積體電路工業的新挑戰-GSI 系統晶片

吳全臨
黃印璽　蔡長委　江尚旻　饒書鈺　許鈺鼎
國立交通大學資訊工程研究所
新竹市大學路 1001 號
clwu@csie.nctu.edu.tw

摘要

當微電子技術可以將單晶片的電晶體數目推到十億個左右時(即 Gigascale Integration, GSI)，其在設計上及應用上就必須有革命性的作法。傳統的單一處理機(Uniprocessor)架構，可能變為不合適；同時，在耗電、速度、及連接方面也必需重新檢討。再者其應用的方法也值得再思考，例如，我們可以把整個系統搬到晶片上來，而形成所謂的系統晶片(System-on-a-Chip)。一些傳統的英特爾(Intel)處理器的設計原則，可能要面臨挑戰；而平行處理架構也許要大行其道。很顯然地，我們面臨一個架構斷層(Architecture Gap)。本文討論 GSI 微處理機架構及分析其功能。

1．簡介

由於面臨數十億個電晶體以上的單晶片的問世，傳統的單一處理機架構，可能變得不合適;而在耗電、速度及連接方面也需重新檢討。另外，應用在多媒體的需求量越來越多，而可攜帶式的電子產品與行動計算的新電器也已經十分普及，例如協助個人處理資料與生活資訊的個人數位助理(Personal Digital Assistant, PDA)。再者，在新趨勢應用需求所形成的系統晶片(System-on-a-Chip, SOC)蓬勃發展下，如何重新訂定微處理機架構以因應 SOC 的需求，也是當前重要的考量。因此，我們面臨所謂的 Gigascale Integration(GSI)的挑戰。

雖然相關論談和研究中指出，GSI 將受限於理論上(熱力學、量子力學、電磁學等)和實際應用上的限制，其中包括 1)fundamental, 2)material, 3)device, 4)circuit and 5)system[10]。此外，在新的競爭趨勢下使得電子資訊產業面臨一個危機:就是仍然不能掌握成熟的方法來做 GSI 晶片的設計，因此在傳統微處理機和新興 SOC 設計需求中間存在一個令我們不得不重視的問題—架構斷層(Architecture Gap)。

影響現存的計算機架構的因素有很多，以現在的半導體技術演進來說，許多專家預測在 2007 年以前，將可以在一個晶片上製作超過十億個以上的電晶體，當然，新的科技將會帶來許多新的問題，在通訊網路應用的需求日益增加之下，人們更希望能夠提升通訊的品質，希望能夠使聲音影像的傳輸大為增加，所以多媒體的應用方面將會是影響現在計算機架構走向的重要因素。

2．設計考量

當 VLSI 進展到數十億個電晶體在單一晶片上(GSI)時，許多技術上的限制對 GSI 的效能有很大的影響，如晶片面積、耗電、連接、時脈週期、匯流排等等。而在微處理機結構盡最大努力下，必須面臨這些限制做重新設計與考量，才能提供最高的效能和應用。

十億個電晶體(billion-transistors)世代的挑戰，最主要的挑戰是如何配置這十億個電晶體，在一個架構的那個部份須要多少電晶體，或是如何以效的利用這十億個電晶體並整合成一個系統在一個晶片上，以求達到所需的功能。以下是未來在製作十億個電晶體在單一晶片時會遇到的最重大影響因素。

一、耗電問題(Power Consumption)：我們知道，越高的頻率，將導致越高的耗電量。為因應現在可攜式電腦與行動計算上面的使用，有越來越多在低耗電量問題上的研究，但是降低耗電量可能要降低處理機的頻

率，是個兩難的問題，所以低耗電量將是未來科技演進後產生的重要問題。

二、記憶體延遲時間問題(Memory Latency)：近來相對於處理器以每年 60%的處理速度在成長，記憶體的處理速度僅僅以7%的小幅度在增加。這將使得處理器與記憶體的距離越拉越遠。為了把這個距離補上，較好的記憶體階層架構例如緩衝器(write buffer)、快取記憶體(caches)、主記憶體系統(memory system)、磁碟輸入與輸出子系統(disk I/O subsystem)與互相連線(interconnection)的部份越顯重要。

三、利用平行度 (Exploiting Parallelism)：想要達到高的表現，除了需要更高的技術，也需要更好的計算機架構。要獲得更高的平行度以達到較好的表現可以從硬體與軟體兩方面來著手。而以硬體的角度以及根據應用需求而言，過去需要編譯器(compiler)的支援轉變為將編譯器的設計包含在計算機架構的設計當中，以求達到高度的指令層平行度(instruction-level parallelism)，但因為如此，可能造成邏輯設計上更為繁多，而繞線的數目也變的更為複雜，而使得延遲的線路(delay path)更長。

四、設計複雜度(Design Complexity)：上等的處理機設計的複雜度、設計時間以及驗證，將會佔掉發展費用的 40%，所以如果能夠簡化晶片中各個模組的交互作用與減少有交互作用的模組數量將會簡化設計複雜度，帶來更大的利益。

五、市場規模(Scale of Market)：如果市場的規模越大，則要生產的產品必定越多，所以設計出來的產品必須跟現在市面上的現有產品有很明顯的優勢，這也會影響到我們的設計。

根據以上的影響因素，我們依(1)應用需求，(2)耗電，(3)接線，(4)功能，(5)時脈週期，(6)匯流排，(7)電晶體個數，(8)微架構內容等設計多維空間(Design Space)，將目前微處理機架構分類討論，比較並分析其優點及極限，提供有系統的比較數據以發展未來GSI架構。

3. 分析比較現行架構

在 GSI 蓬勃發展的趨勢下，從超純量(Superscalar)、多引線(Multithreading)、多純量(Multiscaling)的微架構方法，以及英特爾發展的 IA-64 架構等，能將現有表現提供更好的效能。

多線的處理機架構中，藉著多重引線提供額外的平行度，將工作執行分為(1)將工作分割成引線並將引線放在記憶體中(稱為虛擬引線);(2)選擇一個引線並且將其交給擁有一些暫存器集合來執行(稱為實體引線)。延伸成為同時多線(Simultaneous Multithread)的架構有 Compaq 的 Alpha 處理機、Sun 的 Java 處理機等。

多純量架構是以超純量架構為基礎再利用其指令間的平行度來設計完成的。為了將動態指令序列分割成多個部份來執行而不是以一個部份來執行，多純量架構採用多個超純量處理機在同一時間執行動態指令序列的不同部份，每個超純量處理機可以在所屬的部份指令中搜尋並執行其獨立的指令，和單一處理機比較起來，明顯的增加系統的效能。

在實際的應用上如平行多媒體處理機(Parallel Media Processor)，在單一晶片上整合多個平行運算單元、指令執行以及記憶體架構[7]。而單一指令流多重資料流像素處理機 (SIMD Pixel Processor) 則是一種將平面焦點作影像處理的嵌入式架構系統，在與處理機之間的存取使用一種區域陣列(area-array) I/O 的方式，可以在時脈週期和耗電兩方面的平衡考量下能提供更高的效能[2]。在日益增多的嵌入式系統和行動裝置的需求下，同時多線架構(SMT)的處理機可以提供比單一引線的處理機較少的耗電[16]等等。

4.未來 GSI 架構

未來 GSI 架構針對各種不同的應用，整合處理機與系統的架構，主要可以分為四個主流，將在以下四個小節分別介紹。

4.1. 超大單一處理機系統晶片 (Large Uniprocessor on a Chip)

4.1.1. 超推測性處理機 (superspeculative processors)

超推測性處理機透過積極的控制(aggressive control)和資料的推測(data speculation)以求達到高度的平行度。保守的強烈相依模型(strong-dependence model)是介由使用資料流圖(data-flow graph)與控制流圖(control-flow graph)，透過指出內

部指令相依與真實資料相依，用來避免違反連續性的限制(serialization constraints)。弱相依模型(weak-dependency model)是為了達到超推測性處理機的積極平行度而提出的。包含以下的兩點：

一、如果推測可能有相依的情形發生，則不一定要避免掉這種情形的發生。

二、如果推測可能有相依的情形發生而此時指令已經在執行，則可以在影響整個機器的狀態之前，用重置(recovery)的方式來復原系統。

這種模型的好處在於我們不必先完全了解程式的語意，就可以遇先執行。

超流(Superflow)架構是另外以種接近超推測性微架構的設計方式，利用很大範圍的推測技術，例如追蹤快取記憶體(trace cache)和分支預測器(branch predictor)，來增進整體的流量，包括指令流，暫存器資料流與記憶體資料流。

4.1.2. 路線處理機(Trace Processor)

路線處理機是複製超純量管線(superscalar pipeline)並使用控制與資料階層處理觀念來建立相連的處理元素集。一個高階控制單元將指令流分割成小的路線(trace)，並用特別構造的快取記憶體，也就是線路快取記憶體(trace cache)將這些線路儲存，而處理機一次攫取並解碼一個線路，把一個線路當成一個單元。也就是說，一個線路可以視為一個攫取與執行的基本單元。這種架構必須要倚重預測器(predictor)，因為我們從原本指要定義比較簡單的指令控制流預測器(control flow predictor)，而現在得處理比較複雜的路線流控制器(trace flow predictor)。這可以達到不用更改原本我們輸入的程式，就可以達到比較快的執行速度。

4.2. 單一晶片多處理機系統(Multiprocessor on a Chip)

4.2.1. 晶片多處理機(Chip multiprocessor, CMP)

想要使用單一引線(single thread)的控制以求獲得平行度將會在許多應用程式上產生限制，而且也很耗研發的時間與費用，非常的不經濟。因此平行的使用多個處理機核心來執行多引線(multiple thread)時，晶片多處理機會使用相關簡單的單引線處理機核心，而每一個簡單的單引線處理機核心的平行度也是相當適當的設計。為保證硬體設計又快又簡單，在設計上我們利用軟體來達到這種目的。因為晶片多處理機是真的分成許多各別的處理機核心，而這種設計方式將可大幅降低內部繞線所產生的時間延遲(interconnection delay)，這在十億個電晶體以上的設計，將會是十分嚴重的課題。也因為這樣的設計，處理機核心的電路也會簡化許多，所以如果想達到執行頻率的最佳化也容易的多。而晶片多處理機的最大缺點為如果我們所執行的目標程式是只有一個單一的引線，無法分成多引線(multithread)的控制時，單一而且較複雜的架構就會比晶片多處理機快很多。

4.2.2. 原質處理機(Raw processor, RAW)

這種原質處理機的基本構想為保持硬體上的簡單設計，而針對各種應用程式的需要來設計編譯器(compiler)，以編譯器的編譯硬體，使得得到的硬體可以適合客戶的須求。原質處理機是建立在一些簡易的處理機架構之上，並複製這些架構，而每一個處理機架構都擁有自己的指令流。而設計的主要概念是盡量能夠在單一的晶片上置入越多的處理機越好，也就是說越是簡單的架構，能夠置入的處理機數量越多，而不是為了一些特殊的邏輯架構來製作一些硬體，這些架構如暫存器改名單元(register renaming)，指令排程單元(instruction scheduling)，或是動態指令發出單元(dynamic instruction-issue)。而這些空下來的空間將可以加大記憶體或是運算邏輯(computational logic)，也可以達到較高的執行頻率與減低驗證的複雜度。而原質處理機是用軟體來製作的，這些分散的架構以可程式化，而且有條理的整合線路來作相互連結，當然，還包括可切換的內部連線控制。原質處理機很適合傳統的科學應用計算，這種計算的有許多的資料流要處理。

4.3. 系統晶片(system-on-a-chip, SOC)

SOC 即在單一晶片上結合整個電腦系統(computer system) 如處理機(processor)動態隨機存取記憶體(dynamic random access memory, DRAM)和輸入輸出子系統(I/O subsystem)，整合了記憶體(包括 DRAM、

flash)，類比核心模組(analog cores)，各樣不同功能的 IP Blocks 及最上層的處理器(processor cores)及邏輯運算單元(logic unit)等等。

製程技術的進步，已經進入到了深次微米(Deep Sub-micron)的製程，容許一顆晶片上可負載更多的 gate 及電晶體，且可使得整個設備具有可攜性。SOC 的實現，讓原本只有硬體架構的系統可以融入更有彈性的軟體設計;再者，將大部份的核心(core)放在一塊，可以使得整體的頻寬增大，提高系統的效能，此意謂著各個核心或 IP Block 之間的訊號傳遞不再需要經過複雜及大量的 I/O buffer 及 system board 線路。

4.3.1. 智慧動態隨機存取記憶體(Intelligent RAM, IRAM)

隨著科技的演進，處理機與記憶體的加塊速度卻不層比例，使得處理機比記憶體快的多，產生了距離，這也讓記憶體延遲時間與低記憶體頻寬的問題越顯嚴重。智慧動態隨機存取記憶體的基本精神使是把動態存取記憶體(DRAM)作到單一晶片上，以改善這個影響處理機表現的因素，而不是使用靜態存取記憶體(SRAM)來彌補處理機與記憶體之間的斷層。主要著眼於在同樣的晶片面積的前提下，動態隨機存取記憶體能夠比快取記憶體置入多 30 至 50 倍的單位數量。這種做法的主要優點有三：

一、將記憶體置於單一晶片之上將可以使用很寬的介面來提高記憶體的頻寬與降低記憶體的延遲時間，同時也省去了接腳(pad)與排線(bus)的時間延遲。

二、因為從處理機核心到外部電路的存取次數減少了，耗電量也相對的減少。

三、智慧動態隨機存取記憶體可有擁有多許多倍的有效率介面(stream-lined interface)，因為這些連續的線是直接連到處理機核心來擴大基本輸入輸出的頻寬，而不是透過有限制的接腳(pins)來和記憶體溝通。然而，這種設計的方式將使的當記憶體大小不固定時，設計的結果也要跟著改變。

4.3.2. 在多媒體通訊網路的應用(Application on multimedia communication network)

由於目前 SOC 在影音多媒體通訊網路方面的設計應用很多，而在通訊網路方面的重要特色已經發展成為點對點連續資料流的即時處理(real-time processing)，這與傳統的應用程式多用一般資料的向量處理是迥然不同的，讓我們將這些特色明細如下：

一、即時回應(Real-time response)：多媒體通訊需要即時回應，有的時後為了能夠使輸出的資料更為流暢，不惜丟掉一些來不及處理的資料，特別是在即時影像傳輸的方面。

二、連續媒體資料型態(Continuous-media data type)：多媒體應用程式的輸入資料多由一群取樣類比訊號(sampling analog signal)的資料流所構成，這有別於傳統應用程式處理非連續的資料。

三、顯明精細資料平行度(Significant fine-grained data parallelism)：在以往的計算機架構上，多針對提高指令層的平行度來做設計，以應傳統的程式需要，著眼於高度的超存量指令發出(SuperScaler instruction issue)，指令預測(instruction prediction)或是預先攫取(instruction prefetch)方面的硬體設計，非依序執行(out of order execution)，暫存器改名(register renaming)等。然而，在訊號與圖形處理方面卻是需要處理十分大量相同資料型態的計算機硬體，所以如果使用 SIMD 的方式來作硬體的設計能夠得到比較大的資料平行度，也就是更寬的資料路徑。

四、支援引線平行度(Parallelism for supporting threads)：許多多媒體應用程式是由一些具有時間限制的引線所組成的。這些引線將會帶來相當高的頻率，很深的管線(deep pipeline)和硬體方面多引線(multithread)處理機和對稱(symmetric)多處理機。

五、高指令區域性(High instruction locality)以致小迴圈(small loops)：訊號與圖形的處理應用方面，小迴圈的執行佔了主要的執行時間，所以如果能使迴圈的執行速度增快將大幅增加整體的表現。

六、高記憶體頻寬(High memory bandwidth)：許多的媒體應用程式有非常大的資料量或是工作集，這就是說在系統需要很高的記憶體頻寬前提下會遭受長時間的記憶體延遲，但是快取記憶體對此現象並不會帶來利益，資料的預先攫取或是資料往處理機核心傳遞將會是更有效率的影響因素。

七、擴充資料的重組(Extensive data reorganization)：在 SIMD 的架構上，資料的重新組織是必要的，因為這可以有效的提高純量架構的速度。另外要再提的是，執行這些應用程式所須要的處理機比較不算是一

般用途的處理機，而且特別為訊號處理而設計的。透過簡單架構的支援，新技術將整合處理動態資料到一般性處理機的能力，使得在一般性處理機在多媒體密集處理的能力上帶來明顯的增進。

此外，在 SOC 的架構上，探討其整顆晶片的耗電大致上是以時派功率損耗(clock power consumption)為主要耗電來源。SOC 即是包含多個 IP module 所組成，而各個不同的 IP 其所需要的時脈也不盡相同。在這不同的架構組合之下，需要不同的時脈驅動方式(clock driving method)：在此提出兩種方式：(1)單一驅動(Single Driver Scheme) (2) 分散式驅動(Distributed Driver Scheme)[4]。

SOC 內各個 IP 模組的連線在未來是一個非常重要的議題，就以往的 IC 設計而言，影響延遲的主因已經由邏輯閘延遲(gate delay)變成現在的線路延遲(wire delay)，更何況在 SOC 架構之下，連線(interconnection)更是重要。在此有一新的 Bus 架構更是適用於 SOC 連線，即分割匯流排(Segmented Bus)架構[12]。Segmented Bus 的優點是在大部份的時間中，只有部份的匯流排是正在運作的，所以可以節省耗電，同時每個叢集是獨立的，所以可以增加產能(throughput)。

因為 SOC 的盛行，使得設計趨向於以巨集為基礎(macro-based)方法。這種方法使得在做邏輯驗證及邏輯運算更加有利。在此，IBM 提出了三個對連接核心、函式庫的匯流排架構，即：(1)Processor Local Bus (PLB); (2)On-Chip Peripheral Bus (OPB); (3)Device Control Register (DCR) Bus[13]。

在 SOC 製程中，由於是混合各式各樣的 IP blocks 或 cores ,logic unit 等等，所以在 SOC 的製作中，必須有一個經過混合分析，都適合每一個 IP 的製程。例如，DRAM 的製程不一定符合金屬層的製程。因此尋求一個混合式且都符合不同技術的製程平台是發展 SOC 架構的一個重要議題。[5]

上面的分類是針對現今所存在的處理機及採用的架構所預測的未來 GSI 架構的方向。由上面的分類可以很清楚的看到未來一旦出現十億個電晶體的晶片時，晶片上可能的架構會是處理單一複雜工作的處理機(*如(一)*)，或是由多個相同的單一處理機所組成的一顆處理機，利用 Multi-threading 或 Multitasking 來達到高效能的平行處理(如(2))。或者是大家所熟知將 RAM、I/O sub system、及 processor 放在同一個晶片上的 SOC(*如(3)*)。

4.4. Reconfigurable System

4.4.1. GARP

在 GSI 的廣大範疇之中，我們選出一個較具潛力且有強大功能的系統來介紹，也就是 Reconfigurable system 中的 GARP 系統.

現在大多數的系統都會應用現成的 FPGA 來做為特定應用程式的晶片來加速成品的效率；而一般而言這種架構之下的 Bus System 大多都是依靠 I/O bus 來溝通彼此之間的資料. 然而雖然使用 FPGA 有其好處然而有時它的低頻寬卻成為了有能力執行高速傳速之系統的瓶頸. 鑑於此，Garp 的出現正是可以解決這樣的問題使得系統執行起來更加地流暢. 實際上的 Garp System 是利用一個和處理器更為密切的 reconfigurable hardware 來解決資料流頻寬上的不足，當然，這個 reconfigurable hardware 當然是以特定應用程式運算為主的晶片.

GARP 的架構大致上可分為兩個部份，一為主要處理器，這裡的處理器是以 MIPS 處理器為主而另一部份則是搭配一顆 Reconfigurable Hardware，此 Reconfigurable hardware 主要是用來加速一般運算使用為目的；在此，要發展此架構有幾個重點是我們必須要注意的：

一. 花在傳輸處理器暫存器和 reconfigurable hardware 的多餘時間是必須且可接受的，畢竟 reconfigurable hardware 是我們所注重的，且發生這種浪費時間的情況亦不多.

二. 如同上面所說的，reconfigurable hardware 才是我們的重點，鑑於此，我們必須允許它和記憶體做直接的溝通，就好像是 DMA 一樣，並不需要處理器多餘的關懷，否則處理器將成為一個嚴重的瓶頸.

三. 為了配合程式運算，reconfigurable time 必須短且迅速.

提供 reconfigurable hardware 和記憶體系統之間較寬的匯流排不僅能提供快的資料傳輸，在此，亦可以使 reconfiguration 的時間更有效率；Garp 有提供能處理 reconfigurable hardware 的特定指令，處理器可依需求載入所需的狀態至 reconfigurable hardware 以達到特定的運算結構.

一般而言，reconfigurable hardware 運算時可分為下列四個步聚

一．載入 configuration state

二．利用 coprocessor 指令拷貝初始狀庇暫存器至 reconfigurable hardware

三．開始執行，如果 configurable hardware 正在執行其它運算則必須等候

四．將結果拷貝回處理器

發展一個新的架構，編譯器是絕對少不了的重點，一個好的編譯器可以把一個具有高效能的系統發揮得淋漓盡致．Garp 的編譯程式是以標準的 ANSI C 為主，所以編譯器選擇了既有的 SUIF C 編譯器來將 GARP 系統效能發揮到極致．一個新的架構當然會有新的問題產生，針對 GARP 此架構，編譯器首要面臨的挑戰即是如何利用 GARP 中的 reconfigurable array 來加速運算．在此有兩個非常重要的問題：其一為如何將程式放到 reconfigurable array 內執行，且合乎陣列的規格要求；其二為如何開發程式的平行度，使得管線化工作更加地有效率．

GARP 遇到的問題同樣也發生在 VLIW 這個架構身上，所幸 VLIW 已發展出一套較有系統的編譯流程，也順利地解決了程式大小及平行度的問題．

要如何使得 GARP 更有效率？有下列普遍使用的方法可以使得 GARP 在編譯或執行程式上更加順暢．首先舉出的是“ Speculative loads ”，和某些架構一樣，GARP 為了使程式本身更具有平行度，加了一些硬體或軟體演算法對程式本身做預測，當然這些預測本身後面也是有相當的風險，不過這些軟硬體足以彌補錯誤發生時所造成的傷害；再來就是大家最常見的管線化，故名思義，管線化也是為了開發更多的程式平行度，使程式執行的速率能以倍數的增加；最後一個方法則是利用硬體來模擬存取記憶體系統，這種做法使得原本得用 reconfigurable array 來做的事轉稼到這特別的硬體身上，這種做法讓 reconfigurable array 不需花時間去控制存取記憶體系統，且可用來做其它的事，可以說是事半功倍．

目前為止，reconfigurable hardware 大多是一些理論和數據組合出來的架構，然而要將它製成可以使用的晶片還是有一段路要走；再者，以 Moore’s Law 觀點來看，Silicon size 的密度和電晶體的數目逐漸增加中，加上製程的進步，Gigascale 的晶片系統已是不可擋的趨勢了，趁著這個優勢，發展出具有 Reconfigurable 的晶片系統應該是指日可待了．

4.4.2. RAW

現今 CPU 的架構中，像超純量 (superscalars) 因為本身是利用硬體來做到 dynamic resolving instruction dependency 而進一步達到 parallelism，使得它已接近 performance 及 complexity 的極限。不但耗費硬體資源，和 resolving 本身也有一定的難度，因此這個系統設計上的瓶頸，要讓超純量的平行度再提升是一件很難的工作。

而 RAW 這個系統是一個強調簡單及 wire-efficient 的架構。它在硬體方面能夠隨著 VLSI gate 密度的成長而保持成長，不會受到系統設計上的限制。除此之外，RAW 藉著 compiler 的 static schedule 來達到優質的 parallelism performance。

RAW chip 由眾多相同的 tiles 所組成。每一個 tile 中包含了像 RISC 處理器中有的 ALU、Registers Bank、Data 和 Instruction memory 還有一小部份的 configurable logic 用來調整 tile 內部的溝通。而 tile 與 tile 之間是用一個稱為 Switch Logic 的電路來做連接，它負責連接本身 tile 與東西南北的 tile。Switch Logic 可以經由 compiler 的產生的指令來做 static 的 program。另外 RAW 也提供以 flow-control 的方式針對 compiler 無法決定的正確 static schedule 的情況作動態的調整 switch Logic。

Figure 1.是 RAW 及 Superscalars 的比較圖。

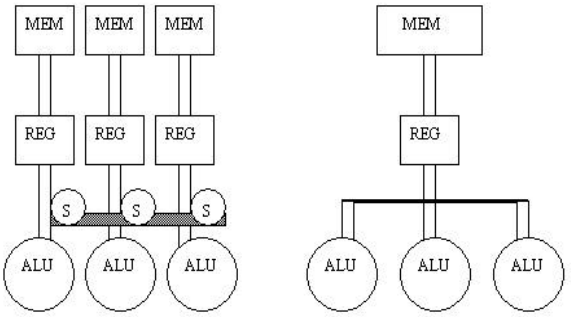

Figure 1. RAW vs Superscalar

RAW 的特色如下：

(1)Simple Replicated Tiles：RAW 由互相連結的 tiles 所構成。RAW focus 在保持每一個 tile 越小越好，但是一個 RAW

chip 上 tile 數量越多越好。可以藉由這種優點來增加平行度及加快時脈速度。

(2)Programmable Interconnect : Switch Logic 連接 tile 與 tile，這樣可以提供 tile 之間高頻寬以及低延遲通訊。由 Figure 2. 可看到 physical wires 是 critical resource。Switch multiplexes 兩種不同的 logic network，分別是 static 及 dynamic network。

Static network 由 compiler 產生的指令控制，它將 switch input 連接到正確的 output 位置。因為 communication patterns 是在 compile time 就完成，因此不須要額外的 header interpretation，也就是沒有多餘的 routing overhead。若是有任何須要 communication 的動作卻不是由 compiler route 及 schedule 的，則使用 dynamic network。經由 static 及 dynamic 的相互合作，可增加 parallelism。

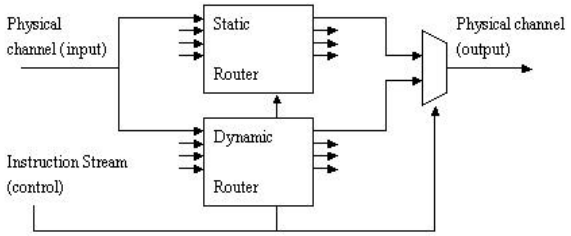

Figure 2. Structure of RAW

(3)Compiler System : RAW 的 compiler 目地是將以 high-level 語言所寫的 single-threaded (sequential)或 multi-threaded (parallel) program map 至 RAW hardware。RAW 清楚的 parallel model 讓 compiler 能夠直接的找出程式的 parallelism 並且描述它。

此外，RAW 相較於其它架構有很多的優點：Tiles 間的溝通是藉由 interconnection 並且是 register level，因此 clock frequency 會較使用 global buses 的 chip 高。

(1)在 1 clock cycle 內所有 signal 所經過的距離不會超過一個 tile 的寬度。

(2)Memory distributed across the tiles 將降低 memory bandwidth 的瓶頸並且在每一個 tile 中提供相當低的 latency 至 memory module。

(3)大量的 distributed registers 將會減少 register renaming 的機率。

(4)相較於 off-chip communication and DRAM latency，RAW 提供了 Higher internal switching speed。

(5)一個簡單的架構有利於快速進入市場。

(6)RAW 架構可以藉由調整 FPGA-like configurable logic 成為 multi-granular 來達到 FPGA computer 能夠做到的平行度。

(7)RAW 相較於 processor in memory 架構使用很多小且低延遲的記憶體。

簡單的一句話就是 RAW 是一個由相同的 processing tiles 所構成的 software-exposed VLSI 架構。Compiler system 能夠在 tile 間 static 的 schedule instruction-level parallelism。

4.4.3. MorphoSys Dynamically Reprogrammable Architecture(動態可重複程式化架構:適應計算的新趨勢)

(一)適應化晶片:這個晶片的目的是使的我們的電腦能夠非常的配合我們所需要的應用，只要藉著動態的規劃這個晶片，就可以讓我們的應用方面達到既快又有效率的目的。

這個架構的提出,包含的幾個新的觀念:

1.動態可重複程式化架構必須在執行的時候因應資料狀態的改變來作重複的程式化。

2.晶片上多個設定的路徑。

3.動態可重複程式化的記憶體陣列。

4.在 CAD 工具方面，也有所改變要有特殊的放置(placement)與平面圖(floorplan)的運算方法，交互連線的部分也是很重要的問題，因為連線有固定的型態。

5.在執行時的記憶體管理中心的設定，與系統表現的量測與最佳化。

6.記憶體與執行資料流的動態再設定。

這個架構的產生，也產生對舊有架構的一些影響:1.系統資源的分配變的比較容易。2.要讓晶片成為特殊用途所用時,要調整比較簡單。3.在執行時,改變資料相依的關係與功能以適應外部的環境，使的硬體可以成為各種不同的應用的專屬硬體。

動態可重複程式化架構的優勢在於它可以適應的應用方面廣泛，而且有較好表現，我們先拿一般用途的運算機器做比較，雖然範圍更大，但是表現卻很差。而 ASIC(Application-specific embedded systems)是為某一特別的應用而製作，當然表現很好，但是他的應用範圍就非常狹窄，

也可以說是比較不經濟，所以動態可重複程式化的新興架構成的最好的選擇，也讓我們的設計變得更容易、更方便。

這個架構的名字叫 MorphoSys，可以先粗略的分成三個部分:可重複設置的處理機陣列(Reconfigurable processor array)，高頻寬資料介面(High-bandwidth data interface)，與主流處理機(Mainstream processor)，如 Figure 3.所示。

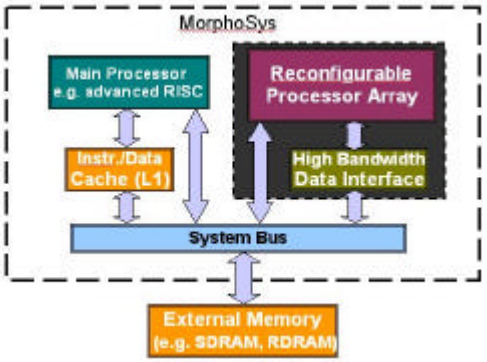

Figure 3. MorphoSys

Figure 3: MorphoSys

談到 MorphoSys 的運作方式，先了解他的資料來源，從外部的 Main Memory Bank 經由一個在晶片內的 DMA 控制器來控制他流到一個 2blocks、16sets 的 context memory，而這個 DMA controller 則是由晶片內的一個 TINY_RISC 的核心處理機(core processor)來控制。而 TINY_RISC 如果要給可重複設置的單元陣列(Reconfigurable Cell(RC) Array)資料的話，則要藉由控制 DMA controller，並將資料給高頻寬的資料介面(High-bandwidth data interface)上的緩衝器(Frame Buffer)，藉由緩衝器將資料傳到可重複設置的單元陣列，或是要求 context memory 將資料傳給可重複設置的單元陣列。如 Figure 4.所示。

再來我們會談到 MorphoSys 的架構主要分成三個部分:可重複設置的陣列處理機(Reconfigurable array processor):包含兩個部分 context memory

與可重複設置的單元陣列(Reconfigurable Cell(RC) Array)。

1. 高頻寬的資料介面(High bandwidth data interface):包含兩個部分直接記憶體讀取單元(DMA controller (Direct Memory Access, controller))與緩衝器(Frame Buffer)。

2. 一般用途處理機(General purpose processor):包含一個小的簡化指令集核心處理機(TinyRISC Core Processor)與資料快取記憶體(Data Cache)。

這三個單元互相都有訊號的傳輸，但是在與外部的單元部分，則是靠資料快取記憶體與高頻寬的資料介面的緩衝器來與外部的主記憶體溝通。

我們先看可重複設置的陣列處理機(Reconfigurable array processor)的部分:

1. 可重複設置的單元陣列(RC array，Reconfigurable Cell Array): 8*8 個 RC array,也可以看成是 4 個 4*4 的 RCS，我們也稱他是 4 個 quads。
2. RC array 與 Context Memory 的關係: context memory 的資料可藉由廣播(Broadcast)的方式傳到每個 Cell，把資料傳到某一列(row)或某一行(column)，再由某一列廣播到其他行，或是由某一行廣播到其他列，所以我們把 Context memory 分成兩個區塊，一個是行區塊(column blocks)一個是列區塊(row blocks)，共有每個區塊都有八組的 RC，也就是說每個區塊控制一個列或是一個行，如此可以達到單一指令多資料(Signal Instruction Multiple Data, SIMD)的目的。
3. 4*4 RC array quad 的連線問題: 第一層是每個 RC 都要連到他的東西南北四個 RC，第二層是每個 column 或是每個 row 的 RC 都需互相連線。

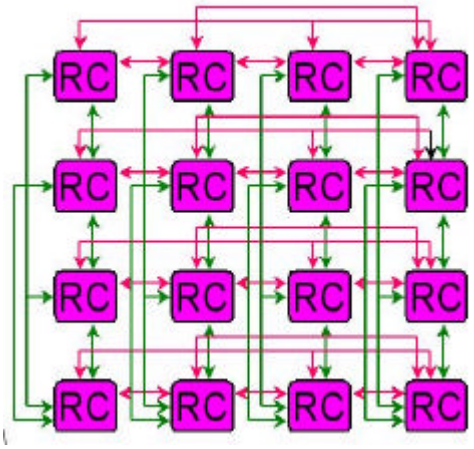

Figure 4. Reconfigurable Cell

4. 整個 RC Array 就可以把四個 4*4 的 RC array quads 連起來，他們的連線方法是讓每個行都用垂直表現巷道(Vertical express lanes)，每個列都用水平表現巷道(Horizontal express lanes)，與三態緩衝器(tri-state buffer)來相連。

5. 每個 RC 的內部構造為兩個來源(source)的運算，source 可由資料匯流排(Data Bus)、鄰居 RC(Neighbor RCs)、暫存器(register file)這三個來源而來，一個簡單的加乘法器(ALU,Multiple)，移位器(shift)與暫存器(register file)。

Figure 5. RC interconnection

我們再看一般用途處理機(General purpose processor)，他很單純，就是一個小的簡化指令集處

理機(TinyRISC)，共切成 4 層:擷取層(Fetch Stage)、解碼層(Decode Stage)，執行層(Execute Stage)，寫回層(Write-Back Stage)。

最後談到高頻寬的資料介面(High bandwidth data interface)。緩衝器(Frame Buffer):分成兩個集合,每個集合又分成兩個堆積(bank).Bank A 提供資料來源給 RC 的 source A，Bank B 提供資料來源給 RC 的 source B。直接記憶體讀取單元(DMA controller (Direct Memory Access, controller)):由 TinyRISC 核心處理機控制，主記憶體與緩衝器的資料直接溝通，也傳資料給 context memory。

根據我們的應用，先定出 RC array function，經過 mView，會產生設定事件(configuration context)，再經過 mLoad 來產生事件庫(context library)。事件庫的資料再用 mSched 轉成 VHDL 或其他硬體描述的檔案，經過 Mulate 與 Morphosim 的模擬後，就可以寫到 MorphoSys 的晶片上了。而我們寫的應用程式，必須經過 MorphoSys “c” compiler 的編譯產生可執行檔來執行。

MSched 是 kernel Scheduler，他要讓事件的再讀取最小化，讓資料的重複使用率增高，而 RC 的運算與資料的移動最大化，經過這些的需求，可以產生最佳化的結果。

而這個架構的目標應用包含如下:影像處理:多媒體影像壓縮、自動辨識、影響分析與醫學診療影像。數位訊號處理:矩陣與陣列運算、FIR filters、Viterbi 編碼解碼、FFT、DCT 與小波轉換。資料加密:DES、IDEA、3-Way。

MorphoSys 設計方法在 CAD 工具上跟現有的要做一個區別，TinyRISC 和 DMAC 可以用標準單位設計(Standard Cell Design)，可是 RC Array 和緩衝器，context memory，資料快取記憶體與暫存器集，卻要用客戶設計(custom Design)，因為它的繞線是固定的(有一定的規則)，如果用一般 CAD 工具的繞線器，是繞不出來的，而且慢又沒效率，所以我們用全客戶方式的繞線，繞出來的結果小，而且又快。而電源訊號的方式是用 H-tree 來繞線，最後做出來的設計延遲時間為 10ns，面積為 1.5mm*1.0mm。

另外我們也可以合併多個 MorphoSys 而產生 meta-MorphoSys Architecture，這又是一個加強運算效能的方法了。

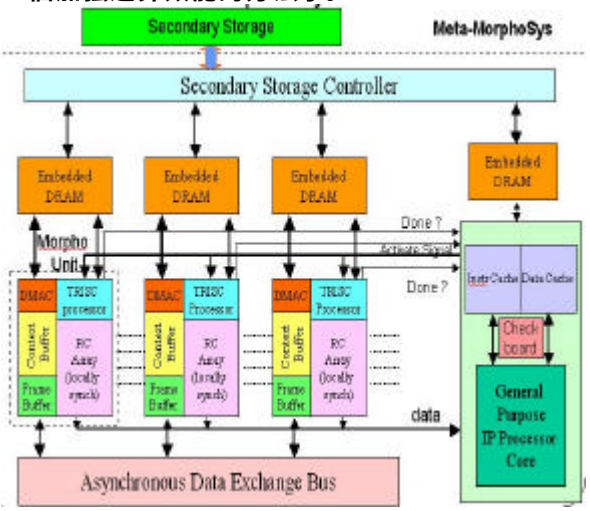

Figure 6. meta-MorphoSys Architecture

5. 結論

可以預見的,未來的 IC designer 所要面對的問題將與今日大大的不同，不論是在 chip 本身 core 的設計方式或是 core 與外部各個 IP block 之間的聯繫，都將因為硬體技術的發展，而面對前所未有的挑戰。不僅所要考慮的層面因為技術層次的不同而有了重大改變，單一晶片或是處理機的功能也將大大不同。設計者必須以全新的觀念與思維來面對;另一方面，隨著硬體技術突飛猛進，”電腦”或是所有的 電子設備(electrical device)都將對我們人類的生活產生越來越重要且不可忽視的影響，人類的生活型態也將因為電腦系統能夠達成的工作日益增加而產生重大的改變，一切都是因為 GSI 時代的來臨，我們得以倚賴功能強大的卻輕薄短小硬體設備來達成許多過去不敢想像的目標，不論是在多媒體、通訊網路或是桌上型設備，能夠為人們帶來的便利絕對是過去的我們無法觸及的世界。

不論是任何的系統晶片，或是 SOC 的整合，甚至 Reconfigurable System，目前都不算達到了真正成熟技術的境界，因為硬體技術目前並沒有辦法看到邊界，所謂的成熟技術也終將成為昨日黃花，而且世代交替的速度也會越來越快，但是相關需要討論的課題卻是越來越多，不僅是因為硬體本身衍生出來的問題，應用方面也將面對更多的可能性，身為現代資訊界的一員，我們所要面對的挑戰正是無限寬廣卻充滿挑戰性。每天都將會有新的技術或是應用問世，除了解決其所衍生出來的新問題之外，如何將這些科技運用在對於生活真正有益的方向才是最重要的課題。畢竟，科技的本身若是不能跟人類的生活作結合，都將只是沒有意義的符號和數字罷了。

參考文獻:

[1]Callahan,T.J.; Hauser, J.R.; Wawrzynek, J.," The Garp architecture and C compiler", Computer , Volume: 33 Issue: 4 , April 2000 Page(s):62-69

[2]Chai, S.M.; Gentile, A.; Wills, D.S.," Impact of power density limitation in gigascale integration for the SIMD pixel processor", Advanced Research in VLSI, 1999. Proceedings. 20th Anniversary Conference on, 1999, Page(s): 57 -71

[3]Chandra, A.; Chakrabarty, K.," System-on-a-chip test-data compression and decompression architectures based on Golomb codes ", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 20 Issue: 3 , March 2001 Page(s): 355 -368

[4]Chen, R.Y.; Vijaykrishnan, N.; Irwin, M.J., " Clock power issues in system-on-a-chip designs", VLSI '99. Proceedings. IEEE Computer Society Workshop On, 1999, Page(s): 48 -53

[5]Daeje Chin," Executing system on a chip: requirements for a successful SOC implementation", Electron Devices Meeting, 1998. IEDM '98 Technical Digest, International, 1998, Page(s): 3 -8

[6]Davis, J.A.; Venkatesan, R.; Kaloyeros, A.; Beylansky, M.; Souri, S.J.; Banerjee, K.; Saraswat, K.C.; Rahman, A.; Reif, R.; Meindl, J.D." Interconnect limits on gigascale integration (GSI) in the 21st century", Proceedings of the IEEE , Volume: 89 Issue: 3 , March 2001 Page(s): 305 -324

[7]Fritts, J.; Wu, Z.; Wolf, W.," Parallel media processors for the billion-transistor era", Parallel Processing, 1999. Proceedings. 1999 International Conference on, 1999, Page(s): 354 -362

[8]Hauser, J.R.; Wawrzynek, J." Garp: a MIPS processor with a reconfigurable coprocessor", Field-Programmable Custom Computing Machines, 1997. Proceedings., The 5th Annual IEEE Symposium on , 1997 Page(s): 12 -21

[9]Hounsell, B.I.; Arslan, T.," Programmable multiplierless digital filter array for embedded SoC applications", Electronics Letters , Volume: 37 Issue: 12 , 7 June 2001 Page(s): 735 -737

[10]Meindl, J.D." Gigascale integration: is the sky the limit?", IEEE Circuits and Devices Magazine , Volume: 12 Issue: 6 , Nov. 1996 Page(s): 19 -24, 32

[11]Meindl, J.D. " Interconnect limits on gigascale integration (GSI)", Plasma- and Process-Induced Damage, 2001 6th International Symposium on , 2001 Page(s): 1 -1

[12]Ohmi, T.; Sugawa, S.; Kotani, K.; Hirayama, M.; Morimoto, A. ," New paradigm of silicon technology", Proceedings of the IEEE , Volume: 89 Issue: 3 , March 2001 Page(s): 394 -412

[13]Remaklus, W.," On-chip bus structure for custom core logic designs ", Wescon/98, 1998, Page(s): 7 -14

[14]Seng, J.S.; Tullsen, D.M.; Cai, G.Z.N.," Power-sensitive multithreaded architecture", Computer Design, 2000. Proceedings. 2000 International Conference on, 2000, Page(s): 199 -206

[15]Shi Yuanyuan; Liu Jia; Liu Runsheng," Single-chip speech recognition system based on 8051 microcontroller core ", Consumer Electronics, IEEE Transactions on , Volume: 47 Issue: 1 , Feb. 2001 Page(s): 149 -153

[16]Singh, H.; Ming-Hau Lee; Guangming Lu; Kurdahi, F.J.; Bagherzadeh, N.; Chaves Filho, E.M. " MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications", Computers, IEEE Transactions on, 2000, Page(s): 465 -481

[17]Vahid, F.; Givargis, T.," Platform tuning for embedded systems design", Computer, Volume: 34 Issue: 3 , March 2001 Page(s): 112 -114

[18]Waingold, E.; Taylor, M.; Srikrishna, D.; Sarkar, V.; Lee, W.; Lee, V.; Kim, J.; Frank, M.; Finch, P.; Barua, R.; Babb, J.; Amarasinghe, S.; Agarwal,A." Baring it all to software: Raw machines", Computer, Page(s): 86 -93

[19]Weinhaudt, M.; Luk, W.," Memory access optimisation for reconfigurable systems", Computers and Digital Techniques, IEE Proceedings- , Volume: 148 Issue: 3 , May 2001 Page(s): 105 -112

[20]Yan Zhang; Wu Ye; Irwin, M.J.," An alternative architecture for on-chip global interconnect: segmented bus power modeling", Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on Volume: 2, 1998, Page(s): 1062 -1065 vol.2

[21]Zarkesh-Ha, P.; Meindl, J.D.," Optimum on-chip power distribution networks for gigascale integration (GSI)",Interconnect Technology Conference, 2001. Proceedings of the IEEE 2001 International , 2001 Page(s): 125 -127