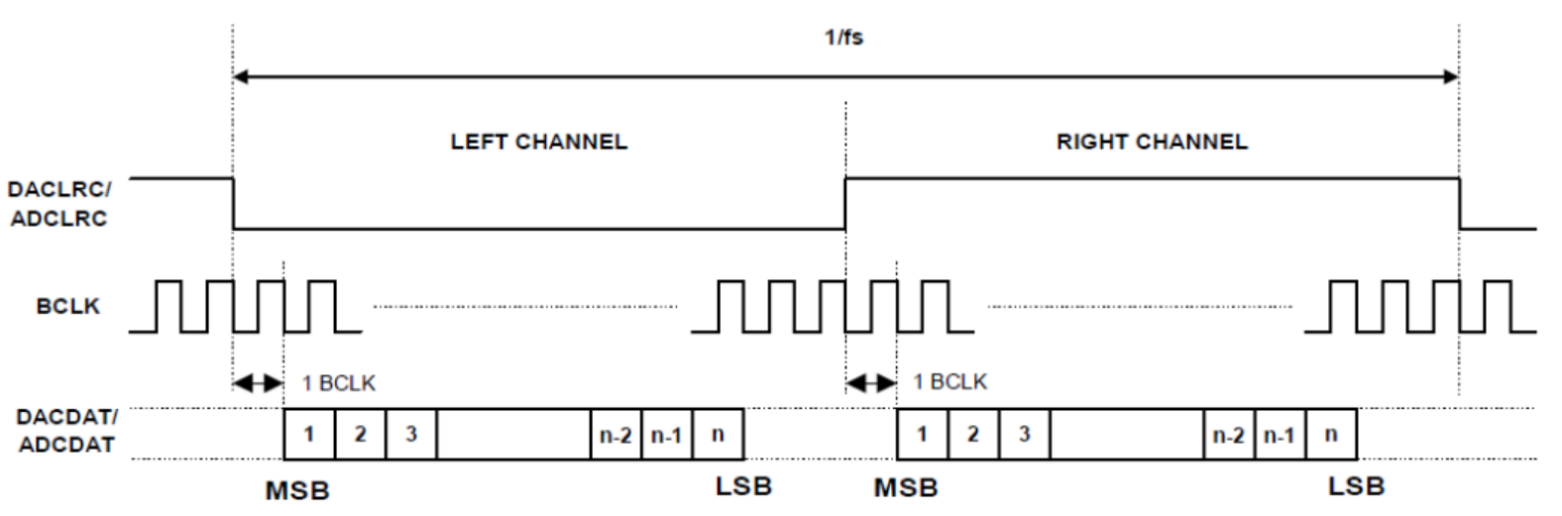
If Pause

Not Pause

需注意：

1. 起始(Start)和終止(Pause)需滿足上述SDA和SCL波形。
2. 只有當SCL為0時才能傳出資料。
3. 執行完S\_IDLE、S\_START、S\_TRANS、S\_FINISH這4個states才會完成傳送24 bits資料，而總共有10筆資訊需要傳送，因此需經10次相同步驟。
   1. I2S

I2S傳輸協定是用來規範與WM8731音效晶片傳輸音訊資料的方法。其中 DACLRC代表由使用者將處理後的音訊送進數位類比轉換器部分，ADCLRC代表使用者由類比數位轉換器接收音訊的部分。在Master mode下，LRC由WM8731傳送給使用者，表示目前在處理的是哪個聲道，低電位代表左聲道，高電位代表右聲道。 DAT則是實際傳送的音訊內容，從LRC改變後的第二個clock開始傳送，直到滿足設定的大小後其餘補0開始傳送，位元數n可自由設定，通常設定為16 bits，而傳輸順序是從MSB到LSB。



在錄音時，由於左右聲道是相同的，所以可以只取一個聲道。但需特別注意的是，傳輸完成後仍會有數個BCLK的時間會繼續給0，因此要藉由LRC的改變來判斷聲道切換，而不要直接計數n次之後就跳轉至另一聲道做讀取。

I2S之Finite State Machine如下圖：

I2C finish

LRC=0

I2C not finish

LRC=0

LRC=1

LRC=1

而上圖僅為傳送過程之大體流程，實際內部操作仍需滿足上述之規範，即LRC剛變換的第一個

cycle不可動作，而之後的16個cycles則依照外部controller的狀態(Record or Play)決定讀取或

寫出資料，完成後需再等待LRC變換後，才可更改當下的STATE。

1. Signal processing：

此部分是此實驗主要的部分，top這個module會和很多modules溝通，包括通訊晶片WM8731、DE2-115裡的SRAM記憶體、輸出錄音機狀態的LCD。

* 1. 從 WM8731接收到音訊後，需對其做數位信號處理。例如錄製時需記錄下儲存的資料尾端位址，以免在播放時出SRAM中沒有使用到的部分。
  2. 調整播放速度：經由改變讀取SRAM記憶體內每筆資料的順序，可以得到不同的播放速度

1. 依序讀取：得到一倍的播放速度；
2. 間隔一取樣點（16 bit）讀取：得到兩倍的播放速度。

根據訊號處理的簡單定律，利用time domain scaling將訊號時間長度減小，會使得frequency domain變寬，也就是聽覺上的音調變高。為使維持不變，可改為間隔多個取樣點讀取。

1. 慢速播放：相鄰兩筆資料中間會有空白需要以內插來填補。

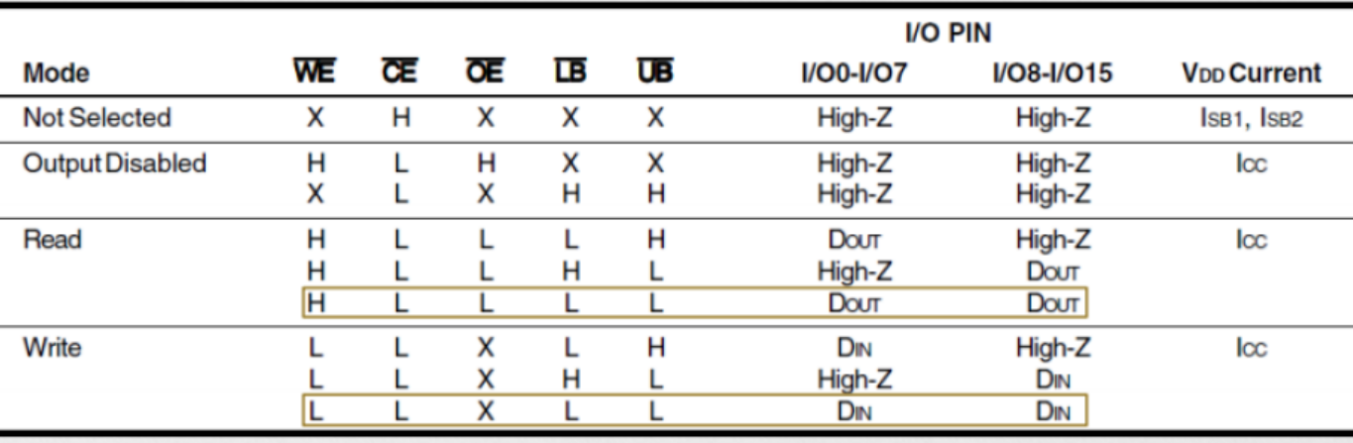
分為零次與線性內插，零次即是以前一筆資料作為空白部分的值；線性內插則是利用前後兩筆資料的線性組合。零次內插處理簡單但失真較大。需要注意的是，在進行線性內插時，需要考慮到正負號以及overflow，可宣告如下：

logic signed [7:0] a, b, c;

c = $signed(a) + $signed(b);

* 1. SRAM

溝通方式如下：

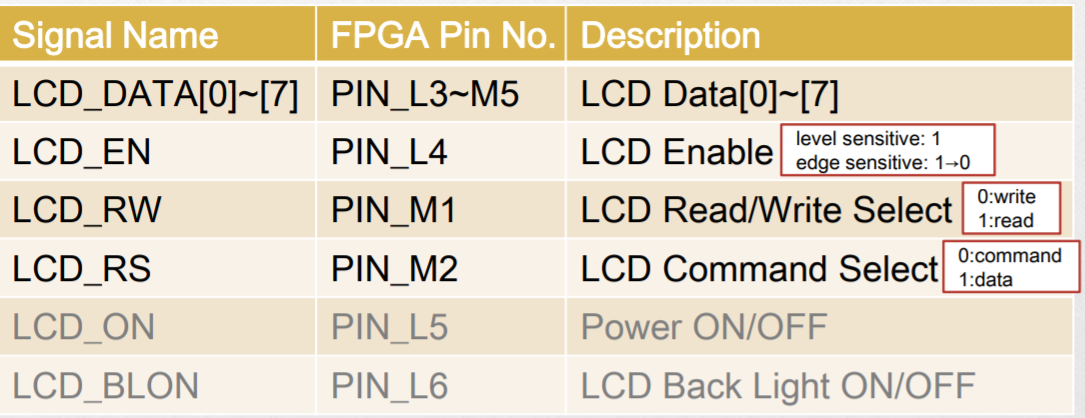


此實驗LB\_n和UB\_n可直接assign為0，因為一次存和取都是16bit，存的時候拉低WE\_n和CE\_n，取的時候拉高WE\_n，拉低CE\_n和OE\_n，sram\_address的話就是要存和去的位置，比較需要注意的是sram\_data為inout port，存入時由top module控制，取的時候由SRAM模組控制，沒有要控制的一方要放開控制，將sram\_address設為High impedance。

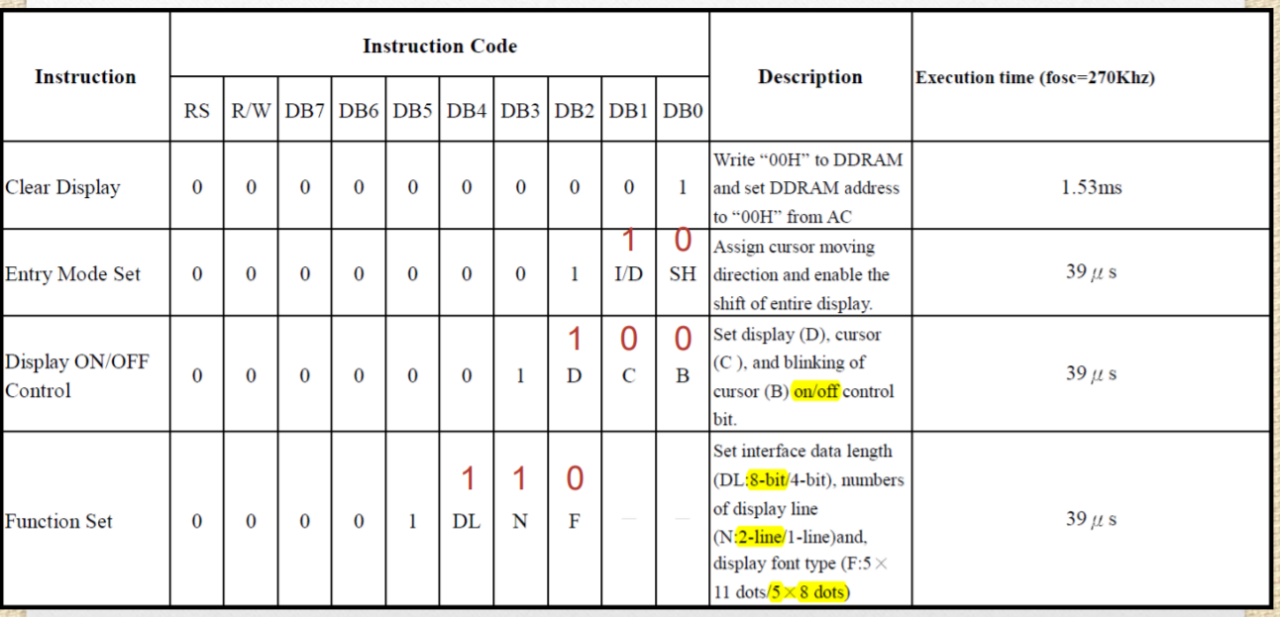
1. Display：
   1. LCD顯示器：

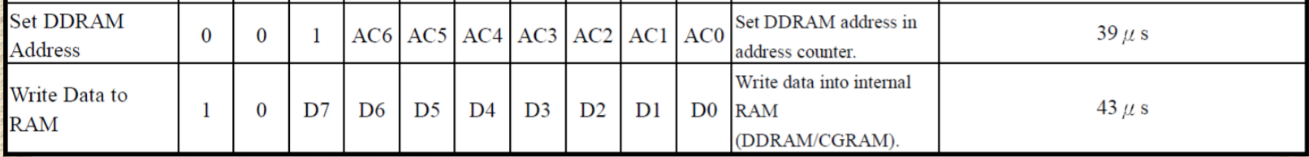
DE2-115的LCD顯示器具有16字元x2行，pin如下圖所示，其中，保持LCD\_ON = 1及LCD\_RW = 0即可，確保顯示器持續開啟以及接受資料的寫入，因為此顯示器無背光功能，故可不理會LCD\_BLON。其餘必須控制訊號為分別：

1. LCD\_EN：在此訊號的falling edge讀取 LCD\_DATALCD\_DATA
2. LCD\_RS：0代表寫入指令，1代表寫入資料
3. LCD\_DATA：欲寫入的訊號，可為指令或資料

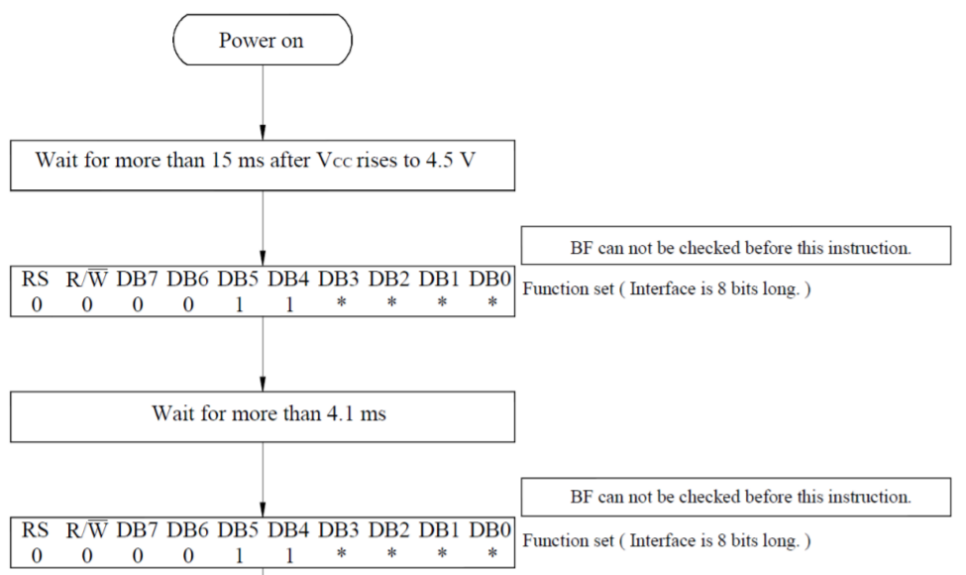


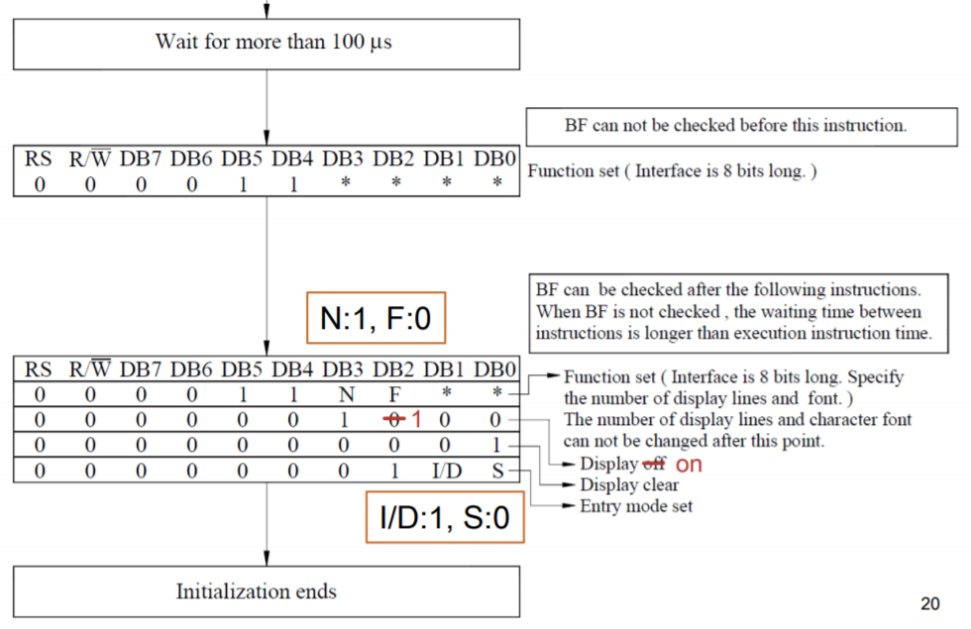
LCD基本指令如下表，可以觀察到每個指令間均需要足夠的時間，因此不建議用頻率太高的clock：





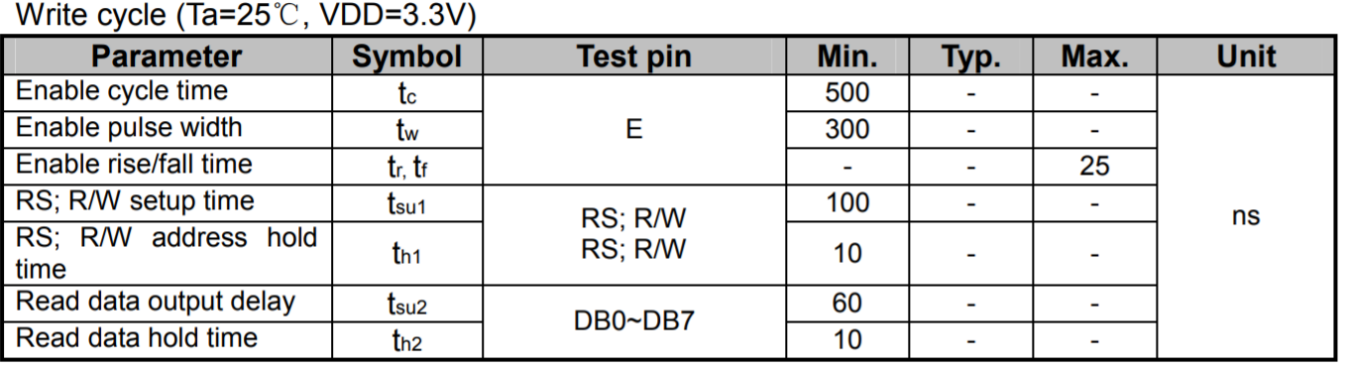
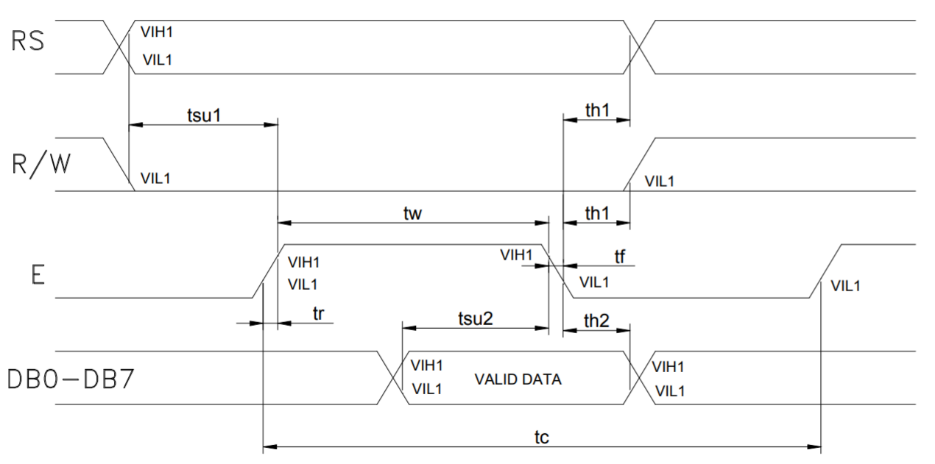
LCD開機時需進行初始化，但之後可直接寫值，初始化流程如下：



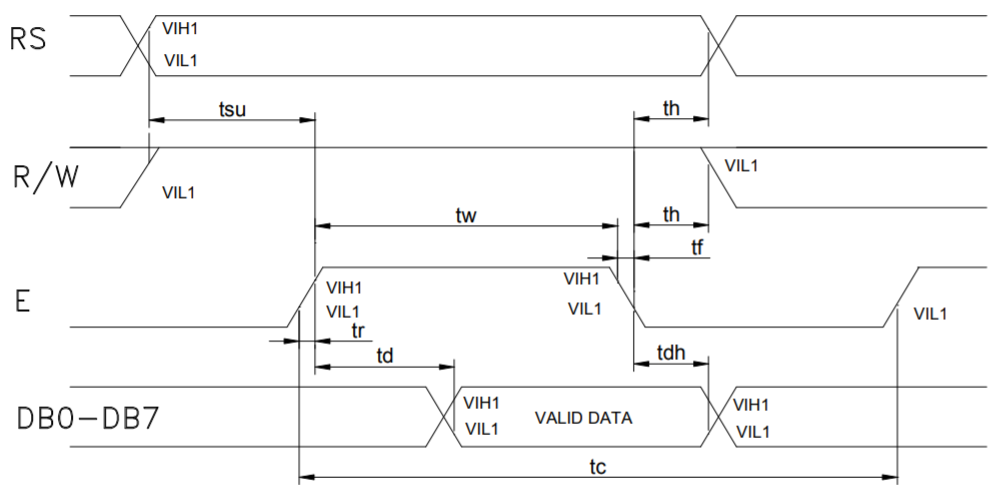
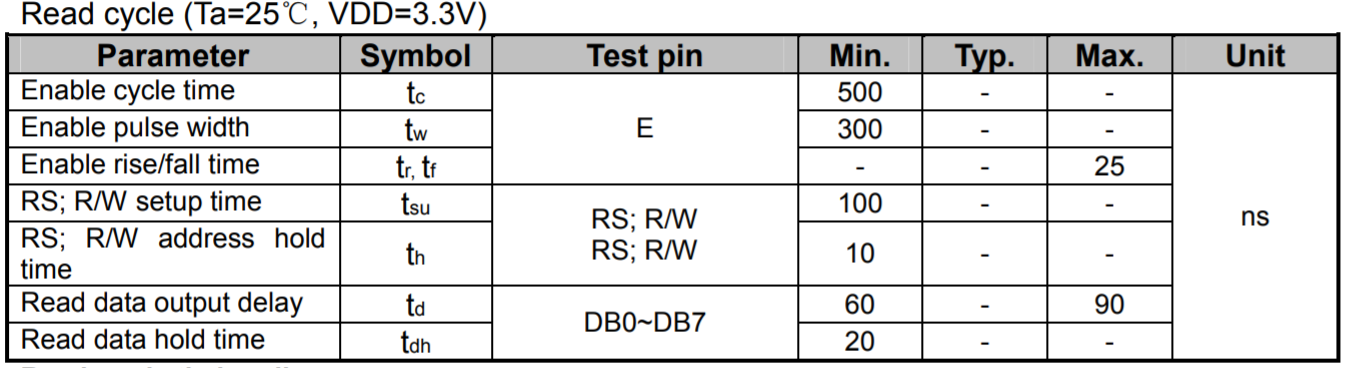


寫值與讀值時同樣需要有充足的時間：

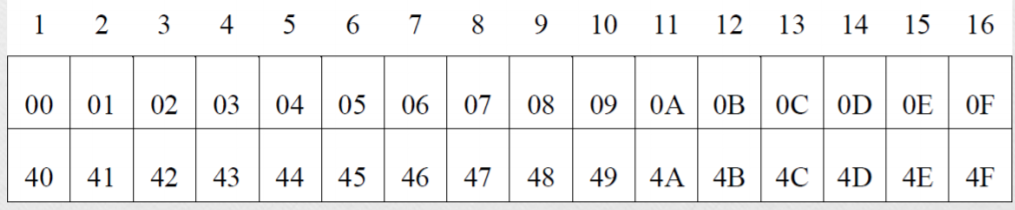
Write cycle



Read cycle

DDRAM addresses與顯示器上對應位置如下：



原廠提供預先設定好的字庫如下，若想自造新字可參考原廠的datasheet：



* 1. Debounce

因為開發平台內建的按鈕並非理想，可能會在0和1之間抖動，所以需要解決按鈕的訊號不夠穩定的問題，電路大致上概念為：在一定時脈週期內，連續出現幾次相同的值才被判有效否則輸出仍維持不變。

## **遇到問題與解決**

* 1. I2C：

這部分需要小心在初始化WM8731時必須遵行一定的規範，SDA和SCL訊號之間的關係要分不同的狀態各自處理，不可出錯，所以在給任何值時，需要謹慎考慮後才可動作，因此花上較多的時間。此外，I2C這部分的程式我們一開始也遇到傳資料時慢了一個CYCLE，導致初始化失敗，而這部分我們是以直接在七段顯示器上輸出BCLK訊號，觀察數值是否有變化來確認初始化是否成功。

* 1. I2S：

I2S我們一開始在設計上有多增設一個除錯機制，多了一個ECHO STATE負責把麥克風接收訊號直接做輸出，就可確認LRC的轉換判斷機制是否正確。而I2S和SRAM之間的溝通我們一開始也遇到一些問題，何時需將enable訊號在兩者間作傳輸，且如何確保enable訊號只會維持一個cycle為high，這部分如果單純用想的很容易就會出錯，因此我們有額外寫testbench去觀測波形，就比較可以確認輸出是在正確的時刻傳出。

* 1. LCD：

LCD是在enable下來時讀值，而每個指令間都需要有足夠的時間，當時間不足時LCD就不會顯示想要的畫面，所以一定要注意datasheet內的規定。