

Computer Organization And Assembly Language

Final Project:

ARM & MIPS 之比較

資工三 B95902087 鄭聖學

動機

在上 ARM 架構的時候，才真正了解到什麼是處理器。長久以來一直以爲所謂的架構是一種實質上的東西 而經過上課後才知道原來更注重的是一種觀念，一種計算機架構處理的原理 也在上完此課後讓我想了解除了泛用的 ARM 以外，還有哪些熱門的處理器系統。

由於是以 ARM 開始認識，因此想了解有哪些是跟 ARM 可說是有類似的功能，而以不同的架構處理。

*底下擷取自 wikipedia:

ARM 架構

(過去稱作進階精簡指令集機器 (Advanced RISC Machine)，更早稱作 **Acorn RISC Machine**) 是一個 32 位元精簡指令集 (RISC) 中央處理器 (processor) 架構，其廣泛地使用在許多嵌入式系統 (embedded) 設計。由於節能的特點，ARM 處理器非常適用於行動通訊領域，符合其主要設計目標爲低耗電的特性。

嵌入式系統

一種完全嵌入受控器件內部爲特定應用設計的專用電腦系統。嵌入式系統通常執行的是帶有特定要求的預先定義的任務。由於嵌入式系統只針對一項特殊的任務，設計人員能夠對它進行優化，減小尺寸降低成本。

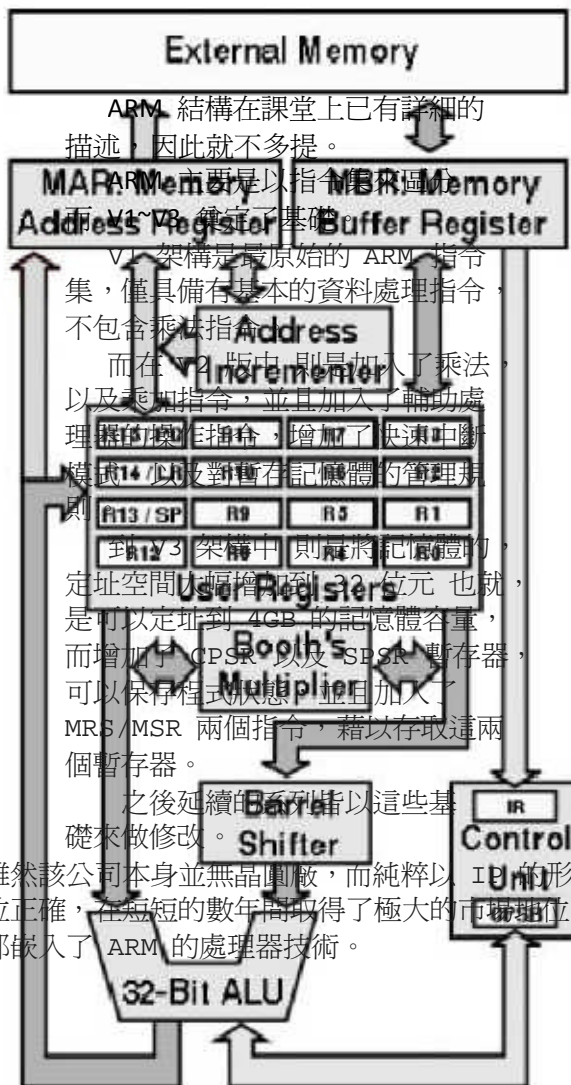
因此，這次的 Project 我以嵌入式系統的微處理器做爲主題，選取了 ARM 以及 MIPS 兩大處理器做比較。

市場主流

目前常見的嵌入式微處理器 (Embedded Micro-processor Unit, 簡稱 EMPU)有 Am186/88、386EX、SC-400、Power PC、Motorola 68000、MIPS、ARM、ARC 系列等。

而這次我們介紹的是市場主流的三大公司 ARM、MIPS 推行的嵌入式架構的處理原理，並在介紹完稍候做比較。

ARM



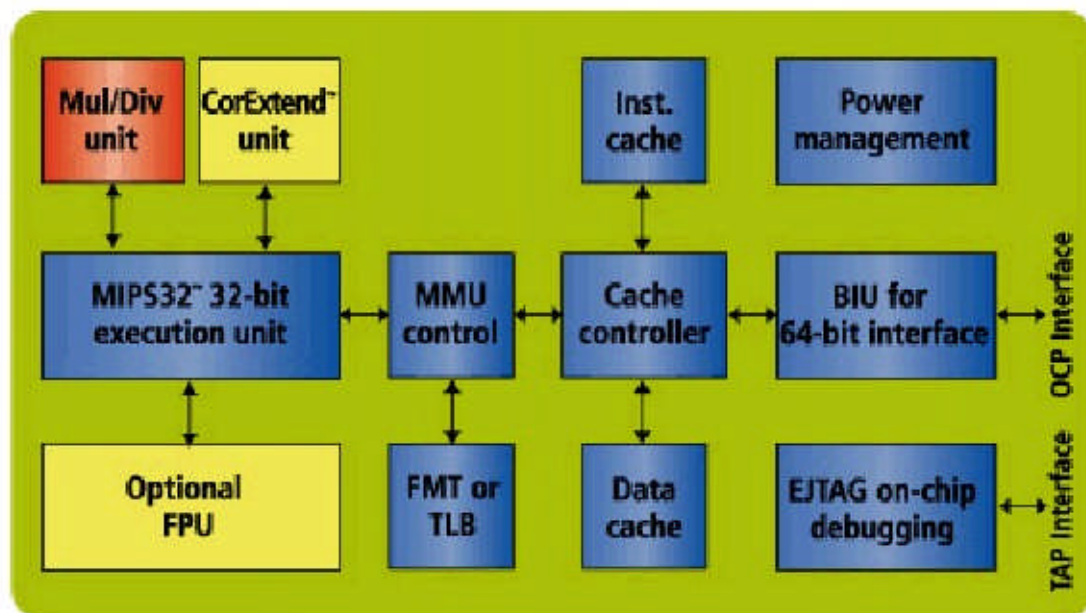
雖然該公司本身並無晶圓廠，而純粹以 IP 的形式出售處理器架構，由於定位正確，在短短的數年間取得了極大的市場地位，全世界絕大多數的手持式裝置都嵌入了 ARM 的處理器技術。



MIPS (Microprocessor without interlocked pipeline stages, 亦即不含 Interlock 機制的管線階層微處理器。)

MIPS 也是一家具有悠久歷史的處理器研發商，同樣的，MIPS 架構處理器也出現在許多日常生活中可見到的產品中，在遊樂器方面，過去的任天堂 64、SONY Playstation 1、Playstation 2，以及新近的 PSP 等產品等是採用 MIPS 架構，而在一般手持式 Windows CE 產品中，也有採用 MIPS 架構，在網通產品方面，MIPS 處理器也被廣泛的應用在 CISCO 的路由器中。

以 MIPS 的架構來說，其發展的歷史比起 ARM 要來得悠久，設計上也有不少過人之處，比如說從 32 位元處理到 64 位元運算的架構延展性，讓 MIPS 處理器可適用於各種用途。



MIPS 處理器是個以管線方式工作的處理器，因此執行程式碼的速度，就相當依賴管線的工作方式。絕大多數 MIPS 指令需要在管線 RD 階段取得足夠的 operands，並且在緊接著 ALU 階段之後產生結果。由於 MIPS 架構中，大多數的指令皆能遵照這樣的運作方式進行處理，因此在指令處理效率上幾乎都能夠達到理論上的最大值。

不過少部份狀況之下，比如說，如果下一條指令必須要靠前一條指令的執行結果來進行運算處理的話，如果在前一條指令處理完之前，另一條指令就搶著進入管線，那麼將會遭遇到不可預料的錯誤。

而由於架構上是以簡單的 RISC 為基礎，因此 MIPS 有較為明顯的兩個缺陷

分支延遲：(branch delay slot)

在所有的 MIPS 處理器中，跟在分支指令之後的指令，即使在與前一個分支指令流向分歧之後，依然會被處理器所執行，因此在之後的 MIPS II 體系中，加入了 Branch-likely 指令，在處理類似的狀況時，在分支指令其後的指令只有在前一個分支被接受時，才會被執行，不過除非自行指定分支之後的指令，在加強後的編譯器的處理下，分支所帶來的延遲將顯得不明顯。

載入延遲：(load delay slot)

在 MIPS I 指令集中 load 指令將無法再次載入才剛被 load 指令本身所載入的資料，若是有需要再度載入，那麼必須在兩個 load 流程之間，使用其他指令來區隔，甚至是使用空指令來空轉一週，以便讓 load 指令可再度進行載入。

浮點運算單元的問題：

由於浮點運算需要耗費多個處理器時脈週期來進行，因此在 MIPS 處理器架構中，大多會有獨立的浮點運算處理管線，構成內部的輔助處理器架構，由於浮點運算單元可以與其後的指令並行處理，因此當並行處理的指令要去存取尚未計算完成的浮點運算結果暫存器時，處理器便會停止執行，因此這部份的處理也需要大量的編譯器最佳化。

架構比較

◆ Pipeline

- MIPS

最簡單的體系結構之一，體積小、耗能比低。但 MIPS 有 "branch delay slot" 以及 "load delay slot" 兩個明顯的缺點，MIPS 使用編譯器來解決上面的兩個問題。因為 MIPS 最初的設計思想是使用簡單的 RISC 及其他軟體技術，來達成 RISC 的完整概念。

- ARM's Shifter

shifter 是 ARM 中很重要的概念，他可以提高運算邏輯的速度，跟同樣功能的 adder/shift register 相比，效率更高，但是也 占用更多的芯片面積。

◆ 指令結構

MIPS 有 32bit 及 64bit architecture，但是 ARM 只有 32bit architecture (ARM11 局部 64 位)

MIPS 是開放式的架構，可在開發的內核中建立自己的指令，而 ARM 是在每個指令結構以 4bit 的 condition code 決定。

◆ 寄存器

由於 MIPS 內核中有 32 個 register,相對於 ARM 只有 16 個，設計天生上的優勢使得同等性能下，MIPS 的芯片面積及功耗比較小。ARM 有一組特殊用途寄存器 cp0-cp15,可以使用 MCR,MRC 等指令控制;相對應的，MIPS 也有 cp0 0-30,使用 mfc0,mtc0 指令控制。

◆ 地址空間 address space

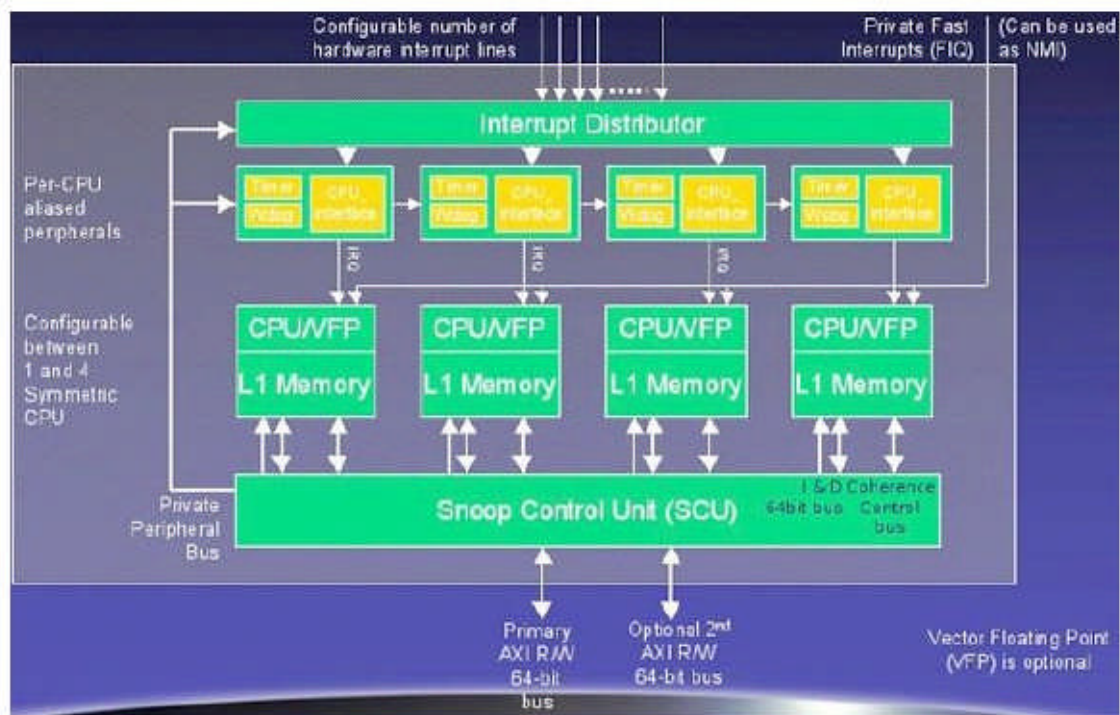
MIPS 起始地址是 0xbf00000,會有 4Mbyte 的大小限制，但一般 MIPS 芯片都會採取一些方法解決這個問題。而 ARM 沒有這種問題。在新版的 MIPS24K 起始地址往後移，有了 16byte 的大小。

發展

爲了提升效能以及改善未來適應性，兩家公司近年來在提升以及研發都有了明確的方向。

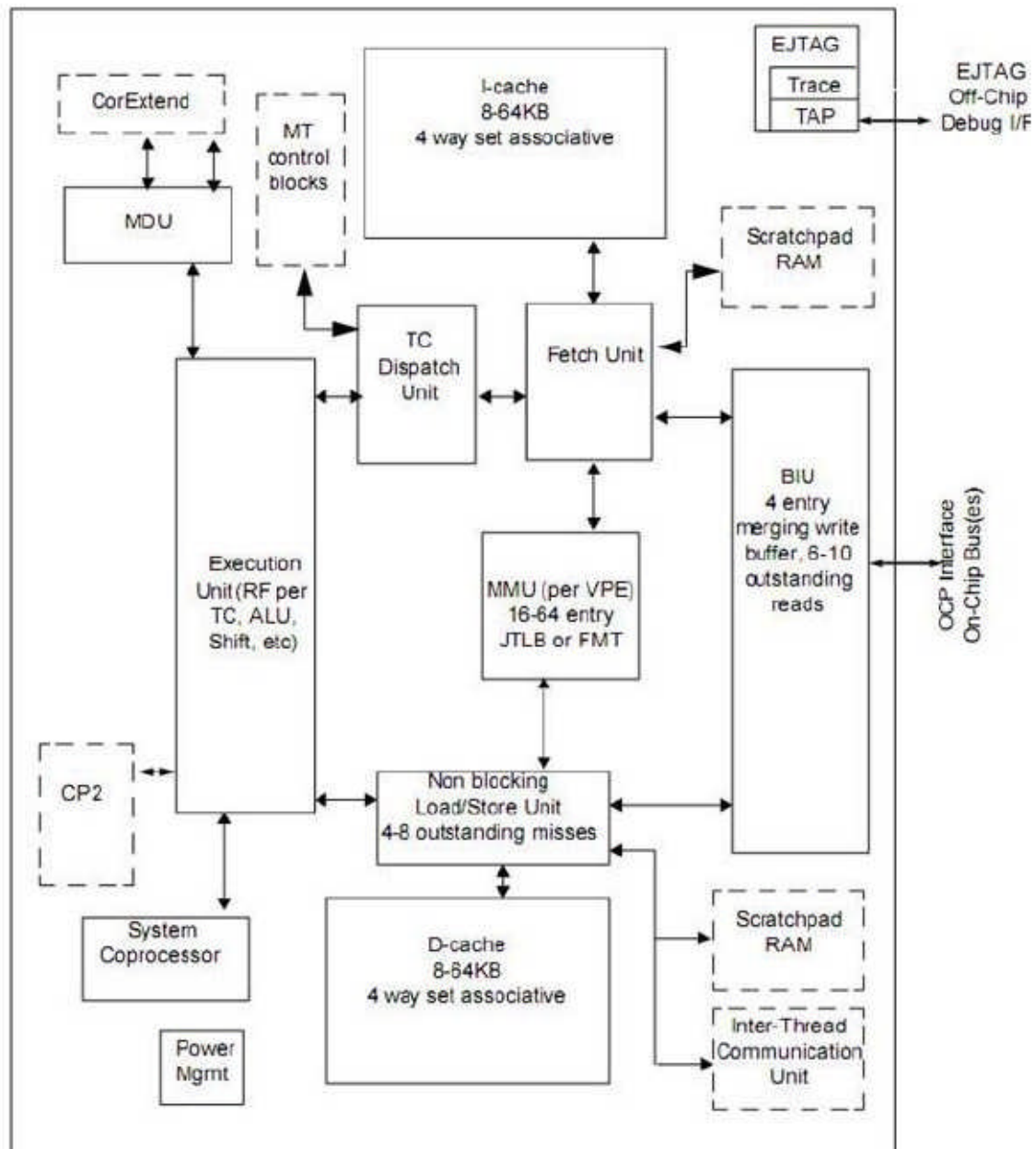
ARM-多核心架構 Multi Processor

以 ARM 處理器-ARM11 MPCORE，根據不同應用的需要，MPCore 可以被配置爲 1~4 個處理器的組合方式，根據官方表示，其最高性能約可達到 2600 Dhrystone MIPS 的程度 MPCore 是標準的同質多核心處理器，。組成 MPCore 的是 4 個基於 ARM11 架構的處理器核心，由於多核心設計的優點是在頻率不變的情況下讓處理器的性能獲得明顯提升，因此可望在多任務應用中擁有良好的表現。



MIPS- 多執行緒 Multi Thread

單一處理器核心在運算的過程中，常會有記憶體存取速度跟不上處理器時脈增加的問題，進而導致快取記憶體錯失（miss）時，形成執行管線長時間閒置的狀。如果利用多執行緒處理概念，適時的將其他執行緒拉過來填補已經造成的閒置狀態，其速度的增長甚至可以達到非常明顯的地步。



多執行緒技術著重於處理單元、記憶體控制器的有效利用，在最大程度上節省電晶體的使用，並且在此前提之下往上提升效能表現，這與多核心架構中，系統效能需求有多少，就複製多少個核心塞進晶片中的浪費作法完全不同，多核心可以取得較為全面的應用廣度，但是稍嫌鋪張浪費，而多執行緒在成本與效能方

面的平衡性表現要來得高明些。但是多執行緒技術有個嚴重的缺陷，那就是多執行緒工作處理過程中，過於頻繁的上下文切換（context switch）將有可能會造成極大的效能耗損。

心得

其實在最早著手於期末報告時，是想以多執行緒以及多核心差別為主的介紹方式。但是由於多執行緒的技術非常複雜，以我的程度還沒辦法好好了解詳述，使得這次的報告內容比我預計的內容還為少。

但是這次的報告，也讓我了解不只是 ARM，還有其它千千種的處理架構系統，也算是一大收穫。而學習本學期的課程的原理，讓我了解到資工系不只要只會學習寫 CODE，而是如何有自己的 IDEA，以及在處理結構能好好掌握才能寫出一個好程式來。

Reference

Wikipedia

<http://zh.wikipedia.org/w/index.php?title=%E9%A6%96%E9%A1%B5&variant=zh-tw>

DIGITIMES 科技網

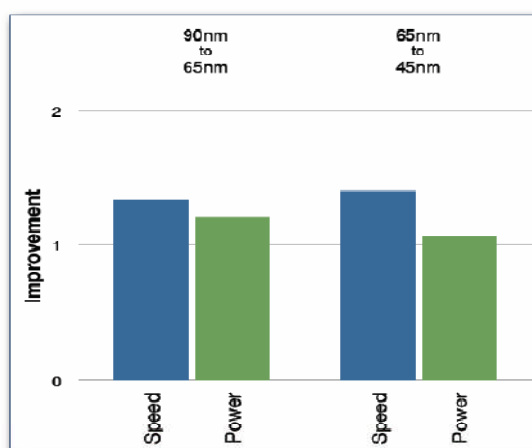
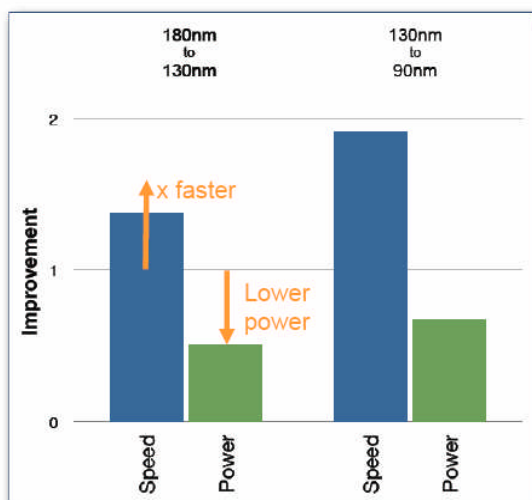
<http://www.digitimes.com.tw/>

台大資工系 計算機組織與組合語言 課程網

<http://www.csie.ntu.edu.tw/~cyy/courses/assembly/07fall/>

以 ARM 技術的發展歷程來看，早期的 ARM7 架構本身能夠滿足一些音效編譯碼應用。在增加 16 位元飽和運算指令和提高 ARM9 核心速度後，完成音效編碼，及大約 15 畫格/秒速度下的 MPEG-4 編碼工作。在 ARM11 V6 指令集架構增加速度和 SIMD 指令後，就可以實現 VGA 分辨率的 H.264 編碼。再進一步到最新的 Cortex A8 與其基於 64 位 SIMD(Single Instruction Multiple Data)架構的 Neon 加速器搭配工作之下，300 MHz 工作頻率的一半效能就可以即時完成 30 畫格/秒的 MPEG-4 VGA 編碼工作。

基本上這些處理器每 2 年有 2 倍計算效能提昇，在這種提昇過程中，早期晶片技術由 180nm~130nm 進展至 130~90nm 時，可在較低能源消耗下同時提高速率，而近期由 90nm~65nm 演進至 65nm~45nm，速率提昇則是在更高能源消耗來達成。



指令管線 Pipeline 複雜度可提升計算頻率，ARM9 有 5 階 pipeline，ARM11 有 8 階，演變到 Cortex-A8 有 13 階，說明複雜度呈現指數增加，費用比效能增加更多，而且照此趨勢預期回收更少。ARM 在此理念下產生 Multicore Processing，雙核心

處理器有雙倍效能也消耗雙倍能源，當 2 task 分由 2 核心處理較共用單核心處理則也更有效率。

ILP

在進一步討論多執行緒前先討論各種平行計算技術。ILP (Instruction Level Parallelism) 是做為量測電腦程式中可同時執行運算之數目，以下列程式 3 組運算為例：

1. $e = a + b$
2. $f = c + d$
3. $g = e * f$

運算 3 依據運算 1、2 結果，因此須前 2 者算出結果後才能計算。但運算 1、2 並不依賴任何其他運算，可同時計算。假設每個運算須 1 單位時間來完成，這 3 個指令總共需 2 個單位時間來完成，得出 ILP 為 3/2。

Compiler 或處理器設計者須能指認出且利用此等 ILP。一般程式撰寫採用循序執行模式，而程式指令也是一個接一個依據 programmer 指定次序執行。ILP 允許 Compiler 或處理器將多個指令重疊執行，甚至改變其執行順序。一個程式能有多少 ILP 完全依應用程式而定。在某些領域像繪圖、科學計算 ILP 值就非常。但在密碼編譯就少有平行運算。用以改善 ILP 之 Micro-architectural 技術有：

- Instruction Pipelining：使多個指令之執行部分可以重疊。
- Superscaler：採用多個 execution units 以平行執行多個指令。一般 superscaler 處理器同時執行之指令次序在原程式是聚在一起。
- Out of order execution：只要不違背數據資料依存性，其指令執行無一定次序。此技術和 Instruction Pipelining、Superscaler 並不相同。
- Register renaming：是指防止因 register 重複使用致使程式需順序執行一種技術，以允許 Out of order execution。
- Speculative execution：允許部分或全部指令，在確認其須執行前就先行執行。Speculative execution 常見之形式為 control flow 後之指令，在 branch 不確定前就先行執行。其他形式有 value prediction、memory dependence prediction、cache latency prediction。
- Branch prediction：用以防止 control dependencies 而停止之技術，此法和 Speculative execution 共同使用。
- VLIW (Very Long Instruction Word)：此技術只適用於極長串指令架構電腦。

TLP

在多處理器系統，平行工作是讓每個處理器各自執行不同執行緒，而執行緒可處理相同或不同資料。在效能指標方面並不是一個程式能產生幾個執行緒，而是在多處理器環境下能有多少執行緒被同時執行，而 TLP 就是此平行執行緒數目之量測。

多緒處理器

所謂多緒處理器指提供有效率執行多個執行緒之硬體支援，和多核處理器差異是所有執行緒須共用單個核心：只有一組的運算單元、CPU caches、TLB (translation lookaside buffer，一種 CPU cache 用於記憶體管理硬體以改善虛擬位址與實體位址間的轉換速度)。多核處理器系統有多個完整處理單元，相對的多緒處理器系統只能透過 TLP、ILP 來提昇單核之使用率。通俗說法，多緒處理器必定重視 ILP，而多核處理器則注重 TLP 之改善。

HyperThreading 範例

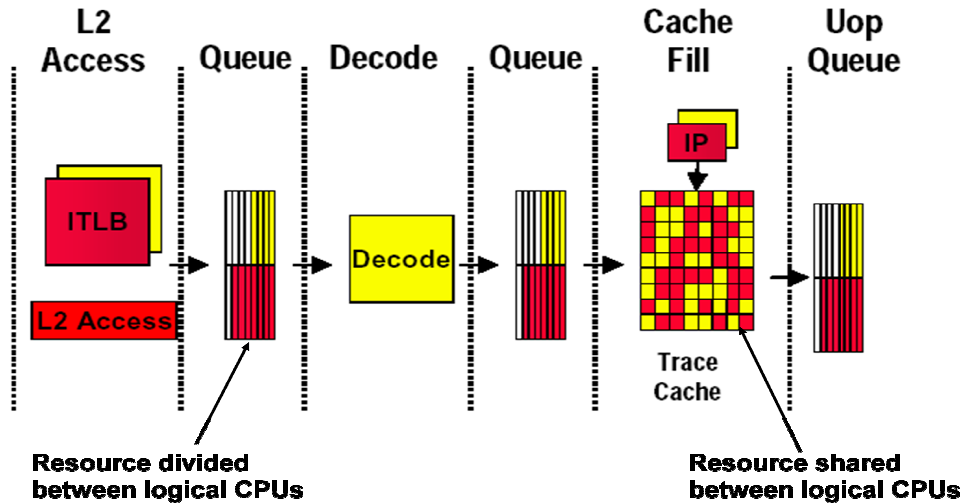
由於 MIPS 的多緒設計細部技術資料不易找到，本文以類似的 Intel HyperThreading 來說明改善 ILP、TLP 實際實施方法。Intel 從 486 進展到 Pentium 4 有超過 75 倍效能提昇，其中 10 倍是歸諸晶片製程改善，6 倍以上採用 Superscaler、Speculative out of order、Trace cache (trace cache 為 cache 特殊應用，將已被擷取、解碼指令之過程儲存，可增加指令擷取頻寬)等 Micro-architecture 技術應用。HyperThreading 實施，具備 SMT (Simultaneous Multi-Threading) 特性，即多於一組以上執行緒之指令可在任一時間於任一 pipeline stage 執行。SMT 重點在“共用機構資源”將既有資源做最大利用。並不需對處理器基本架構做多大改變，主要增加為在一週期內從多個執行緒擷取指令，更大之 register file 以收容多緒資料。

HyperThreading 和前幾代 CPU 其最大改變為：

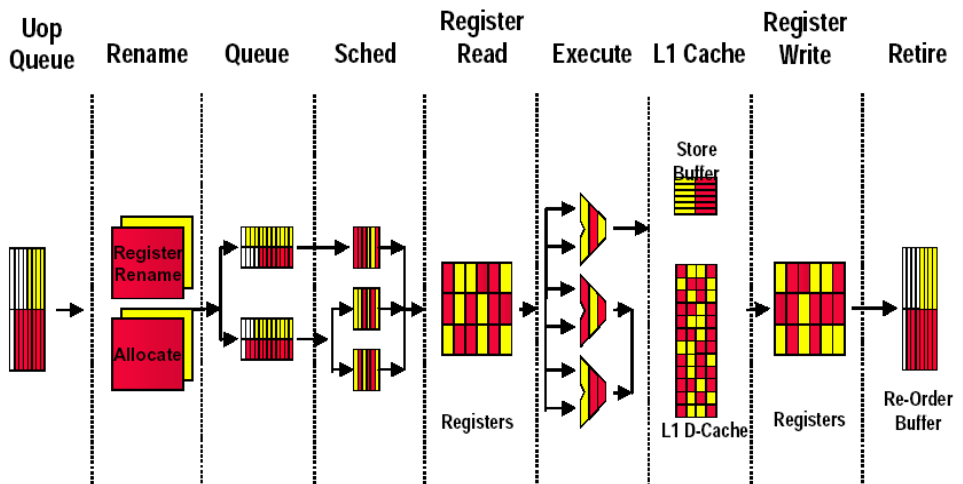
- 複製機構資源
 - 所有每個CPU之架構狀態(architecture state)，但不增加execution resources
 - Instruction Pointers, renaming logic
 - 部分較小機構資源
 - return stack predictor, Instruction TLB, etc
- 機構資源進一步切割
 - 增加數個buffers (Re-order buffer, load/store suffers, queues, etc)
- 共用大部分機構資源

- Out-of-Order execution engine
- Caches

依據Intel，單個實體CPU只增加5%晶片面積，就可維持2個邏輯CPU平行運算，提高效能15-30%。



Pentium 4 HyperThreading Front End



Pentium 4 HyperThreading Execution Pipeline

多核處理器

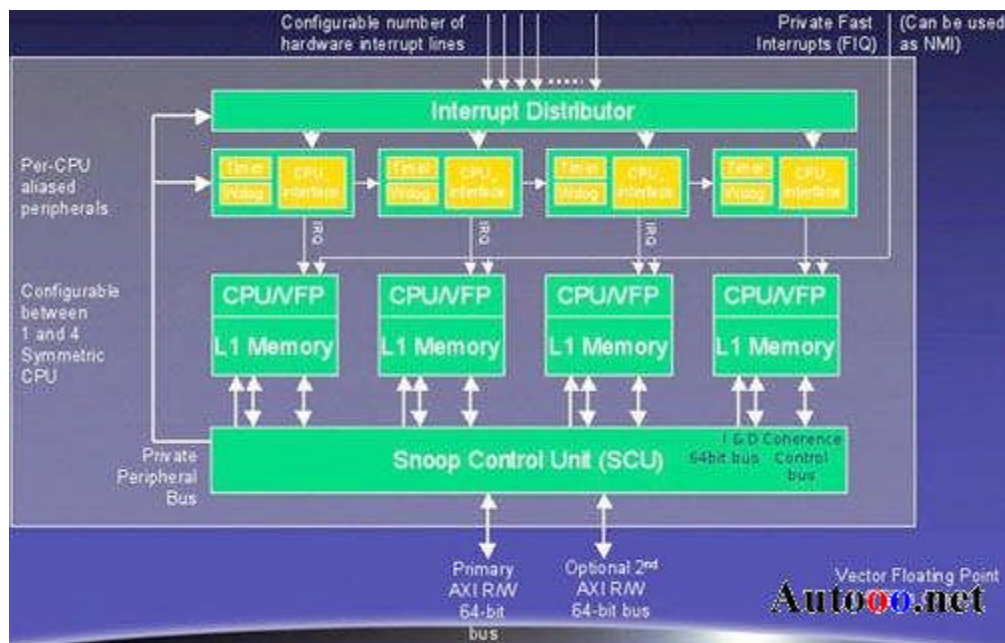
多核處理器是將 2 個以上核心(通常為 CPU)組合在同一片晶片包裝內者。在多核處理器之核心可能共用一組 coherent cache (例如，Intel Core 2) 或是各有其獨自之 cache (例如，現有 AMD dual-core processors)。此類處理器也可共用介面到其他系統。每個核心各自獨立執行 superscalar execution、pipelining、multithreading 最佳化。n 核系統當 n 個或以上之執行緒能同時執行為有效率。多核處理器所能增加效能

是依據待解決問題特性、演算法、及實際軟體程式撰寫方式。但此種處理器在多工環境將表現較好，因為每個核心各處理一個程式，可同時運跑 2 組程式。

ARM TLP

ARM11 MPCore 乃是在 ARM11 核心的基礎組成，架構上屬於 V6 指令體系。根據不同應用的需要，MPCore 可以被配置為 1~4 個處理器的組合方式。MPCore 是標準的同質多核心處理器，最高性能達到 2600 Dhrystone MIPS，運算能力與 10 倍耗能以上之 1GHz Pentium III 處於同一水準。MPCore 多處理器可支持高達 4 路共享快取結構的對稱多處理器（four-way cache coherent symmetric multiprocessing，SMP）、或者是 4 路不對稱多處理器（four-way asymmetric multiprocessing，AMP）以及 4 路兼有對稱/不對稱的混合式多處理器系統。其設計理論上可以滿足各種跨領域應用對運算性能的彈性需求，確保系統可獲得一流的響應能力或數據吞吐量。

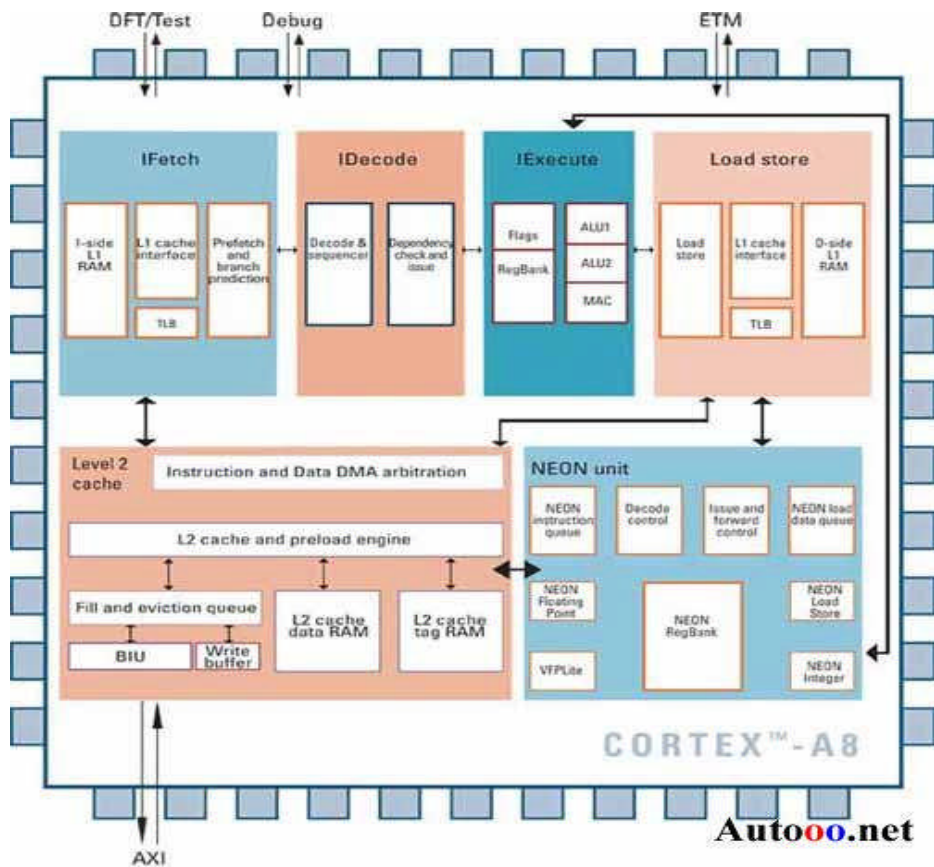
多處理器必然意味着晶片大幅提升，功耗也隨明顯增大。為緩解功耗這個問題，MPCore 引入 Adaptive Shutdown 技術將暫處於閒置狀態的處理核心關閉，讓 MPCore 以單核心的節能方式進行，及 Intelligent Energy Management 智能电源管理技術，根據任務的負荷情況對處理器的電壓進行動態調節，達到節能目的。



圖說：ARM11 MPCore 的結構示意圖。

ARM 處理器架構 Cortex-A8，基於最新的 ARM v7 體系，並且整合了 1 個 64 位 DSP 處理單元，對數位信號處理應用具備極佳的加速能力，因此非常適用於類

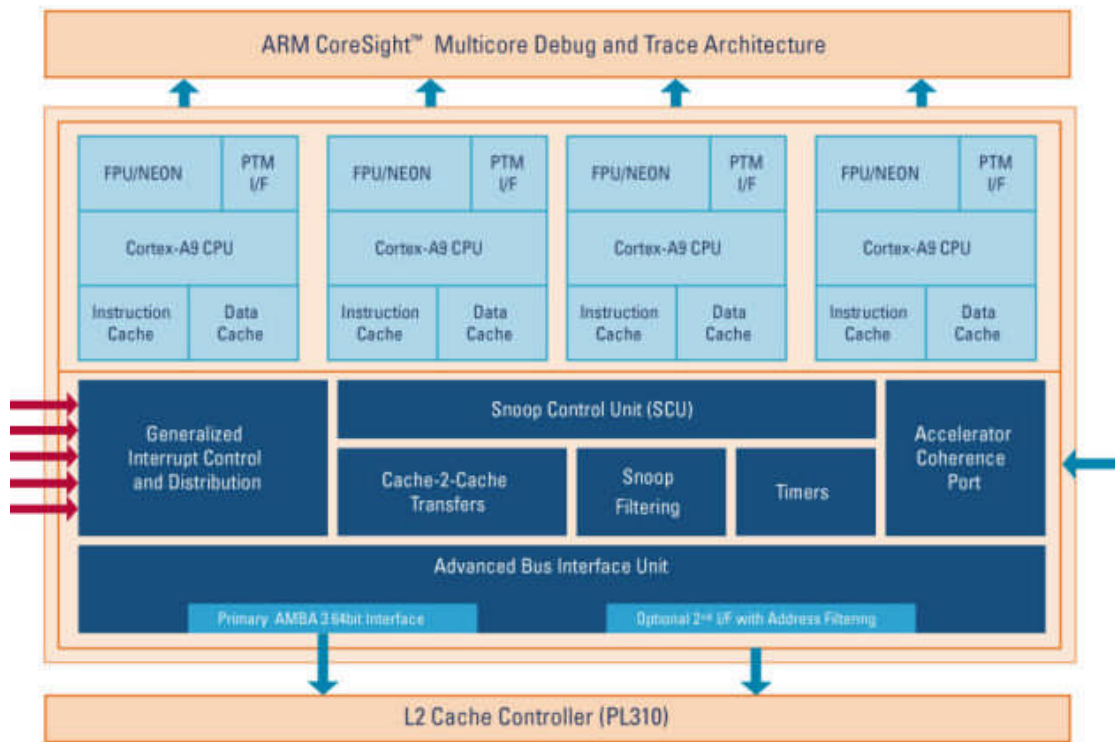
MID 掌上型裝置的多媒體、甚至是遊戲應用。Cortex-A8 算多核心體系之一，但是其架構與 MPCore 之類的同質核心不同，而是採用 1 個通用處理器核心，並搭配個 DSP 核心而成的異質多核心處理器。(本圖只顯示單核，但著重在顯示 ILP 多緒設計之改善。)



圖說：Cortex-A8 的結構示意圖

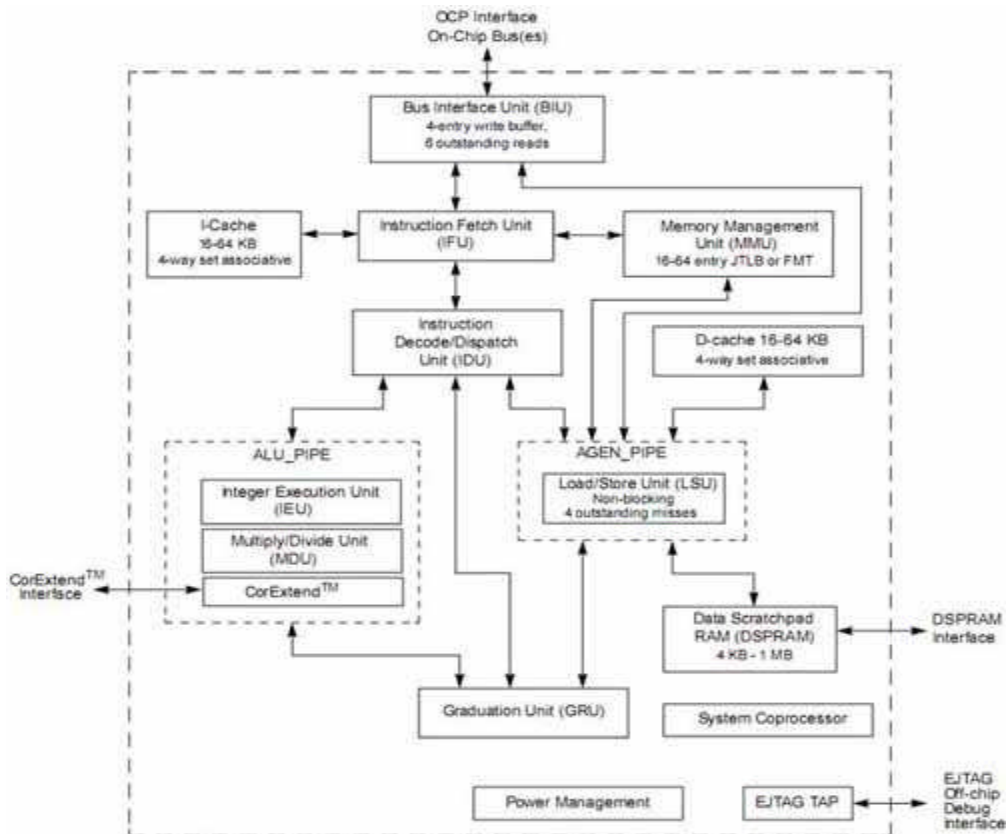
2008年最新發佈之ARM Cortex-A9，宣稱為第二代之MPcore multicore processor，從其宣傳資料有如下特性：

- 引進 out-of-order instruction issue and completion
- Register renaming 允許 out of order execution
- Non-blocking memory system with load-store forwarding
- Fast loop mode in instruction prefetch



圖說：Cortex-A9 MPCore 的結構示意圖

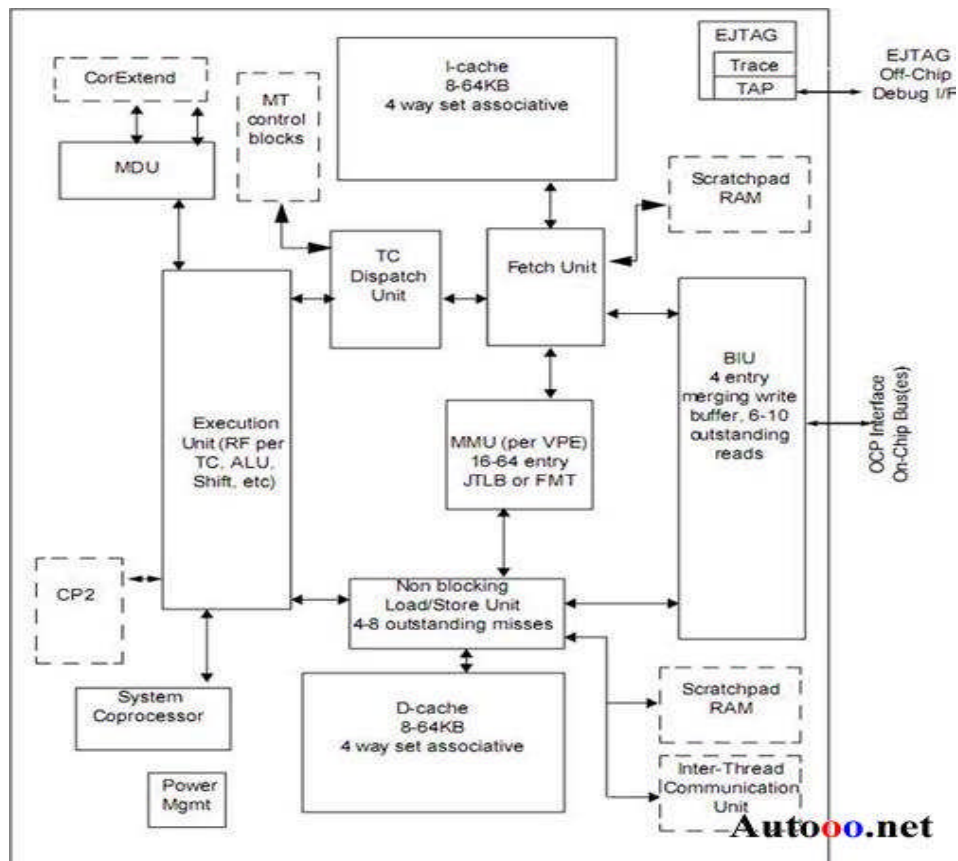
MIPS ILP



圖說：MIPS 74K 處理器結構示意圖。

MIPS 主張多緒架構設計，其主要訴求是在單一個處理器核心之內改善就可以達成更多的工作。在 SoC 晶片裡面，則是在不大幅增加晶片面積前提下，增加處理器的處理能量。其達成的方式為，設計相對應數量的暫存器以及程式計數器，在只增加少量的晶片面積，仍可達到更大的每時脈指令吞吐量。

MIPS 公司產品線，分別是單執行緒的 24K 與 74K 系列，以及多執行緒的 34K 系列。74K 在 65nm 製程下，其運作頻率已經超越 1GHz，採用通用處理器搭配 DSP 核心的設計，不過總體效能與功耗表現略遜於類似架構的 ARM Cortex-A8。多執行緒處理器的主角—34K 系列，該處理器核心能設定 1 或 2 個虛擬處理組件 (VPE) 以及最多 5 個執行緒內容 (TC)，提供充分的可配置彈性。兩個 VPE 的作法就是將單顆核心模擬為 2 個核心，使 34K 核心能同時執行兩個獨立的操作系統，或是一個雙路的對稱式多重處理器操作系統。



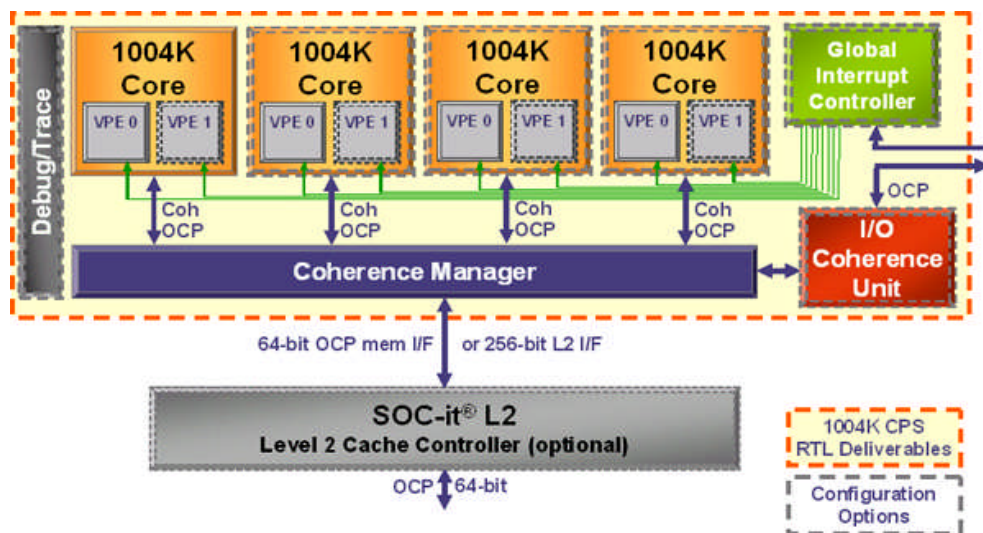
圖說：MIPS 34K 處理器結構示意圖。

根據 MIPS 自家的估算，與同家族的 24K 系列產品相較起來，34K 在 2 個 VPE 以及 2 個 TC 的組態設定之下，可以將效能提升到超越 24K 處理器 60% 的程度，晶片面積大略增加 14%，而因為多執行緒作業所導致的高速緩存失誤比率則是

由4.41%增加到5.16%，算是在可接受的範圍之內。不過與單核心74K 相較起來，34K 反而更不適用於網絡或多媒體串流的密集計算環境，而 VPE 和 TC 單元的增加，同樣也會加大晶片的面積。雖說 MT 技術的局限性，使其不適合用於多媒體編譯碼應用上，但是在汽車電子方面，已經有廠商成功利用 2 顆 34K 處理器組成雙核多執行緒處理器，並提供的相當優秀的執行效能。

於2008年4月 MIPS 也開始對工業界提供多執行緒、多處理器 MIPS32 1004K，支援四路同質多處理器及晶片多執行緒(four-way symmetric multiprocessing (SMP) and chip multithreading)，全面實施可組成相當於八路同質多處理器。

MIPS 1004K Coherent Processing System (CPS)和先前他家多處理器產品不同之處在於特別組合的 coherent SMP 及 hardware multithreading，"coherent" 此處旨在強調 MIPS 硬體在執行緒處理有較高凝聚性之效率。1004K core 主要是架在 MIPS32 34K 處理器上，再具備 SMP 同質多處理器。



MIPS32 1004K 架構

先前 MIPS 多核心設計是採用多個獨自之 MIPS 處理器來組成，1004K 提供至多 4 個處理器，由多執行緒、多核心技術整合，具備單個處理器之 4 倍效能。

內嵌式處理器方面，早期 ARM 式著重在多核、執行緒層級平行的設計，MIPS 則是注重多緒、指令層級平行設計，目前則都是朝著多核與多緒方向發展，以大幅提昇整合的計算能力。

心得

由於我是重修生，所以這份報告以以前的為主，再加上上次結尾提到想要多探討的地方，就是這次所說的多執行緒跟多核心的差異。