

行政院國家科學委員會專題研究計畫 期中進度報告

可動態擴充之數位訊號處理核心於系統單晶片內之整合架構研究(2/3)

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-009-024-

執行期間：93年08月01日至94年07月31日

執行單位：國立交通大學資訊工程學系(所)

計畫主持人：鍾崇斌

共同主持人：單智君

計畫參與人員：楊惠親、傅世彰、吳建宏、唐心磊、廖源漢

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 94 年 6 月 1 日

中文摘要

伴隨著 IC 製程的進步，現今的在一個晶片中塞入數百萬以上電晶體以經是件滿容易的事，其中可能包含 CPU、Memory、DSP、Flash 等，形成完整系統，即所謂晶片系統(System-on-Chip, SOC)，亦可稱為系統晶片。然而若將 SOC 應用到嵌入式系統時，電源消耗將是一個最直接遭遇到的課題。通常對於 SOC 系統的電源消耗而言，處理器往往是佔最大的比重。因此，在本報告中，我們針對嵌入式處理器提出兩個減低電源消耗的方法。分別為 ARM9 微處理機之省電設計與低耗電指令快取記憶體 (Instruction Cache, I-Cache)。

根據分析 ARM 程式在 ARM9TDMI 處理機架構執行流程發現，處理機中的運算單元 (乘法器、ALU、Shifter) 並不會同時使用到，也就是說運算單元不需一直保持於忙碌狀態，這種狀況提供了省電設計的可能。因此，若能將不需使用的運算單元關閉，則可節省不必要的能量消耗。成果顯示平均而言可以節省 30~60% 之 ALU 與 Shifter 的 Switching Power，另外可節省 99% 之 Multiplier 的 Switching Power。

在 I-Cache 的使用上，程式在執行時 Cache Line 的使用個數是不多的，因此我們也可把這些沒有使用到的 Cache Line 轉成低電壓模式，以降低 I-Cache 的靜態功率消耗。然而調低 Cache Line 電壓所付出的代價是，當系統需要存取該 Cache Line 時，則需要較多時間延遲。因此，若我們能準確的預測每一條 Cache Line 需要被存取的時機，我們就能在耗電與存取時間上取得一個較好的平衡。模擬的結果顯示，我們所提出的方法可以節省約 80% 的 I-Cache 靜態功率消耗。

關鍵字：SOC、低耗電快取記憶體、嵌入式處理器

英文摘要

The power dissipation is usually the first challenge of integrating a system-on-chip (SOC) into an embedded system. Since all of these embedded systems are battery-operated, reducing the SOC power dissipation to extend the battery life therefore is important. Most power is consumed by embedded processor in most SOC system. In this report, we propose 2 schemes, are ARM9 low power mechanism and low power instruction cache (I-Cache) respectively, to reduce power consumption in the embedded processor.

Most operation units, such as ALU, multiplier and Shifter, would not be used at same time according to application profiles from ARM9. Such phenomenon provides a feasible way to reduce processor power dissipation. In this report, we propose a method to turn on/off unused parts of operation units to reduce processor power dissipation. Our simulation shows that our proposed method reduces 30~60% and 99% switching power in ALU/shifter and multiplier respectively.

In our observation, only one cache line is accessed in the I-Cache at any cycle. Other unused cache line can hence be turned to low power mode to reduce I-Cache's static power dissipation. But the cache accessing time would increase, if the processor accesses a cache line in low power mode. Thanks to branch target buffer (BTB), the processor easily predicts the next one and two instruction address and turns these cache lines corresponding to this instruction addresses to active mode in advance. By the same way, the processor predicts whether a cache line will be used in the future. If no, this cache line would be turned to the low power mode. Our simulation shows that our proposed method reduces 80% static power in I-Cache.

Keywords: SOC, low power processor, low power cache

1. 前言與研究目的

由於科技的進步，現今的在一個晶片中塞入數百萬以上電晶體以經是件滿容易的事，其中可能包含 CPU、Memory、DSP、Flash 等，形成完整系統，即所謂晶片系統(System-on-Chip, SOC)，亦可稱為系統晶片。另一方面，由於可攜式行動裝置的盛行，如手機、個人數位助理 (PDA) 與 MP3 隨身聽等。但是這類的裝置往往受限於體積與重量的限制，其電池的容量通常是相當有限的。因此若將 SOC 應用到嵌入式系統時，電源消耗將是一個最直接遭遇到的課題。通常對 SOC 系統的電源消耗而言，處理器往往是佔最大的比重。因此，在本報告中，我們針對嵌入式處理器提出兩個減低電源消耗的方法。分別為 ARM9 微處理機之省電設計與低耗電 I-Cache。

根據研究發現，在執行某段程式過程中，運算單元會執行一段時間後閒置，也就是說運算單元不需一直保持於忙碌狀態，這種狀況提供了省電設計的可能。因此，若能將不需使用的運算單元關閉，則可節省不必要之能量消耗。成果顯示平均而言可以節省 30~60% 之 ALU 與 Shifter 的 Switching Power, 另外可節省 99% 之 Multiplier 的 Switching Power。

另一方面，根據研究指出在嵌入式處理器中，Cache 大約消耗了處理器中近 50% 的電力。舉例來說，在 StrongARM SA-110 的 Cache 佔了 43% 的耗能 (其中 I-Cache 27%，D-Cache 16%) [1]。因此，若能減少 Cache 的耗電量也就直接的減低嵌入式系統的電力消耗。對於根據[2]中指出，程式在執行時 Cache Line 的使用個數是不多的。這意味著說，如果我們就可以把不需要用到的 Cache Line 調至低電壓模式，以降低 I-Cache 的靜態功率消耗。模擬的結果顯示，我們所提出的方法可以節省約 80% 的 I-Cache 靜態功率消耗。

2. 文獻探討

對於 Cache 的耗能可以分為動態耗能及靜態耗能。其中，動態耗能是指當我

們在存取 Cache 時所消耗的能量。靜態耗能則是指在 Cache 在不動作時，因電晶體本身漏電流(Leakage Current)所造成的耗能。根據不同型態的耗能，降低 Cache 耗能的方法可以分成兩大類：

1. 減少動態耗能：藉由降低 Cache Miss Rate，便可減少處理器對於外部記憶體的存取次數，進而降低耗能。[3]是利用 Compiler 的技術來降低 Cache Miss Rate 而達到省電。[4] 是針對只執行特定軟體之嵌入式系統，利用 Index Bits 的重新調整來減少 Cache Miss Rate，如此可以減少對外部記憶體的存取而達到省電。另外，Cache Access Energy 則是降低每次存取 Cache 所需的耗能。[5]提出在處理器與 L1 Cache 間增加一個小容量的 L0 Cache，如此可以讓約 60%~85% 的資料都從這個 L0 Cache 獲得，減少去存取 L1 Cache 的次數，以達到省電的效果。缺點則是因為 L0 Cache 的增加會導致 1 個 Cycle 的存取-延遲。[6]則在關聯式 (Set-association) Cache 中利用 Way 預測的方式來推測可能使用的 Way，減少標籤 (Tag) 的比較，達到省電的目的。
2. 減少靜態耗能：因為靜態耗能是快取記憶體不存取時也會消耗的能量，所以只跟電壓及快取記憶體大小有關，那最基本的想法就是把電壓降低，或是把部分不使用的快取記憶體關掉。Gated-Vdd[7]則提出把不使用的 Cache 關閉來減低功耗，但缺點是在 Cache 中的資料會消失。Drowsy Cache[8] 則設計一個可操作在雙電壓的 Cache，當某 Cache Line 不用時便將該 Cache Line 切換至低電壓模式 (Drowsy Mode)，這時資料是被完整的保存在 Cache 中，但是無法存取。一旦有需要時，再將該 Cache Line 切換回正常電壓，其缺點是會造成存取延遲。

3. 研究方法

3.1 ARM9 微處理機之省電設計

在本報告中，我們以 ARM9TDMI 為例，提出以下三個方向作為低耗電處理

機設計的目標：

1. 設計適時關閉不需要使用之運算單元的方法
2. 設計低耗電 Dummy Operations 處理流程
3. 設計 False-Condition 指令處理流程

(1) 適時關閉不需要使用之運算單元

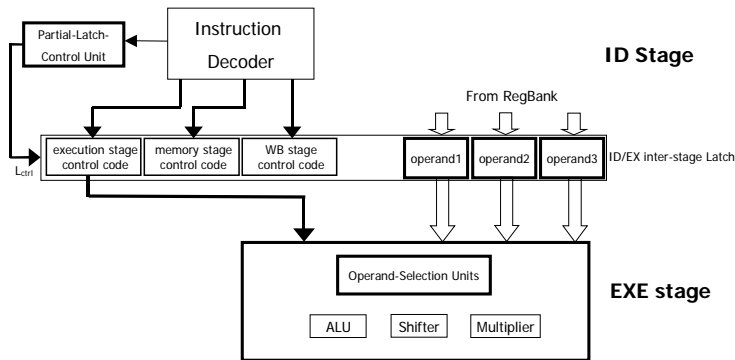


圖 1：ARM9TDMI Pipeline 修改示意圖

對於目前指令不需使用的運算單元保留其前一道指令的 Control Signals 與 Operands，使得運算單元內部的 Switching Power 在該 Cycle 內為零。此目標需要修改 Decode Stage 以及 Execution Stage。圖 1 為 Pipeline 修改示意圖，我們在 Decode Stage 加入 Partial-Latch-Control Unit 以及 Execution Stage 加入 Operand Select Units 支援本設計。

在 Decode Stage 中，設計 Partial-Latch-Control Unit，其目標有二，對於目前指令不需使用的運算單元，其一是保留前一道指令對該運算單元的控制訊號，其二是產生控制訊號使該運算單元使用前一道指令之 Operands 當成 Inputs，使得該運算單元內部的 Switching Power 在該 Cycle 內為零。

在 Execution Stage 中，設計數個 Operand Select Units，放在各運算單元前，以保留該指令的 Operands 值，當目前指令被 Partial-Latch-Control Unit 告知不需使用某運算單元時，則使用前一道指令保留下來的 Operand 當成 Input，以支援 Partial-Latch-Control Unit 之設計。

(2) 低耗電 Dummy Operations 處理流程

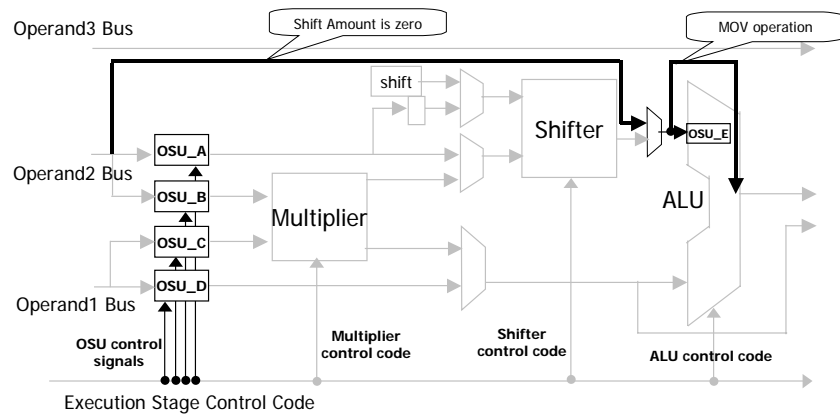


圖 2：修改後的 Execution Stage

所謂的 Dummy Operations 指僅傳遞 Register 內容的運算，例如 Move Operations。Dummy Operations 亦可關閉不需要使用的運算單元使得該運算單元再執行 Dummy Operations 時，Switching Power 為零。圖 2 為我們修改 Execution Stage 支援低耗電 Dummy Operations 設計的示意圖。其中對於 Move Operation，因其不需要使用 ALU，因此可直接使用 ALU 的 Input 當 Output。另外對於 Shifter Amount 為 0 的 Shifter Operations，則不需要使用 Shifter，因此可直接使用 Operand2 Bus 的值當成 Shifter 的 Outputs。

(3) False-Condition 指令處理流程

ARM 支援 Conditional Execution。因此對於並不是每道指令都要執行完所有的 Pipeline Stage，若發現該指令為 False Condition，則停止該指令的執行，以節省 Power。

3.2 低耗電 I-Cache

本報告中，我們所要降低的是 I-Cache 的靜態耗能。對此，我們將採用 Drowsy Cache 來動態調整 Cache Line 的運行模式以達到省電的目的。Drowsy Cache 有兩種模式，高 VDD 的一般模式 (Awake)，另一種較低 VDD 的睡眠模式 (Drowsy)。Drowsy Cache 在 Drowsy 模式時的功率消耗只有一般模式的 25%，但從睡眠模式改變成一般模式時，需要額外的電源消耗，以及 1 至 2 時脈 (Cycle) 的延遲。

在大部分的處理器中，若利用分支預測緩衝區(Branch Target Buffer, BTB)，則 Instruction 讀取順序的預測準確度將可以高到 95% 以上。因此，如果我們利用 BTB 來得知 Instruction 的讀取順序，將可以準確的判斷每一條 Cache Line 需要被存取的時機。對此，我們便可在某一 Cache Line 需要被讀取前，先把該 Cache Line 由 Drowsy 模式轉成 Awake 模式時，以減少讀取時間的延遲。相對的，當 Cache Line 需要由 Awake 模式轉成 Drowsy 模式時，則是得視該 Cache Line 在未來的 146 個 Cycle[2] 是否需要用到。如果是，此時關掉 Cache Line 將會造成額外的電源消耗，因為這樣一開一關所消耗的電量比起把 Cache Line 一直維持在 Awake 模式時還要來的多。

倘若每次判斷 Cache Line 是否得由 Awake 模式轉成 Drowsy 模式就得看未來 146 個 Cycle 的 Instruction 讀取順序，這樣一來預測的準確度勢必會大幅的下降。再者，若還得紀錄 146 個 Cycle 後的 Instruction 讀取順序，則必須有個相當大的暫存器來儲存，這將會造成系統過多的額外耗電。對此，我們的作法就是利用一個固定大小的暫存器，稱之為 Instruction 順序暫存器 (Instruction Sequence Buffer, ISB) 來儲存部分的 Instruction 讀取順序。根據我們實驗的結果顯示，當 ISB 長度，也就 ISB 所能記錄的位址個數大於 8 時，就有著滿不錯的效能。

ISB 為一 FIFO Buffer，在每次處理器的 Instruction Fetch 時，ISB 會把 Buffer 中的第一筆位址傳給處理器的 Program Counter (PC)，並把 Buffer 內的位址資料往前 Shift 一位。接著 ISB 就利用 Buffer 中倒數第二筆位址作為 Index，然後到 BTB 查詢該位址是否會發生 Branch。若是，則把 Target 位址入 Buffer 的最後一個位置；若否，則將倒數第二筆位址加 4 後填入。一旦處理器發生 Branch 預設錯誤時，或是在處理器初始與重置 (Reset) 時，ISB 就會 Reset 一次。此時，ISB 的中的第一筆位址就是處理器的 PC 值加 4 (假設處理器為 32 位元)，並根據此位址利用循序的方式把 Buffer 填至倒數第二位，再把倒數第二筆位址作為 Index 到 BTB 中重新把最後一位的位址找出後填入。

4. 結果與討論

在本節中我們將對上述的兩個方法進行相關的模擬。在 ARM9 微處理機之省電設計方面，我們使用 MediaBench [9]當成 Benchmark 與 ARM STD2.5 當成 Execution Trace 產生工具，並且建立 Trace Driven Simulator，用來實驗各種設計下可節省的運算單元之 Switching Activities。其綜合實驗結果如圖 3。成果顯示平均而言可以節省 30~60%之 ALU 與 Shifter 的 Switching Power,另外可節省 99%之 Multiplier 的 Switching Power。

在低耗電 I-Cache 的部分,我們修改 SimpleScalar [10],並利用 MinneSPEC [11]作為測試軟體。在此,我們僅選用 4 個程式作為本方法的模擬,其中包含 ammp、ampp、lucas 與 mgrid。而在 Cache 的設定方面,為了加快系統執行的速度,我們把第二層 Cache 與轉換表緩衝區 (Translation Look-Aside Buffer, TLB) 都加以關閉。I-Cache 與 D-Cache 的設定皆為:512 條 Cache Line、每條 Cache Line 有 32 個位元組 (Bytes) Direct Mapping、LRU (Least-Recently-Used)。在此,我們先假設 Instruction 讀取順序的預測是 100%準確的。圖 4 所示不同機制與程式中 I-Cache 所能降低的靜態耗能,其中 4、6、7、8、12 與 24 表示不同的 ISB 所能記錄的位址個數。我們可以發現到當 ISB 為 8 時,I-Cache 所能降低的靜態耗能都可以超過 80%。

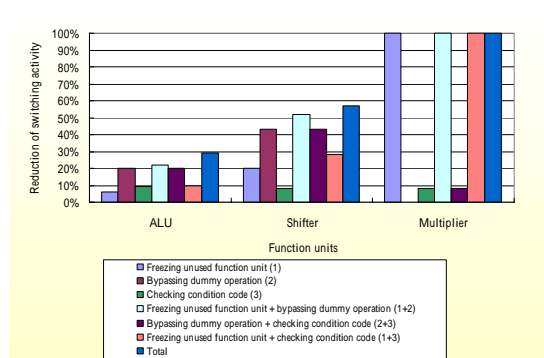


圖 3：ARM9 微處理機之省電模擬結果

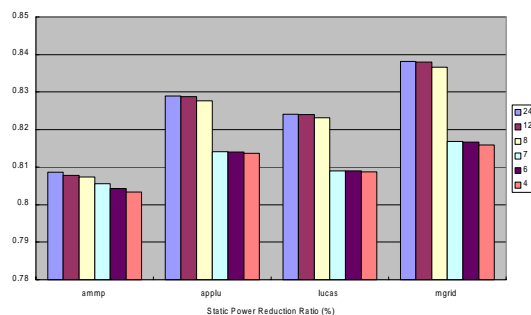


圖 4：不同程式中 I-Cache 所能降低的靜態耗能

參考文獻

- [1] K. Flautner et al., "Drowsy caches: Simple techniques for reducing leakage power," In Proc. the 29th Anchorage, AK, May 2002
- [2] J. Montanaro et al., "A 160 MHz, 32b 0.5W CMOS RISC Microprocessor," in IEEE ISSCC, Digest of Papers, 1996.
- [3] N. Hajj, Univ. of Illinois at Urbana-Champaign, " Architectural and Compiler Support for Energy Reduction in the memory Hierarchy of High Performance Microprocessors", 1998 International Symposium on Low Power Electronics and Design
- [4] T. Givargis. Improved Indexing for Cache Miss Reduction in Embedded Systems. Design Automation Conference (DAC), Pages: 79-84, June 2003.
- [5] J. Kin, M. Gupta and W. Mangione-Smith, "The Filter Cache: An Energy Efficient Memory Structure," Int. Symp. on Microarchitecture, pp. 184-193, Dec. 1997.
- [6] D.H. Albonesi., Selective Cache Ways: On-Demand Cache Resource Allocation. Proceedings of the 32nd International Symposium on Microarchitecture, Pages: 248–259, 1999.
- [7] M. Powell, S. H. Yang, B. Falsafi, K. Roy and T. Vijaykumar, "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories," in Proc. of International Symposium on Low Power Electronics and Design, 2000.
- [8] K. Flautner et al. Drowsy caches: Simple techniques for reducing leakage power. In Proc. the 29th Anchorage, AK, May 2002.
- [9] Chunho Lee, Potkonjak, M. and Mangione-Smith W.H., "MediaBench: a tool for evaluating and synthesizing multimedia andcommunications systems," In Proc. 30th International Symposium on Microarchitecture, pp. 330-335, 1997
- [10] D. Burger and T. Austin. The simplescalar tool set, version 2.0. Technical report, Computer Sciences Department, University of Wisconsin, June, 1997.
- [11] AJ Klein Osowski and David J. Lilja, "MinneSPEC: A New SPEC Benchmark Workload for Simulation-Based Computer Architecture Research", Computer Architecture Letters

計畫成果自評

本研究主題著重在嵌入式處理器上的省電機制設計，意旨在減少嵌入式系統的耗能，進而延長系統的使用時間。此外，根據研究結果顯示，我們所提出的方法有著不錯的省電效果。並且在實做上只需修改部分的系統設計即可。因此除了學術研究上是一大進展之外，對於實際應用層面亦有很大的幫助。