

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

(3/3)

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-009-001-

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：蔣昆成、陳治璋、賴敬中、黃國晉、黃富群、林聖勳、莊富元、  
黃士嘉

報告類型：完整報告

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 94 年 10 月 31 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(3/3) **Reconfigurable Computing for Complex Arithmetic Systems** — **Using Rendering as an Example (3/3)**

計畫編號：NSC 91-2213-E-009-062

執行期限：93 年 08 月 01 日至 94 年 07 月 31 日

計畫主持人： 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、陳治瑋、賴敬中、黃國晉、黃富群、  
林聖勳、莊富元、黃士嘉

### 中文摘要

面對可攜式裝置對於多媒體運算的需求急遽的提升，一個可以支援多媒體運算能力的單晶片架構成為產品成功與否的關鍵因素。然而，在大部分的應用上一般用途處理器無法完全滿足即時運算的要求。雖然特殊應用的加速器可以獲得執行效能上的有優點，但缺乏彈性的缺點導致其嚴重受到應用的廣泛性與短暫的產品生命週期而使其用途受到限制。因此，我們的研究致力於開發可重組式計算架構來執行 3D 繪圖顯像的應用。

為了在可重組式環境模擬 3D 繪圖顯像的運作，我們建立一個執行平台包括：一個一般用途處理器、一個處理單元陣列、與一個共用的記憶體提供前兩者間的資料傳輸用途。除此之外，我們使用一個區域性的記憶體於處理單元陣列中來儲存處理單元的執行組態與執行過程中的暫存資料。而組態控制器與資料存取控制器則負責提供必備的控制訊號來切換運算單元的工作模式與運算單元間資料的傳輸路徑。

我們選擇 3D 圖形顯像上的應用，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，同時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。

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。在第一年度中具有以下四個研究成果：(1)軟硬體程式切割技術；(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。

在第二年的計畫執行中，我們進一步設計可重組式架構及其算術運算單元。利用此架構，我們將 3D 繪圖顯像的應用在此架構上進行模擬並且發現這種設計可以有效提升運算的效率與邏輯電路使用率。除此之外，我們採用具有類似運算結構的 MPEG-4 編碼流程，配合有效的排程演算法執行於本架構上發現該應用亦可有效提升編碼執行效率。在第二年度的計畫執行中，我們展現出可重組式架構在應用上的彈性。

第三年的計畫執行中，我們對於算數運算單元與可重組式架構的設計以 Verilog 硬體描述語言進行實作，針對應用設計不同的運算單元與架構，分析其在執行時間與硬體資源上的優劣。另外，在傳輸網路的繞線設計上，我們設計一個配置與繞線的演算法來實作運算單元間的資料繞線。

**關鍵詞：**可重組式計算、系統單晶片、繪圖顯像、MPEG4

## Abstract

With the dramatically increasing demands for multimedia processing capabilities in portable electronic devices, the architectural supports for various multimedia computations in one platform are crucial to the success of a product. However, a general-purpose processor can not meet the real-time criteria in many cases. Although application-specific accelerators have performance advantages, their inflexibilities seriously impede the applicability and shorten product lifetime. Therefore, the aim of our research is to design a reconfigurable platform, and to execute 3D-rendering as an example application.

In order to simulate 3D-rendering in a reconfigurable environment, we develop a platform consisting of a general-purpose CPU, an array of processing elements, and a shared memory to transfer data between them. In addition, a memory local to the array is used to store array configuration contexts and temporary data, and two additional controls, the configuration controller and the data fetch controller, are employed to provide essential control signals while executing applications in different stages. Processing elements can be assigned different function and their interconnections tuned to suit different applications.

We choose 3D rendering as an example. It has the features of longer computation time, large amount of data with highly parallelism, and pipelining computation structures. If we develop a dedicated hardware for it, it can achieve the benefit of highly parallel computations. We also need the technology of compiler that can cooperate with the reconfigurable hardware tightly. In the three-year project, we spilt the jobs as following. In the first year, we analyze the benchmark and design a technology compiler. In the second year, we design the reconfigurable architecture and processing element. And in the last year of the project, we implement the reconfigurable architecture and analyze its performance.

We analyze the data flow graphs, loop features and computation features of the benchmark of 3D rendering application. According to these features we design associated compile technology for them. In the first year, we have four results as follows. (1) Hardware/software program partition (2) Data flow graph partition (3) program scheduling and (4) highly routing utilization for placement and route.

Furthermore, we design the architecture of reconfigurable computation and processing element in the second year of the project. By using this architecture, we simulate the application of 3D rendering and find out that the performance of execution is improved and the utilization of hardware circuit is also better. In addition, we use the application of MPEG-4 encoding on our

reconfigurable architecture. The MPEG-4 has almost the same features as 3D rendering on computation time and large amount of data. To cooperate with high efficient scheduling technology, the architecture can also achieve high performance of execution. In this year, we demonstrate that the reconfigurable architecture have higher flexible on multi applications products.

In the third year of the project, we design and implement the processing elements and reconfigurable architecture in verilog hardware description language. For different designs of processing elements and reconfigurable architecture, we analyze the execution time of it and the use of hardware resource. In addition, for the routine on the transportation network, we design a placement and route algorithm for executing the routines of between processing elements.

**Keywords :** Reconfigurable Computing, SoC, Rendering Pipeline, MPEG4

# 目 錄

中文摘要 .....	I
Abstract .....	II
目 錄 .....	IV
一、前言 .....	1
二、研究目的 .....	2
三、研究方法 .....	2
1. 可重組式架構之設計 .....	2
2. 可重組式運算單元之設計 .....	3
3. 利用運算排程提高 3D 繪圖顯像執行效率 .....	4
4. 軟硬體程式切割技術 .....	5
5. 資料流程圖切割技術 .....	5
6. 程式分割排程技術 .....	5
7. 高繞線效率的硬體配置架構 .....	6
8. 應用在 MPEG-4 的排程方式 .....	6
四、結果與討論 .....	7
五、參考文獻 .....	8

# 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(3/3) **Reconfigurable Computing for Complex Arithmetic Systems** — **Using Rendering as an Example (3/3)**

計畫編號：NSC 91-2213-E-009-062

執行期限：93 年 08 月 01 日至 94 年 07 月 31 日

計畫主持人： 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、陳治璋 …

### 一、前言

在過去十年中，可攜式行動裝置已成為最重要的消費性產品之一，例如：數位相機、行動電話、手持式電動玩具與個人數位助理。為了使這種產品提高對消費者的吸引力，特色化與個人化的屬性成為產品的成功主要因素。人們希望能夠除了利用行動電話來打電話之外，同時希望它具有照相、瀏覽圖片與聽 MP3 音樂的功能。無庸置疑地，多媒體應用是可攜式裝置中非常重要的功能。隨著液晶螢幕 (LCD, Liquid Crystal Display) 尺寸的增加與消費性產品中各式各樣的需求，3D 圖學在多媒體應用的趨勢上是不可避免的。3D 圖學的運算處理需要大量的數學運算，因此通常運用專屬特殊的加速器來提供即時顯示的能力[1, 2, 3]。

有許多計算架構被提出並用來提升 3D 顯示的運算執行速度，這些架構結合了中央處理器與一個 3D 繪圖引擎來實現[4]。Chung 等人[5] 整合了 3D 繪圖處理管線與記憶體控制單元以同時處理多重且經過分割的資料。而另一方面，Takemoto 等人[6] 結合 3D 繪圖引擎、MPEG4 編碼/解碼引擎、JPEG 編碼/解碼引擎與兩千萬位元

的動態隨機存取記憶體成為單一晶片以支援多媒體應用的多樣需求。雖然這些計算架構可以達到即時運算的需求，其硬體資源的運用將會因為系統執行不同的應用而造成其他運算引擎的停滯與浪費。其他被提出來的架構結合向量運算 (vector processing) 的技術與動態隨機存取記憶體來促進多媒體應用的執行效能。Kozyrakis [7] 提出向量式智慧型記憶體 (Vector-IRAM)，它結合了向量處理器與嵌入式動態隨機存取記憶體來加速多媒體運算的速度。雖然，向量處理器在具大量運算特性的應用上擁有非常高的執行效能。另一方面，可重組式架構提供達成即時運算的需求與易於減少開發上的錯誤、易更換應用、易重覆使用運算元件及易於產品差異化之方法。可重組式計算架構被提出來解決系統單晶片 (SoC) 上日益增加的多媒體應用之需求 [8]。根據可重組式架構的運算單元 (Processing Element, PE) 的運算能力大小 (即，其複雜度)，可分為 (1) 小運算單元架構 (Fine-grained architecture)，如：可重組式邏輯陣列 (Field Programmable Gate Array, FPGA) [9-11]，與 (2) 大運算單元架構 (Coarse-grained architecture)。相較於

大運算單元架構，小運算單元架構具有較高的功能彈性但是較低的執行效能。

許多小運算單元架構被提出來加速大量資料與多媒體應用[12-16]，不僅如此，Altera[9]與 Xilinx[10]以經生產了結合精簡指令集處理器 (RISC processor) 與可重組式邏輯陣列的可重組式開發平台 (Reconfigurable platform)。小運算單元架構具有易於實作電路與編譯器支援的優點[17-19]，然而，他們的傳輸往路的繞線面積通常佔大量的整體晶片面積以致於不利於低價的消費性電子產品。大運算單元架構中的運算單元具有較強大的運算能力與較少的傳輸繞線面積。因此，它大量地減少晶片的面積[20]。然而，大運算單元架構由於其複雜的運算結構與多變的執行時間，不易開發相對應較複雜的編譯器[21]。

## 二、研究目的

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[22]、Combinatorial search[23]、與 Stream processing[24]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 [22, 23, 25, 26, 27]，利用 High level synthesis 技術[28]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 [25]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[27]，(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，並運用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：設計一個可重組式管線系統 (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。

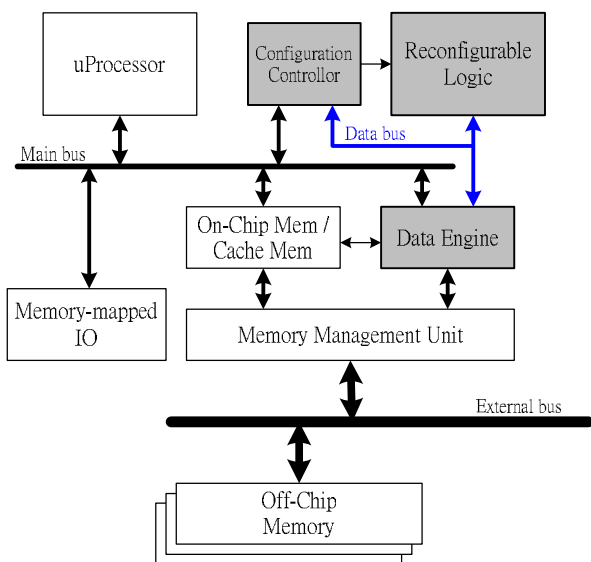
## 三、研究方法

### 1. 可重組式架構之設計

如圖一所示，可重組式系統單晶片由以下單元所組成：(1)32 位元微處理器，

如 MIPS, ARM7 等一般用途微處理器(2)快取記憶體 / 晶片內記憶體 (Cache Memory/On-chip Memory)及記憶體控制單元 (Memory Management Unit) (3)記憶體映像輸出/輸入介面 (4)可重組式架構則包括：硬體配置控制器 (Configuration Controller)、可重組式邏輯 (Reconfigurable Logic) 與資料引擎 (Data Engine)。

應用程式由編譯器在編譯階段決定軟體與硬體執行程序的分割，決定那些運算需由可重組式架構執行。應用程式執行時，由微處理器將程式由晶片外記憶體載入並開始執行。當應用程式執行到硬體執行的程序之前，微處理器會向硬體配置控制器要求配置可重組式邏輯，並在配置完成後視需要提出執行的要求。



圖一、可重組式系統單晶片

硬體配置控制器 (Configuration Controller) 負責接收 CPU 所發出的新的硬體配置需求，根據此需求向資料引擎要求所需的資料並將這些資料配置到可重組式邏輯中準備執行。直到微處理器發出執行要求後，控制可重組式邏輯開始執行。

可重組式邏輯 (Reconfigurable Logic) 由運算單元陣列為主要組成，加上相對應的資料儲存/讀取單元輔助其資料的需求。由於此架構以 32 位元 CPU 為核心因此其運算單元的設計重點為執行一個 32 位元運算、兩個 16 位元運算以及四個 8 位元運算等三種模式以符合原始應用程式的編譯流程。其目標為取代一連串的 32 位元指令的執行。

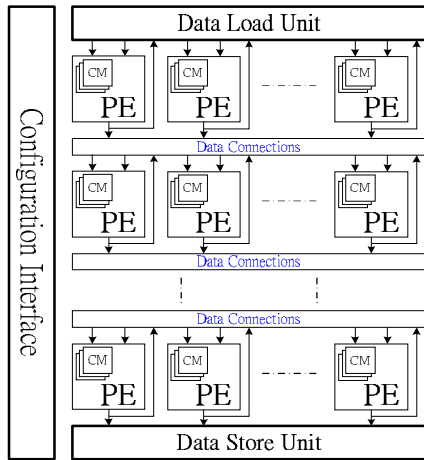
資料引擎 (Data Engine) 則是負責存取硬體配置資料以及計算資料，提供硬體配置控制器與可重組式邏輯的資料需求。同時，維持 CPU 的快取記憶體與可重組式邏輯間的資料一致性與相依性。

## 2. 可重組式運算單元之設計

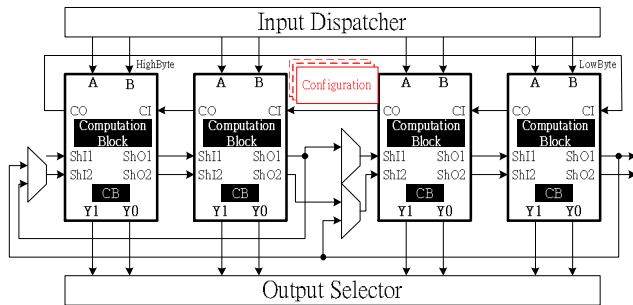
可重組式邏輯由 (1)可重組式運算單元陣列(2)硬體配置介面，及(3)資料儲存/載入單元所組成。陣列中的基本運算單元經由資料連結網路形成運算單元陣列，編譯程式可以根據實際的陣列配置產生相對應的資料流 (Data Flow)。在執行階段時，硬體配置介面 (Configuration Interface) 根據硬體配置控制器的指定將配置資料寫入相對應的運算單元以及連結網路。透過改變運算單元的功能執行與連結網路的連接方式達成不同的運算。

在可重組式運算單元設計方面，為了配合 32 位元微處理的架構，一個運算單元是由四個可串接的 8 位元計算方塊 (Computation Block) 所組成。各個方塊具有獨立運算的能力並且可經由兩個為一組或四個為一組執行 16 位元或 32 位元的運算，如圖三所示。同時亦可將兩個或以上的可重組式運算單元串接為 64 位元或以上的運算，在這種高位元的運算情況之下則必須適度的降低執行頻率以配合高位元運算所產生的訊號延遲。





圖二、可重組式邏輯架構圖

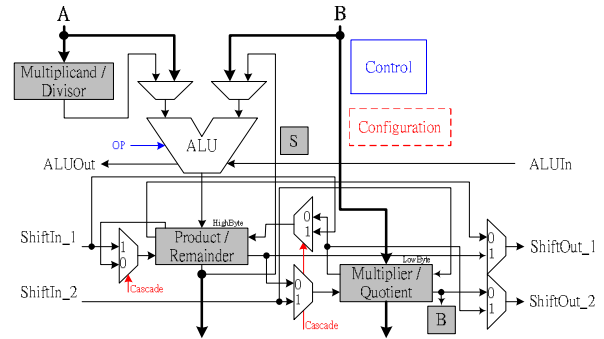


圖三、可重組式運算單元

如圖四所示，八位元計算方塊由一個算數邏輯單元 (ALU)、數個八位元暫存器及控制邏輯所組成。當硬體配置的需求被寫入到計算方塊中，ALU 及控制單元透過解碼瞭解應用程式所要求的功能，並執行指定的運算。此單元可以執行加、減、乘、除及位移等運算，根據運算的複雜度執行時間為一個或多個時脈週期。

### 3. 利用運算排程提高 3D 繪圖顯像執行效率

3D 圖像是由大量的物件存在於不同的向量與位置所組成，因此 3D 繪圖顯像程式需要大量的數學運算且資料之間的平行性極高。由於每筆資料處理的時間不盡相同，使用微處理器來執行的程式效能無法有效提升，因此通常必須設計一個特定的硬體來加速運算效能。

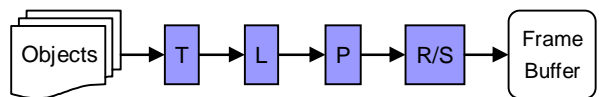


圖四、八位元計算方塊架構圖

為了增加硬體應用的彈性與硬體資源的有效利用，我們使用可重組式運算架構進行大規模的運算，使硬體結構直接切合實際的需求，達到高效能的目的。在探討 3D 顯像程式如何在可重組式系統上執行，透過：

1. 程式分割：提高最大平行性的前提下，將程式分割成大量的程式片段。
2. 工作排程：利用平行特性來重用硬體和並且減少重組的次數進而降低總執行時間。

傳統的 3D 繪圖顯像程式包括以下的流程：Transformation, Lighting, Projection, Rasterization and Shading。每一個 3D 物件必須透過這五種運算產生影像配置到視框緩衝器 (Frame Buffer)，且每個物件間的運算為互相獨立。



圖五、3D 繪圖顯像管線

工作排程的方法可分為 FCFS (First-Come-First-Serve) 與優先權 (Priority) 等方式。FCFS 即是以工作指派的時間為優先順序的依據，這種方法也是最直接簡單的方法。優先權排程方式，則是以五種運算工作間的佇列 (Queue) 目前的需要等待運算的個數為優先權的依據。

根據我們的實驗結果，優先權排程方式可以得到較好的系統效能，而且當硬體配置時間較長時，此種排程方式可以隱藏配置時間對效能的影響。在相同可重組式硬體的前提下，這種排程方式可以減少 30% 至 80% 的執行時間。而在相同的執行時間的條件下，硬體的需求則只需原來的 75%。

#### 4. 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。(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。(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。

#### 5. 資料流程圖切割技術

我們修改軟體設計流程，使得在 Multi-FPGAs 架構下 Stream 形式資料的產量可以有效的提升，而 Software Design

Flow 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 and FDS 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)：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，利用此排程的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。

#### 6. 程式分割排程技術

在部分可重組式系統 (Partial Reconfigurable System, PRS) 中，可以利用不同程式分割 (partition) 間的重組與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 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。

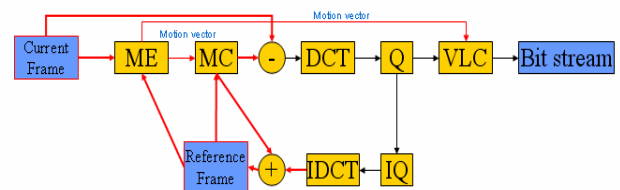
### 7. 高繞線效率的硬體配置架構

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 [29] 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 [29] 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。

### 8. 應用在 MPEG-4 的排程方式

可重組式架構提供一個彈性的硬體運算, 因此本架構並不侷限在 3D 的應用。更進一步地, 我們可以將可重組式架構延伸到各種需要大量運算的應用之上。例如: Digital Signal Process, 語音應用及影像應用等等。在本計畫, 我們嘗試利用架構執行具有類似運算結構的應用上, 以驗證可重組式架構在多元應用的可行性。



圖六、MPEG-4 執行流程

如圖六所示, MPEG-4 的壓縮過程必須經過 Motion Estimation (ME)、Motion Compensation (MC)、Discrete Co-Sin Transform (DCT)、Quantization (Q)、Variable Length Coding (VLC)、Inverse Quantization (IQ)、Inverse DCT (IDCT) 等步驟。其中有大量的矩陣運算與 3D 顯像運算有相當的同質性。因此, 我們將這個應用也納入我們的研究範圍。

我們的研究將工作區分為以上七種步驟並以這些為基本運算單元。我們發現其中大部分的運算時間為 ME, 如果 ME 的演算法為 Full Search 的話, 則 ME 的運算時間佔總時間的百分之九十以上。因此, 我們根據此特性將排程的方法設計為以下兩種方式:

- (1) Partial Reconfigurable scheduling: 不論何種工作均以分時多工的方式排程。
- (2) ME Dedicated scheduling: 由於 ME 的執行時間為整體時間的關鍵, 因此我們規劃一部分的可重組式邏輯專供 ME 使用, 其餘的工作以分時多工的方式共同利

用可重組式邏輯。

根據我們的實驗結果，當程式在 ME 步驟採用演算法 Three-step search (TSS) 時，則 Partial Reconfigurable scheduling；而 Full Search 演算法時，ME Dedicated scheduling 可以有較大的好處。在資料匯流排寬度為 64 位元的情況下，TSS 演算法需要 69 個 PE，Full search 演算法需要 37 個 PE。

#### 四、結果與討論

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。此外，我們針對編譯的技術上的研究，針對大量資料運算的應用研究相關的技術並發表三篇碩士論文：(1)在可動態重組系統中利用預先抓取及合併的技術降低重組負擔[30]、(2)多 FPGA 之可重組式系統中資料流處理的資料流程圖切割方法設計[31]、(3)在應用執行過程中可重新設定組態系統中重新設定時間的減少與隱藏[32]。這些碩士畢業論文針對標竿程式進行分析並設計適用編譯技術對後續的計畫研究有長足貢獻。

在第二年的計畫時程中，我們更進一步地設計可重組式架構及運算單元，使其符合現今消費性產品的需求。這樣的硬體架構可以有效的減短產品的研發時程及降低因為重新設計新晶片所需的風險，可大量提高產品的可行性及獲利率。3D 繪圖顯像應用是未來消費性產品的主流，它讓電腦的應用從平面的限制擴展到三度空間的擬真環境。我們藉由可重組式架構來實做 3D 加速引擎不僅可以加速應用程式的執行

效率，更進一步地當此應用未來有所更新規格時，我們亦可同時更新應用程式與編譯程式達到相同的硬體即可適用於未來相同領域產品的相容性。對於快速的發展原形機有非常大的助益。除了 3D 繪圖顯像的應用之外，我們對於影像應用的 MPEG-4 編碼流程進行應用的分析與架構設計。我們發現這些需要大量資料運算的應用可以有效的利用此可重組式架構來達成，由這一點可以展現出可重組式架構除了在研發時程的優越性更擁有多樣化的應用，對於系統單晶片的設計上，是一個非常適用的架構。針對 3D 繪圖應用的執行排程上我們發表一篇碩士論文：在可重組系統中使用動態重組排程方式增加 3D 顯像程式的效能[33]。

在第三年的計畫執行中，我們根據今年對於可重組式架構系統單晶片及可重組式運算單元之設計，進行硬體實做的工作。以硬體描述語言(HDL)，Verilog，描述各部分硬體的實際動作。將整個系統製作完成，並在硬體電路模擬環境下，驗證系統的正確性。此外，在 3D 繪圖顯像的應用上，將其應用程式編譯為執行程序以相同的模擬環境驗證其演算法及功能的正確性。在本年度的中我們發表了兩篇碩士論文與一篇國際性會議論文：(1)針對一個固定集合的應用設計一個有效減少線路面積的可重組式硬體[34]，(2)針對安全性嵌入式系統之彈性管線化設計與實做[35]，(3) Run-time Reconfiguration Scheduling of 3D-Rendering on a Reconfigurable System[36]。

綜觀為期三年的計畫執行，我們根據初始的規劃發展編譯技術及基礎硬體架構。在編譯技術部分已經發表三篇碩士畢業論文，在排程技術上發表一篇碩士畢業論文及一篇國際性會議論文，而在硬體繞

線技術上發表一篇碩士論文，在其他應用的可行性上的研究上針對 MPEG4 應用與安全性演算法上，我們發表了一篇碩士論文。我們不僅僅對於可重組式架構中編譯技術、繞線技術、執行排程技術與運算單元設計上發表具體的研究。對於其他應用的發展亦有具體的成果發表，並期望這些成果對於國內發展可攜式消費電子產品上提供具體的助益。

## 五、參考文獻

- [1] Alan Watt, 3D Computer Graphics, Third edition, Addison-Wesley, USA, 2000.
- [2] K. L. Ma and S. Parker, "Massive Parallel Software Rendering for Visualizing Large-Scale Data Sets," IEEE Computers and Applications, Vol. 21, No. 4, pp. 72-83, July/August 2001.
- [3] H. Hoppe, "View-Dependent Refinement of Progressive Meshes," Proceeding of SIGGRAPH '97, pp. 189-198, 1997.
- [4] A. Dasu and S. Panchanathan, "A Survey of Media Processing Approaches," IEEE Transactions on Circuit and System for Video Technology, vol. 12, no. 8, August 2002.
- [5] Chris Y. Chung, Ravi A. Managuli, and Yongmin Kim, "Design and Evaluation of Multimedia Computing Architecture Based on a 3D Graphics Pipeline," Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'02), 2002.
- [6] Takashi Takemoto, Yasuharu Takenaka, Tsutomu Minagawa, Tomohiro Koizumi, Yasuyuki Ushijima, Naoaki Yanagida, Yasuo Ohara, Kouichi Tannka, and Yasuhiro Fujita, "T4G: Media Processor Including 3D Graphics for Mobile Set Based on Configurable Processor", 2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, Aug. 4-5, 2004.
- [7] C. E. Kozyrakis, "A Media-Enhanced Vector Architecture for Embedded Memory Systems," Technical Report UCB/CSD-99-1059, UC Berkeley, July, 1999.
- [8] Francisco Barat, Rudy Lauwereins, and Geert Deconinck, "Reconfigurable Instruction Set Processors from a Hardware/Software Perspective," IEEE Transactions on Software Engineering, Vol. 28, No. 9, September 2002.
- [9] S. Brown, and J. Rose, "Architecture of FPGAs and CPLDs: A Tutorial," IEEE Design and Test of Computers, vol. 13, no. 2, pp. 42-55, 1996.
- [10] ALTERA INC., Stratix II Device Handbook, <http://www.altera.com>, San Jose, CA, March 2005.
- [11] XILINX INC., Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet, <http://www.xilinx.com>, San Jose, CA, March 1, 2005.
- [12] R. Razdan and M.D. Smith, "A High-Performance Microarchitecture with Hardware-Programmable Functional Units," Proceeding of 27th int'l symp. Microarchitecture (MICRO 27), pp. 172-180, Nov. 1994.
- [13] M. J. Wirthlin, and B. L. Hutchings, "A Dynamic Instruction Set

- Computer," Proceeding of Workshop FPGAs and Custom Computing Machines (FCCM '95), pp. 99-107, 1995.
- [14] R. D. Witting and P. Chow, "OneChip: An FPGA Processor with Reconfigurable Logic," Proceeding of Workshop FPGAs and Custom Computing Machines (FCCM '96), pp. 126-135, 1996.
- [15] J. R. Hauser and J. Wawrzynek, "Garp: A MIPS Processor with a Reconfigurable Coprocessor," Proceeding of Workshop FPGAs and Custom Computing Machines (FCCM '97), pp. 12-21, 1997.
- [16] S. Hauck, T. Fry, M. Hosler, and J. Kao, "The Chimaera Reconfigurable Functional Unit," Proceeding of Workshop FPGAs and Custom Computing Machines (FCCM '97), pp. 87-96, 1997.
- [17] Timothy J. Callahan, John R. Hauser, and John Wawrzynek, "The Garp Architecture and C Compiler," *IEEE Computer*, vol. 33, no. 4, pp. 62-69, Apr. 2000.
- [18] Z. A. Ye, N. Shenoy, and P. Banerjee, "A C Compiler for a Processor with a Reconfigurable Functional Unit," Proceeding of Eighth Int'l Symp. Field-Programmable Gate Arrays (FPGA '00), 2000.
- [19] M. Budi u and S. C Goldstei n, "Fast Compilation for Pipelined Reconfigurable Fabrics," Proceeding of Seventh Int'l Symp. Field-Programmable Gate Arrays (FPGA '99), pp. 195-205, 1999.
- [20] S. C. Goldstein, H. Schmit, M. Moe, M. Budi u, A. Cadambi, R. R. Taylor, and R. Laufer, "PipeRench: A Coprocessor for Streaming Multimedia Acceleration," Proceeding of 26th Int'l Symp. Computer Architecture (ISCA '99), pp. 28-39, May 1999.
- [21] S. C. Goldstein, H. Schmit, M. Budi u, S. Cadambi, M. Moe, and R. R. Taylor, "PipeRench: A Reconfigurable Architecture and Compiler," *IEEE Computer*, vol. 33, no. 4, pp. 70-77, Apr. 2000.
- [22] S. D. Haynes, et. al., "Video image processing with the Sonic architecture," *IEEE Computer*, Vol. 33, No. 4, pp. 50-57, 2001.
- [23] M. Platzner, "Reconfigurable accelerators for combinatorial problems," *IEEE Computer*, Vol. 33, No. 4, pp. 58-60, 2001.
- [24] 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
- [25] V. Bets and J. Rose, *Architecture and CAD for Deep Submicron FPGAs*, 1999.
- [26] J. Hauser, and J. Wawrzynek, "Garp: A MIPS Processor with a Reconfigurable Coprocessor," *Proc. IEEE Symp. Field-Programmable Custom Computing Machines*, 1997.
- [27] Andr' e DeHon: Reconfigurable Architectures for General-Purpose Computing, *A. I. Technical Report No. 1586, Massachusetts Institute of Technology*. 1996
- [28] Michael I. Gordon, William Thi es, 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
- [29] 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
- [30] 任軒, "在可動態重組系統中利用預先抓取及合併的技術降低重組負擔/Reconfiguration Overhead Reduction Using Prefetch and Merge Techniques in Run-Time Reconfigurable System", 碩士論文, 2003
- [31] 陳美璇, "多 FPGA 之可重組式系統中資料流處理的資料流程圖切割方法設計/Data Flow Graph Partitioning for Stream Processing in Multi-FPGA Reconfigurable System", 碩士論文, 2003
- [32] 曾啟華, "在應用執行過程中可重新設定組態系統中重新設定時間的減少與隱藏/Reconfiguration Overhead Reduction and Hiding of Run-Time Reconfigurable System", 碩士論文, 2003
- [33] 李孟道, "在可重組系統中使用動態重組排程方式增加 3D 顯像程式的效能/Using a run-time Reconfiguration Scheduling to improve performance for 3D Rendering on Reconfigurable System", 碩士論文, 2004
- [34] 盧惠真, "針對一個固定集合的應用設計一個有效減少線路面積的可重組式硬體/Designing a Wiring Area-efficient Reconfigurable Hardware for Multimedia Applications", 碩士論文, 2005
- [35] 陳治璋, "針對安全性嵌入式系統之彈性管線化設計與實做/Design and Implementation of a Flexible Pipeline for Secure Embedded Systems", 碩士論文, 2005
- [36] Kuen-Cheng Chiang, Meng-Tao Lee, Jean Jyh-Jiun Shann, Chung-Ping Chung, "Run-time Reconfiguration Scheduling of 3D-Rendering on a Reconfigurable System", 3rd International Conference on Computing, Communications and Control Technologies (CCCT '05), pp.30-25, July 24-27, 2005 - Austin, Texas, USA.