行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※　　　低功率高效能精簡指令集處理器　　　※
※　　　　　　核心之設計研究(II)　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：■個別型計畫　　□整合型計畫
計畫編號：NSC89－2215－E－002－031
執行期間：88 年 11 月 1 日至 89 年 10 月 31 日

計畫主持人：　　龐台銘 教授

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：國立臺灣大學電機研究所

中　華　民　國　　90 年　3 月　8 日

低功率高效能精簡指令集處理器核心之設計研究(II)
ARM7 RISC CPU Design

計畫編號：NSC 89-2215-E-002-031
執行期限：88 年 11 月 1 日至 89 年 10 月 31 日
主持人：龐台銘　台灣大學電機工程研究所教授
計畫參與人員：張景祺 台灣大學電機工程研究所碩士班研究生

摘要

A.　適用於個人電腦的處理器

隨著多媒體與網路的快速發展，個人電腦(PC)已經成為不可獲缺的生活必須品。然而 PC 的澎勃發展則仰賴在各軟硬體部份不曾停息的邁進潮流中，而硬體部份當中最受全球注目的就是中央處理器單元(CPU)了。包括有 Intel, AMD, Cyrix 等等，隨著 VLSI 的發展與半導體技術的演進其所研發的 CPU 在功能、速度上真可用出神入化來形容。

B.　精簡指令集處理器

事實上處理器的應用無所不在，小如手錶、計算器，大如汽車、飛機、軍事中心的控管等，有電器的地方就有處理器。面對神化般的個人電腦用 CPU，我們不僅要問: 真有需要把 CPU 做得如此複雜嗎?真有需要投資如此巨大的經費與人力來研發嗎?答案是否定的。在 1980 年由 Patterson 與 Ditzel 所寫的論文”
The Case for the Reduced Instruction Set Computer ”展新的將經簡指令集電腦(RISC)的觀念帶入了萌芽階段。由 Patterson 所率領的研究群開始研發所謂的 Berkeley RISC 的計畫，後來成為 SPARC station (SPARK 工作站) 的基礎。其友人 Hennessy 則在 Stanford 率領研究群開始研發 MIPS RISC 處理器，後來則成為 MIPS 電腦公司。由於精簡指令集電腦(RISC)其處理器設計的觀念就是快速、經巧、簡單、可信度高，因此在近二十年內不斷有新成員加入 RISC CPU 的設計潮流，如 Apple, ARM (Advanced RISC Machines)等,其所研發的處理器均以高效能、低功率消耗、低成本所著名。

一. 計劃緣由與目的

「可以快為何要慢慢來，可以簡單為何要這麼複雜，可以精巧又為何要這樣龐大」這是簡指令集電腦(RISC)設計者共通的理念，而在眾多的 RISC CPU 之中以 ARM 所生產的處理器除了擁有 RISC CPU 的特色外，還具有多元化的應用性以提供客戶式積體電路(CSICs)及應用式積體電路(ASICs)的特色，另外中心電路(CPU core)因面積小，因此易與其它積體電路如:RAM, ROM, DSP 等合成單一晶片以達成 SOC (Syste On Chip 單一晶片系統)的目標。在其生產的處理器中以 ARM7 同時具有商品性及學術研究性，在學術研究性方面:它運用了多種提高效能的處理器設計方法如平行處理(Parallel processing)、管路處理(Pipeline Processing)、固線式控制(Hardwired Control)、布斯乘法器(Booth’s Multiplier)、前置加法器(Carry Look Ahead Adder)、多週期結構(Muticycle implementation)等都是最先進的設計方式。在產品結合方面:GSM 終端控制器、通訊協定轉換、掌上形電腦、引擎控制器、通訊加密(Smart Card)、及 JPEG 控制器等。因此瞭解、改良及重新設計 ARM7 core 是進入精簡指令集電腦(RISC)處理器領域的最佳路徑。

二. 研究方法與成果

改進過的 ARM7 core 輸出入規格如 Fig 1.所示，共有 139 支 I/O 輸出入腳，包括有 32 支輸出腳、32 支輸入腳、24 支位址腳、及其他的控制腳。

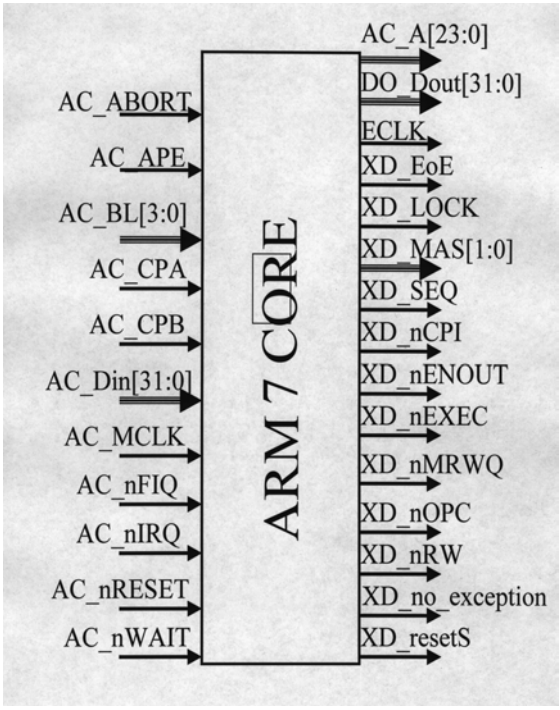


Fig 1. ARM 7 CORE I/O SPEC

ARM7 core 的功能區塊圖在 Fig 2.

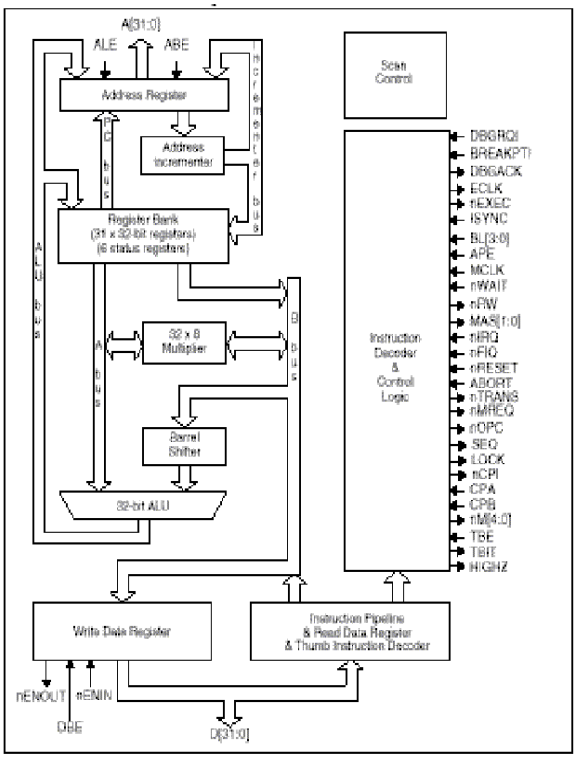


Fig:2.ARM7 core Block diagram

主要功能區塊簡介:

❑ Address Register (AR)

具有 24 條輸出線並提供位址內容的暫存器，除了有可做位址自動遞增的回授線外，AR 亦可接受來自 ALU 計算出的結果當做新的位址輸出。

❑ Address Increment (AI)

即上面所述用以實現 pc = pc + 1 的功能，其中 pc 為程式計數器(Program Counter)。

❑ Register Bank

由高速 CMOS 暫存器所組成，可暫存運算過程中的資料或運算結果資料，其也代表 RISC CPU 中具有適量的暫存器之特色。在原本規格列為 ARM 公司的專利，也因此在本區塊重新做了很多的設計及改變。

❑ Booth's Mutiplier

即 32 bits Booth's 乘法器，可實現高速乘法。

❑ Barrel Shifter

可實現整體向量同時移位，不同於傳統移位器必須做漣波式移位。因此加快了向量元素個數的倍數。

❑ 32 bit ALU

實現 32 bit 算數與邏輯運算功能。

❑ Instruction Decoder & Control Logic

即一般統稱的 CU(Control Unit)，是控制處理器所有運作的控制器。由於採 RISC CPU 的設計方式，因此其面積為時下 CISC (Complex Instruction Set Computer)處理器的一半以下。

❑ Write Data Register (WDR)

暫存要被輸出至記憶體的資料，在本處理器是採單向方式，與下面要簡介的 RDR 分開配置。

❑ Instruction Pipeline & Read Data Register

暫存由記憶體送來的資料，並當做管線運作時的暫存器。

三. 設計流程

A. 使用 C 語言

設計流程是先從系統模擬開始，以 C 程式語言描述電路的行為，並設計一 ARM assembler 語言程式，其包含所有的指令及會造成處理器可能無法正常運作的指令等。另外還設計一供處理器使用的計憶體單元，是由軟體描述的方式亦可供硬體完成時模擬之用。

B. 使用 Verilog (Behavioral model)

將 C 語言改成 Verilog behavioral 的格式來擬模硬體的運作行為，並與 C 語言做比較。主要的目的是取得程式執行過程所有訊號線的內容以作為將來硬體完成時其執行過程資料的比較，如此可節省大量的除錯時間。

C. 使用 Verilog (Hardware model)

將 Verilog Behavioral model 改為合成軟體可接受的模式，以區塊的方式一一將 Verilog Behavioral model 替換為 Hardware model 並隨時作驗證。

D. 使用 Synopsis 來完成合成

將完成且驗證無誤的電路描述語言經由 Synopsis 合成為邏輯閘階層的電路且再作驗證。

E. 使用 Cadence 來完成 Layout

Verilog In ➔ Floorplan ➔ Power Ring ➔ Power Stripe ➔ Create Row ➔ Place ➔ Followpin ➔ Route ➔ Fix error

F. 使用 DRACULA 驗證

DRC/ERC ➔ LVS ➔ Fix error

G. 使用 EPIC tools 驗證

TimeMill ➔ PowrMill ➔ Fix error

模擬結果

標準 ： 運作時脈: 40 MHz
功率消耗: < 100 mW
程式執行: 須與軟體模擬一致
須同時達到以上三點表模擬成功。

實際電路實現:

A. 使用 VL4 模擬

運作時脈 ： 40 MHz ➔ OK
最快正常運作時脈 ： 45.56 MHz

B. 使用 DRACULA 模擬

DRC/ERC ： 有 ERROR,解釋如下:
D1E 60: 檢查時看不到standard cell的 substrate contact。
LVS ： ➔ OK

C. 使用 EPIC TimeMill

運作時脈 ： 40 MHz ➔ OK
最快正常運作時脈 ： 50 MHz

D. 使用 EPIC PowrMill

運作時脈 ： 40MHz
平均功率消耗 ： 7.1 mW
RMS 功率消耗: 63.9 mW

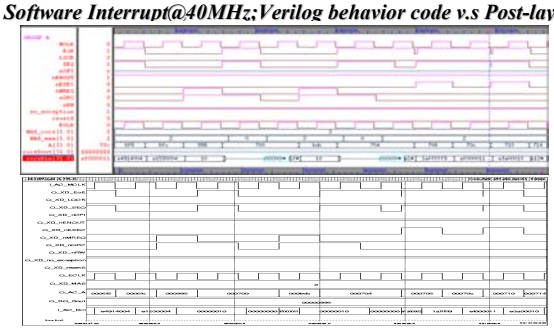
運作時脈 ： 50MHz
平均功率消耗 ： 8.32 mW
RMS 功率消耗: 70 mW

模擬結果完全合乎標準甚至超標準。

四 .模擬結果與測試報告
(A) pre-simulation
所有的設計與模擬均以軟體程式階段的資料為標準，不但節省大量除錯時間也由於測試資料量大，大大提高晶片的完全正確性。

Technology	ARM 7 CORE
Cell library	Compass High Performance
Die size	5388 X 5388 um^2
Transistor count	85642
Total pins	139 (144CQFP)
Work frequency	40MHz
Power supply	5V

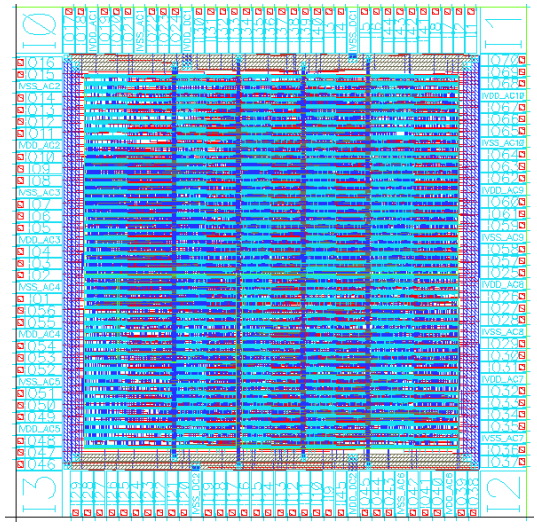
Software Interrupt@40MHz;Verilog behavior code v.s Post-layout



錯誤說明 ：

DRC/ERC check部份:
D1E 60: 檢查時看不到standard cell的 substrate contact。

(B) 佈局平面圖



(C) 測試方法與測試儀器

本 ARM7 core CPU 是以 IMS 晶片測試儀器來做各晶片功能的模擬，主要就是以 CPU 工作情況來做實際執行程式的模擬。

首先準備 CIC 測試部門所提供的 Socket Card (QFP 144) 以繞線棒及繞線將 CPU 上接腳所對應到 Socket card 上的銅棒之 power 及 ground 各別接到 VDD 及 VSS，接著請 CIC 工程師幫忙設定好 IMS 工作站的環境，並將先前準備的程式傳送到申請的帳號下，接著將晶片安裝在 Socket 上就開始進行測試，現在就直接由附件一來看：從一開始執行一直就是與正確值相同一直到第 704 個指令就發生第一個錯誤(連續三個)，接著又一直有正確值又來到第 733 個指令又發生同樣的情況，一直到第 1271 個指令之間一直都是偶爾有幾比輸出值

錯誤，依照輸出的結果來分析，前半段約 700 個正確指令及當發生錯誤後，又有一段正確的資料再接著又發生錯誤再又繼續正確的資料，由此可知 CPU 的控制電路方面(CU)是正確的;當發生預得資料錯誤時，CPU 觸動中斷(interup)，載入中斷處理程式來解決資料錯誤的問題，所以有接下來一段正確的資料。再由正確的資料可判斷得知 CPU 運算單元(ALU)的部份是正確的。

問題出在哪呢？從錯誤有遞移的效應(error propagation) 可得知正確的資料寫回暫存器(register file)時有發生錯誤的現象，因此問題應發生在暫存器的存取或暫存器存取的控制電路上。

針對這樣的猜測幾天後我們又準備了另二套程式再到 CIC 測試中心去測試，再煩請 CIC 工程師為我們準備好測試環境後就接著做更進一步的分析，此程式(附件二)的特色在於 CPU 做任何的運算均直接將資料送到輸出(output bus)，而不回寫到暫存器組(register bank)，很明顯可發現本 CPU 可做任何的運算而不會出錯，包括算數運算、邏輯運算、程式控制等..均得到正確的結果。第二個程式則只針對暫存器組做模擬，從編號零暫存器(Reg0)到編號三十一(Reg31)個別做讀取與寫入的動做，而且資料首先保持不互相參考，接著再部份參考以便交差比對，如此以便確認何者暫存器組或暫存器控制電路在存取編號何者暫存器時發生了錯誤，由附件三 IMS 輸出的結果分析得知錯誤發生在第七號暫存器(程式碼第 Seq183、Seq200、Seq382、Seq1632)，而且是發生在第一位元(Bit 1)的位子，就是倒數第二位元的地方恆為零，或者是剛好成互補的現象，如程式碼的輸出值 Seq183 的位子，正確值是
(00000F67)而輸出值是(00000F65)，倒數第貳位原應是”1”卻輸出位元”0”， 程式碼的輸出值 Seq1632 的位子，正確值是(0001FFFF)而輸出值是(0001FFFD)，倒數第貳位原應是”1”卻輸出位元”0”，如果此位元應是輸出 “0”則結就是正確的，但事實上原因在此位元在做運算動作時恆為零。

原因為何呢？首先我們懷疑是 clock skew 的問題，在當初設計時我們是以標準原件來自行設計時脈樹(clock tree)，所以重新 trace 程式碼，但發現每個葉端的緩衝器其負載(load)是機乎相同的也就是有達到當初設計的目標，而且同一暫存器組(reg 7)是由同一時脈來控制的，不應會有如此錯誤的現象(只有 bit 1 錯),
我們再去看原先的 Layout 圖也難發現有何原因會造成這樣怪異的錯誤，那原因為何呢？是否是制程公司光罩的問題呢？由於八棵已包裝的晶片都是這樣的問題，而且由程式來看無法找出會發生錯誤的原因，再加上再由 Verilog 及 VL4 仍模擬出正確的結果，及始將此晶片超頻在 45MHz(設計目標是 40MHz，而晶片實際操作在 33MHz)。

教授的意見：教授仍認為是設計者的錯，但不能完全怪設計者，因為在 VL4 模擬時此晶片只能操作在 45MHz 才可得到正確值，而 TimeMill 模擬時卻顯示此晶片可操作超過 50MHz，很顯然 model 兩者並不一制而且錯誤卻在晶片操作小於 33MHz 時就已出現，因此可能種種誤差累積造成設計者無法掌握的現象。

我個人意件：我比較同意教授的看法，但努力一年的結果為如此仍有些許的安慰，畢竟此晶片並非完全失敗，只要軟體程式設計師在設計程式時或修改 ARM7 編碼器(program compiler)不要使用到編號 7 的暫存器，此 CPU 便能正常運作。而至於何種原因造成如此的結果，我想還是設計者的錯，但目前還未完全了解是何種原因罷了。

五. 參考文獻

[1] David A. Patterson, John L. Hennessy, “ Computer Organization & Design“1994.

[2] John P. Hayes, “Computer Architecture and Organization “, Second Edition 1992.

[3] Steve Furber “ARM System Architecture”
1996.