



## 以單晶片微處理器實現反或閘型快閃記憶體分類測試器

陳逸塵\*

高苑科技大學電機工程系

2010 年 6 月 15 日收稿；2010 年 10 月 15 日修訂稿；2010 年 12 月 31 日接受

### 摘要

本文描述以單晶片微處理器實現反或閘型(NOR) 串列週邊介面(Serial-Peripheral-Interface, SPI) 快閃記憶體(Flash Memory) 之分類測試器。以單晶片微處理器內建 SPI 介面來控制讀寫快閃記憶體, 達到比對檢測快閃記憶體 IC 的目地, 並依測試所得容量的大小來分類。此外, 所設計的測試機也能像燒錄器般由樣品 IC 讀取資料後儲存至待燒錄的 IC 內, 同時逐位元組比對資料, 不僅可達到燒錄的目地而且資料的正確性也比目前燒錄器只做加總(Check Sum)比對的方式來得高。雖然所採用的單晶片微處理器只有一組 SPI 介面, 由於寫入資料的時間遠高於讀取資料的時間, 在設計上以同時寫入資料然後逐一分別比對的方式, 可以做到同時比對檢測、燒錄入顆快閃記憶體 IC, 而所用的時間只比檢測單顆 IC 約 2.5 倍的時間。所設計的測試機由於是以單晶片微處理器的架構完成, 電路簡單、低成本但功能卻不遜於市面上的燒錄器, 在 SPI 介面的快閃記憶體的未來逐漸普及的應用上提供了一方便可靠的工具。

**關鍵字：**單晶片微處理器、串列週邊介面、快閃記憶體

### 1. 簡介

快閃記憶體(Flash Memory)由於應用廣泛, 在短短數年內僅次於動態隨機存取記憶體(Dynamic Random Access Memory) 外, 成為最大宗的半導體記憶體產品。快閃記憶體由基本的記憶單元結構可分為反及閘(NAND)與反或閘(NOR)兩大類。NAND Flash 由於結構密度較高, 通常容量高但讀寫速度慢, 多應用於大量資料之儲存如記憶卡、隨身碟、固態硬碟等。也常用於手機、MP3、MP4 等行動電子產品上來儲存媒體資料。而 NOR Flash 則為求高速的讀取所以結構密度較 NAND Flash 低, 容量較小, 多用於 DVD 播放機、機上盒等作為應用程式軟體或韌體之儲存及其他需更新軟體或韌體之電子產品內。近年由於價格低廉, 有開始取代可抹除式唯讀記憶體(EEPROM)成為 BIOS 儲存裝置的趨勢。而 NOR Flash 又因對外接腳的不同可分為串列式(Serial)及並列式(Parallel)兩種。串列式的 NOR Flash 則因為腳數少封裝價格較低, 另也因為讀寫速度提高到可以接受的程度, 故取代並列式的成為目前的主流產品。

本文中則是針對串列式的 NOR Flash, 以單晶片微處理器實現一記憶體之分類測試機。將以單晶片微處理器, 透過內建的 SPI 介面下指令給快閃記憶體, 將測試資料先寫入快閃記憶體積體電路內, 再讀出來與原資料比對, 再反覆以不同資料測試比對, 之後便可檢測快閃記憶體 IC 的記憶單元是否能完整的記錄資料。並依測試所得有效記憶容量的大小分類, 透過 LCD 模組顯示給操作者來完成分類動作。此外, 所設計的測試機也能像燒錄器般由樣品 IC 讀取資料後儲存至待燒錄的 IC 內, 同時逐位元組比對燒錄資料的正確, 不

僅可達到燒錄的目地而且資料的正確性也比目前市場上的燒錄器只做加總(Check Sum)比對的方式來得好。

由於寫入資料的時間遠高於讀取資料的時間[1], 雖然所採用的單晶片微處理器只有一組 SPI 介面, 在設計上我們以同時寫入資料然後再逐一分別讀取比對的方式可以做到同時比對檢測、燒錄入顆快閃記憶體 IC, 而所用的時間只用了檢測單顆 IC 約 2.5 倍的時間。對大量的 IC 檢測、燒錄而言, 可大幅的縮短作業的時間。所設計的測試機由於是以單晶片微處理器的架構完成, 具有電路簡單、低成本的優點, 應可應用於現有快閃記憶體 IC 測試產業上。若能結合自動 IC 處理器(auto handler), 則更能達到高效率的要求。

本文其他章節結構如下。在第二章中我們簡述了 NOR Flash 記憶體的結構及 SPI 的指令功能, 同時也介紹有單晶片微處理器所內建的 SPI 介面與設定方式。第三章則是詳細介紹以單晶片微處理器實現 NOR Flash 記憶體讀寫比對方法及依有效記憶體大小分類之作法。同時也說明如何利用多工方式同時檢驗 8 顆記憶體 IC 以提升使用效率。另也擴充功能使該測試機也具有燒錄比對的功能。而於第四章則是硬體電路說明及相關實驗的結果。第五章則為結論及未來發展方向。

### 2. 相關技術

在本章我們將簡述了 NOR Flash 記憶體的結構方式, 同時也介紹單晶片微處理器所內建的 SPI 介面與設定與 NOR Flash 記憶體連接方式。

NOR Flash 記憶體的結構如圖 1 所示：

\* 通訊作者：陳逸塵

電子郵件信箱：t20034@cc.kyu.edu.tw

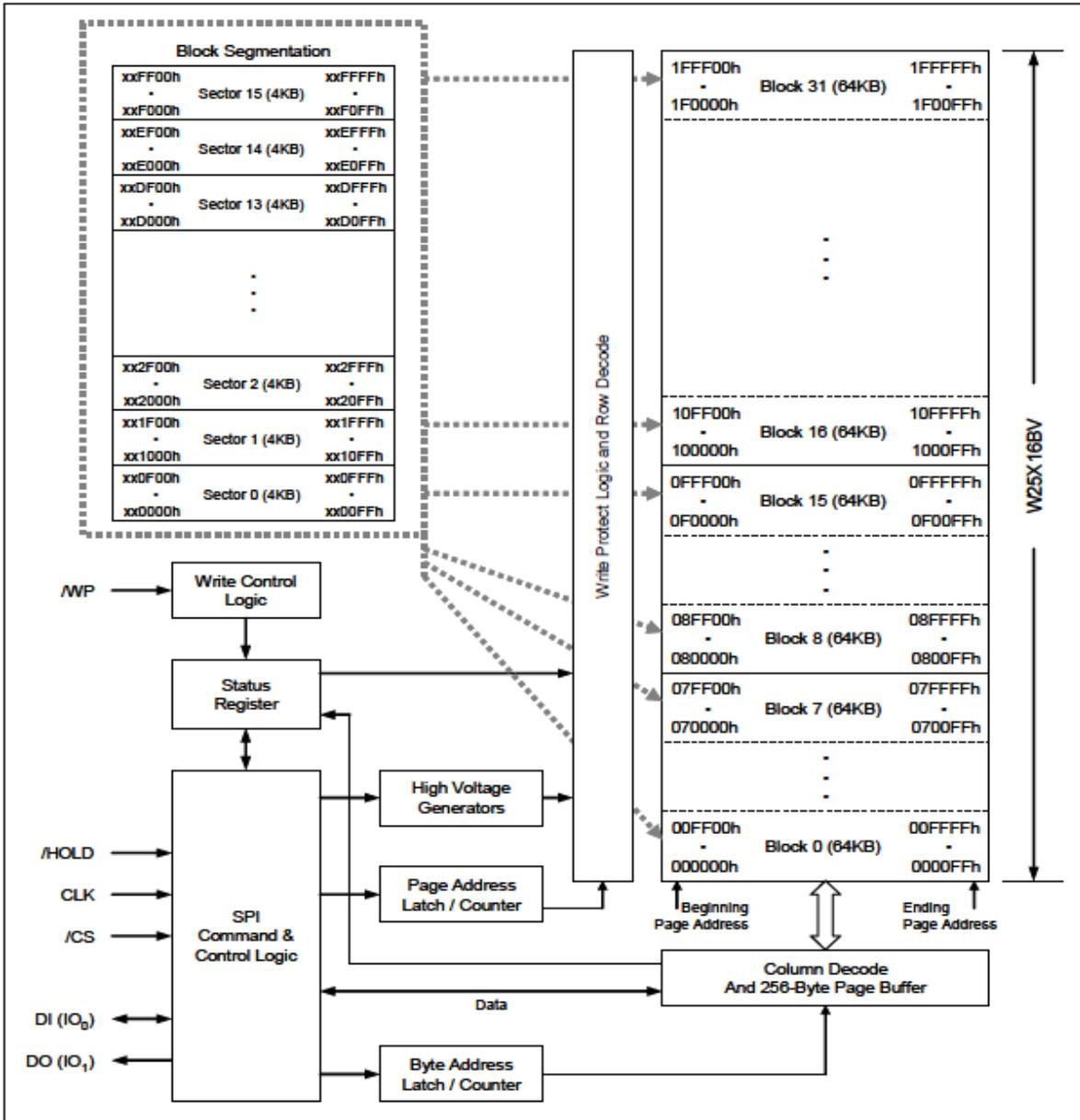


圖 1. NOR Flash 記憶體結構，資料來源：[1]

每一個 Page 由 256 個位元組所組成，Page 為資料的寫入單位，經由 Page Program 指令，一次可寫入 256 個位元組。另外為方便快速抹除資料，通常把 16 個 Page 組合成一個 Sector，16 個 Sector 組合為一 Block。使用者可以下 Sector Erase /Block Erase 指令來抹除 4K/64K 個位元組。甚至可以下 Chip Erase 來一次抹除整個 IC 的資料。

以 WINBOND 的 W25X16BV 為例，其結記憶體容量有 16M 位元或 2M 位元組。其具有 32 個 Block，512 個 Sector 或 8192 個 Page。當使用者下 Page Program 指令時寫入資料會先儲存在 Page Buffer 內，再置入指定的 Page Address 內。有時為防止意外的把程式更改，設計者可利用軟體或硬體的防寫保護來設定防寫區。軟體的防寫保護方式為在狀態暫存器的 Block Protect Bit (BP0, BP1, BP2)及上下區防寫 TB 分別設定程式防寫的範圍，如圖 2 所示。硬體的防寫保護方式則是在 IC 的第三腳有提供防寫開關，如圖 3 所示，該腳接到低電

位時防寫保護開啓，若接到高電位則防寫護關閉。

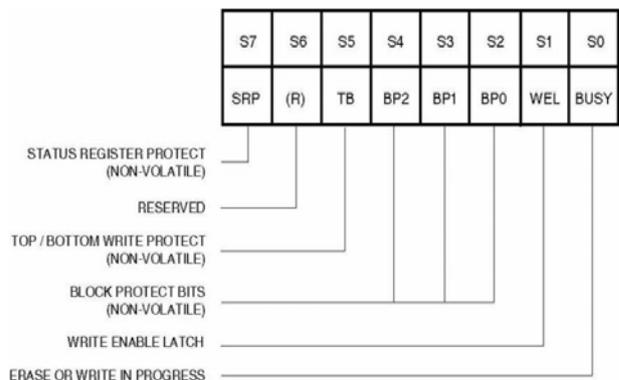


圖 2. 狀態暫存器，資料來源：[1]

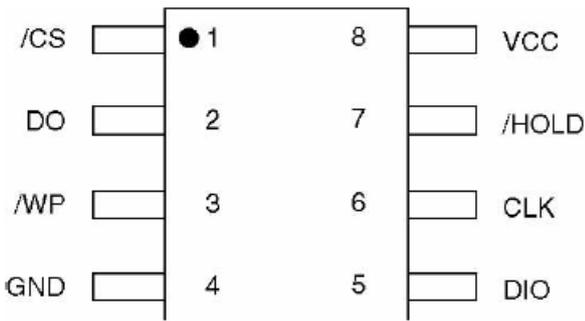


圖 3. IC 接腳定義，資料來源：[1]

該 IC 提供 16 個 SPI 指令，包含 Program、Erase、讀取 Flash ID、讀／寫狀態暫存器及寫入制／禁能(Write Enable/Disable) 五大類。在下 SPI 指令之前須先將 Chip Select(CS)腳設定為低電位表示該 IC 被選擇，否則 IC 將拒絕所有的指令。在下 Program、Erase 或是寫入狀態暫存器指令之前，必須先寫入制能指令，使狀態暫存器的 WEL 位元為 1，IC 才會接受 Program、Erase 或是寫入狀態暫存器指令。當 IC 在執行 Program、Erase 或是寫入狀態暫存器的指令時，IC 會拒絕除了讀取狀態暫存器之外的指令，所以設計者必須在下了上述指令之後檢查狀態暫存器的 Busy 位元，若 Busy 位元為 1 便表示 IC 還在執行指令，無法接受其它指令；若 Busy 位元為 0 則表示 IC 已經將前一個指令執行完畢，可以再下達其它指令。

SPI 介面是一種串列雙向傳輸介面，採用主從式架構(Master/Slave)，連接時可單一 Master 對單一 Slave 連接或單一 Master 對多個 Slave 連接，如圖 4、圖 5 所示。

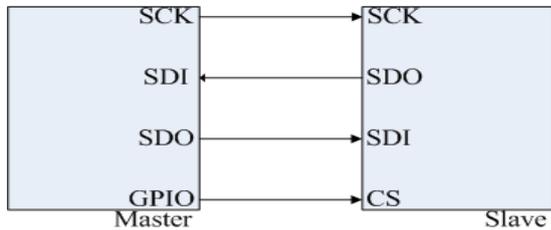


圖 4. 單一 Master 對單一 Slave 連接

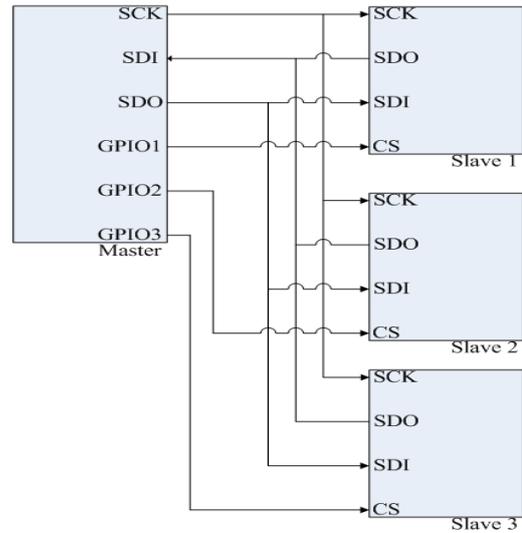


圖 5. 單一 Master 對多個 Slave 連接

SPI 定義了四種信號：SCK(Serial Clock)、SDI(Serial Data In)、SDO(Serial Data Out)及  $\overline{CS}$  (Slave Select)。在通訊初始時，由 Master 端提供 SCK 信號並選擇目標 Slave 端，再由 Master 端發出指令開始傳送／接收資料。在單一 Master 對單一 Slave 連接時，Master 端與 Slave 端的四條信號線是一對一連接，但在單一 Master 對多個 Slave 連接時，Slave 端的 SCK、SDO、SDI 信號是並聯在一起的，所以在 Master 端要寫入資料到 Slave 端時，可以先將所有的 Slave 都選擇後再將資料一次寫入，如此可節省逐一寫入資料所花費的時間。因為這個特點，所以在本論文中我們使用單一 Master 對多個 Slave 的連接方式。以 WINBOND 的 W25X16BV 為例，其 Read Data 指令的時序如圖 6，和單晶片微處理器以 SPI 連接時，單晶片微處理器為 Master，NOR Flash 為 Slave，Master 控制  $\overline{CS}$  並提供 SLK 信號，一開始 Master 將  $\overline{CS}$  設定為低電位，接著提供 CLK 信號及由 Master 端的 SDO 輸出 Read Data 指令(0x03)和要讀取的位址 24 個位元給 Slave，Slave 便在位址位元輸出後的下一個時脈將指定位址的資料由 Slave 的 SDO 傳回給 Master，最後 Master 在讀取了需要的位元數後便可將  $\overline{CS}$  設定為高電位結束 Read Data 指令。

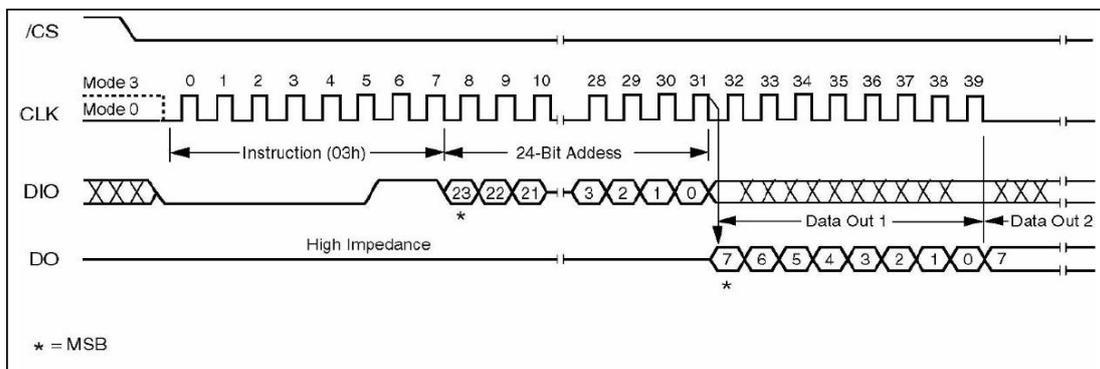


圖 6. Read Data 指令時序，資料來源：[1]

目前市面上已經有許多單晶片微處理器有內建 SPI 介面，如所示。  
以 Microchip 的 PIC 單晶片微處理器來說，其 SPI 模組架構

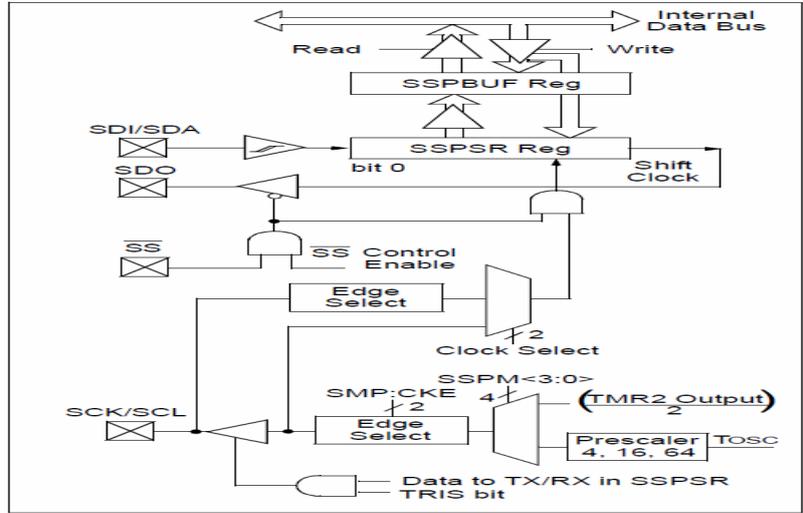


圖 7. SPI 模組架構，資料來源：[2]

我們只要設定單晶片微處理器為 SPI Master 模式及 SPI Clock 就可以開始使用 SPI 模組。資料在傳送時先將資料放在 SSPBUF 暫存器中，單晶片微處理器會自行將 SSPBUF 暫存器的資料放在移位暫存器 SSPSR 內，如此一來在每個 Clock 週期 SSPSR 暫存器會傳送 1 個位元給 Slave 端，在 8 個 Clock 週期後，SSPSR 將其內的資料都傳送完時，會改變狀態暫存器 SSPSTAT 的 BF(Buffer Full)位元代表資料已傳送完成，使用者可再將下一個位元組的資料放入 SSPBUF

暫存器繼續傳送資料。資料接收時則相反，Slave 端會在每個 Clock 週期傳來 1 個位元的資料給 SSPSR 暫存器，當 SSPSR 暫存器接收了 8 個位元時會將資料放在 SSPBUF 暫存器，並改變 SSPSTAT 暫存器的 BF 位元代表 SSPBUF 暫存器內已經接收到資料，使用者必須自行將 SSPBUF 暫存器內的資料移到記憶體中，否則再過 8 個 Clock 週期後 SSPSR 暫存器會將新的資料覆蓋過去。

在 Microchip 的 PIC18F45K20 中 SPI 組態的設定如下所示：

R/W-0							
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

<b>Legend:</b>			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)  
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared by software)  
0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit<sup>(1)</sup>  
**SPI Slave mode:**  
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared by software).  
0 = No overflow
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit<sup>(2)</sup>  
1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins. When enabled, the SDA and SCL pins must be configured as inputs.  
0 = Disables serial port and configures these pins as I/O port pins
- bit 4 **CKP:** Clock Polarity Select bit  
1 = Idle state for clock is a high level  
0 = Idle state for clock is a low level
- bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits<sup>(3)</sup>  
0101 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control disabled,  $\overline{SS}$  can be used as I/O pin  
0100 = SPI Slave mode, clock = SCK pin, SS pin control enabled  
0011 = SPI Master mode, clock = TMR2 output/2  
0010 = SPI Master mode, clock = Fosc/64  
0001 = SPI Master mode, clock = Fosc/16  
0000 = SPI Master mode, clock = Fosc/4

圖 8. PIC18F45K20 中 SPI 組態的設定：[2]

### 3. 硬體架構及系統流程

在本論文的系統架構如圖 9 所示，我們使用單晶片微處理器為 Microchip 的 PIC18F45K20，該單晶片微處理器有一個 SPI 通道、3 個可中斷計時器及 5 組 8 位元 I/O 可供我們使用[2]。此外在硬體架構上還包含樣品 IC，與待測 IC Flash\_1 至 Flash\_8。樣品 IC 我們稱為 Golden Sample，除了提供給測試機待測 IC 的 ID 及規格外，要測試的樣本資料或要燒錄的資料也會存放在樣品 IC 內。另為方便操作，在系統中我們也加入了二列 16 字元的 LCM 來顯示分類結果及 3 個功能按鍵與一個重置鍵。

樣品 IC，與待測 IC Flash\_1 至 Flash\_8 採共用匯流排的方式，也就是如圖 Master 對多個 Slave 連接方式。待測 IC

Flash\_1 至 Flash\_8 的  $\overline{CS}$  腳位則分為接在 PORT\_D 的 0~7 腳上。在實作上我們若要同時寫入 8 個待測 IC，因為 SCK、DI、DO 均並聯在一起，我們只要將 PORT\_D，也就是 8 個待測 IC 的  $\overline{CS}$  都設定為低電位，就可以同時下指令給 8 個待測 IC，在指令完成後再改為高電位即可。所以 Chip Erase、Block Erase、Page Program 及 Write Enable 會對待測 IC 做寫入動作的指令都可利用這種方式減少測試所需要的時間。而當需分別對其中一 IC 動作時則把 PORT\_ROTATE 變數中的值設為 0xFE(1111 1110)，再將 PORT\_ROTATE 的內容移到 PORTD 便可選擇 Flash1。在完成 Flash1 的動作之後，我們將 PORT\_ROTATE 的資料左移為 0xFD(1111 1101)，如此就可以循序處理每個待測 IC 了。

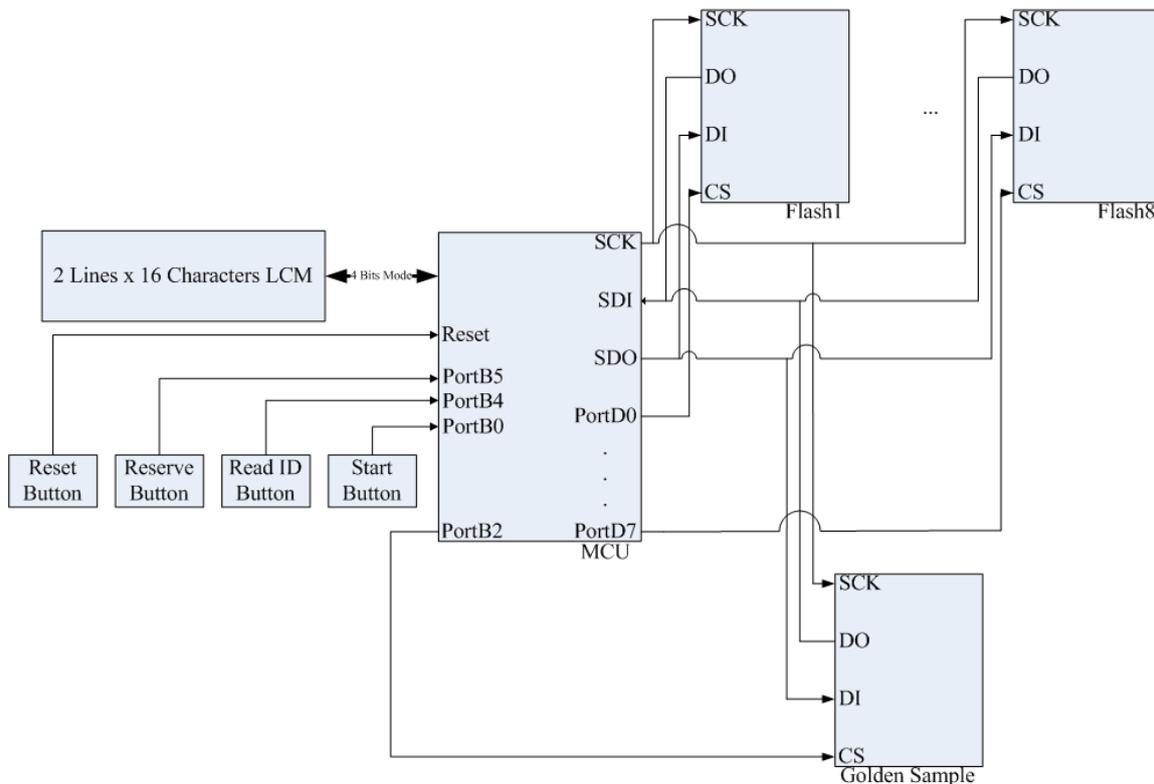


圖 9. 系統架構

在本論文中我們除了測試 Flash 內每個位元的讀寫是否正確，也會測試 Flash 的 ID、防寫功能及 Erase/Program 的反應時間。Flash 的 ID 有製造商 ID (Manufacturer ID, MID)、裝置 ID (Device ID, DID) 及 JEDEC ID (JID)，以 WINBOND 的 W25X16BV 為例，其 ID 如表 1 所示，製造商 ID 0xEF 代表 WINBOND，裝置 ID 0x14 代表容量，JEDEC ID 的 0x30 為記憶體類型，0x15 則是容量。該 Flash 有 3 個指令可以讀取 ID，Release Power Down/Device ID 指令可傳回裝置 ID，Read Manufacturer/Device ID 可傳回製造商 ID 及裝置 ID，JEDEC ID 指令則是傳回 JEDEC ID，在測試時我們為了確保 Flash 功能的正確，上述 3 個指令都會進行測試。

表 1. W25X16BV 的 ID

Manufacturer ID	0xEF
Device ID	0x14
JEDEC ID	0x3015

測試流程如圖 10 所示，首先我們會先讀取樣品 IC 的製造商 ID、裝置 ID 及 JEDEC ID，由樣品 IC 的 ID 我們可以由查表得知容量及 Block 數目。接著檢查待測 IC 的 VCC 是否有對 GND 短路，再分別讀取 8 個待測 Flash 的製造商 ID、裝置 ID 及 JEDEC ID，這三種 ID 只要其中之一與樣品 IC 相同我們便可知待測 IC 和樣品 IC 有相同容量，若待測 IC 的三種 ID 都和樣品 IC 的 ID 相異時，代表該待測 IC 讀取 ID 的功能不良或是待測 IC 與樣品 IC 的容量不同。

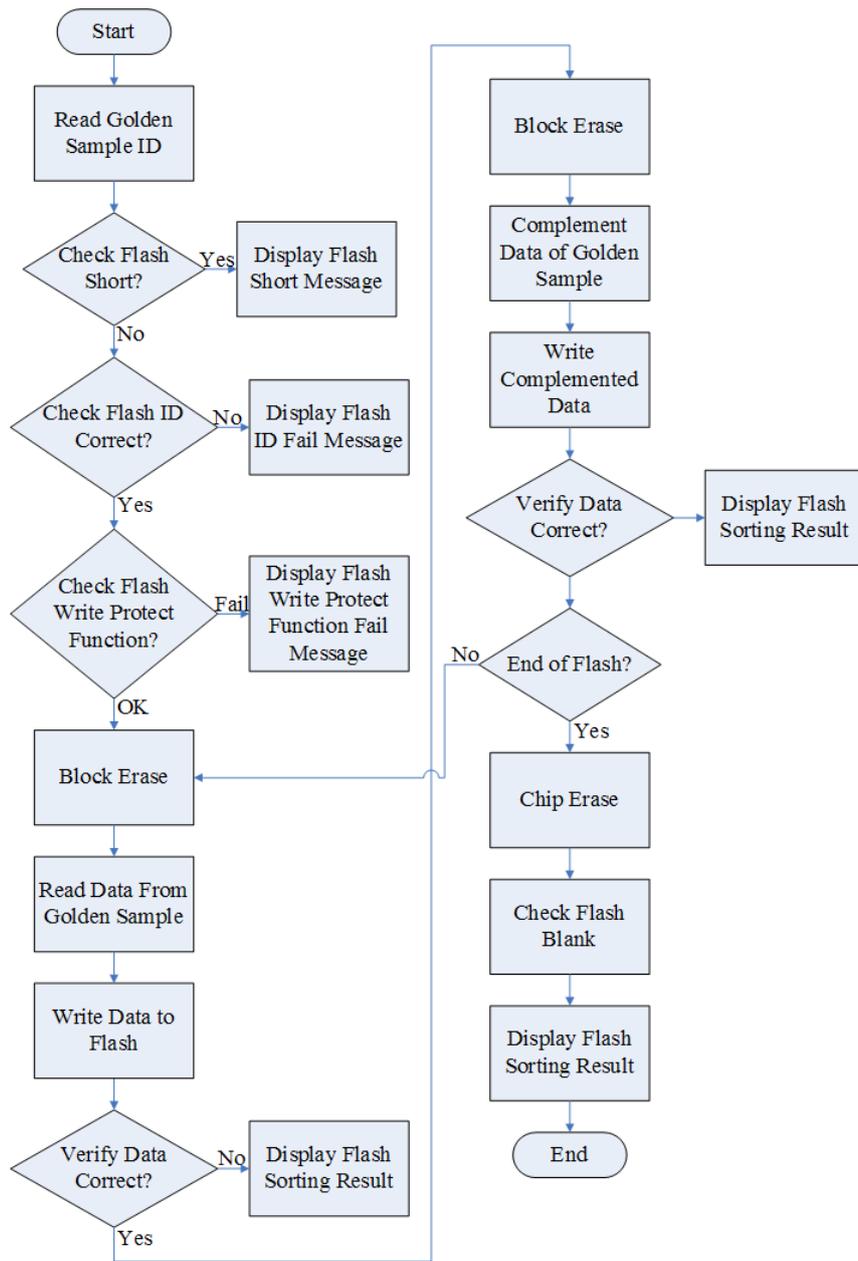


圖 10. 測試流程

接下來我們同時對 8 個待測 IC 下 Block Erase 指令，將第一個 Block 內的資料清除，等到 8 個待測 IC 都 Erase 結束後，從樣品 IC 第一個 Block 的第一個 Page 讀取資料到單晶片微處理器的記憶體中，然後同時對 8 個待測 IC 下 Page Program 指令，待寫入完成後再逐一比對各待測 IC 內的資料是否與樣品 IC 相同。重覆上述動作 256 次後第一個 Block 寫入及比對完成，我們再將 8 個待測 IC 的第一個 Block 內的資料清除一次，重新從樣品 IC 第一個 Block 的第一個 Page 讀取資料，並做 1 的補數運算再存到單晶片微處理器的記憶體中，重覆寫入與比對的動作直到第一個 Block 結束。如此待測 IC 的第一個 Block 內所有的位元都有寫過 0 與 1，可確保各位元不管要寫入何種狀態都可正確無誤，若待測 IC 若

在寫入或比對時發生不良，便記錄該待測 IC 不良發生時 Block 數，以便最後計算分類結果。對待測 IC 其它 Block 重覆以上動作將全部的 Block 都測試過後，我們再對各待測 IC 下 Chip Erase 指令並檢查待測 IC 內的資料是否清除成功，即可知道待測 IC 的各功能是否正常，資料是否可正確寫入或清除，最後再顯示分類結果如圖 11 所示，第一列表表示待測 IC 位置，第二列為分類結果。每個位置的待測 IC 佔用 2x2 個 LCM 顯示位置，分類結果如表 2 所示，Bin 1 代表待測 IC 為正常良品，從 Bin 2 開始可用容量每一級降一半容量，若是讀取製造商 ID、裝置 ID 及 JEDEC ID 都失敗便分類到 10，狀態暫存器的防寫位元或是硬體上的防寫腳位功能不良導至 IC 無法寫入則分類為 20。

待測IC位置	1	2	3	4	5	6	7	8
分類結果	1	1	0	1	3	2	0	2

圖 11. LCM 顯示分類結果

表 2. 分類結果

分類結果	說明
Bin 1	全容量可用
Bin 2	1/2 容量可用
Bin 3	1/4 容量可用
Bin 4	1/8 容量可用
Bin 5	1/16 容量可用
Bin 6	1/32 容量可用
Bin 7	1/64 容量可用
10	讀取 ID 失敗
20	防寫無法關閉

以寫入資料的 PageProgramAll 函式為，例如圖 13，我們先從樣品 IC 讀取一個 Page 的資料到 Data Memory 的 0x200 到 0x2FF 中，如圖 12，再呼叫 PageProgramAll 函式。在 PageProgramAll 中先將 PORTD 輸出 0x00 選擇 8 個待測 IC，接著對待測 IC 下 Page Program 指令，將 Data Memory 中的 0x200 到 0x2FF 的資料寫入全部的待測 IC，最後再設定 PORTD 為 0xFF 取消選擇待測 IC 就完成同時寫入資料的動作。依[1]所記載的資料，Page Program 的時間最多為 3ms，考量到一般使用者的使用狀況並不會要需如此快的反應時間，我們在寫入的指令結束後會等待 1 秒，若 1 秒過後待測 IC 還處於 Busy 狀態，便判定該 IC 在 Page Program 的功能有異常並顯示其分類結果。

```

SUB SourcePageRead
  SET PORTB2 to 0
  SPI OUTPUT 0x03 // Read Data
  SPI OUTPUT Address
  FOR Data Buffer = 0x200 to 0x2FF
    SPI INPUT to Data Buffer
  END FOR
  SET PORTB2 to 1
END SUB
    
```

圖 12. SourcePageRead 虛擬碼

```

SUB PageProgramAll
  SET PORTD to 0x00
  FOR Data Buffer = 0x200 to 0x2FF
    SPI OUTPUT 0x02 // Page Program
    SPI OUTPUT Address
    SPI OUTPUT Data Buffer
  END FOR
  SET PORTD to 0xFF
  CALL CheckTimeout
END SUB
    
```

圖 13. PageProgramAll 虛擬碼

但是在下 Read Data、Read Status Register、Release Power Down/Device ID、Read Manufacturer/Device ID 及 JEDEC ID 這類讀取指令時，若採用相同的方式一次讀取全部待測 IC 則會因為資料線都並接在一起產生資料碰撞的問題，所以在讀取 IC 時必須逐一個別下指令。

以比對資料的 PageCompareAll 為例，如圖 14，初始時我們將 0xFE(1111 1110)放在 PORT\_ROTATE 變數中，再將 PORT\_ROTATE 的內容移到 PORTD 便可選擇 Flash1，接下來我們就可以針對 Flash1 下 Read Data 指令，讀取 1 個 Page 的資料到 Data Memory 的 0x100 到 0x1FF 內，同時和 Data Memory 的 0x200 到 0x2FF 內的資料做比對，便可得知寫入的資料是否正確。在比對完 Flash1 之後，我們將 PORT\_ROTATE 的資料左移為 0xFD(1111 1101)，如此就可以循序比對每個待測 IC 的資料。測試時為了確定 Flash 中每個位元都可正確定入高電位及低電位，在測試完 1 個 Block 後，我們會將樣本資料做 1 補數運算後再重新測試一次，如此一來便可確保 Flash 各位元的功能是否正常。

```

SUB PageCompareAll
  SET PORT_ROTATE to 0xFF

  FOR Flash = 1 to 8
    SET PORTD TO PORT_ROTATE
    FOR IC Data Buffer = 0x100 to 0x1FF, Data Buffer = 0x200 to 0x2FF
      SPI OUTPUT 0x03 // Read Data
      SPI OUTPUT Address
      SPI INPUT to IC_Data_Buffer
      // Compare IC Data Buffer with Data Buffer
      IF IC Data Buffer != Data Buffer THEN
        DISPLAY SORTING RESULT
      END IF
    END FOR
    SET PORTD to 0xFF
    PORT_ROTATE ROTATE LEFT
  END FOR
END SUB
    
```

圖 14. PageCompareAll 虛擬碼

常用的測試樣本資料有 0x00FF (先寫低電位再寫高電位)、0xFF00 (先寫高電位再寫低電位)、模擬高頻信號的 0x55AA 或 0xAA55、反覆寫入 0x15(0001 0101)及 0x01(0000 0001)的高低頻信號或是採用亂數資料。我們只要在程式修改於 Data Memory 的 0x200 到 0x2FF 內放入各式測試樣本資料，便可執行寫入與比對。

#### 4. 實驗結果

實作出的測試器如圖 15 所示，CPU 使用 Microchip 的 PIC18F45K20，4 個功能按鍵及一個 2 列 x16 字的 LCM 顯示分類結果和錯誤訊息。開發環境使用的是 Microchip MPLAB IDE[3]以組合語言[4]開發完成，最後使用 Microchip ICD3[5]將編譯後的二進位碼燒寫至 PIC18F45K20 內。

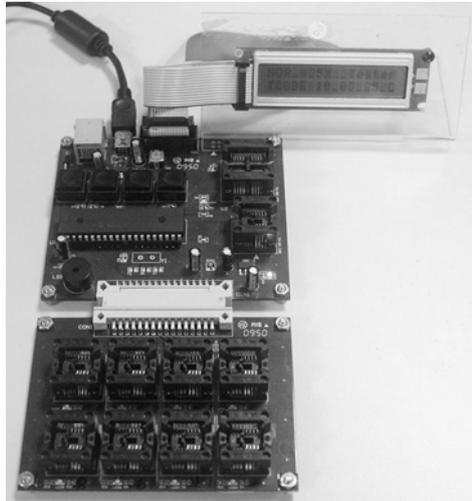


圖 15. 測試器實體

實驗時使用的 NOR Flash 為 WINBOND 的 W25X16BV，容量為 16Mbit。這些 IC 都先以市面上常見的燒錄機測試過，燒錄所用的資料為亂數內容，測試結果 OK 的有 90 個，Fail 的有 2 個，數量共有 92 個。接下來以本論文所實作出的測試機對以上 IC 加以分類，結果如表 3 所示：

表 3. 與一般燒錄機測試結果比較

測試平台	測試結果				
	OK				Fail
一般燒錄機	90				2
本論文的測試機	Bin 1	Bin 2	Bin 4	Bin 6	Bin 2
	84	4	1	1	2

一般燒錄機上測試為 Fail 的 IC 在我們的測試機上分類為 Bin 2，我們進一步使用 MPLAB IDE 的除錯模式(Debug Mode)可以看到從待測 IC 讀回來的資料及發生錯誤的位址，確定這 2 個 IC 是有些位元的狀態無法改變。但是在一般燒錄機上測試為 OK 的 IC 於我們的測試機上測試出有 4 個 Bin 2、1 個 Bin 4 及 1 個 Bin 6 共 6 個不良的 IC，我們在除錯模式中觀察發現，這些 IC 都是在下了 Block Erase 的指

令後，程式在等待 IC 的 Busy 狀態回復時超過了我們預定的 1 秒所造成的逾時。

由於 Block Erase 的時間依[1]所標示為 1 秒，所以我們將 Block Erase 的等待時間延長至 2 秒後再做一次實驗，以下 Block Erase 等待 1 秒的版本為 V1，等待 2 秒則為 V2，實驗結果如表 4 所示：

表 4. 將 Block Erase 等待時間延長後的測試比較

測試平台	測試結果				
	OK				Fail
一般燒錄機	90				2
本論文的測試機 V1	Bin 1	Bin 2	Bin 4	Bin 6	Bin 2
	84	4	1	1	2
本論文的測試機 V2	Bin 1				Bin 2
	90				2

在延長 Block Erase 的等待時間後，在一般燒錄機上測試為 OK 的 IC 在分類後都是 Bin 1，而在一般燒錄機上測試為 Fail

的 IC 則因為其不良狀況並不是 Block Erase 所造成，所以分類結果維持在 Bin 2。由此我們可以得知一般市面的燒錄機

在設計上為了穩定性，都將等待的時間延長以減少不良的產生。

我們也嘗試使用其它的測試樣本（如：0x55AA、0xAA55），實驗結果發現在所測試的 IC 內結果均相同。最後我們比較測試所需要的時間，結果如下：

**表 5. 將 Block Erase 等待時間延長後的測試比較**

測試平台	測試時間
一般燒錄機	2 分 56 秒
本論文的測試機 V2	1 分 46 秒

可以看出本論文的測試機的測試時間比一般燒錄機快 1 分以上，並且可以將 IC 依可用容量加以分類，不同於一般燒錄機只有燒錄 OK 與 Fail 的二分法，大大提高測試的效率及 IC 的可用性。

## 5. 結論及未來發展方向

在本論文所設計的測試機由於是以單晶片微處理器的架構完成，具有電路簡單、低成本的優點，但功能卻不遜於市面上的燒錄器；在 SPI 介面的快閃記憶體的未來逐漸普及的應用上提供了一方便可靠的工具。

另由於 SPI 介面的快閃記憶體在未來將朝雙通道、四通道等多通道高速 SPI 介面前進，也將持續研發適用相關 NOR Flash 記憶體的檢測機。

## 6. 參考文獻

1. Winbond, 16M-BIT SERIAL FLASH MEMORY WITH 4KB SECTORS AND DUAL OUTPUT SPI, 2009
2. Microchip, PIC18F23K20/24K20/25K20/26K20/43K20/44K20/45K20/46K20 Data Sheet, 2009
3. MPLAB IDE User's Guide, Microchip, 2009
4. 趙春棠，PIC 單晶片學習祕笈，全威圖書，台北縣、台灣，2007。
5. MPLAB ICD3 User's Guide, Microchip, 2008



高苑學報 第十七卷 第一期 第 47-56 頁 民國一〇〇年  
Journal of Kao Yuan University Vol. 17, No. 1 (2011) 47-56

ISSN: 2075-745X

**高苑學報**  
**Journal of Kao Yuan**  
**University**

---

## Implementation of Micro-Controller-Based NOR Flash Sorting Tester

Y. C. Chen\*

Department of Electrical Engineering, Kao Yuan University, Kaohsiung, Taiwan, R. O. C.

Received 15 June 2010; Revised 15 October 2010; Accepted 31 December 2010

---

### Abstract

This paper describes the implementation of a micro-controller-based tester for NOR-Flash memory. Using the embedded SPI interface, we may program then read back to compare the testing patterns. Therefore, Flash IC's are sorted according to their effective memory size. The implemented tester can also be used as a pre-loader to read the data in source Flash IC then program into target IC's. The byte-to-byte comparison algorithm, instead of check-sum used in market available pre-loader, is used to guarantee the preload quality. Since the adopted microcontroller has one SPI interface only, the methodology of together programming then compares one by one, makes it possible to test or program 8 pieces of Flash IC at the same time. When comparing with testing single Flash IC, only 2.5 times testing period is required. The tester is implemented using single chip microcontroller, it will be cost effective. We believe that it will be a useful tool in NOR-Flash Memory applications.

Keywords: Micro-controller-based, SPI interface, Flash Memory

---

---

\*Correspondence to: Y. C. Chen

E-mail address: [t20034@cc.kyu.edu.tw](mailto:t20034@cc.kyu.edu.tw)