

## FPGA Base 之單軸伺服馬達系統開發與建置

黃秉宏<sup>1</sup>、吳文傑<sup>1</sup>

Ping-Hung Huang<sup>1</sup>, and Wen-Chieh Wu<sup>1</sup>

<sup>1</sup>財團法人金屬工業研究發展中心

<sup>1</sup> Metal Industries Research & Development Center, Taichung, Taiwan, R. O. C.

E-mail: pinghuang@mail.mirc.org.tw

E-mail: wcwu@mail.mirc.org.tw

### 摘要

本文主要是講述使用現場可程式邏輯閘陣列晶片(FPGA)嵌入軟核心處理器,來架構出 FPGA Base 之單軸伺服馬達系統開發與建置。透過 FPGA 來驅動伺服馬達,內部主要有兩個模組,第一個模組在 FPGA 晶片內以數位硬體方式實現,主要執行速度控制迴圈計算、光學編碼器信號偵測及數位/類比介面轉換電路等功能。第二個模組在 NiosII 處理器內以軟體程式實現,主要執行運動命令之計算。將其兩個模組進行整合後,使控制單軸伺服馬達平台得以實現。

**關鍵字詞：** FPGA、伺服馬達、NiosII

### Abstract

This paper is about the use of field programmable gate array (FPGA) and an embedded soft-core processor, Nios II, to structure the FPGA Base of uniaxial servo motor system. Through the FPGA to drive the servo motor, the internal two modules, the first module in the FPGA chip implement digital hardware, including execution speed control loop computing, optical encoder signal detection and digital / analog interface conversion circuit and other functions. The second module that mainly implement the calculation the motion command achieves the software program in NiosII processor. After integrating two modules, it can be achieved to control a uniaxial servo motor platform.

**Keywords:** FPGA、SOPC、Servo Motor

### 1. 前言

伺服馬達之應用已廣泛使用於工業自動化、製程、產業機械、品管與量測等相關自動工業控制領域,例如工廠中常見的 CNC 車床,就可以見到伺服馬達的蹤跡。目前既有的控制方式大多都是透過 PLC 或 PC\_Based 控制器等等方式進行馬達指令傳輸與控制。但其控制架構、運算速度與邏輯法則卻常受到既有控制器規格與模式上的限制,使客製化開發更加困難、因此也比較難以創新功能和模組整合,使其獲得突破。

在本文中將運用 FPGA 現場可程式邏輯閘陣列晶片,將其具有的可程式規劃設計的功能及高能效之數值計算之能力,來發展和設計出適合該行業客製化之應用軟硬體,以提昇原來設備之品質與功能,更使之達到小型化、網路化、低成本、

省元件、智慧型與高性能的目的。此外因整個系統開發都在數位系統架構下與類比系統比較上,數位系統設計之控制器更具有控制法則調整之彈性、成本低及體積小等優點,並且可利用數位系統執行複雜之計算與各種演算法則進行結合,因此相當適合發展高性能之控制器。而 FPGA 擁有上述之特性,且具有硬體可程式化、快速開發與驗證,相較於 ASIC(application specified integrated circuit)硬體開發週期更為縮短。

因此本文將以 Altera FPGA 開發板來建置相關環境與平台,來發展伺服單軸馬達速度控制系統。開發過程主要包含了三個階段,第一階段為周邊電路之研究開發,第二階段則是在 Nios II 處理器內以軟體程式實現並執行運動命令之計算;第三個模組則是在 FPGA 晶片內以硬體描述

語言 VHDL 進行撰寫程式內容，分別有速度估測計算、光學編碼器信號偵測和數位轉類比控制時序撰寫等功能。最後，在本文中將實務架設實驗系統，包含 PC、FPGA 控制器、伺服馬達驅動器與伺服馬達之運動平台進行速度控制實驗之功能驗證測試，以此簡要介紹說明其開發方式與應用模式並驗證所提控制技術之正確性。

## 2. FPGA Base 平台控制系統晶片架構

### 2.1 FPGA Based SOPC 嵌入式系統介紹

FPGA 是一款介於特殊 ASIC (application specified integrated circuit)與一般功能處理器之元件，它具有硬體可程式化、較短的設計週期、耗電量低、以及產品可以快速驗證後，快速進到市場等優點，因此極非常適合數位系統之實現。

因此在 FPGA 應用面上，已廣泛應用在工業領域馬達控制電路之開發上。而在單純 FPGA 架構下，大部份研究方向均為運用 FPGA 之數位硬體來進行換流器及馬達控制電路計算因為速度較快，但較少涉及複雜控制法則處理，使得上位控制如進階回授控制邏輯、運動軌跡規劃等功能較難與驅動器控制電路直接於單一晶片進行整合，在無形中降低晶片加值功效並且增加了整合困難度。因此在嵌入軟核架構下，使用者發展之應用 IP 及 Nios II 處理器 IP 可共同下載到 FPGA 晶片內，以提供使用者進行軟硬體整合設計。將需要快速計算電路但較不複雜者，可在 FPGA 以硬體實現；而演算法則較為複雜者，可運用 Nios II 以多核型態實現。以軟硬體共同設計開發之方式，更為增加了可程式設計能力、使數位系統設計更具有彈性，如下圖 1 所示。

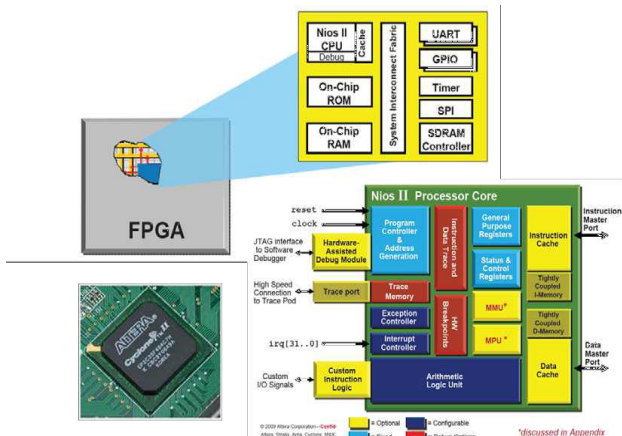


圖 1. FPGA 嵌入軟核 Nios II 構成 SOPC 之架構

因此基於前述 FPGA Based SOPC 架構，以 Altera FPGA 晶片內嵌入一顆 Nios II 嵌入式軟核

心處理器，用於建置伺服馬達速度控制系統晶片控制模組，架構規劃如下圖 2 所示：系統實驗設計主要進行伺服馬達速度控制以及連結現行平台後之運動控制。

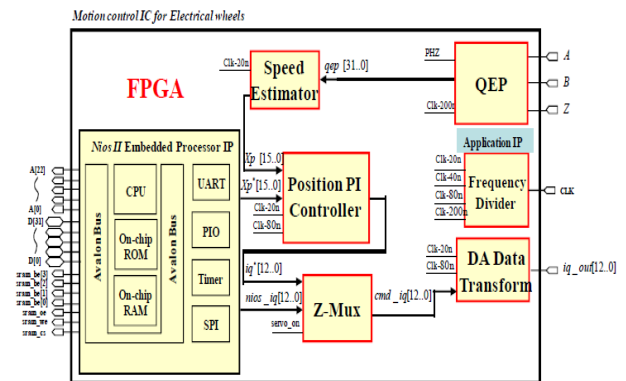


圖 2. 伺服馬達速度控制晶片架構圖

在圖 2 中可以看到此晶片內主要包含兩個模組，第一個模組在 FPGA 內以 VHDL 開發邏輯電路硬體程式 IP，其功能主要執行伺服馬達速度迴路之控制電路，也就是伺服馬達 PI 控制器所需要速度估測、QEP 編碼器計數器、DAC 時序控制 IP 介面電路等，都將以 HDL 硬體數位電路實現。第二個模組在 Nios II 處理器內則以軟體程式 C 語言來撰寫實現，其主要功能為擷取伺服馬達數據和下達速度命令。從圖中可以看到速度迴路主要是利用輸入轉速命令與經過受控體後所回授後之轉速達成一個閉迴路控制系統，將馬達編碼器上的資訊回授到方形脈波編碼器(QEP)來計數脈波，再將計數後的脈波送進速度估測公式轉換求得轉速。

### 2.2 速度估測 IP 介紹

在本文中採用的馬達編碼器轉動一圈會輸出 2500 個脈波，透過乘 4 後解碼電路後每轉會輸出 10000 脈波數之規格，且速度估測電路採用 2KHz 之速度取樣頻率，因此其速度估測之計算公式如下所示：

$$rpm = \frac{\text{圈數}}{\text{分鐘}} = \frac{\Delta pulse / 10000}{1 / (2K * 60)} = \Delta pulse * \frac{2K * 60}{10000} = \Delta pulse * 12 \quad (1)$$

其中， $\Delta pulse$  為兩次取樣之脈波數之差，而 rpm 為計算之馬達轉速。在此公式中可以看到在脈波值跨越 10000 之臨界值時必需修正。例如當馬達順時針運轉，脈波數向上計數，假若兩次取樣的數值分別為 9900 與上數到 10000 後歸零跨越到 150，則所計算出之  $\Delta pulse$  兩次取樣之脈波

數之差將為負，而造成錯誤的輸出結果，故需透過加值補償方式進行修正，如下圖 3 所示。

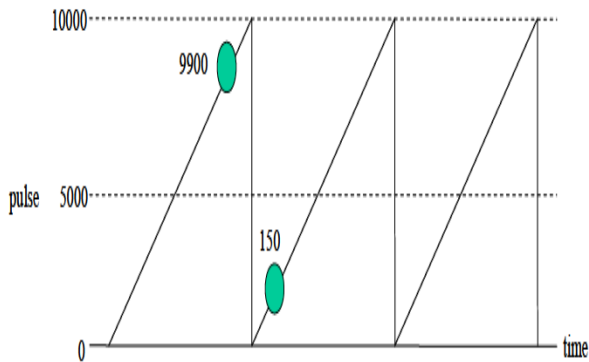


圖 3. 速度估測誤差修正視意圖

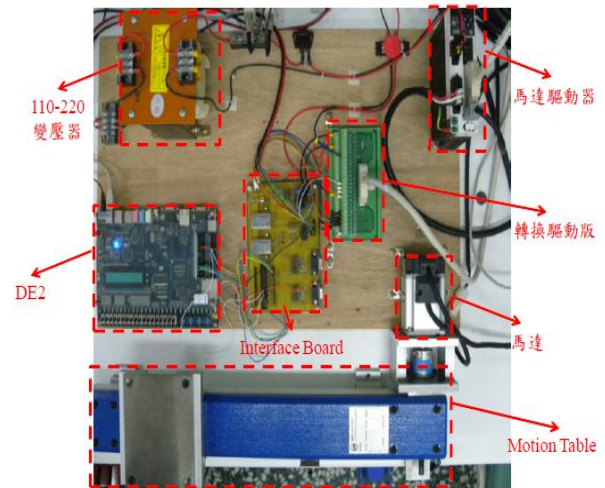


圖 4. 實驗平台架構圖

### 3. 實驗設備與結果

在本文中 FPGA Base 之單軸伺服馬達系統開發與建置的相關實驗設備與結果如下圖 4 中，可以清楚看到並且一一介紹。

#### 3.1 實驗設備

- (1)一顆伺服馬達及驅動器：在本實驗中所使用之 Panasonic 伺服馬達，輸入電壓為 90V，馬達之極數為 8 極，功率為 200W，額定電流 1.6A，額定轉速 3000rpm，具光學編碼器感測轉子位置。而驅動器使用 Panasonic MADDT1207，輸入電壓為 200-240V，單相輸入，頻率為 50/60Hz，額定電流 2A。
- (2)筆記型電腦一台：主要作為 VHDL 軟體撰寫、NiosII 撰寫、PC 與晶片間程式下載和擷取數據使用。
- (3)一組 FPGA 開發板：此開發板為整個系統之核心，此開發板內採用的 FPGA 為 Altera Cyclone II EP2C35 晶片。具有 475 個可用 I/O 腳位、35 個嵌入式乘法器、483,840 位元之 RAM 以及 33,216Les 邏輯閘元件。
- (4)一組轉換電路板：本電路板內含有 DAC8814 12 位元轉換 IC，QEP 光學編碼器比較電路各一組以及 RS232 通訊電路。

### 3.2 實驗結果

因此在本實驗中主要是進行單軸伺服馬達速度控制其架構如下圖 5 所示，伺服馬達之速度迴路控制以數位硬體電路實現，包括速度估測、速度控制器(PI 控制器)、QEP 解碼電路、DA 轉換控制等電路；速度命令則在 Nios II 內以 C 語言撰寫。在速度模式控制下初始設定速度迴路之控制取樣頻率設計為 2kHz 而 PI 控制器參數為  $I = 0.00152$ 、 $P = 0.45$ 、 $\text{Error Gain} = 64$ 。如下圖 6 中可看到 500 至 1000rpm 之方波命令與伺服馬達轉速下的速度步階響應，在此結果驗證了馬達在速度控制上有良好之追蹤性。

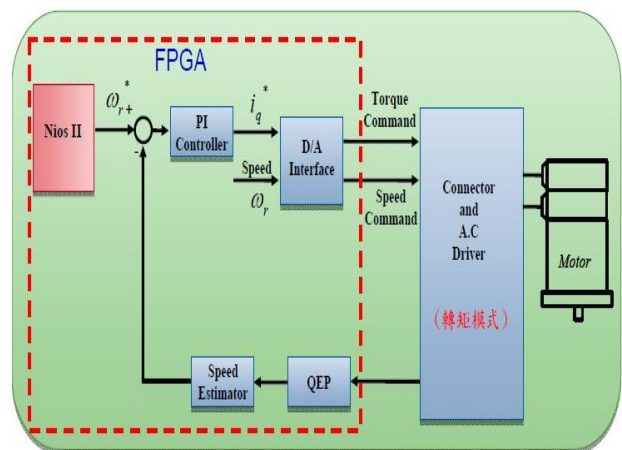


圖 5. 伺服馬達速度控制系統圖

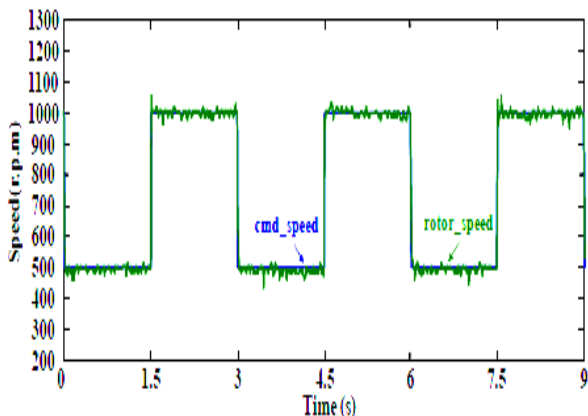


圖 6. 速度模式下 500-1000RPM 之速度響應圖

#### 4. 結論

本文介紹在 FPGA Base 技術下來發展單軸伺服馬達系統之開發，內有伺服控制晶片和運算控制用 IP，以利單軸伺服馬達速度控制及平台定位運動控制系統之用。並架設實際的實驗系統，包含了 PC、FPGA 控制器、商用馬達驅動器與馬達、運動平台進行功能與性能驗證測試。基於此技術之發展應用，未來將可用於工業自動化、製程、產業機械、品管與量測等應用以利伺服運動控制微型模組與便於快速彈性實現客製控制方法或邏輯驗證。此外此系統具有優於一般 PLC 或 PC\_Based 控制系統之邏輯運算速率與執行穩定性。因此在本文中所提到，PI 控制器、QEP 解碼電路、DAC 等功能，全部都在一顆 FPGA 以軟硬體共同設計之方式來實現。

在此文中在最後速度步階響應實驗結果中，證實所提 FPGA Base 下伺服控制之正確性及可行性。

#### 5. 參考文獻

1. 賈證主，“VHDL 數位系統設計與應用”，台科大圖書股份有限工司,2004.
2. 唐佩忠，“VHDL 與數位邏輯設計”，高立出版社,1999.
3. Web-site: [www.altera.com](http://www.altera.com)
4. M. F. Tsai and C. P. Chen, “Design of a Quadrature Decoder/Counter Interface IC for Motor Control Using CPLD,” IECON’02, Proceedings of the 28th Annual Conference of IEEE Industrial Electronics, pp. 1936-1941, 2002.
5. Altera Corporation, ” Designing with the Nios II Processor and SOPCBuilder”, 2009.

6. Z.Zhou, T. Li, T. Takahahi and E. Ho, “FPGA realization of a high-performance servo controller for PMSM,” in Proceeding of the 9th IEEE Application Power Electronics conference and Exposition, 2004, vol.3, pp. 1604-1609.
7. Y.S. Kung and M.H. Tsai, “FPGA-based Speed Control IC for PMSM Drive with Adaptive Fuzzy Control”, IEEE Transaction on Power Electronics , Vol. 22, No. 6, November, 2007, pp. 2476-2486.
8. Y.S. Kung, R.F. Fung and T.Y. Tai, “Realization of a Motion Control IC for X-Y Table Based on Novel FPGA Technology”, IEEE Transactions on Industrial Electronics. Vol. 56, No. 1, Jan. 2009, pp. 43-53.