

# 基於 Avalon Bus 實現 8051 之同步化多重 I/O 控制

吳秋宏

淡江大學電機工程學系

Email: 697450772@s97.tku.edu.tw

李世安

淡江大學電機工程學系

Email: lishyhan@gmail.com

李維聰

淡江大學電機工程學系

Email: wtlee@mail.tku.edu.tw

**摘要**—近幾年來，微處理器一直被廣泛的應用在嵌入式系統與簡單手持式系統上。大多數的嵌入式系統，並不需要強大的處理能力。只需要高適應、高擴充、高移植、低成本、低耗能即可。因此 8051 仍然被廣泛的使用在簡單的單一系統上。然而，其內部 I/O 埠的限制與單一匯流排的限制，使其擴充性與操控性一直無法提升。

本篇論文主要的研究目的，在於改進 8051 內部 I/O 埠不足的因素與單一匯流排的限制。將其外部記憶體定址的架構，透過我們所設計的 System Interconnect Fabric 模組來連接至 Avalon Interface 上。經由 memory-mapped I/O 裝置，以提升其擴充性；並透過 Avalon 匯流排的特性，以達成同步化控制 I/O 的能力來提升其效能。

**關鍵詞**—8051, Avalon Bus, Computer Architecture, FPGA, Verilog HDL

## 一、前言

微處理器在現今一直被廣泛的使用在許多的應用上，例如：嵌入式系統。在大多數的應用裡，都必需要求微處理器可以運作在低耗能與不同的作業環境裡。

8051 微處理器是隸屬於 MCS-51 家族，是 Intel 公司在 1980 年所設計的 8 位元處理器。它包含了 4KB 的內部程式記憶體(外部可擴充到 64KB)、128B 的內部資料記憶體(外部可擴充到 64KB)、4 組 8 位元的 I/O 埠、2 組 16 位元的計時/計數器等[6]。

由於 8051 是複雜指令集(CISC)且指令長度不一致，與 MIPS 或 OpenRISC1200 是精簡指令集(RISC)且指令長度一致，有著很大的差異。因此，在指令集解碼的設計或管線式指令集解碼的設計，都會變的複雜且困難 [2][4][7][9]。

在傳統的匯流排架構裡，一個中心仲裁器控

制著多個主裝置與從裝置之間的溝通。然而，這樣的結構會產生出一個瓶頸，就是在任何時刻只能有一個主裝置能造訪系統匯流排。而 Avalon 匯流排的開關則是使用一種從裝置仲裁技術(slave-side arbitration)，它能允許多個主裝置控制器能真正地同步操作。當有多個主裝置造訪同一個從裝置時，從裝置仲裁器將會決定哪個主裝置會獲得造訪權[3][5][8]。

因為 8051 的內部 4 組 I/O 埠限制與單一匯流排的限制，使得 8051 無法同時對 I/O 裝置進行同步化的控制。因此，我們希望透過 Avalon Interface 的 memory-mapped 方式，將 I/O 映射至外部記憶體，使其周邊裝置不再受內部 4 組 I/O 的限制，而是以記憶體映射周邊裝置的方式。所以，我們將 8051 透過我們所設計的 System Interconnect Fabric 模組並連結至 Avalon Interface。透過 Avalon 匯流排的特性，可以使得 8051 達成真正同步化控制 I/O 的能力並提升其效能。

在本篇論文裡將分為以下說明：第二節將介紹 Avalon Interface 規格；第三節將說明 System Interconnect Fabric 模組的設計；第四節將說明模擬環境與模組實現；最後是總結以及未來可以再改進的部份。

## 二、Avalon Interface 規格

Avalon Interface 是被設計成可以在可程式化單晶片系統(SOPC)環境上，能夠以簡單的設計方式連接微處理器和周邊裝置的一種匯流排協定。這個協定能提供周邊裝置的設計者，依據記憶體位址的讀寫界面描繪，來建立其周邊裝置的主從端(master/slave)。依據這個協定，可以定義周邊裝置與開關結構之間的連結轉換結構和時序的狀態。基於 Avalon Interface，一個典型的系

統可以結合多種功能模組，也可稱為 Avalon 周邊裝置。Avalon 交換結構是一個晶片連結的邏輯單元，可以一起連接 Avalon 周邊裝置，已構成一個更大的系統。圖 1.顯示出 Avalon Interface 經由 Avalon 交換結構連結多個 Avalon 周邊裝置 [1]。

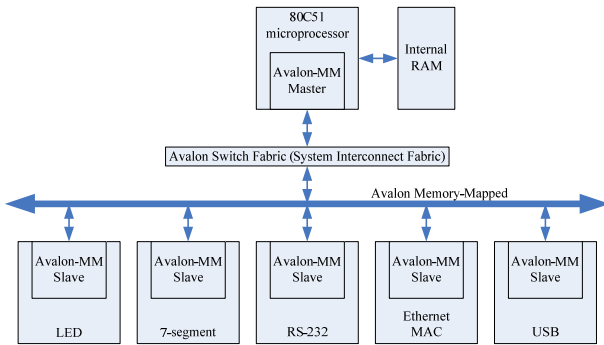


圖 1. Avalon Interface 結構

另外，Avalon Interface 也擁有的卓越特性，Avalon Interface 的訊號設定由表 1.說明：

- 具有獨立的位址和資料控制線，不需要再花費時脈週期去解碼是位址還是資料。
- 具有同步的運算能力，簡化了 Avalon Interface 的時序和容易整合高速的周邊裝置。
- 動態匯流排寬度，Avalon Interface 支援的資料寬度最大到 128 位元。Avalon Interface 可以很容易連接不同資料寬度的周邊裝置。

表 1. Avalon Interface 訊號說明

	Avalon Interface
Reset	reset : Asynchronous reset
Clock	clk : Synchronization clock
Interrupt	irq : Interrupt request
Data Access	readdata : Read data
	writedata : Write data
	address : Address
Control	chipsselect : Chip select
	read : Read request
	write : Write request
	byteenable : Enable byte lane
Handshake	waitrequest : Stall the bus

### 三、System Interconnect Fabric 設計

本節將會分成兩個小節說明，第一個小節會說明系統的架構；第二個小節則說明 System Interconnect Fabric 的設計方式。

#### (一) 系統架構

由於 8051 內部僅有 4 組 I/O 埠，擴充不易。為了使 8051 可以連結至更多的 I/O 裝置，本論文將 8051 的外部記憶體(16-bits 共 64KB)連結至 Avalon Memory-Mapped Interface，並透過位址的方式映射至 I/O 裝置，使得 8051 可以達到高擴充與分時多工的特性。

系統架構包括了以下的功能區塊：8051 Microprocessor、System Interconnect Fabric、Avalon Memory-Mapped Interface、I/O。圖 2.顯示出系統的架構。

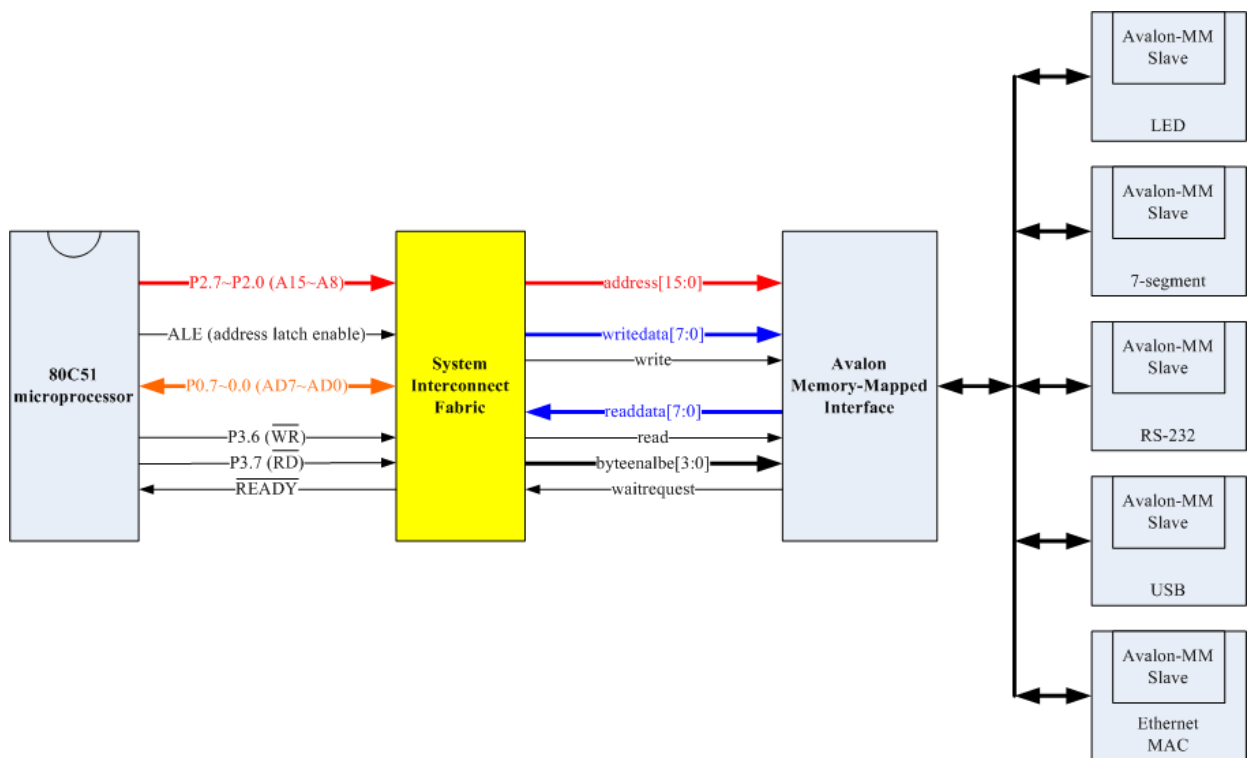


圖 2. 系統架構

- P0.7~P0.0 (AD7~AD0)：當 8051 為 16 位元定址之外部記憶體存取時，P0.7~P0.0 先當做 A7~A0 將位址送出定址後，再當做 D0~D7 來存取資料。
- P2.7~P2.0 (A15~8)：當 8051 為 16 位元定址之外部記憶體存取時，P2.7~P2.0 當做 A8~A15 來定址。
- P3.6 ( $\overline{WR}$ )：當 8051 欲將資料寫入外部記憶體時，此腳會產生負緣訊號。
- P3.7 ( $\overline{RD}$ )：當 8051 欲從外圍記憶體讀取資料時，此腳會產生負緣訊號。
- ALE (Address Latch Enable)：當外部記憶體存取時 ALE=1 且負緣，可鎖住 A0~A7。
- $\overline{READY}$ ：將等待狀態插入 8051 的時序中。如果  $\overline{READY}=0$  時，8051 會進入等待狀態，並保持閒置。

## (二) System Interconnect Fabric 架構

由於 8051 與 Avalon Memory-Mapped Interface 是兩個不同的系統，為了使兩個系統可

以完整的連結，本論文提出 System Interconnect Fabric 的設計方式。圖 3.顯示 System Interconnect Fabric 的架構。

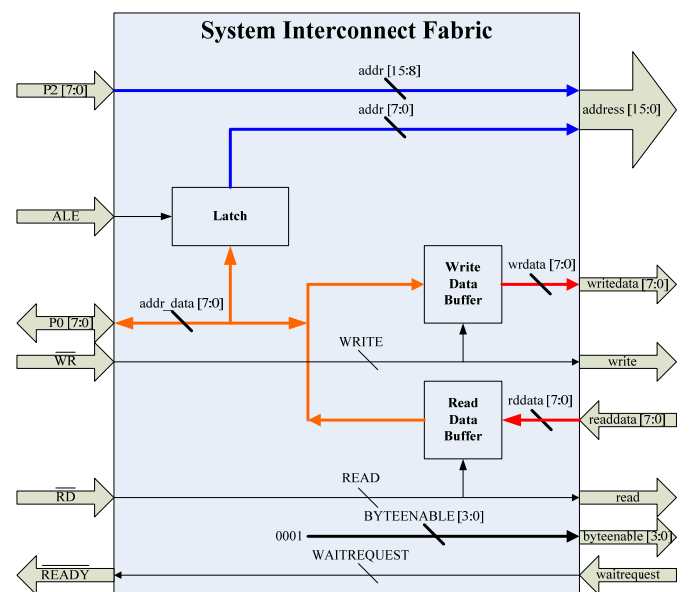


圖 3. System Interconnect Fabric 架構

- addr [15:8]: 用以接收 8051 外部記憶體的高 8 位元位址，並將高 8 位元的位址傳送至 address [15:8]，做為 Avalon Memory-Mapped Interface 的高 8 位元位址。
- addr [7:0]: 用以接收 8051 外部記憶體的 8 位元位址，並將低 8 位元的位址傳送至 address [7:0]，做為 Avalon Memory-Mapped Interface 的低 8 位元位址。
- addr\_data [7:0]: 用以傳送或接收 8051 與 Avalon Memory-Mapped Interface 之間的低 8 位元位址或是資料。
- wrdata [7:0]: 用以傳送 Write Data Buffer 的資料至 writedata [7:0]，做為 Avalon Memory-Mapped Interface 的資料寫入。
- rddata [7:0]: 用以接收 Avalon Memory-Mapped Interface 至 readdata [7:0] 的資料，做為 Read Data Buffer 的資料寫入。
- BYTEENABLE [3:0]: 由於 Avalon Memory-Mapped Interface 的 writedata 或 readdata 一次是 32 位元，而 8051 為 8 位元的處理器。當 BYTEENABLE 為 0001 時，表示僅要 32 位元中最低 8 位元的資料。
- WRITE: 用以告知 Avalon Memory-Mapped Interface 或 Write Data Buffer，8051 將寫入資料。
- READ: 用以告知 Avalon Memory-Mapped Interface 或 Read Data Buffer，8051 將讀取資料。
- WAITREQUEST: 用以 Avalon Memory-Mapped Interface 告知 8051，資料未準備完全，先進入等待狀態，並保持閒置。

#### 四、模擬環境與模組實現

本節將分成兩個小節說明，第一個小節會說明系統模擬的環境；第二個小節則說明模組的實現方式。

##### (一) 模擬環境

本論文提出的 System Interconnect Fabric 是

實現在 ALTERA Cyclone II EP2C35F672 FPGA 平台，並且該系統執行在 FPGA 平台上，運作頻率為 50Mhz。

##### (二) 模組實現

由於 8051 在讀取外部記憶體的資料時，P0 埠既是位址線也是資料線，所以必須用 ALE 與  $\overline{\text{READY}}$  這兩個控制訊號來隔離位址與資料。其動作順序如下說明，而動作時序如圖 4.所示。

1. 8051 自動由 P2 送出高位址。
2. 8051 自動由 P0 送出低位址，並利用 ALE 的負緣將其閃鎖在 Latch。
3. 8051 的  $\overline{\text{READY}}$  送出 0，使其等待外部記憶體。
4. 8051 由  $\overline{\text{RD}}$  或  $\overline{\text{WR}}$  決定 P0 是讀入外部記憶體資料，或是寫出資料至外部記憶體，並將  $\overline{\text{READY}}$  恢復為 1。

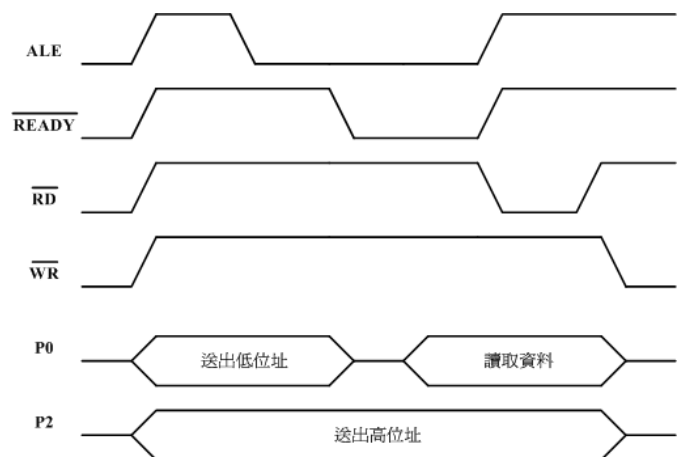


圖 4. 8051 讀寫外部記憶體資料之時序

因此，當外部記憶體的資料尚未準備完全時，必須使用  $\overline{\text{READY}}$  訊號告知 8051 先進入等待狀態；當外部記憶體的資料準備完全後，即可告知 8051 已可讀取資料。圖 5.顯示 System Interconnect Fabric 的控制流程。

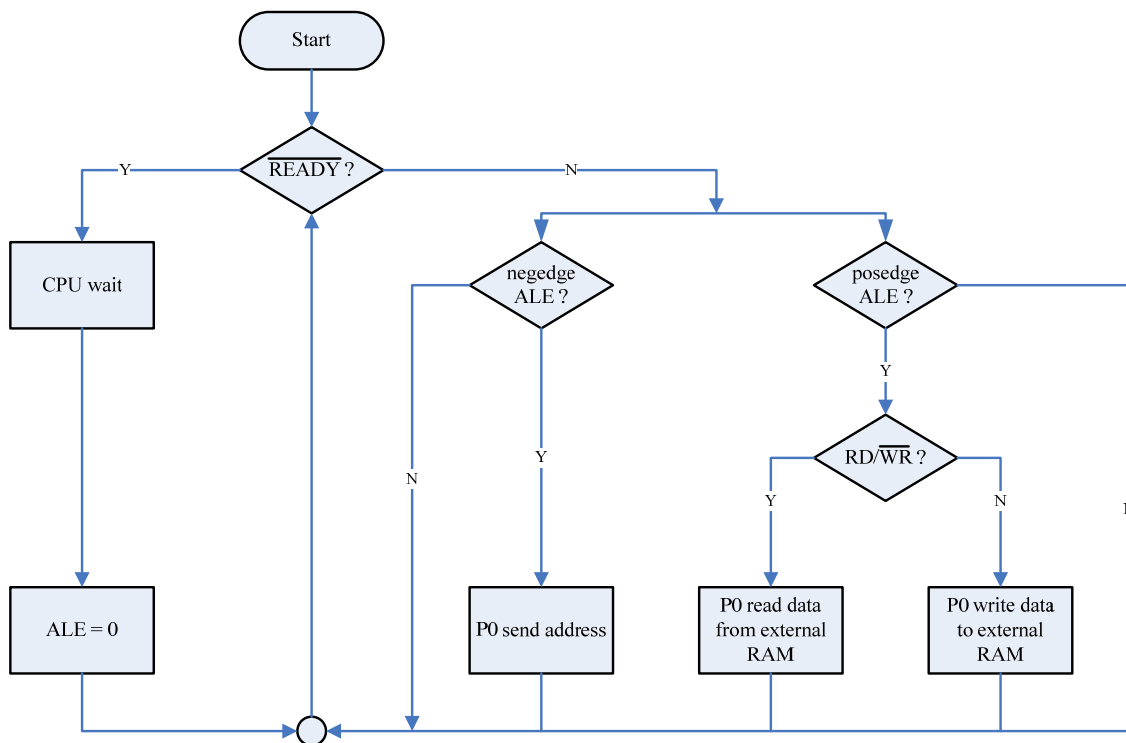


圖 5. System Interconnect Fabric 控制流程

由波形模擬結果可知，當 avm\_m1\_export\_ale=1 且負緣時，可鎖住 avm\_m1\_export\_p0\_i，並與 avm\_m1\_export\_p2\_i 合成 16 位元的位址後傳送至 avm\_m1\_address，做為 Avalon Memory-Mapped Interface 的位址。當 avm\_m1\_export\_ale=0 且正緣時，若 avm\_m1\_export\_rd=1 時，表示要從 Avalon Memory-Mapped Interface 讀取資料至 8051 裡，因此將 avm\_m1\_readdata 的資料讀取至 avm\_m1\_export\_p0\_o 裡；若 avm\_m1\_export\_wr=1 時，表示要從 8051 寫出資料至 Memory-Mapped Interface 裡，因此將 avm\_m1\_export\_po\_i 的資料寫入至 avm\_m1\_writedata 裡。圖 6. 顯示 System Interconnect Fabric 波形模擬。

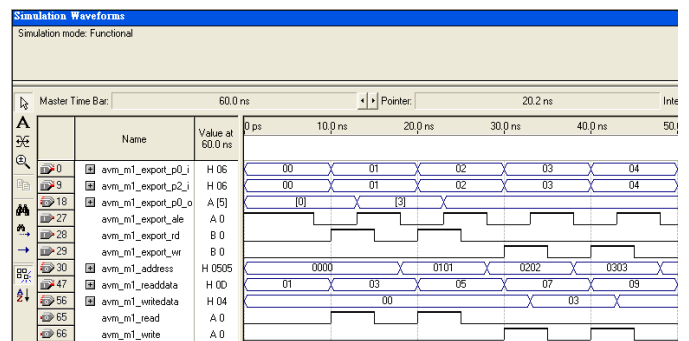


圖 5. System Interconnect Fabric 波形模擬

## 五、結論與未來工作

本論文所提出的 System Interconnect Fabric 模組，可以經由 ALTERA Quartus II Version 9.0 編譯，成功的與 8051 Core 和 Avalon Interface 結合。由於使用 Avalon Interface 的 memory-mapped 方式，能夠使 8051 得以應用在一些特殊的 I/O 裝置上，例如：USB 裝置、MAC 裝置、SD CARD 等，不在只是局限於 4 組內部 I/O 埠的簡單應用。而透過 Avalon 匯流排的特性，使得原本 8051 單一匯流排無法同步化控制 I/O 裝置的能力，變

成可以同步化的控制 I/O 裝置，從而提升了系統匯流排的吞吐量。

在未來的工作裡，我們將持續對 8051 Core 內部做改善，例如：以 Avalon Interface 設定多處理器運算，以達成多核心系統。以 Avalon Interfaced 的 memory-mapped 外部記憶體的方式來減少內部 I/O 單元，以達成低成本的設計。修改其記憶體定址架構，以解決 64K 記憶體的限制。研究其指令集架構，使其能達成超純量處理器。並期望能夠發展出一套低成本、高應用性、高適應性的微處理器。

## 六、參考文獻

- [1] Altera Corporations, “Avalon Interface Specification”, 2008.
- [2] Chang-Jiu Chen, Wei-Min Cheng, Tuan-Chieh Wang, Yuan-Teng Chang, Hung-Yue Tsai, “Instruction Decoder Implemented with Balsa for an Asynchronous Pipelined 8051 compatible Microcontroller”, International Computer Symposium, 2008.
- [3] Chia-Ying Tseng and Yen-Chih Chen, “Design and Implementation of Multiprocessor System on a Chip (MPSoC) Based on FPGA”, International Computer Symposium, 2008.
- [4] Damjan Lampret, “OpenRISC1200 IP Core Specification”, OpenCores.org , Rev. 0.7, September 6, 2001.
- [5] Feng Lin, Haili Wang, Jinian Bian, “HW/SW interface synthesis based on Avalon bus specification for Nios-oriented SoC design”, Field-Programmable Technology, pp.305-306, 2005.
- [6] Jaka Simsic, Simon Teran, “8051 Core Specification”, OpenCores.org , Rev. 0.1, August 14, 2001.
- [7] Lee, J.-H. and Kim and Y. H. and Cho, K.-R. ; “A Low-Power Implementation of Asynchronous 8051 Employing Adaptive Pipeline Structure”, Circuits and Systems II: Express Briefs, Vol. 55, pp.673-677, 2008.
- [8] Yang Xu, Min Xiang, “Design a New Type PWM Peripherals in Nios II”, Computer Science and Information Engineering, Vol. 2, pp.442-446, 2009.
- [9] YunZhu Xiang, YueHua Ding, “Instruction Decoder Module Design of 32-bit RISC CPU Based on MIPS”, Genetic and Evolutionary Computing, pp.347-351, 2008.