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(1/3)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-009-062-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立交通大學資訊工程學系

計畫主持人：鍾崇斌
共同主持人：單智君
計畫參與人員：喬偉豪

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 5 月 31 日

中文摘要

SOC 的設計，是將數顆 processor cores embedded 在一起，也在當今 VLSI 產品中扮演重要的腳色。隨著 embedded processor 快速發展，省電考慮也日益重要。針對目前 embedded processor 在 instruction bus 上的省電需求，我們提出一同時兼具 register relabeling 以及 instruction encoding 優點的 algorithm，使 encoding 完後的 code，在 instruction bus 上傳送時的 bit toggle 數更小。實驗數據顯示，我們提出的 algorithms 比起單純只有 register relabeling 多出約 50%的 bit transition 減少量，比起單純只有 instruction encoding 多出約 19%的 bit transition 減少量。整體而言，這項研究成果在 embedded processor 上有更進一步的省電效果。

關鍵字: SOC, embedded processor，省電，register relabeling

英文摘要

SOC design approaches, typically embedding a number of processor cores, have played an important role in VLSI products. As the rapid growth of embedded processor, low power consideration assumes a much heightened importance. For the low power requirement on instruction bus of current embedded processors, we present an algorithm with both advantages of register relabeling and instruction encoding scheme proposed by other researches. The simulation results showed that our algorithm has more 50% of bit transitions than register relabeling design only and more 19% of bit transitions than instruction encoding only. We can conclude with certainly that our research results may have more power saving opportunities.

Keywords: SOC, embedded processor, low power, register relabeling

1. 前言與研究目的

SOC 的設計，是將數顆 processor cores embedded 在一起，也在當今 VLSI 產品中扮演重要的腳色。隨著 embedded processor 快速發展，省電考慮也日益重要。許多 mobile devices 都很重視省電的相關設計主題。而本研究焦點是針對 SOC 架構中 instruction bus 上，針對指令進行 encoding 的動作，使得指令傳送時兩兩之間的 bit toggle 數較原來小，達到省電的目的。過去研究這類主題的相關文獻中，指令 encoding 的方法[2]，以及 register relabeling 的方法[3][4]效果都有限，我們的目標希望結合這兩種設計的優點，在有限的硬體支援下，使 encoding 完後的 code，在 instruction bus 上傳送時的 bit toggle 數更小。其相關的文獻探討請見下一章節。

2. 文獻探討

Bus-invert [1]適合運用在比較窄的匯流排上，若一般指令匯流排為 32bits，這方法效果並不佳。BITS[2]當 signal probability 是分布不均的時候能有較好的效果。Register Relabeling[3][4]由於暫存器欄位這些可以更動的欄位所佔整體程式並不會超過 50%，以致這個方法在整體指令的 bits transition 減少上幫助並不大。而 Petrov’s encoding method[5]的 decoder 較為複雜，而且所需要額外的 table 較大。

3. 研究方法

3.1 Basic Block Selection Algorithm

Basic Block Selection Algorithm 的目標，是要在 benchmark trace 中挑出最值得 encoding 的 basic blocks，以便接下來的 Register Relabeling and BIBITS Encoding 可針對這些 basic blocks 進行相關的編碼工作。

Each basic block has a ***contribution*** value measured as showed in EQ1. With the contribution of each basic block, we can apply the greedy algorithm to help us select which basic block should be encoded. 圖 1 是此 greedy selection algorithm 的一個範例，Table 1 則是所選出的 Basic Block 的 ***contribution*** value 列表，我們則從大到小一一將該 basic block 選入 encoding set 中，直到該 set 限定的值滿了為止。

$$Contribution = \frac{ExecutionCounts \cdot BitToggles}{Length} \quad \text{(EQ1)}$$

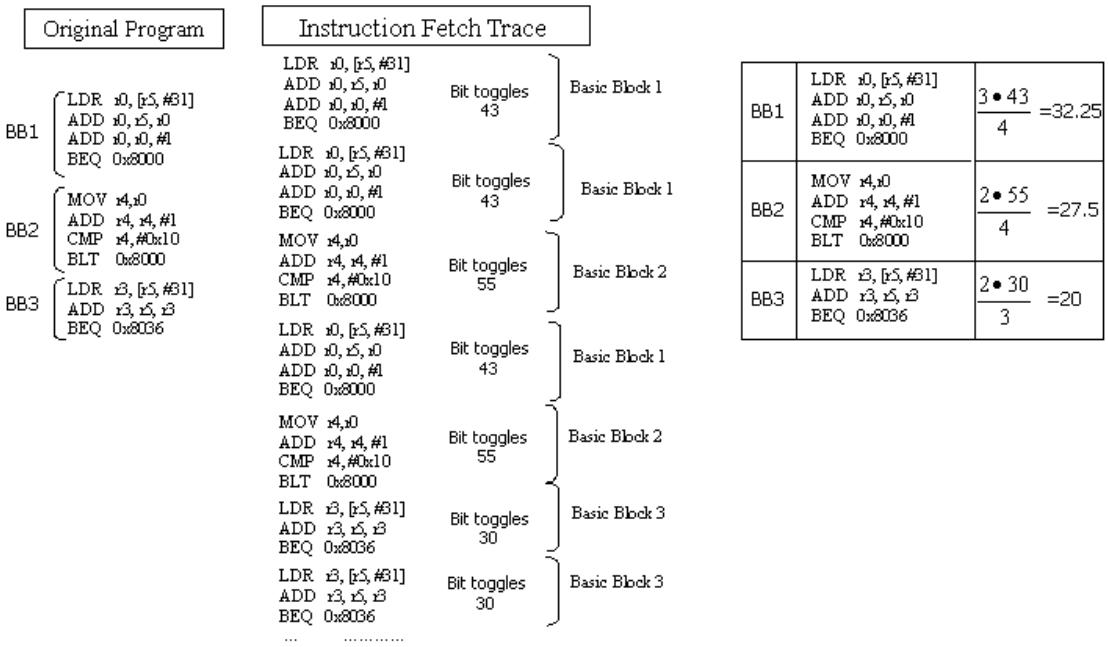

圖 1: Example of greedy-basic -block selector algorithm

Table 1: Computing contribution Ratios of basic blocks

	Frequency	Bit Toggles	Length	Contribution	CR
BB1	3	43	4	129	32.25
BB2	2	55	4	110	27.5
BB3	2	30	3	60	20

3.2 BIBITS Encoding Scheme

目標

BIBITS Encoding 的目標，是要指令編碼成相鄰指令間的 bit toggle 數較少的指令格式。我們將分成兩個步驟來做，首先是切割適當之 encoding field 的大小，再來將切好的 encoding 欄位選擇適當的 encoding function。

方法

由於我們的最終目標是要將 register relabeling 後的指令進行編碼，因此，切割適當之 encoding field 的方法會受限於 target instruction set architecture 的 register field 位置，以 MIPS 為例我們的切割方式如圖 2，灰色部份即為要 encoding 的單位。

圖 2: BIBITS encoding method

再來我們根據是否容易 decode 以及是否適合作 register relabeling 等考量選擇四種 encoding function，分別為 identify，invert，XOR，and XNOR。

Hardware Support

如圖 3，此編碼設計需要一些相關的硬體支援電路，包括 Basic Block Identification Table (BBIT)為了辨認是否為 encoding 過的 instruction，Transformation Table(TT)用來查出解碼用的控制訊號，Decoder 則依照控制訊號將 encode 過的指令還原。

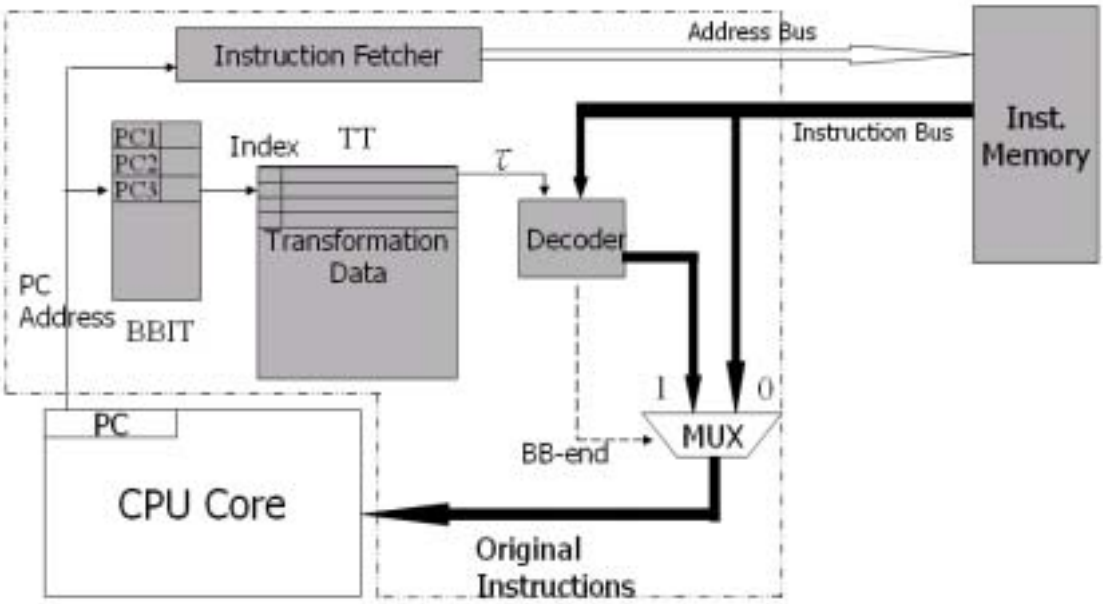

圖 3: System architecture with BIBITS encoding

3.3 BIBITS Encoding Scheme with Register Relabeling

目標

主要目標是設計可使 BIBITS Encoding 發揮最大效益的 register relabeling algorithm。換句話說，就是先做 register relabeling，但 relabel 的 register，選用可使未來 BIBITS Encoding 時 bit toggle 數最小的 register。因此，BIBITS Encoding 方法不變，但須特別介紹 register relabeling 的方法。

Register Relabeling for BIBITS encoding

我們修改[4]所提出的 register relabeling 方法，將其中選擇被置換 register pair 的部份改用依照 EQ2 的 score function 所計算出來的大小來選擇可用的 register。而至於選擇用來置換的 register pair 則依照其置換過後總 bit toggle 數最小為原則。

$$Score = Frequency \times Min\ (\ H.D.\ (R_x,\ R_y),\ 5 - H.D.\ (R_x,\ R_y))\quad (EQ2)$$

4. 結果與討論

在這小節中我們評估每個標竿程式在不同的技術方法應用下的 bit transition 數量。我們利用觀察指令記憶體到 CPU 間的資料匯流排上的 transition 減少數量來判斷這個方法的功效。我們採用一般典型的嵌入式處理器架構來當作比較基準架構。我們挑選了六個跟數位訊號處理與數值運算相關的評估標竿程式，這些都是在大部分的嵌入式產品中的程式常會出現的運算，而且這些程式的主要特徵都是包含了大量的迴圈的執行。下表為這六個標竿程式的說明。

mmul	A matrix multiplication of (100x100)-element matrices
sor	Successive over-relaxation on a (256x256)-element matrix
ej	Extrapolated Jacobi-iterative method on a (128x128)-entry grid
fft	Fast Fourier transform with 256-bit sample block size
tri	Tri-diagonal system solver on a (128x128)-element matrix
lu	Lower/upper triangular matrix decomposition algorithm on a (128x128)-element matrix

我們模擬的環境採用 MIPS SDE-LITE toolset，它採用了 MIPS R4000c 指令集架構，用來分析與統計程式執行的狀況。將這些獲得的資訊輸入我們寫了一個後端工具，將原本的標竿程式採用各種所提出的方法加以調整與編碼，之後再將這些修改過的程式進行模擬，進一步計算出指令匯流排上的 transition 數量。

我們實驗了七個不同的技術在不同的標竿程式模擬，並且是針對程式所有的 basic block 都進行編碼，而這七個不同的技術為：Bus-invert、BITS、register relabeling、Petrov's encoding method、BIBITS encoding scheme、BIBITS with classical register relabeling 和 BIBITS with my register relabeling。後面的三個為我們提出的方法。

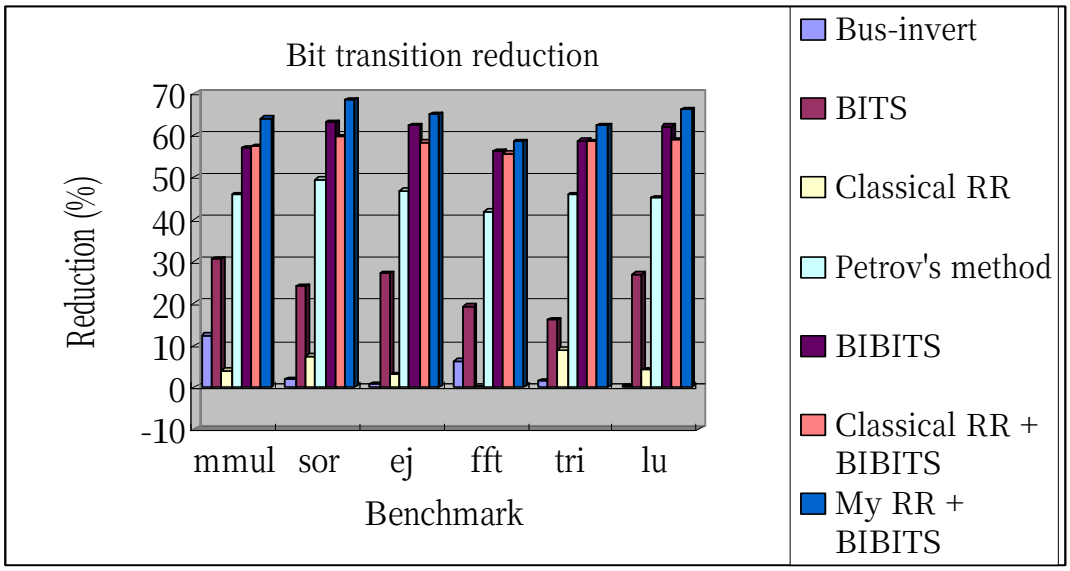

圖 4. 不同技術的 Transition Reduction

從圖 4 實驗的結果可得知我們提出的 BIBITS encoding 方法在 bit transition 減少的量約 56%到 63%。而在 BIBITS encoding 方法前加上 classical register relabeling 大部分狀況下都比單純只做 BIBITS encoding 方法還差。因此最後一個我們提出的方法就是來改前一個方法的缺點，從實驗結果看來確實得到改善，使得這個方法可以在不再增加額外硬體負擔下比單純只做 BIBITS encoding 可以再減少 3 到 7 個百分比。而且我們提出的方法跟 Petrov 的方法比，只需要它使用的 table 大小的一半，而且 decoder 也較為簡單。

參考文獻

[1] M.R. Stan, W.P. Burleson, “Bus- invert coding for low-power I/O,” IEEE Trans. on VLSI Systems, Vol. 3, No. 1, pp. 49-58, Mar. 1995

[2] Huzefa Mehta, Robert Michael Owens, Mary Jane Irwin, Rita Chen, Debashree Ghosh, “Techniques for Low Energy Software,” Low Power Electronics and Design, 1997. Proceedings.,International Symposium on , Aug. 1997 Pages:72 – 75

[3] P. Petrov, A. Orailoglu, “Compiler-Based Register Name Adjustment for Low-Power Embedded Processors,” International Conference on Computer Aided Design (ICCAD), November, 2003.

[4] Seungdo Woo,“A Low-Power Register Relabelling Technique,”

[5] Peter Petrov and Alex Orailoglu, ”Application-Specific Instruction Memory Customizations for Power-Efficient Embedded Processors,” IEEE Design and Test of Computers magazine Jan 2003 pages : 18-25

計畫成果自評

本研究主題著重在 instruction bus 上的省電機制設計，當前 SOC 架構上亦含有此類 bus，因此仍有高度相關程度。此外，研究結果顯示，其省電效果相較於同類型的研究有更好的表現，實做上只需對原始 code 預先處理以及在 processor 外增加少量的解碼電路支援，並不困難。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。