

植於 FPGA 之雙核心微控制器設計

胡永楠

大葉大學

ynhu@mail.dyu.edu.tw

林銘晟

大葉大學

R9503048@mail.dyu.edu.tw

洪子家

大葉大學

R9603044@mail.dyu.edu.tw

摘要

本文主要是基於 FPGA 之優勢，以設計一顆可植入於 FPGA 之雙核心 MCU。而處理器核心是以 MIPS32 為基礎去設計出來的核心，除了有相容於原本 MIPS32 的指令集外，還增加了自行設計的指令，並配合一些週邊功能電路，如 PWM、UART 和一些 I/O 埠來構成雙核心 MCU。

本文製作的流程是先用 Verilog 硬體描述語言製作出似 MIPS32 的核心及周邊電路，然後再用 ModelSim 及 SynaptiCAD 去模擬各個模組及整體雙核 MCU 架構的正確性，然後再以 FPGA 作為實體驗證，進而把擁有雙核心架構之微控制器植入一顆晶片當中。

關鍵詞：FPGA、Verilog、ModelSim、SynaptiCAD。

Abstract

This research is mainly the advantage on the basis of FPGA, can plant it into in dual-core MCU of FPGA while designing one. The core of the processor is a core that is designed out based on MIPS32, cooperate in some peripheral circuit, If PWM, UART and some I/O ports come to form dual-core-MCU.

The procedure of this research is using Verilog hardware describe language to make MIPS32 core first, and then use ModelSim and SynaptiCAD softwares to simulate the correction of the whole dual-core-MCU and take FPGA as entity testify. After that, dual-core-structure's micro-controller then can plant into a chip.

Keywords: FPGA、Verilog、ModelSim、SynaptiCAD。

1. 前言

隨著半導體製程技術的增長，製程的層級越來越高，如今已達到 65 奈米，可看到的是在一顆晶片裡，能塞入的處理器核心，將會是越來越多，在這種情勢下，處理器的效能將會達到前所未及，進而把整個電腦領域帶入了另一個世代。以前透過單核電腦難以實現的，現在透過多核架構的電腦得以實現。隨科技的增進，人們對 3C 用品的需求量越來越高。行動電話就是一個很好的例子，從早期只能聽講電話，到現在能上網、聽 MP3，甚至收看數位電視，而手機裡的晶片所要處理的資料量越來越多，如還只採用單一核心架構的話，將不足應付這麼多的影音資料量；而目前市面上所販售的電視遊樂器如 Microsoft 的 XBOX360 及 SONY 的 PlayStation-3 所採用的處理器都是多核架構，這才足以應付相當龐大的 3D 繪圖資料量。而現在搭配雙核心處理器的電腦已是相當普及，可想而知在過個些許時間，更多核心的處理器將會陸續登場。

而把微控制器製作成多核心的架構則可在同一時間內控制不同事件。可能其中一顆核心在負責溫度的監控而另一顆核心在控制廣告看板，而其他的核心又在控制其他的事項；以前這些事都要被同一顆處理器所控制，事件和事件之間都是依序執行，現在可透過多核心架構來達到平行處理，這不僅是執行的速度和效率都能大大的提升。以前龐大的超級電腦，只有在特殊的場合才看的到，但現在超級電腦是相當有可能的直接植入在一顆晶片當中，而這種趨勢將會一直蔓延下去。

2. 多處理器知識背景

Flynn 在 1966 年提出，以電腦的指令流和資料流作為依據，而把電腦分成四大類。指令流是指被計算機所執行之指令數量，資料流是指計算機運算時，需要從記憶體所存取之資料數量。而利用這種指令流和資料流來區分電腦的方法，可區分為四大類，如下：

1. SISD(單一指令，單一資料)
2. SIMD(單一指令，多重資料)
3. MISD(多重指令，單一資料)
4. MIMD(多重指令，多重資料)

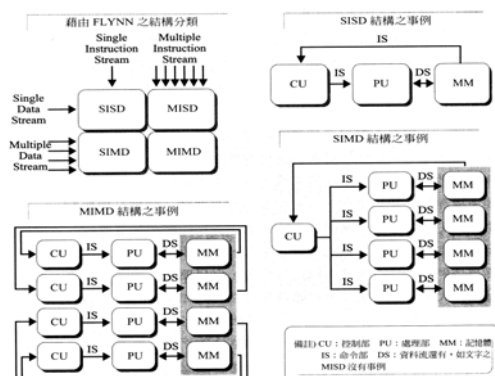


圖 2.1 Flynn 分類[1]

現今的多處理器架構是屬於 MIMD 電腦。

2.1 多處理器架構

多處理器架構主要可分為集中式和分散式兩種，而這樣的分類，主要是依據處理器和記憶體的實體連接方式來做區分的，如以記憶體存取方式來區分，可分為記憶體共享式和非共享。集中式架構中，所有處理器通常都是共享一個記憶體，而分散式架構中，各自的處理器通常都有自己獨立的記憶體，如以存取方式來看分散式架構也有共享記憶體的類型，圖 2.2 為集中式架構，圖 2.3 為分散式架構。

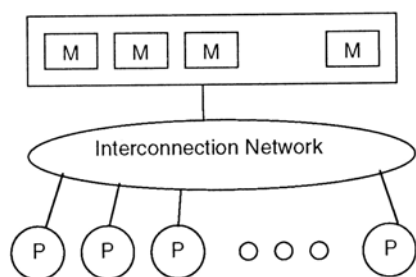


圖 2.2 集中式多處理器[8]

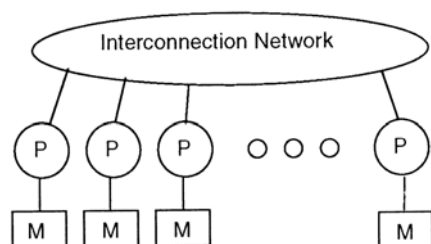


圖 2.3 分散式多處理器[8]

在集中式多處理器架構中以對稱式多處理器(Symmetric Multiprocessors, SMP)居多，圖 2.4 為一用匯流排連接的 SMP。而現今市面上的桌上型多處理器系統以 SMP 較多。

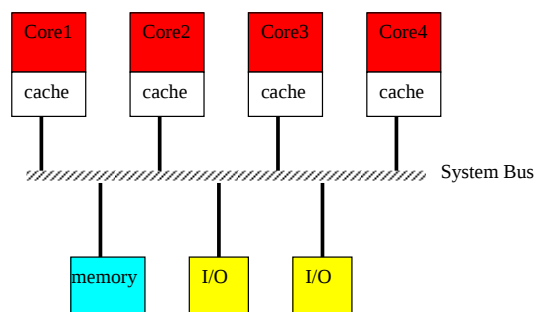


圖 2.4 SMP 架構

2.2 快取一致

當程式在多處理器架構中運行時，多顆處理器有可能會存取到同一個變數，而這些變數都會先儲存在處理器各自的快取記憶體中，假如這時候其中有一顆處理器改變了這個變數值，但是這次的更改只有更改到本身快取中的變數，在其他處理器的快取中還是沒更改過的舊值，等這些處理器要讀取這個變數值時，讀到的卻是這個沒更新過的舊值，這時程式的執行就發生錯誤了。所以為了確保整個程式執行的正確性，所有處理器的快取記憶體和主記憶體必需保持相同變數一致性，而這就是快取一致性。圖 2.5 為快取一致，每個快取中相同的變數，經更新前、後都須維持一致。

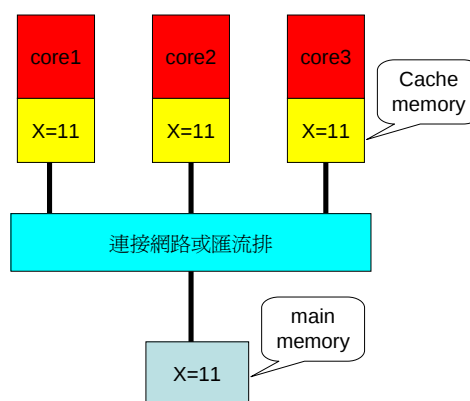


圖 2.5 X 在所有快取和主記憶體中都一致

2.3 多處理器工作原理

當我們所使用的電腦是單一處理器時，假設有很多的使 用者程式或著是任務工作一起執行時，將使 CPU 的工作排班 量過於龐大，

使 CPU 負荷不了這麼多的任務工作，而造成使用者等待程式執行完成的時間過久，甚至造成電腦當機，而這都是因為我們的 CPU 只有一顆所有程式或工作任務都是依序執行的，工作任務之間必須不斷的切換使用 CPU 資源的權限，當電腦裡 CPU 數量增加時，我們就可以把這些要一起執行的工作交由個別的 CPU 去執行，每個工作任務各自在不同的 CPU 中執行，這會使的電腦的效能大量的提升，上述的問題將不會發生；當 CPU 越多，能配於的工作任務就越多，這就是我們所說的多工處理(圖 2.6)；然而多處理器電腦不僅能做多工處理，還可以進行平行處理(圖 2.7)，而平行處理就是把單一的工作任務切割成數等分的小工作交由不同的 CPU 去同時執行，這會使的工作任務的執行更快更有效率，這就是多處理器系統給我們帶來的好處。

然而要在多處理器電腦上運行平行處理，軟體本身必須支援多處理器系統，軟體支不支援，就要看它本身被設計時，是不是被設計成支援多處理器系統。而現今除了一些伺服器軟體和作業系統有支援多處理器外，其餘的應用軟體很少有支援多處理器，但隨著多核處理器的問世，用於桌上型電腦的應用軟體和遊戲軟體，都開始往支援多核處理器的方向去設計，而新推出的軟體都會是支援多核處理器的軟體。

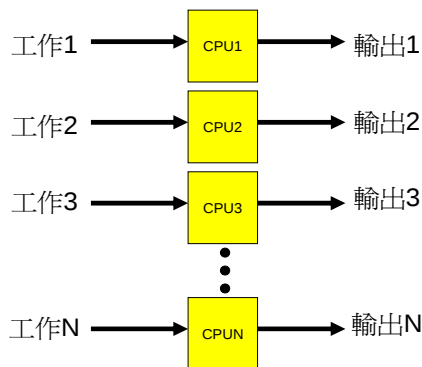


圖 2.6 多工處理

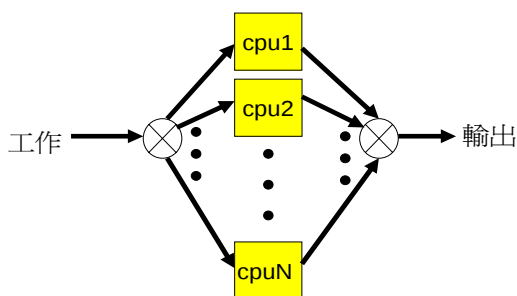


圖 2.7 平行處理

3. 系統架構

本文所設計之 MCU 主要是採用由本文自行設計的似 MIPS32 dual-core 為主要的處理器核心，它是集中式多 CPU 架構；然後再配合一些週邊的功能電路來實現這一顆 MCU，圖 3.1 為此 MCU 的內部元件。當此 MCU 開始運作時，程式和資料都是預先儲存在 RAM 裡面，然後再經由 CPU 將指令或資料抓入 L2 快取記憶體中，而後再到各核心所對應之 L1 的指令快取和資料快取中，經由雙 CPU 運算後去控制周邊電路。

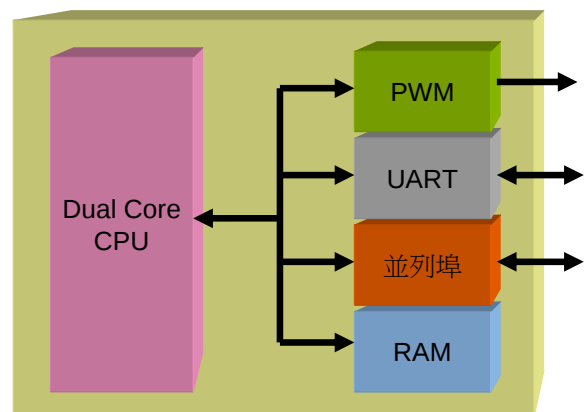


圖 3.1 Dual Core MCU

3.1 MIPS32 Dual Core

此處理器核心擁有相容於 MIPS32 的指令集外還擁有本文所設計的指令，而內部是採用四階管線化的方式運作，每一核心內部各自擁有第一層的指令快取(L1-I)和資料快取(L1-D)，而這些第一層的快取連接到一個共享快取記憶體，此記憶體為第二層快取(L2)，L1 和 L2 之間有一快取記憶體控制單元，他負責控制整個處理器內的快取機制，包括讀取及寫入，命中跟失誤時該如何處理，還包括了快取一致的功能，以確保在兩顆核心的快取中，擁有共享變數時，能維持資料的正確性；此快取控制只負責處理器內部快取一致，外部有另一個記憶體控制器會負責資料更新時 RAM 也一致，而這兩顆核心也藉由 L2 的共享快取記憶體和快取一致機制來達到互相溝通的目的。

而快取一致的策略是採用窺探式的寫入更新法來實現，快取記憶體則是採用 4 Way-Set-Associative 方式製作，置換策略則採用 LFR(Least Frequently Used)，這部份包含在快取控制單元內。圖 3.2 為此處理器和記憶體連

接的情形，此記憶體是一個多埠記憶體，可同時存取兩個不同位址的資料；另外有兩個記憶體控制器，負責當 CPU 對 L2 的快取存取命中或失誤時如何的對外部的記憶體做存取及資料的更新。

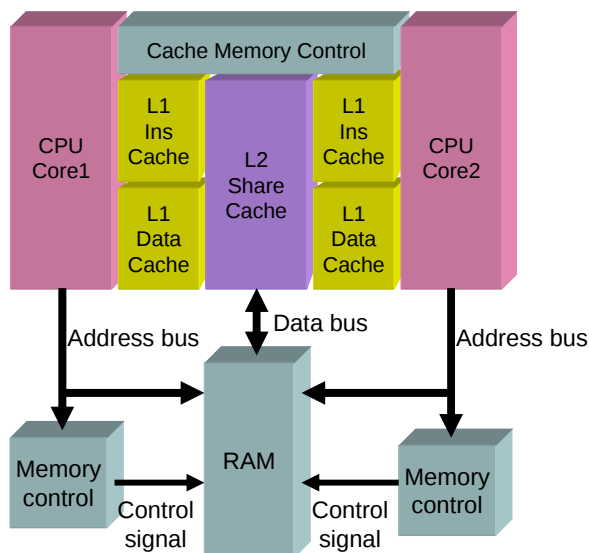


圖 3.2 dual core cpu

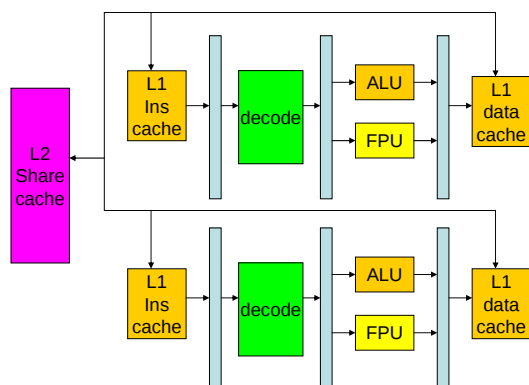


圖 3.3 dual core 內部架構(概略)

周邊電路元件的功能設定，採用暫存器映射的方法，每一個周邊都有它相對應的暫存器，當我們要設定某周邊或對輸入埠存取資料時，就直接指定到它相對應的暫存器。而我們把原先 MIPS 暫存器檔案中的 32 個暫存器重新規劃，把其中幾個暫存器規劃為特殊功能暫存器以作為設定周邊模組用。由於 MIPS 的暫存器是 32 位元，所以設計輸入埠時可從 8bit 擴展到 32bit，這些特殊功能暫存器也可設計成 8 到 32bit。

3.2 製作流程

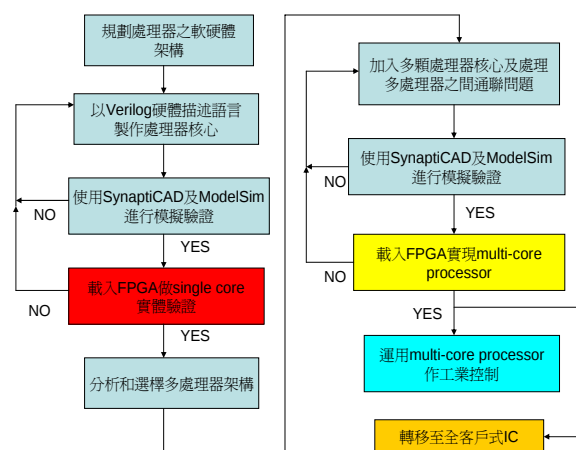


圖 3.4 製作流程

首先以 Verilog 硬體描述語言製作已規劃好的處理器軟硬體架構，然後再用 ModelSim、SynaptiCAD 等工具去做模擬驗證，以測試功能的正確性，正確無誤後再載入 FPGA 做實體的功能驗證，在這我們可以確保單一的核心是正確的，隨後我們選擇好要以什方式來建構多核處理器，核心之間要以什方式做溝通之後就再度回到與前面相同的步驟依序的將它完成；當處理器都無誤後，便加入一些週邊模組，最後把它載入 FPGA 之內做整體系統的實體驗證，並把它拿來做一些工業上的控制。

4. 系統驗證

首先以兩個 LED 的控制程式來執行多工處理情形，然後再以一個簡單明瞭的累積成程式來觀看分別以單核和雙核執行後的結果，以比較兩者的性能差異，最後在一個實際的工業控制作為實體驗證。此驗證只是為了要測試 Dual Core MCU 設計的正確性。

4.1 多工處理

在這裡我們分別用核心 1 執行跑馬燈程式，用核心 2 執行動態點矩陣 LED 的程式。跑馬燈從 portA 輸出，點矩陣 LED 需要送出字型碼和掃描信號分別從 portA2 和 portB2 輸出。圖 4.1 為模擬執行結果，其中 portA 為 16bit 的跑馬燈輸出，portA2 及 portB2 則輸出顯示一動態字型 MLB 的信號。

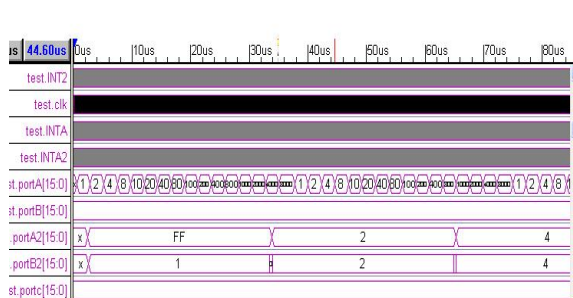


圖 4.1 多工處理模擬執行



圖 4.2 由核心 2 所控制點矩陣 LED

4.2 平行處理

分別用單一核心和雙核心去執行一個由 1 乘到 10 工作，先把原始的乘法程式載入到單一核心中執行，隨後把這個乘法程式改為雙核心的寫法，然後交由雙核心執行；我們可看到用雙核心執行會比單核心快了大約一倍左右。單核執行得出答案 3628800 的時間約為 5 us，雙核執行得出答案時間約為 2.8us。

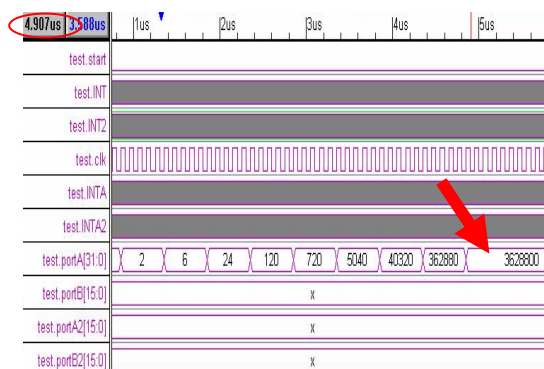


圖 4.3(一) single core 執行結果

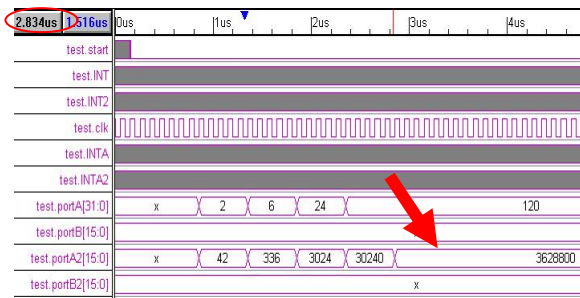


圖 4.3(二) dual core 執行結果

4.3 溫度控制

這個溫控系统如圖 4.4，AD590 為一溫度感測器，溫度信號經由 A/D 轉換器轉換成數位信號後進入 MCU，當溫度超過 45°C 時 MCU 就會啟動風散進行散熱，此刻加熱器是停止的，當溫度低於 30°C 時 MCU 就會啟動增溫器以加熱，此時風散是停止運轉的。

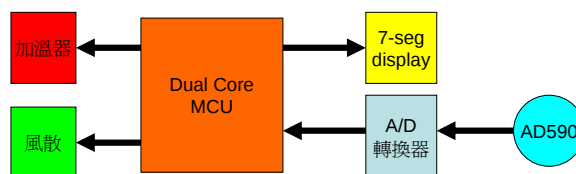


圖 4.4 溫控系统

4.3.1 程式流程

主程式由 CPU1 執行，中斷副程式由 CPU2 執行如圖 4.5 所示。

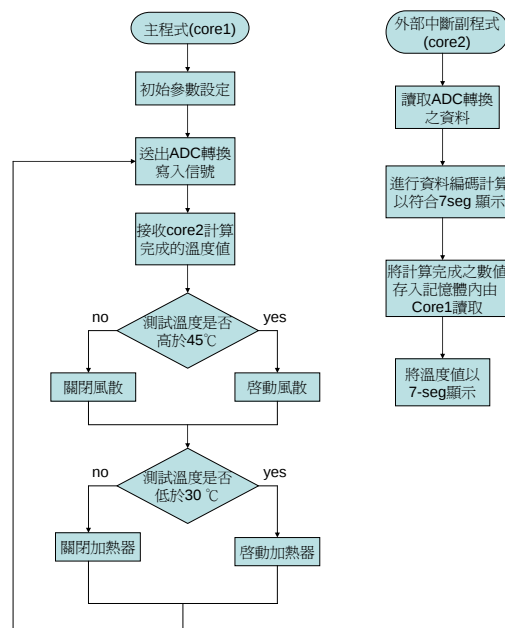


圖 4.5 程式流程

4.3.2 實際運作情形

圖 4.6 為溫度四十度時情形，風散和加熱器都是停止的，加熱器透過一個繼電器去做開關的切換。圖 4.7 為溫度四十七度時，溫度過高導致風散開始運作。當溫度過低 29 度時，則導致繼電器閉合，使加熱器運作，如圖 4.8。

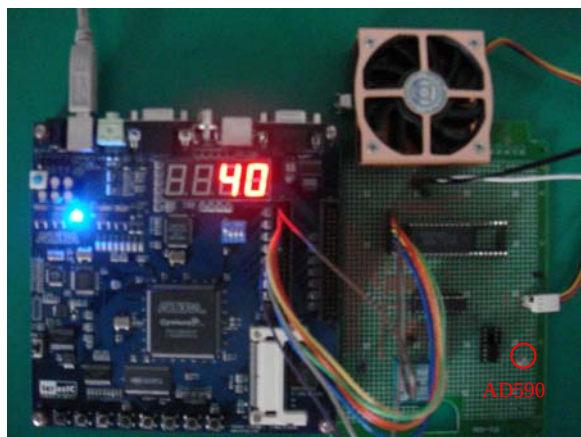


圖 4.6

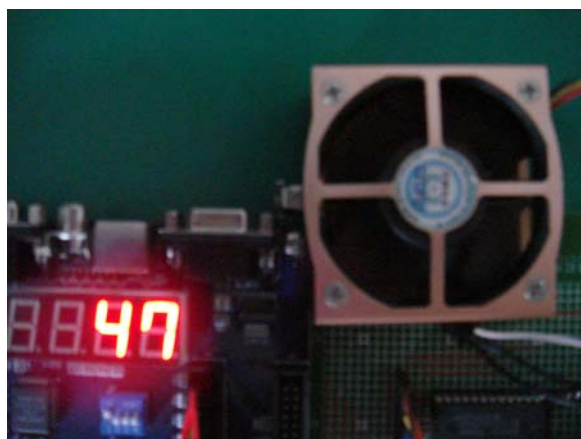


圖 4.7

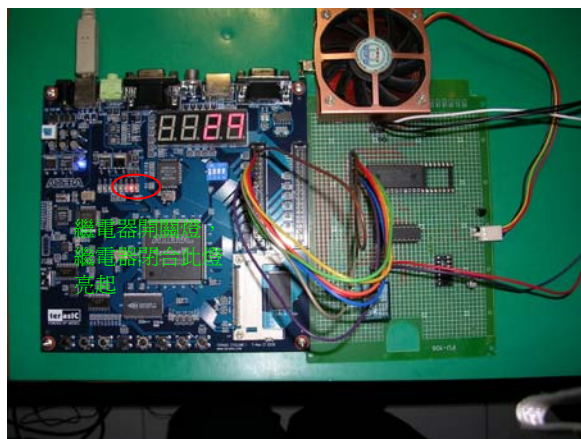


圖 4.8

4. 結論與未來展望

多核心能給我們帶來的好處就是處理效能的提升，主要可以並行處理和平行處理的方式運作；當我們要在同一時間內，同時執行多項事情時就選擇並行處理，這也是所謂的多工處理；而當我們要更快速的完成一件事時就採用平行處理；但是要以平行處理方式運作，程式在撰寫時，就必須以平行處理的觀念去設計，而往往平行處理程式的設計是困難的，是相當具有挑戰性的，這其中會包括一些變數相依性的危機，所以必須很小心設計。

本研究在往後將繼續把核心數目增加，以及加入更多的周邊電路，這樣此 MCU 將能做更多事，能發揮更多的效益，而將它植於 FPGA 中的好處是我們可以隨時的修改電路，增減核心數目，這是相當有彈性的。

參考文獻

- [1] 卓聖鵬，嵌入式系統開發技術，全華科技圖書。
- [2] 楊明豐，8051 單晶片 C 語言設計實務，基峯。
- [3] 鍾富昭，8051/8052 系列原理介紹與產品設計，全華科技圖書。
- [4] 何信龍、李雪銀，” PIC16C7X 入門與應用範例”。
- [5] William Stallings,”Computer Organization and Architecture Design for Performance “,Sixth edition.
- [6] David A.Patterson, John L.Hennessy “ Computer Architecture: A Quantitative Approach”,Third edition.
- [7] David A.Patterson, John L.Hennessy “ Computer Organization and Design”Third edition.
- [8] Abd-El-Barr,Hesham El-Rewini,Mostafa “Advanced Computer Architecture and Parallel Processing”.
- [9] M.Morris Mano,”Computer System Architecture “ Third edition.
- [10] Wayne Wolf,”FPGA-Based System Design”.
- [11] Samir Palnitkar,”Verilog HDL: A Guide to Digital Design and Synthesis”.