

行政院國家科學委員會專題研究計畫 期中進度報告

可重組式計算於複雜算術系統的應用--以圖形顯像為例

(1/3)

計畫類別：個別型計畫

計畫編號：NSC91-2213-E-009-062-

執行期間：91年08月01日至92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

報告類型：精簡報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中 華 民 國 92 年 6 月 2 日

行政院國家科學委員會專題研究計畫成果報告

可重組式計算於複雜算術系統的應用--以圖形顯像為例(1/3)

Reconfigurable Computing for Complex Arithmetic Systems -- Using Rendering as an Example (1/3)

計畫編號：NSC 91-2213-E-009-062

執行期限：91年8月1日至92年7月31日

主持人：鍾崇斌 國立交通大學資訊工程系

共同主持人：單智君 國立交通大學資訊工程系

計畫參與人員：蔣昆成、陳美璇、任軒、盧惠貞、曾
啟華、
李孟道

一、摘要

隨著 SoC (System-on-Chip) 的出現並逐漸成為消費性電子產品的主流，系統單晶片的設計已由單純的效能提昇而逐漸涵蓋到應用的彈性、系統的整合度、以及與周邊設備的互動等重要觀念。本計畫針對繪圖顯像之應用對可重組式計算進行相關研究，提出適當可行的可重組式架構以及配合硬體架構之相關的編譯技術。

根據上述的想法，我們選擇針對圖形顯像上的應用，是因其具有較長的運算時間、大量且平行性資料及管線式運算結構的性質，特別適合於專屬硬體實作以發揮高平行性計算的優點，以及需要設計相關的編譯技術做緊密的結合。在三年期計畫中，工作項目區分如下：第一年，標竿程式分析與編譯技術之設計；第二年，算術單元與可重組式管線之設計；第三年，算數架構與可重組式管線之實作與效能分析。

在第一年度中，我們針對圖形顯像之標竿程式進行資料流程圖、迴圈特性及算術特性上的研究分析，並設計相關的編譯技術。研究成果有以下四個方面：(1) 軟體程式切割技術；(2) 資料流程圖切割技術；(3) 程式分割排程技術；(4) 高繞線效率的硬體配置架構。以上的成果，將可作為未來兩年研究規劃與實作設計之參考數據與相關分析工具。

關鍵詞：可重組式計算、系統單晶片、高效率繞線、即時重組、資料流程圖

Keywords: Reconfigurable Computing, SoC, Wiring area-efficient, Run-time reconfigurable, Data flow graph

二、緣由與目的

隨著積體電路系統的應用範圍快速擴展，可重組式計算(reconfigurable computing)的觀念已被提出，且成為重要的研究課題。由於近年來嵌入式系統與消費性電子產品的迅速發展，造成產品週期愈來愈短，而效能需求卻愈來愈高。在研發時程與效能需求的雙重壓力下，可重組式架構的觀念日顯重要。可重組式架構包含可重設運算功能的處理單元，及可重設資料路徑的連結網路。透過重設硬體功能與資料路徑，可重組式架構具備下列功用：(1) 縮短產品研發時間；(2) 使硬體架構直接切合應用程式的需要；(3) 以有限的晶片面積進行大規模運算。目前可重組式架構已成功的應用於實現即時影音播放[1]、combinatorial search[2]、與 stream processing[3]等多項應用中。

可重組式計算的研究，包含硬體架構設計與軟體編譯技術二方面。目前最常見的系統是：以 FPGA (field programmable

gate array) 作為硬體架構 [1][2][4][5][6]，利用 High level synthesis 技術 [7] 將軟體或硬體描述語言的應用程式編譯為應用專屬電路，再以 placement and route [4] 方法將電路對應至可重組式架構的處理單元與資料路徑上。

儘管已有諸多技術被提出，當可重組式計算方式要被應用到更複雜的對象上時，卻遇到下列困難：(1) 於硬體設計時應用對象尚不明確，以至於繞線佔用過多晶片面積 [6]，(2) 缺乏成熟的編譯技術來規劃有限的硬體資源在不同時間點如何利用，造成許多硬體資源閒置，計畫中將探討上述困難的因應之道，並? 可重組式計算方式開發更複雜的應用對象—以 3D 繪圖程式為代表的複雜算術系統。

本計畫將針對以繪圖顯像(rendering) 程式為代表的複雜算術系統，探討可重組式計算系統的軟硬體共設計方法。研究時最佳化的方向，是儘可能提升每單位晶片面積能增加的執行效能。欲達成的目標，在硬體與軟體方面分別條列如下：

(1) 在硬體方面：設計一可重組式管線系統(reconfigurable pipelined system) 架構，以硬體實現多種不同形式的 rendering pipeline。

(2) 軟體方面：提出編譯技術設計方法，以充分利用有限的硬體資源，並發揮硬體管線高資料處理率的潛能。

我們並以效能模擬與實作部份電路的方式，對所提各項設計方案進行評估。

三、結果與討論

本計畫第一年度的主要項目如下：

1. 軟硬體程式切割技術

在一般處理器與可重組式硬體的環境裡，把一應用中迴圈部份利用可重組式硬體加速時，所需要的重組時間會產生兩個問題：(1) 部份迴圈在軟硬體切割(HW/SW partitioning) 的步驟中因為重組的時間大於可重組式硬體的加速利益，而選擇不利用可重組式硬體加速；(2) 決定由可重組式硬體執行的迴圈都需要至少一次的重組，花費在重組的時間很長。

為了克服上面所提及的問題，以下將

提出 HW/SW partitioning procedure，步驟如下：(1) 比較重組的時間和利用可重組式硬體所減少的時間 [15]。(2) 針對尚未決定由可重組式硬體或一般處理器執行的 candidates，提早在一般處理器執行時，進行重組的動作 [16]。(3) 在不違反可重組式硬體的大小和合併前的執行時間的狀況下與已經決定切割的迴圈，合併在同一次的重組中。

在圖形顯像的應用中，具有大量迴圈及大量資料平行性的特性。因此，對於迴圈的硬體化及分割排程有助於大量提高運算的效能。

2. 資料流程圖切割技術

我們修改軟體設計流程，使得在 Multi-FPGAs 架構 [9] 下 Stream 形式資料 [7] 的產量可以有效的提升，而 Software Design Flow [8] 的主要工作便是把應用程式對應到 Reconfigurable Hardware 上執行，其流程可分為兩大部分：

(1) 分割及預先排程 (Partition and Pre-Schedule)：首先我們使用 ASAP [10] and FDS [11] 演算法來對由行為規格書 (Behavioral Specification) 產生出來的 Data Flow graph (DFG) 排程，產生 Scheduled DFG (SDFG)。接下來，我們根據硬體資源和運算執行時間來對 SDFG 作切割，得到 Sub-SDFGs。

(2) 排程 (Schedule) [10]：我們根據資料相依性，來決定每一個 FPGA 的執行順序，得到 scheduled Sub-SDFGs，然後將之轉成硬體描述語言，使其產生實際的硬體以便實際執行。

在實際的應用上，因為可重組式架構具有動態重組硬體運算架構上的優點，因此，不會也不必要將整體應用需求的硬體同時製作於可重組式硬體上。分析整體應用的需求時間性可分割為獨立的 DFG，利用此排程方式可大幅減少因為更換硬體組態造成的效能減少並大幅增加可重組式硬體的利用率。

3. 程式分割排程技術

在部分可重組式系統 (Partial Reconfigurable System, PRS) [13] 中，

可以利用不同程式分割(partition)間的重組和執行時間的可重疊性來隱藏減少所需的總執行時間，我們在這裡提出 Partitions-to-Blocks scheduling (PBS)。

在 PBS 中採用 Cluster-Based [14] 分割法來切割 DFG，此 schedule 主要有三個步驟：(1) *Combination Candidate Generation*：針對已被重組之硬體的所有可能狀態，以重組時間(Reconfiguration Time)為考量來選擇程式分割(partition)和可重組硬體區塊(block)之間的組合。(2) *Combination Candidate Replacement*：以分割的可執行時間來考量步驟(1)中所產生的組合是否改變。(3) *Reconfigured Blocks Execution Sequence*：將已決定好的可重組硬體區塊，產生執行序列並執行之。

PRS 為新一代的重組式硬體架構，此架構解決了重組時間對效能上的影響，相對地其有效的排程及硬體分配將可使重組時間大幅減少且不影響硬體的執行時間。在重組式架構上將在效能上大幅趕上特殊應用 IC (ASIC) 並同時具有應用彈性的優點。

4. 高繞線效率的硬體配置架構

在減少繞線面積方面，我們設計一個可以減少繞線面積的接線方法 (interconnection)，使得晶片的尺寸較小且資源利用率高，設計流程為：(1) 從特殊領域的應用分析主要的迴圈；(2) 把這些迴圈轉成 DFGs，並且用 [12] 的方法把他們合併在一起；(3) 分配上一步驟中產生的 DFG 裡面每一個運算應用到哪一個 logic block；(4) 根據 cost-function 來選擇適當的路徑。經過上述的步驟，並應用 island style FPGA [12] 來當我們的硬體結構。

由於製程技術的不斷進步，使得運算單元的設計更為緊密不佔空間，但是運算單元之間所需要的接線面積，卻隨著製程的進步而有大量的減少，若無法更進一步精簡繞線，將使得繞線面積在晶片上佔去過多比例；因此本計劃對於硬體的設計，本年度著重於繞線數量的精簡，除了能縮小晶片尺寸之外，亦能使資源利用率提高，為未來提升每單位晶片面積能增加的

執行效能作準備。

四、計畫成果自評

在第一年的計畫執行過程中，我們針對圖形顯像標竿程式的特性進行一系列的分析，並找出圖形顯像應用中最常出現的程式特性並對於這些情況加以分析模擬，使得硬體設計前的規劃與實作設計更趨完整與有效率。在未來的兩年計畫中，我們將針對這些特性，設計更加有效率、有彈性及節省面積的 SoC 平台。對於未來的 SoC 發展相信可以提供一個快速有效率的設計方法。

五、參考文獻

- [1] S. D. Haynes, et. al., "Video image processing with the Sonic architecture," *IEEE Computer*, Vol. 33, No. 4, pp. 50-57, 2001.
- [2] M. Platzner, "Reconfigurable accelerators for combinatorial problems," *IEEE Computer*, Vol. 33, No. 4, pp. 58-60, 2001.
- [3] E. Caspi, et. al., "Stream computation organized for reconfigurable execution (SCORE): Introduction and Tutorial," *Proceeding of 10th Int'l Conf. Field-Programmable Logic and Applications*, 2000
- [4] V. Bets and J. Rose, *Architecture and CAD for Deep Submicron FPGAs*, 1999.
- [5] J. Hauser, and J. Wawrzynek, "Garp: A MIPS Processor with a Reconfigurable Coprocessor," *Proc. IEEE Symp. Field-Programmable Custom Computing Machines*, 1997.
- [6] Andr'e DeHon: *Reconfigurable Architectures for General-Purpose Computing*, A.I. Technical Report No. 1586, Massachusetts Institute of Technology. 1996
- [7] Michael I. Gordon, William Thies, Michal Karczmarek, Jasper Lin, Ali S. Meli, Andrew A. Lamb, Chris Leger, Jeremy Wong, Henry Hoffmann, David Maze, and Saman Amarasinghe, "A Stream Compiler for Communication-Exposed Architectures", *ASPLOS*, 2002
- [8] Katherine Compton, Scott Hauck,

- “*Reconfigurable Computing: A Survey of Systems and Software*”, ACM Computing Surveys, Vol. 34, No. 2, June 2002, pp. 171–210.
- [9] Vinoo Srinivasan, Sriram Govindarajan, and Ranga Vemuri” *Fine-grained and coarse-grained behavioral partitioning with effective utilization of memory and design space exploration for multi-FPGA architectures*”, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, February 2001
- [10] Karthikeya M. Gajjala Purna, and Dinesh Bhatia, “*Temporal Partitioning and Scheduling Data Flow Graphs for Reconfigurable Computers*”, IEEE Transactions On Computers, June 1999
- [11] D. D. Gajski, “*High-Level Synthesis: Introduction to Chip and System Design*”, 1992
- [12] N. Moreano, G. Araujo, Z. Huang, S. Malik, ”Datapath Merging and Interconnection Sharing for Reconfigurable Architectures”, Proc. of the 15th. ACM/IEEE International Symposium on System Synthesis, 38-43, October 2002
- [13] Temporal Partitioning combined with Design Space Exploration for Latency Minimization of Run-Time Reconfigured Designs, Meenakshi Kaul, Ranga Vemuri Laboratory for Digital Design Environments, Department of ECECS University of Cincinnati, Cincinnati, OH 45221-0030
- [14] Temporal Partitioning and Scheduling Data Flow Graphs for Reconfigurable Computers, Karthikeya M. Gajjala Purna, Student Member, IEEE and Dinesh Bhatia, Member, IEEE
- [15] Hardware-Software Co-design of Embedded Reconfigurable Architecture Yanbing Li, Tim Callahan, Ervan Danell, Randolph Harr, Uday Kurkure, Jon Stockwood, 2000 ACM
- [16] Configuration Prefetch for Single Context Reconfigurable Coprocessors Scott Hauck, ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 1998