行政院國家科學委員會專題研究計畫　期中進度報告

低功率系統晶片設計的智慧型整體架構(1/3)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-002-056-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：賴飛羆

計畫參與人員：張延任、蔡坤霖、鍾玉芳、陳立偉、鄭昂旻、張志鵬

報告類型：完整報告
報告附件：國際合作計畫研究心得報告
處理方式：本計畫可公開查詢

中　華　民　國 93 年 5 月 24 日

行政院國家科學委員會補助專題研究計畫 **□ 成 果 報 告**
☑期中進度報告

低功率系統晶片設計的智慧型整體架構(1/3)

計畫類別：☑ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 92－2213－E－002－056－
執行期間： 92 年 8 月 1 日至 93 年 7 月 31 日

計畫主持人： 賴 飛 羆　　教授
共同主持人：
計畫參與人員： 張延任、蔡坤霖、鍾玉芳、陳立偉、鄭昂旻、張志鵬

成果報告類型(依經費核定清單規定繳交)：□精簡報告　☑完整報告

本成果報告包括以下應繳交之附件：
☑赴國外出差或研習心得報告一份（附件一）
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立台灣大學資訊工程學系暨研究所

中　華　民　國　93　年　5　月　31　日

摘要

功率消耗在目前的處理器設計上是一個很急迫的問題。因為系統晶片上的快取記憶體通常會消耗大量的電功率，而減少快取記憶體所造成的功率消耗是我們對於減低能量耗損的其中一個目標。在本期的計畫當中，我們設計了一個雙層式架構的濾器(two-level filter)，由 L1 與 L2 濾器所組成。因為在傳統的快取設計上，快取記憶體運作時會有很多不必要的動作。因此，我們用一個單一的區塊緩衝器(block buffer)當作 L1 濾器來減少不必要的快取存取。在 L2 濾器中，我們提出了一個警示標籤(sentry-tag)的架構來更進一步地過濾 L1 所遺漏的運作。我們用 *SimpleScalar* 模擬 SPEC2000 測試程式並執行 HSPICE 去評估我們所提出的架構。實驗結果顯示出雙層式的濾器可以有效的減少大量非必要動作所產生的能量消耗。以 L1 濾器跟雙層式架構相比，雙層式架構可以減少 30%的功率消耗。同樣地用過去傳統的架構與單一 L1 濾器相比，則可以節省約 46%的功率消耗.

關鍵詞：區塊緩衝區、濾器架構、低功率快取記憶體、功率消耗、
非必要性快取行為

Abstract

Power Consumption is an increasingly pressing problem in modern processor design. Since the on-chip caches usually consume a significant amount of power, it is one of the most attractive targets for power reduction. This project presents a two-level filter scheme, which consists of the L1 and L2 filters, to reduce the power consumption of the power consumption of the on-chip cache. The main idea of the proposed scheme is motivated by the substantial unnecessary activities in conventional cache architecture. We use a single block buffer as the L1 filter to eliminate the unnecessary cache accesses. In the L2 filter, we then propose a new sentry-tag architecture to further filter out the unnecessary way activities in case of the L1 filter miss. We use *SimpleScalar* to simulate the SPEC2000 benchmarks and perform the HSPICE simulations to evaluate the proposed architecture. Experimental results show that the two-level filter scheme can effectively reduce the cache power consumption by eliminating most unnecessary cache activities, while the compromise of system performance is negligible. Compared to a conventional instruction cache (32 KB, two-way) implemented with only the L1 filter, the use of a two-level filter can result in roughly 30% reduction in total cache power consumption. Similarly, compared to a conventional data cache (32KB, four-way) implemented with only the L1 filter, the total cache power reduction is approximately 46%.

Keywords: Block buffer, filter scheme, low-power cache, power consumption, unnecessary cache activity.

目錄

一、前言

由於晶片上的快取記憶體能夠有效地減少處理機和主記憶體間速度上的差異，因此幾乎現代的微處理器都普遍使用快取來提升系統的效能。在高時脈頻率下，這些晶片上的快取記憶體通常是由高密度靜態隨機存取記憶體所排列而成的陣列來實做。因此，快取記憶體中的電晶體總數便在晶片整體電晶體數目當中佔了一個很重要的部分。而隨著系統晶片的快取記憶體尺寸不斷地變大，快取記憶體在系統晶片中所消耗的電功率也顯得更為重要。隨著處理器變得更複雜以及提供更高的效率，這個趨勢也會一直持續下去。

如同上面所提到的，快取記憶體是在減少電功率消耗上最具有吸引力的目標之一。在減少晶片快取記憶體的功率消耗上有好幾個技術。與傳統平行存取流程不同，Hasegawa [3] 提出可連續存取的階段式快取，首先是標籤比較，之後跟隨著資料陣列讀取，使其能夠讀出實際上需要的資料陣列。然而，這個階段式快取會經歷更長的快取命中時間。Choi [4]用一個新的標籤架構和標籤省略的技術，減少非必要查詢標籤的次數。濾器快取[5]，L 快取[6]，區塊緩衝[7]，和多線緩衝器[8]等，透過放置小型快取或輸出拴鎖在處理機和 L1 快取之間來試圖減少功率消耗。如果 L0 快取或者輸出拴鎖能夠供應多數 L1 快取的要求，則能夠大大地減少 L1 快取活動，從而節省功率消耗。在 [7] 和 [8] 中也介紹分區快取的概念，在其中資料記憶快取陣列被分成幾個區塊。在每一次快取存取中，只有含有資料的那些區塊才會被讀出。在[9]中，Albonesi 利用集合關連式快取記憶體的分區陣列，並且在不需要整塊快取來提高效能的時候，提出一條選擇性存取路徑方法來阻斷一個區域之內的快取路徑。然而除了硬體修正之外，選擇快取路徑方法需要許多軟體支援，包括特別指令和為了專門分析應用快取所需要的軟體。這個路徑預測的集合關連式快取記憶體透過存取單一預測的快取路徑而不存取所有快取路徑來減少功率消耗。整個快取會因為預測錯誤變的跟傳統的架構沒有差異，所以效率與所節省的

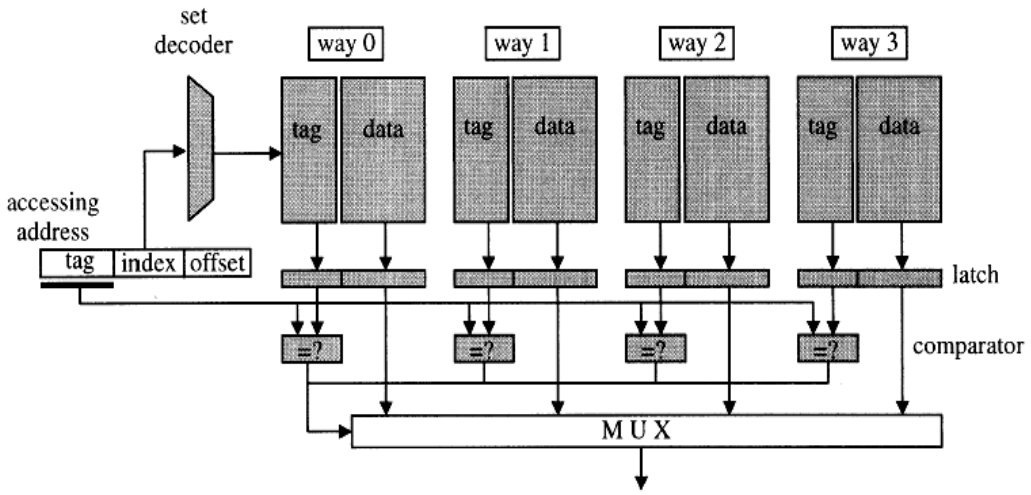
功率會大大的依據所預測的準確度而定。

在本計畫中，我們對研究低耗電成本的方法感到興趣並藉以減少快取記憶體功率消耗，而它與軟體無關，僅需要一些硬體的額外成本以及不大的架構修正。我們提出二層次的濾器規劃結合一個區塊緩衝器與一個新警示標籤架構。在 L1 濾器中，用這個區塊緩衝器利用參考的空間位置以減少不必要快取存取。區塊緩衝器是一個著名的技術，但是它只利於具有較高空間區域性的應用程式。因而，在第二層濾器中，我們提出這個警示標籤以防 L1 濾器遺漏過濾掉這些不必要的存取。透過用這個 L2 濾器來存取有可能的擊中路徑而全部存取，能夠更進一步減少這個快取的功率消耗。為了瞭解 L2 濾器的架構，我們發展了一個分析模型，並用實驗結果加以證實。與先前的研究[9]，[10]比較，所提出的雙層次的濾器規劃不需要任何軟體支援和保留固定的快取存取時間。這個研究與先前研究[11]的主要差別是我們發展一個分析模型以評估我們所提出架構的效率,並且在本文中提供更多有關能量及效率的細節。

本文的其餘部分組織如下。第二部分我們說明傳統關聯式快取記憶體執行時的問題。部分三，描述我們提出的雙層次的濾器規劃快取架構的細節，並為濾器效能提供一個分析模型。我們也為所提出的架構提供一個詳細的能量評估模型。實驗結果與結論則是在第五部份。

二、背景

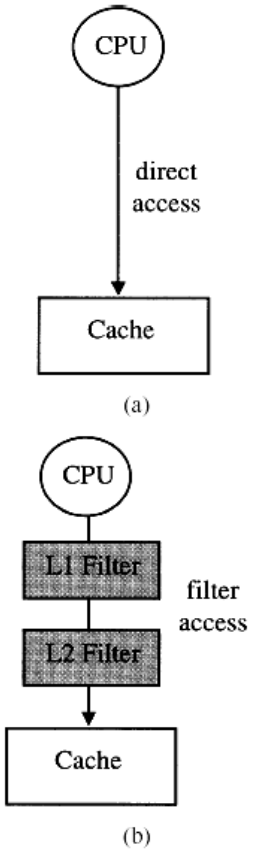
圖一顯示了傳統四路集合關聯式快取記憶體的實作。中央處理器發出一個由三部份組成的位址到快取,即標籤,索引,和位移。考慮一路的關聯式快取，全部大小是 C 位元組而一個區塊尺寸是 B 位元組. 因為區塊的數目是 C/(B*A) 索引的長度是 log(S)位元，用來給第幾集合區塊做索引，來得到這個資料。位移的長度是 log(B/4)，用來在一個區塊之內選擇適當的字組(1 字組=4 區塊)。最後，用這個標籤部分來檢查目前的存取是否命中或者失誤。更進一步將存取時間延遲減到最少，資料快取陣列同時用標籤存取陣列，然後用標籤比較的結果來選擇這個需要的區塊。換句話說，在四路的集合關聯式快取中，總是為四路的存取活動，如同由灰色區塊表示那樣。傳統平行存取規劃用於集合關聯式的快取有益於效率，但是，它並沒有從功率消耗的觀點來作最佳化。這是因為若在知道標籤比較的結果作平行資料陣列存取，則會造成許多不必要的存取和功率消耗。例如，設想存取位址的標籤是(x, …, x1)。選擇的集合含有四個區塊(這是四路的集合相聯的快取), 而標籤的內容陣列分別(x, …, x0) (x, …, x0) (x, …, x0) (x, …, x1)。很明顯地，需要的資料不在路徑 0, 1, 2 之內，因為標籤的最後一位元總是 0，若不等於存取位址(假設為 1)。那麼，路徑 0, 1 和 2 是這個途徑中的非必要活動。如果我們在開始作這個傳統快取存取前知道這個結果，我們才可能允許存取路徑 3 而不存取這個整個快取。隨著關聯的程度變得更大，非必要路徑活動的次數將會增加以及功率消耗也會增加。



圖一、傳統的四路集合關連式快取記憶體架構。

三、研究方法

在這個部分中，我們提出一個簡單和有效的雙層次濾器規劃以減少非必要的快取活動次數和非必要的功率消耗。不直接存取(如同在圖 2(a)顯示那樣)，我們同時用兩個濾器來減少不必要快取活動的次數(如同在圖 2(b)顯示那樣)，在第一層(L1)濾器和第二層濾器(L2)分別是一個單一塊緩衝器和崗哨的標籤。



圖二、(a) 傳統快取架構 (b)具備我們所提出的雙層式濾器方法的快取架構

I. 第一層(L1)濾器：單一區塊緩衝器

就之前章節中描述的傳統快取記憶體架構中的單一塊緩衝器來說，快取記憶體存取的單元是一個區塊。區塊尺寸的範圍在目前處理機中通常是 4 到

16 字組。為使其應用具有空間區域性，下一次將存取的資料很有可能位於上一次存取的相同區塊中。因此，我們能夠利用空間區域性的特質，增加一個輸出拴鎖器來減少非必要快取的存取次數。換句話說，如果要被存取的快取區塊目前仍存在於緩衝器中，則所需要的資料便可以從區塊緩衝器直接取出而不需要進行一般的快取存取。Su 和 Despain 提出了一個單一區塊緩衝器的快取[7]。而有研究[8]指出，一定數量區塊緩衝器的使用可減少快取記憶體的功率消耗時並且很有效率。他們證明當使用八個區塊緩衝器時，40% 50%的功率可以被節省下來。而我們所期望的是，當區塊緩衝器的數量增加時，所節省的功率消耗也會變多，但一旦使用超過一個區塊緩衝器，所節省的功率並不如用一個緩衝器所節省的量。事實上用一個區塊緩衝器就可以節省 40%的功率，所在本計畫中我們就只用一個區塊緩衝器。

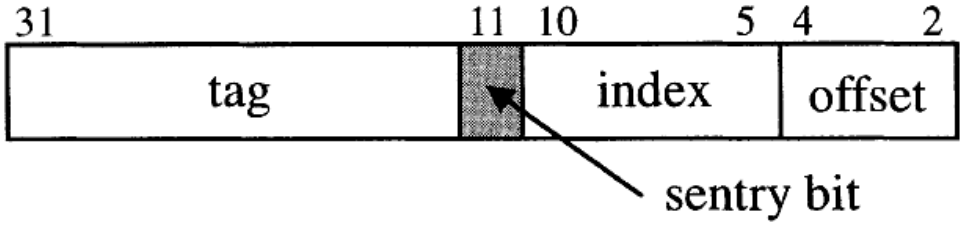
II. 第二層(L2)濾器：警示標籤

從功率和效能改進方面來看，一個區塊緩衝器的使用的確有其效能，但是，功率節省的多寡取決於程式行為。存取串流具有的空間區域性越高，能夠被節省的功率也就越多。這個特性不見得有益於具有低空間區域性的那些程式。我們提出的 L2 濾器架構的關鍵想法是要在區塊緩衝失誤的情況下減少這些非必要的存取行為，因而進一步減少快取記憶體的功率消耗。

警示位元可定義為對每一個快取區塊的標識符號。首先我們得選擇一些標籤位元做為警示位元，然後把他們從標籤陣列撤下，並移到警示標籤儲存陣列。藉由預先比較存取位址的警示位元與選定集合中的警示標籤，這個 L2 濾器的規劃可以有效的識別出哪一個路徑活動不是必須的，然後在往後的存取中便可以阻擋不需要的快取路徑。警示標籤的內容將會被更新在快取沒有命中且需要的區塊從低階記憶體被重新載入時。

例如，讓目前存取位址的警示位元為 1，而被選擇的集合含有四個區塊並且那些警示位元分別為 “0”, “1”, “0”, “0”。很清楚地可以知道，我們不可能存取到路徑 0，路徑 2，與路徑 3，我們能夠使這三條路徑被阻擋。對第一條路徑，因為警示位元是 1，意味著路徑 1 潛藏有需要的資料。因而，在這種情況下，我們能夠把路徑由 4 變成 1 並減少能量消耗。不同於[9]在選擇快取路徑上需要軟體的支援去分析應用軟體。因此我們可以應用所提出的警示標籤在處理器上而不需修改作業系統還有指令集。

必須注意的是，一次以上的警示位元比較是可能的。它意味著我們的規劃不保證所有非必要快取活動的消除。任何的標籤位元可以被當作警示位元，由於參考的空間區域性特質，標籤的低次序位元比更高次序位元更敏感。最簡單的選擇是要把標籤部分的最低位元用作 1-B 的崗哨位元 (e.g 圖三中的 A[11])。越多的位元當作標籤位元，則過濾的過程將會越正確。在接下來部分，我們會對於不同的警示位元評估 L2 濾器的效率。



圖三、具有一個警示位元的 8-kB 四路快取位址空間分割

III. 警示標籤的分析模型

理想上，給幾個位元的警示位元(S)，在每次存取中的每一個路徑活動(平均活動次數為 W_{Ave})，能夠被表示成命中率(HR)跟快取路徑的多寡，如(1)式所示。對於每一次快取行為，會有兩個可能的結果：命中或失誤。首先，在快

取命中的時候，被命中的路徑應該要被啟動，而在剩下的(W-1)路徑中，因為警示的 S 位元的關係，應該要有$(W-1)/2^S$個路徑被啟動。因此在每一次的快取命中平均路徑活動是 $HR*(1+(W-1)/2^S)$，也就是(1)式的前半部。同樣的，萬一沒命中的話，平均路徑活動應該是 $MR*W/2^S$ (如同第(1)式的第二部份)。然而沒有命中的機率為(1-HR)。舉個例子，如果 S 為零，平均路徑活動為 W。也就是說，所有的路徑應該都要被存取在快取中而沒有所提出的警示標籤。在另一個的例子，如果 S=1，W=4，則平均的路徑活動為 2+0.5HR。因此我們在每一存取之中可以節省 4-(2+0.5HR)非必要的活動，如下：

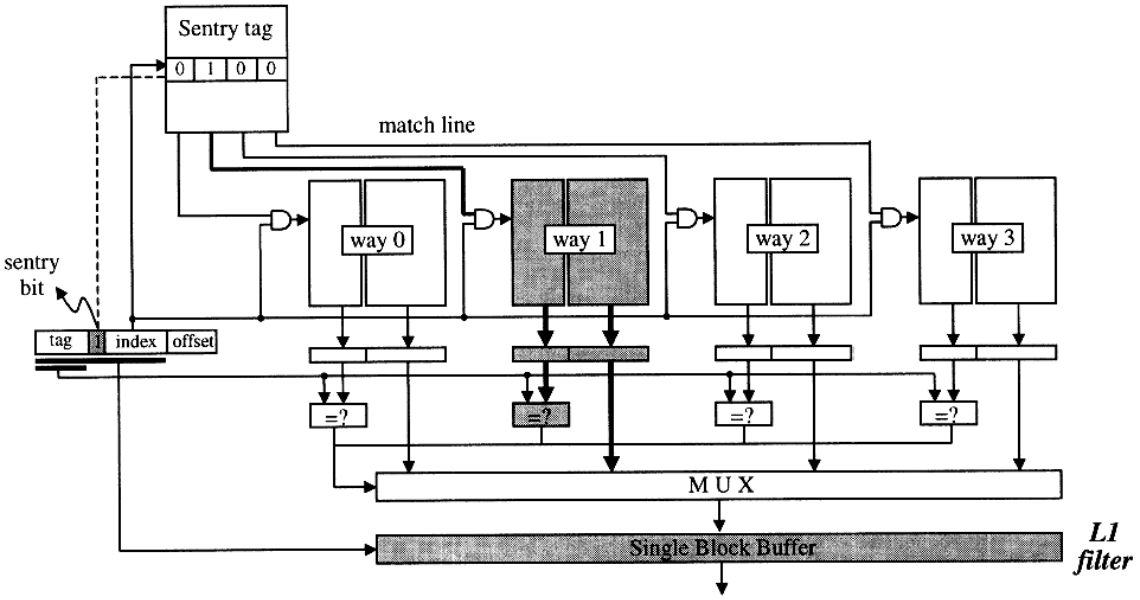
$$\begin{aligned} W_{Ave} &= HR\times(1+(W-1)\times\frac{1}{2^S})+(1-HR)\times W\times\frac{1}{2^S} \\ &= HR+\frac{HR\times W}{2^S}-\frac{HR}{2^S}+\frac{W}{2^S}-\frac{HR\times W}{2^S} \\ &= \frac{W}{2^S}+(1-\frac{1}{2^S})\times HR \end{aligned} \qquad (1)$$

我們接下來定義平均過濾率(FR)作為平均非必要路徑活動與快取路徑的比率。根據定義，平均過濾率是第(2)式。比較高的 FR 代表警示標籤在過濾非需要的路徑活動上是比較有效率的。從(2)，給定 HR 跟 W，過濾比率將會隨著警示位元的增加而增加。這證明了如果越多的位元被當作警示位元，在過濾不需要的存取路徑將會變的較準確。假設，命中率是 0.98，一個有 2 位元警示標籤四路徑的平均過濾率是 0.56。如果我們增加到 3 位元，比率會達到 0.66。在第下一章，一個更精確的平均過濾比率分析模型會經由實驗結果被驗證：

$$\begin{aligned} FR &= \frac{W-W_{Ave}}{W} \\ &= 1-\frac{\frac{W}{2^S}+(1-\frac{1}{2^S})\times HR}{W} \\ &= (1-\frac{1}{2^S})\times(1-\frac{HR}{W}) \end{aligned} \qquad (2)$$

IV. 快取架構與雙層濾器

圖四描述一個具有雙層濾器的四路集合關聯式快取記憶體。與一般的架構相比，硬體的增加包含了一個區塊緩衝器，一個警示標籤，和控制電路。我們用電晶體數量當作量測底下硬體或面積負擔的分析。

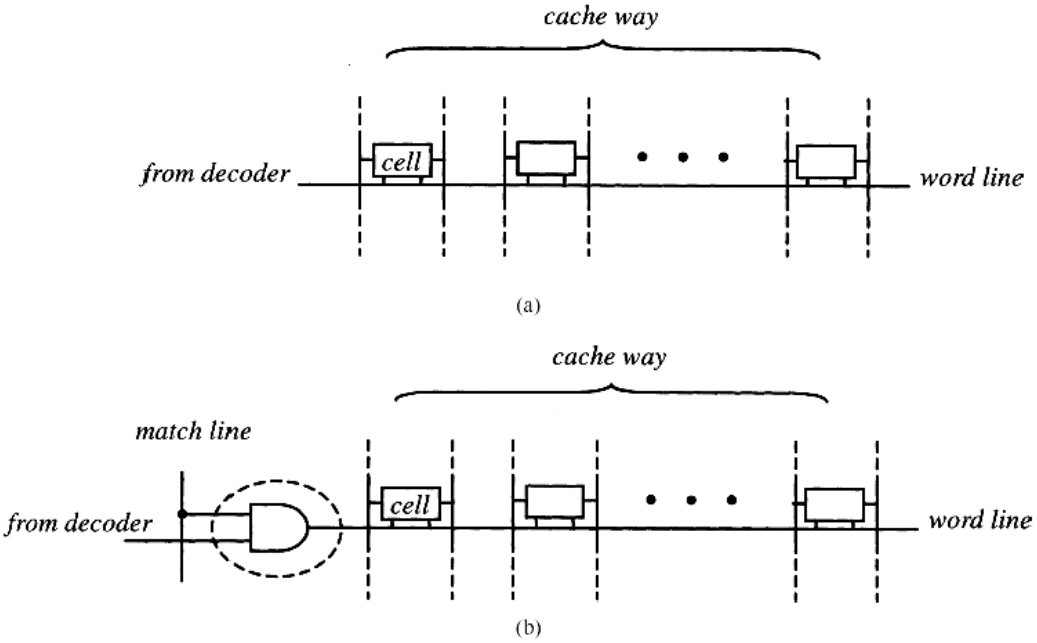


圖四、雙層濾器架構。一個具有區塊緩衝區與 1 位元警示標籤的四路集合關聯式快取記憶體架構。(灰色圖形表示正在運作的元件)

1) 在區塊緩衝器中，我們用一個 9T 的內容可定址記憶體(CAM)去實作出標籤部份(寬度是 27 bits)，資料部份則可以用 8T 拴鎖器實作出來，寬度跟區塊數量一樣(本文是 256 位元)。因此，這個區塊緩衝的面積負擔大約是(9*27)+(8*253)=2291 電晶體.

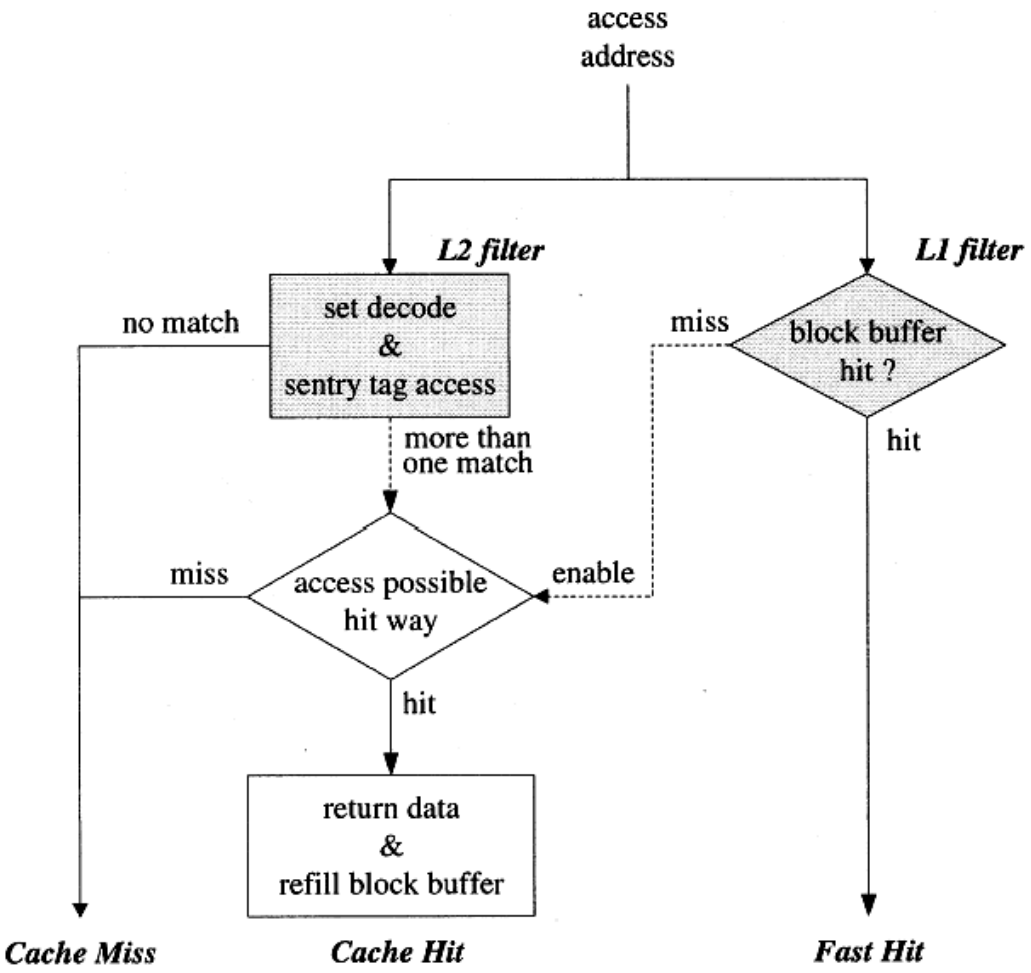
2) 我們必須把警示位元從標籤陣列移至警示標籤儲存區。為了要減少比較延遲，我們用一個 9T 的 CAM 去實作警示標籤。因此警示標籤的面積負擔是 $3*N_{ST}*N_B$ 個電晶體，其中 3 這個數值是 9T CAM 細胞與 6T SRAM 細胞的差距. N_{ST} 是警示位元數而 N_B 是快取區塊數.

3） 我們需要一個額外的控制電路去開啟或關閉快取路徑。一般的快取如圖五(a)所顯示，當字組線(word line)被插入時，快取路徑會被存取。值得注意的是，字組線是直接從解碼器衍生而來。我們可以加一個 AND 邏輯閘去控制選定的字組線存在與否。如同圖五(b)所示，當解碼線與符合線被同時啟動時，快取路徑會被存取。警示標籤的輸出是符合線，如圖四所示。AND 邏輯閘所用到的數量是 S*A，其中 S 是快取集合的數量，A 是快取的關聯性。因此，面積的負擔大約是 6*S*A 個電晶體。



圖五 (a)傳統快取 (b)具有警示標籤快取 的控制電路

對一個有 32bytes 區塊的 32KB 二路快取，在快取中的標籤與資料陣列面積大約為(1024*18*6)+(1024*256*6)=1683456 個電晶體。1024 指的是區塊數目。如果警示位元數目是 3 個位元，根據上面的面積分析(a)-(c)，面積額外負擔在我們的雙層濾器規劃大約是 2291+(3*3*1024)+(6*512*2)=17651 電晶體。因為額外負擔只佔差不多 1%的快取面積，所以是可以忽略的。用雙濾器架構作單一路徑的快取存取如圖六所示，我們在底下加以說明：



圖六、具備雙層濾器機制的快取架構之存取流程

Step 1)**存取位置被同時傳給** L1 **跟** L2 **濾器。我們用** L1 **濾器來檢查是否所需要的資料仍然存在於區塊暫存器。在同一時間，集合的解碼與警示位元的比較會依照次序在第二層濾器被完成。這裡為了要減少在過濾掉非必須的快取存取還有路徑活動的時間拖延所造成的負擔，我們將** L1 **濾器與** L2 **濾器重疊。**

Step2)Case1**：如果** L1 **濾器命中的話，這是一個快速的命中。我們可以跳過這個快取存取，並且可以直接從區塊暫存器讀出所需要的資料。**Case2**：萬一** L1 **沒有命中，我們必須要用在** L2 **濾器比對出來的結果去驅動所相對應的路徑來讀出一些可能存有資料區塊。在** L2 **濾器存有兩種可能：如果在** L2 **裡沒有比對符合的情**

況發生，這個存取一定是沒有命中。我們可以放棄接下來的快取存取並從低階記憶體重新載入區塊。否則第三步必須被執行。

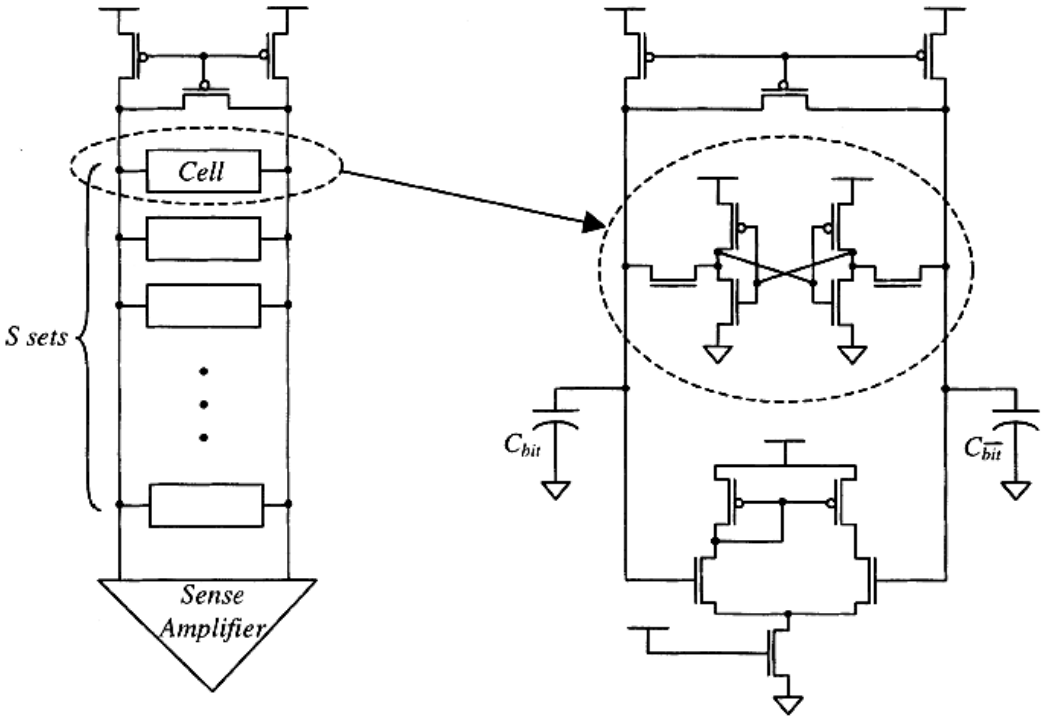
Step3)執行後續的標籤比對，就像在一般的集合關聯式快取記憶體一樣，我們並沒有平行比較存取位置的標籤跟從標籤陣列取出的 A 輸出(用一個與 A 獨立的比較器)，我們只單純檢驗那些可能存有需要字組的標籤。

與傳統的存取流程相比，我們的雙層規劃將會導致時間延遲的額外負擔，因為我們必須要在正常的存取之前過濾掉一些非必須的快取活動。詳細的分析將會在結果章節討論。不像路徑預測方法[10]，它的快取存取時間是變數，而在我們所提出的架構是固定的。在路徑預測方法中，如果預測正確的話，快取存取可以在一個時脈週期之內完成。若預測不正確，則需要多個時脈才能完成。其付出代價在最差的情況將不會減少。相反的，我們是固定的存取時間，所以可以簡化處理器的實作。

四、功率估算

在這個章節，我們對於所提出的雙層濾器架構中各部份的硬體提供了一個仔細的功率預估。為了精確量測快取中每個元件的功率消耗，我們用 0.18 微米的製程，配合 1.8V 的電壓，進行 HSPICE 的模擬。如同圖四所顯示的，有三個主要的組成硬體在我們的架構之中，亦即單一區塊緩衝器，警示標籤，及快取記憶體。因為他們彼此獨立，我們可以分開去分析。更精確一點，在快取記憶體中能量耗損可以被簡化為 $P_{Cache} \sim P_{way}*A$，其中 P_{way} 是每個快取路徑的功率消耗，A 是集合關聯程度(路徑數目)。根據[12]的結果，位元線跟放大器很明顯地在快取之中是最耗電的部份。他們占全部快取所消耗功率超過 70%。因此，為了簡化，我們只考慮位元線跟放大器.

I. 每一個快取路徑的能量消耗



圖七、欄電路

圖七(a)顯示出一個欄電路由兩條位元線組成(位元跟位元條)，S 個記憶單元跟一個放大器，S 是集合的數目。通常 S 很大，而且我們在這邊也不考慮水平分開資料陣列以取得更短的位元線。為了簡化，並非全部的記憶體單元，我們可以用相同的負載電容去估計每一欄的功率消耗。因此圖七(a)可以被更進一步簡化為圖七(b)。根據[13]，預先充電時有效的位元線負載電容，也就是 C_{bit}，為

$$C_{bit} = C_{\overline{bit}} = (S-1)\times(\tfrac{1}{2}\times C_{d,pass} + C_{bitmetal})$$

其中 S 為集合的數目。$C_{d,pass}$ 是通過電晶體汲端電容，$C_{bitmetal}$ 是在一個單一位元單位所包括的範圍中金屬線的電容。對於每一個通過電晶體的汲端電容被分割成兩部份因為它是被在兩個垂直且鄰近的單位所共享。當快取大小增加或關聯性減低，集合數 S 會增加，同時每一欄的功率消耗也會增加。對於各種的集合數，我們可以用 HSPICE 量測出每一欄的功率消耗，其結果在表一。很明顯地，讀取的功率消耗(RP_{col})比寫入的功率(WP_{col})還要稍微再多一點。這個結果可以被[14]所證實。雖然能量的差距在讀跟寫之間是少量的，為了得到更精確的結果，我們在本文中分開討論。在一般架構跟我們所提出的架構，平均每一個快取路徑所消耗的平均能量為:

$$\begin{aligned} P_{WAY\_Conv} &= [(RP_{col}\times N_{col})\times RR] + [(WP_{col}\times N_{col})\times WR] \\ &= [RP_{col}\times(T + B\times 8)\times RR] + [WP_{col}\times(T + B\times 8)\times WR] \end{aligned} \quad (3)$$

$$\begin{aligned} P_{WAY\_2L} &= [(RP_{col}\times N_{col})\times RR] + [(WP_{col}\times N_{col})\times WR] \\ &= [RP_{col}\times((T - N_{ST}) + B\times 8)\times RR] + [WP_{col}\times((T - N_{ST}) + B\times 8)\times WR] \end{aligned} \quad (4)$$

T 跟 N_{ST} 是標籤位元與警示位元的數量，而 B 是區塊大小。注意到欄的大小(N_{col})包含兩個部份，也就是標籤跟資料。RP_{col} 跟 WP_{col} 是每一欄讀跟寫所消耗的能量。RR 是讀取動作對所有存取次數的比例。WR 是寫入動作對所有

存取次數的比例。在指令快取(IC)，RR 對 WR 的比率是 1:0(所有的快取存取是讀取)。但是在資料快取(DC)，RR 對 WR 的比例大約是 2:1。事實上，這兩個能量方程式之間的差距是可以忽略的，當 N_{ST} 很小的時候。

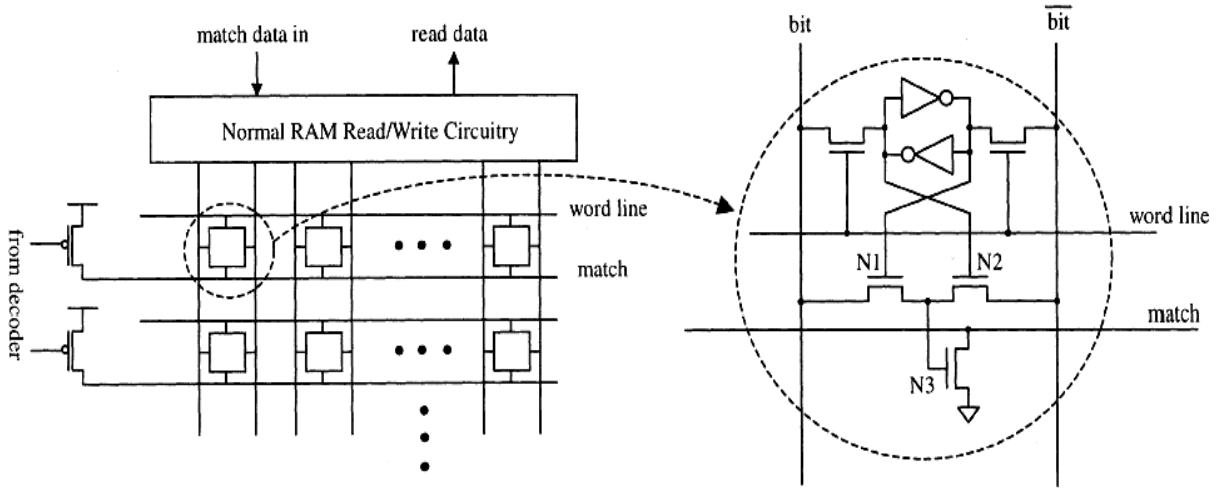
表一、各種集合尺寸之欄功率消耗

S	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}
RP_{col}(mW)	0.2233	0.2377	0.2246	0.2254	0.2286	0.2338	0.2454	0.2702	0.3298	0.4385	0.7025	1.3135
WP_{col}(mW)	0.2144	0.2148	0.2133	0.2119	0.2126	0.2174	0.2258	0.2459	0.3002	0.3946	0.6323	1.1821

II. 區塊暫存器的功率耗損

在我們提出的架構中，使用 L1 與 L2 濾器會產生額外的功率消耗。我們首先分析在 L1 濾器之中的功率消耗，P_{L1}。事實上，P_{L1} 包含了因比較而消耗的功率以及資料輸出的功率。而這些值可以從 HSPICE 的模擬當中測量出為 0.6 與 7.75mw，所以 P_{L1} 為 8.35mw.

III. 警示標籤與控制電路的功率消耗



圖八、警示標籤架構。

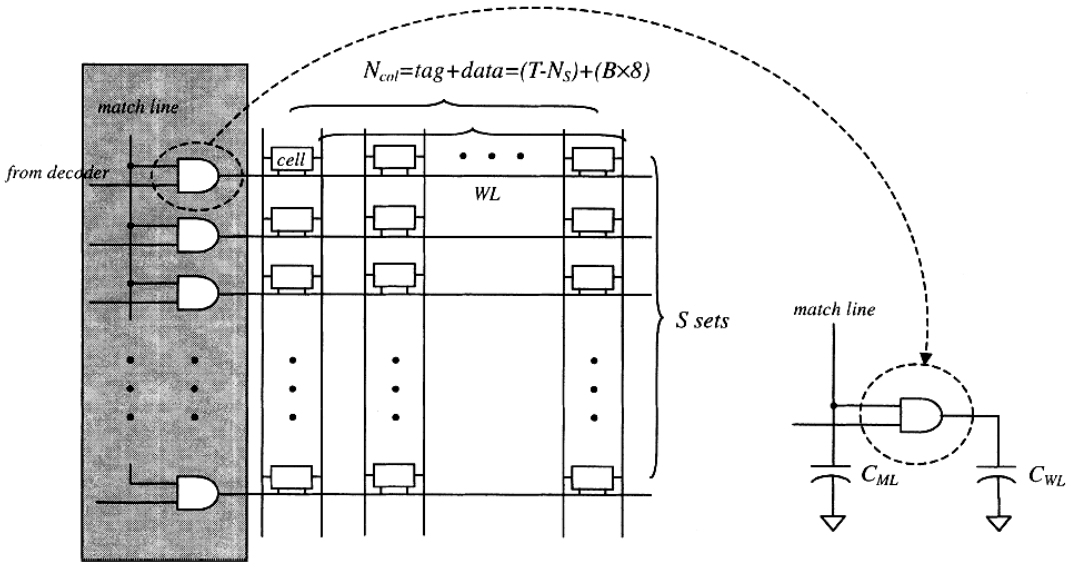
在 L2 過濾中的功率消耗，P_{L2}，因為在我們所提出的規劃當中警示位元的比較是非常重要的，因此我們用 CAM 去實作出警示標籤。一個典型的 CMOS CAM 記憶體單元如圖八所示。一個比對的運算是指把資料放在位元線之後來做比對，而不是宣告字組線。如果它們不相等，比對線將會被 N_3 放電。否則，它將會在預先充電的階段，也就是高電位。如圖八所示，比對線一直是高位元若且為若在所有的單位中發生了一個"比對相符"。

注意到比對訊號是用來觸發這個警示位元所對應到的快取路徑並且使他啟動成可以被存取。這些在 L2 濾器額外的控制電路也產生了功率消耗。我們必須一起考慮警示標籤還有控制邏輯電路。圖九(a)顯示出在我們架構中的控制邏輯電路。很顯然的在控制電路中大部分的功率消耗是字組線(WL)還有比對線(ML)。字組線電容(C_{WL})大約等於在每一列中每一個記憶體單元的閘電容總和，而比對線電容(C_{ML})等於每一欄中的 AND 邏輯閘的閘極電容總和。因此在圖九(a)可以被簡化成圖九(b)。藉由快取的設定，我們可以計算出電容 C_{WL} 與 C_{ML}，並且估計有控制電路的警示標籤之功率消耗。

對於每一個路徑，一個有控制電路的警示標籤的功率消耗可以整理如表二。在這個模擬當中，我們用了 32KB 雙路徑快取的基礎線，還有警示位元的大小是從 1 到 8。因此，所有的警示標籤的功率消耗(P_{ST})在一個單路徑快取為 $P_{ST}=P_{ST\_1}*A$。

表二、包含控制電路的警示標籤功率消耗

N_{ST}	1	2	3	4	5	6	7	8
P_{ST_1}(mW)	0.8347	0.8378	0.8444	0.8498	0.8552	0.8606	0.8659	0.8716



圖九、警示標籤中的控制電路。

五、結果與討論

在整個計畫中，我們用 SimpleScalar[16]**去模擬** SPEC2000 **測試程式。為了得到較佳的** CPU **密集和** memory **密集的混和負載，我們隨機選取** 8 **個** CINT2000 **和** 4 **個** CFP2000 **測試程式。表** 3 **提供這些測試程式的描述，並且也顯示對每一個負載所需的指令數和資料數。**

表三、測試程式描述。

Category	Benchmark	Description	Instr. Count	Data Count
CINT2000	164.gzip	Compression	81641529094	26630126869
	175.vpr	FPGA Circuit Placement and Routing	214237074291	100616950911
	176.gcc	Programming Language Compiler	78423865621	31384113066
	181.mcf	Combinatorial Optimization	132862206988	67879944204
	197.parser	Word Processing	623496981528	333797230413
	253.perlbmk	PERL Programming Language	43730351658	20095105101
	255.vortex	Object-oriented Database	168619585094	88466806040
	256.bzip2	Compression	159926907421	80349377343
CFP2000	177.mesa	3-D Graphics Library	492137176762	246401727733
	179.art	Image Recognition/Neural Networks	181402289419	78280973858
	183.equake	Seismic Wave Propagation Simulation	597511951360	235108186746
	188.ammp	Computational Chemistry	1924889017223	542422636930

I. Baseline Cache Configurations

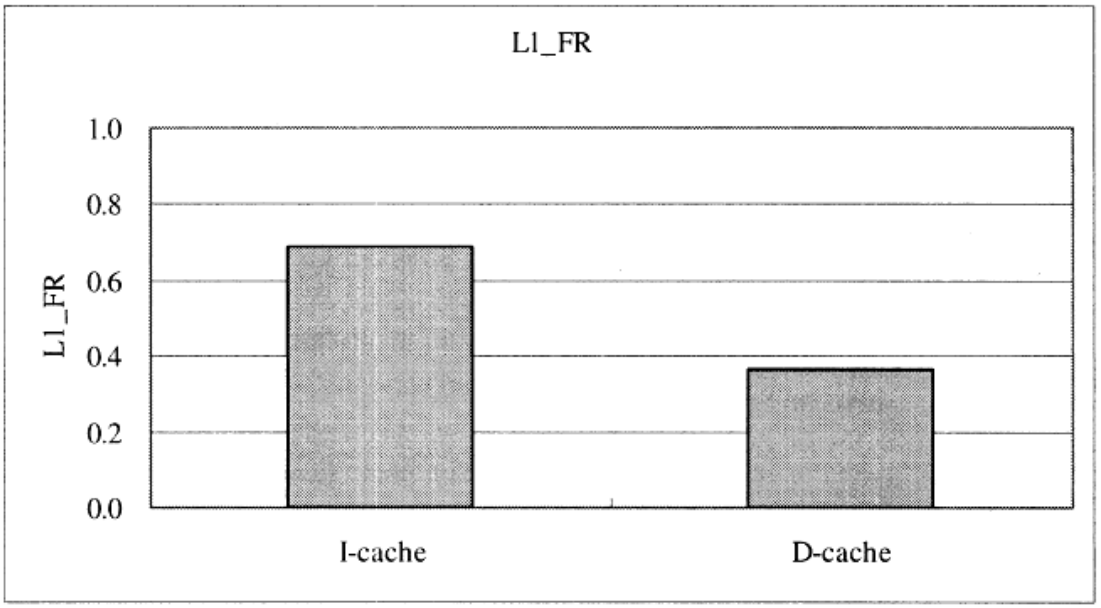
在整個計畫中，我們使用具有分離指令和 DC **的** on-chip **快取架構，其為一個** 32KB **的二路** IC **和一個** 32KB **四路** DC**。每一個快取的區塊大小是** 32KB**。為了避免結果數量的擴展，位址空間被固定在** 32-b **的寬度。**

II. Results and Discussions

在以下的討論，我們用過濾率，平均路徑活動，功率節省，存取延遲等

來比較傳統架構中與雙層式濾器方法的基礎線快取。為了有公正的比較，我們也要比較我們的架構跟只有用 L1 濾器的結果。因為 CINT2000 和 CFP2000 在模擬結果的差異很小，所以我們對於這兩組測試程式不分開討論。

L1 的過濾率：我們先定義 L1 濾器(L1_FR)的過濾率為區塊緩衝器命中數與快取存取數的比例。L1_FR 越高表示 L1 濾器在濾除非必要的快取存取越有效率.

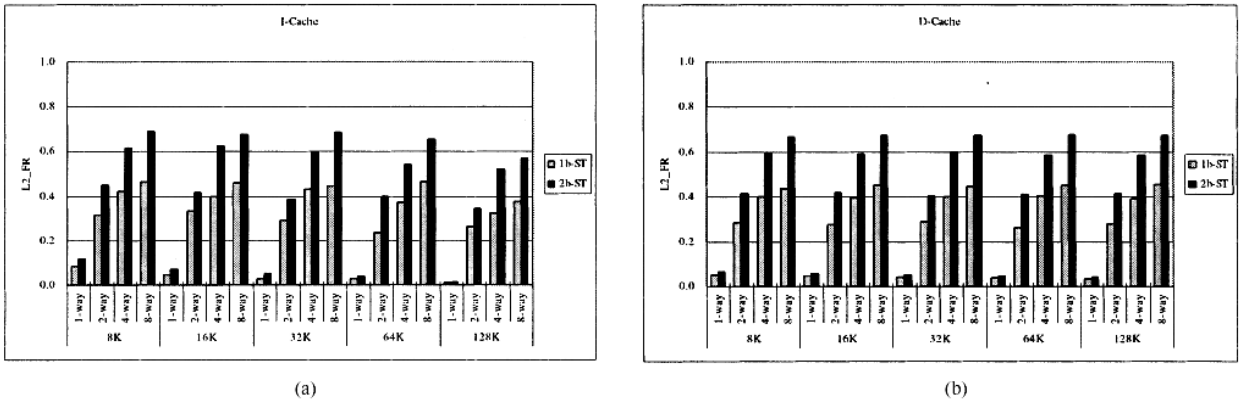


圖十、IC 與 DC 的 L1_FR 情形

圖十表示一個單一的區塊緩衝器 L1_FR 的成長。很明顯地，L1_FR 的值對不同的快取組態還是固定的。在使用了單一個區塊緩衝器上，我們可以減少大概 69%跟 37%的 IC 和 DC 的快取存取。由於區域性很低，在 L1 濾器裡用單一區塊緩衝器在 DC 比在 IC 裡不利。

L2 的過濾率：我們先定義 L2 濾器(L2_FR)的過濾率為在 L1 濾器失誤時，不必要的行動跟所有行動的比例。L2_FR 越高代表濾器在過濾不必要的動作上越有效率。圖十一顯示在 L1 過濾後 DC 和 IC 的 L2_FR。在這個模擬中，

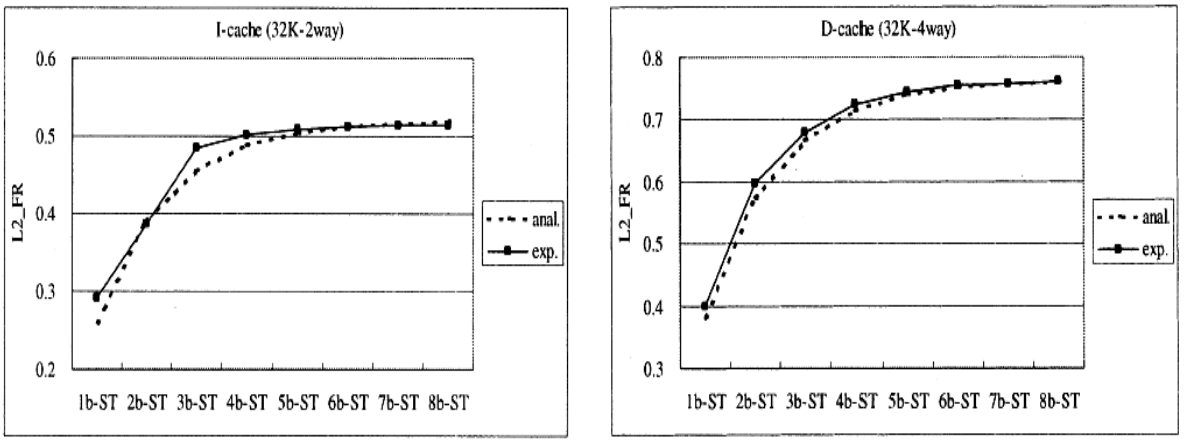
為了未來的研究我們考慮兩種不同的警示標籤組態。一個是 1-b 警示標籤，我們可以用最小比重的位元的標籤的部分當作警示標籤。(如圖 3 的 A[12:11]).



圖十一、具 1-b 和 2-b 警示標籤的 IC 與 DC 之 L2_FR (a) IC (b) DC

從圖十一我們得到一些重要的觀點。第一在這個圖中的每個情況，IC 和 DC 的 L2_FR 都跟快取的組合數成正比。這個趨勢也可以在一路快取觀察到，但是沒有那麼明顯。這是因為快取路徑越多，我們有越大的可能可以過濾掉不必要的路徑存取行為。因此，這個結果讓人想起 L2 濾器是很值得被實作在超過一個組合的快取，特別是高組合數的快取常常用在嵌入式系統。舉例來說 32-和 64-way 組合數的快取常被採用[17][18]。第二，使用 2-b 警示標籤會比用 1-b 警示標籤導致更大的 L2_FR。除了 one-way 快取，在早先的情況會導致後來的情形有 10-20%的 L2_FR 增進。這裡有一個 1-b 和 2-b 包含關係直接的結果，也就是 2-b 警示標籤命中隱含 1-b 警示標籤命中，因為前面提過的警示位元選擇的方法。

對未來研究增加在 L2_FR 的警示位元數的影響，基礎線快取被我們的雙層式濾器方法實作，L2 濾器的警示位元數從 1 到 8。圖十二表示模擬的結果。我們可以從(2)觀察實驗的結果(實心線)和解析的值(虛線)幾乎是一樣的。這證明我們所提出的解析模式的精準度是正確的.



圖十二、在多種警示位元下的 L2_FR**。實線表示實際實驗結果，虛線表示根據公式**(2)**所分析出來的值。**(a) IC (b) DC

從圖十二來看，L2-FR **會跟警示位元一起增加，即使這不是線性的。很特別地，**3-b **警示標籤在** ID **與** DC **上都是曲線的轉折點。換言之，當我們用超過** 3 **個位元的警示位元，**L2_FR **會持續的增加，但是幅度很不明顯。關鍵是使用小數目的警示位元，這樣可以很容易的過濾掉不必要的動作。**

***平均路徑活動*：我們定義平均路徑活動是在每一次存取快取時，平均可以存取的路徑數。很清楚地，傳統快取的平均路徑活動(**W_{Conv}**)是快取的關聯性。舉例來說，一個傳統四路快取的平均路徑活動是** 4**。對一個快取只有一個濾器而言，平均路徑活動的式子在**(5)**。同樣的對一個有我們雙層式濾器的快取，平均路徑活動的式子可寫成**(6)**：**

$$W_{1LF} = \text{cache associativ ity} \times (1 - L1\_FR) \qquad (5)$$

$$W_{2LF} = \text{cache associativ ity} \times (1 - L1\_FR) \times (1 - L2\_FR) \qquad (6)$$

使用上面的式子，可以得到表四的 W_{1LF} **和** W_{2LF} **的結果。我們觀察用雙層式濾器比用** L1 **濾器在減少平均路徑活動來得有效率，特別是對區域性很低的** DC **來說。舉例而言，在一個** 32KB **八路** DC**，使用雙層式濾器方法可以減少平均**

路徑活動從 8 到 1.088，但是 L1 濾器只能從 8 減少到 5.07。

表四、具有 L1 濾器與雙層濾器方法的基礎線快取實做之平均路徑活動情形

I-cache	L1_FR	L2_FR	W_{1LF}	W_{2LF}
1-way	0.688	0.060	0.312	0.293
2-way		0.484	0.624	0.322
4-way		0.670	1.249	0.412
8-way		0.784	2.497	0.540

D-cache	L1_FR	L2_FR	W_{1LF}	W_{2LF}
1-way	0.366	0.053	0.634	0.600
2-way		0.473	1.267	0.668
4-way		0.678	2.535	0.816
8-way		0.785	5.070	1.088

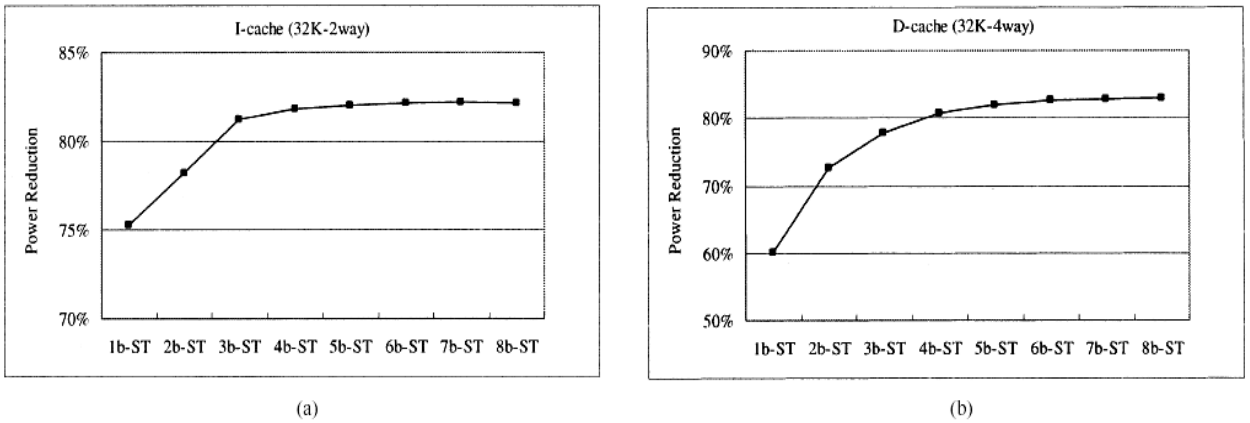
功率節省：根據在先前提到的功率消耗模式，對存取傳統快取平均功率消耗(7)和用 L1 濾器(8)還有我們提出的方法(9)可以用下列式子表示：

$$P_{Conv} = W_{Conv} \times P_{WAY\_Conv} \quad (7)$$

$$P_{1LF} = P_{L1} + (W_{1LF} \times P_{WAY\_Conv}) \quad (8)$$

$$P_{2LF} = P_{L1} + ((W_{2LF} \times P_{WAY\_2L}) + P_{ST}) \quad (9)$$

P_{WAY_Conv} 和 P_{WAY_2L} 是傳統快取和我們提出架構的一路快取的功率消耗，而 P_{ST} 是警示標籤的總消耗功率。P_{L1} 是在 L1 濾器的區塊緩衝器的消耗功率，從 HSPICE 模擬出來的值趨近於 8.35mW。用(9)，對不同警示位元數的電量減少曲線顯示在圖十三，跟圖十二的過濾率的曲線很像。這是因為警示標籤是一個很小的儲存，跟一路快取的消耗功率(P_{WAY_2L})比較，PST 的值不是很顯著。從表 2 來看，即使我們用 8-b 警示標籤，IC 和 DC 的 P_{ST} 分別為 1.7 和 3.5 mW。因此(9)可以被化簡成 $P_{2LF} \sim P_{L1}+(W_{2LF}*P_{WAY\_2L})$。所以功率減少曲線跟過濾率曲線幾乎很像。從圖十二、十三的結果，我們決定用 3-b 警示標籤去實做基礎線快取的 L2 濾器。



圖十三、多種警示位元下得功率節省情形 (a) IC (b) DC

結合(7)-(9)**跟表四的結果，用** L1 **濾器實作的基礎線快取和雙層式濾器的每次存取快取平均功率消耗顯示在表五，以毫瓦特為單位。在模擬中，我們只考慮基礎線和感測放大器的功率消耗。**

表五、具有 L1 **濾器與雙層濾器方法的基礎線快取實做之每次存取的部分功率消耗情形**

I-cache	P_{conv}	P_{1LF}	P_{2LF}
1-way	242.2	83.9	81.1
2-way	364.4	122.1	68.8
4-way	597.0	194.7	71.6
8-way	1085.0	347.0	83.2

D-cache	P_{conv}	P_{1LF}	P_{2LF}
1-way	233.9	156.6	152.2
2-way	353.3	232.2	129.7
4-way	578.8	375.2	129.8
8-way	1055.5	677.2	155.3

我們觀察到雙層式濾器的方法在過濾非必要的行為上很有效率，而且可以節省很多功率消耗。使用 L1 **濾器可以減少功率消耗，但是它不會有跟我們雙層式濾器架構有一樣的功率節省結果，特別是對區域性低的** DC **而言。**L2 **濾器用在兩階層式濾器方法裡可以很有效率去減少因** L1 **濾器失誤所造成的功率消耗。表六顯示使用雙層式濾器方法的功率消耗改進情形。表六**(a)**顯示快取功率的部分改進。為了得到全部快取的改進功率，表六**(a)**的結果必須乘上**

70%。這是因為考慮的部分元件大概佔全部消耗功率的 70%以上[12]。因此，表六(b)顯示整個快取的整體改進。

表六、使用雙層濾器方法的功率消耗改進程度一覽。基礎線 IC 為 32K 二路快取，DC 為 32K 四路快取。Conv.表示傳統快取，Conv.+L1 表示具有 L1 濾器的傳統快取。(a) 部分改進 (b) 整體改進。

Partial Improvement	I-cache	D-cache
Compared to Conv.	81.13%	77.58%
Compared to Conv.+L1	43.68%	65.41%

(a)

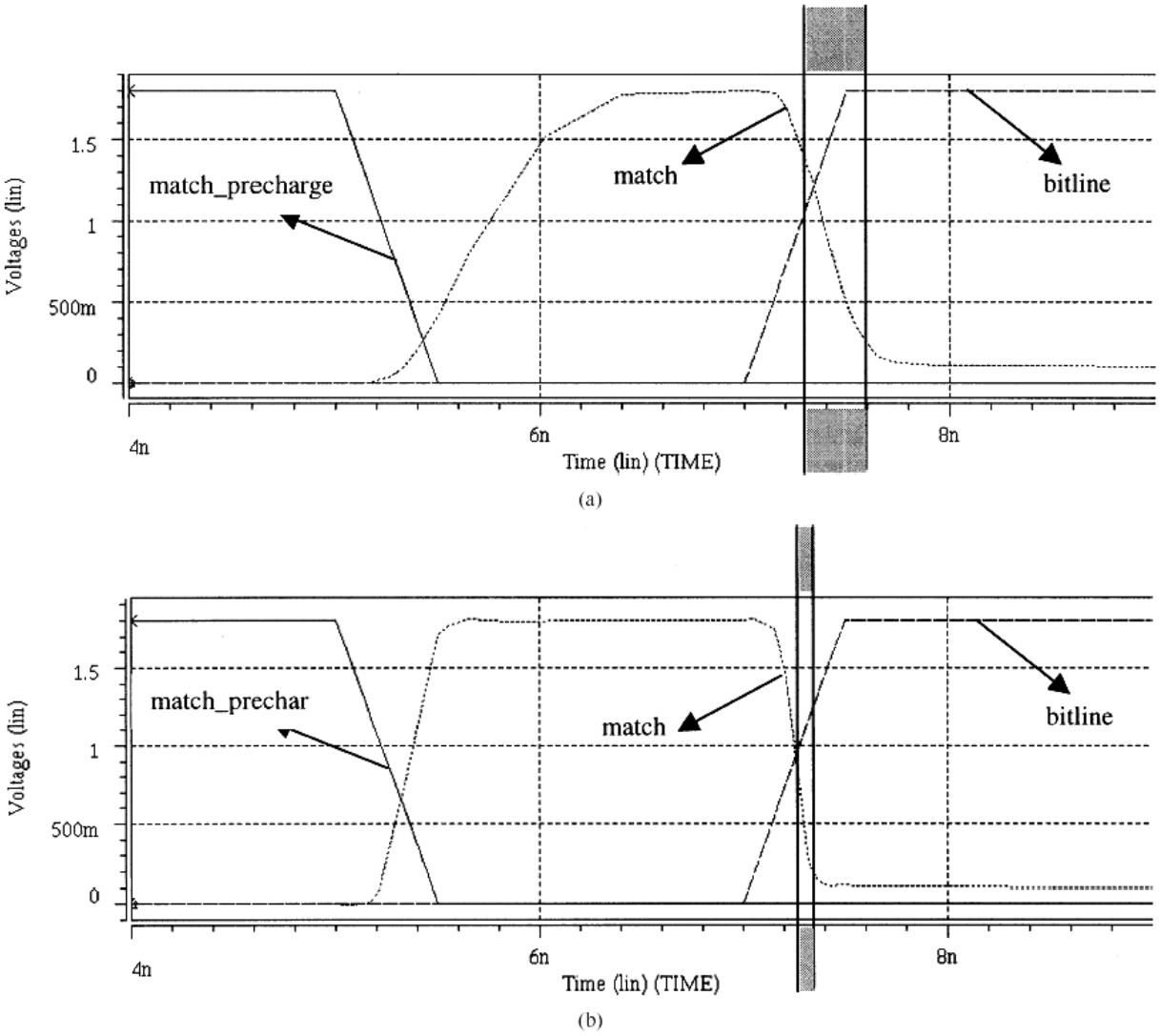
Total Improvement	I-cache	D-cache
Compared to Conv.	56.79%	54.31%
Compared to Conv.+L1	30.57%	45.79%

(b)

延遲負擔: 關於這點，我們研究提出的方法在功率消耗上的影響。另一個重要的因素是快取存取的時間。在我們的方法中，由於在正常快取前使用雙層式濾器方法會造成額外延遲負擔。雖然 L1 濾器後緊接著 L2 濾器，去縮小在過濾非必要快取存取和行動所造成的延遲，我們可以將 L1 濾器和 L2 濾器重疊，如先前描述的。我們用 L1 濾器去檢查需要的資料是否還會在區塊緩衝器內。在相同的時間後，在 L2 濾器的集合解碼和警示位元的比較也按照順序的完成。

為了去測量 L1 濾器的時間，我們採用 HSPICE 時間模擬。從圖十四(a)去決定 L1 濾器命中的時間大約是 0.3ns。而 L2 濾器的時間，因為它由集合解碼和警示位元比較組成，藉由用 CACTI[13]，我們先估計對 base IC 和 DC 時間大概是 0.35 和 0.31ns，在加上警示位元比較的時間[大約是 0.1ns, 圖十四(b)]，所以 L1 濾器的時間會被 L2 濾器的時間隱藏起來。因為在 L2 濾器的集

合解碼的時間是必要的對傳統快取和我們的方法，所以真正的延遲負擔是警示標籤的比較，最好的情況大約是 0.1ns。



圖十四、HSPICE 波形圖 (a) L1 濾器與(b)3-b 警示標籤

III. 結論

在新進製程的設計中，on-chip 快取常用來增進系統的效能。然而 on-chip 快取通常會消耗一定比例的功率。在這個計畫中，我們著重在硬體階層對節省快取電量的技術去發展。傳統集合關聯式的快取問題先前已經提過，因而我們提出一雙層式濾器的方法去減少非必要的快取行為，因此可以節省電量。提出的方法跟軟體是獨立的而且需要一些硬體的額外負擔。在 L1 濾器中, 我們用單一區塊緩衝器去減少不必要的存取快取，在 L2 濾器上，當 L1 濾器失誤時我們提出新的警示位元去減少不必要的行動。藉由 L2 濾器去存取可能命中的地方的結果，快取電量可以進一步地減少。所提出的方法在效能和消耗電量作一個取捨，與傳統的快取比較，我們的方法會導致存取快取時間增加約 0.1ns。實驗的結果顯示我們的方法比傳統的方法消耗更少的電量。因為雙層式濾器方法是依據 L1 濾器而來的。為了公平比較，我們比較傳統快取和只有使用 L1 濾器，使用雙層式濾器會導致減少大約 30%的電量消耗。同樣的對 baseline DC(32KB, four-way)，總電量減少大約有 46%。最大的電量節省是依據程式的行為，和快取的組態而來。其中雙層式濾器方法對區域性低的 DC 比較好。而且在關聯性超過 1 的快取上很值得實作，特別是那些常使用高度關聯性快取的嵌入式處理器。

參考文獻

[1] J. F. Edmondson *et al.*, "Internal organization of the Alpha 21164, a300-MHz 64-bit quad-issue CMOS RISC microprocessor," *Digital Tech. J.*, vol. 7, no. 1, pp. 119–135, 1995.

[2] J. Montanaro *et al.*, "A 160 MHz, 32 b 0.5 W CMOS RISC microprocessor," in *IEEE Int. Solid-States Circuits Conf. Dig.*, 1996, pp. 214–215.

[3] A. Hasegawa, I. Kawasaki, K. Yamada, S. Yoshioka, S. Kawasaki, and P. Biswas, "SH3: High code density, low power," *IEEE Micro*, vol. 15, pp. 11–19, Dec. 1995.

[4] H. Choi, M. K. Yim, J. Y. Lee, B. W. Yun, and Y. T. Lee, "Low-power four-way associative cache for embedded SOC design," in *Proc. 13th Annu. IEEE Int. ASIC/SOC Conf.*, 2000, pp. 231–235.

[5] J. Kin, M. Gupta, and W. H. Mangione-Smith, "The filter cache: an energy efficient memory structure," in *Proc. 30th Int. Microarchitecture Symp.*, Dec. 1997, pp. 184–193.

[6] N. Bellas, I. N. Hajj, C. D. Polychronopoulos, and G. Stamoulis, "Architectural and compiler techniques for energy reduction in high-performance microprocessors," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 317–326, June 2000.

[7] C. L. Su and A. M. Despain, "Cache design for energy efficiency," in *Proc. 28th Int. System Sciences Conf.*, 1995, pp. 306–315.

[8] K. Ghose and M. B. Kamble, "Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation," in *Proc. Int. Low Power Electronics and Design Symp.*, 1999, pp. 70–75.

[9] D. H. Albonesi, "Selective cache ways: on-demand cache resource allocation," in *Proc. 32nd Int. Microarchitecture Symp.*, 1999, pp. 248–259.

[10] K. Inoue, T. Ishihara, and K. Murakami, "Way-predicting set-associative cache for high performance and low energy consumption," in *Proc. Int. Low Power Electronics and Design Symp.*, 1999, pp. 273–275.

[11] Y. J. Chang, F.Feipei Lai, and S. J. Ruan, "An efficient two-level filter scheme for low power cache," presented at the *IEEE/ACM 11th Int. Logic and Synthesis Workshop*, New Orleans, LA, June 4–7, 2002.

[12] G. Reinman and N. Jouppi, "An integrated cache timing and power model," Compaq, Palo Alto, CA, WRL Summer Internship, 1999.

[13] S. E. Wilton and N. Jouppi, "An enhanced access and cycle time model for on-chip caches," DEC, Palo Alto, CA, WRL Res. Rep. 93/5, July 1994.

[14] P. Shivakumar and N. Jouppi, "CACTI 3.0: An integrated cache timing, power, and area model," Compaq, Palo Alto, CA, WRL Res. Rep. Feb 2001.

[15] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A QuantitativeApproach*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1995.

[16] D. C. Burger and T. M. Austin, "The SimpleScalar tool set, version 2.0," *Comput. Architecture News*, vol. 25, no. 3, pp. 13–25, June 1997.

[17] S. Santhanam *et al.*, "A low-cost 300-MHz RISC CPU with attached media processor," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1829–1839, Nov. 1998.

[18] *ARM920T Technical Reference Manual*, ARM Ltd., Cambridge, U.K., 1999.

計畫成果自評

I. 原訂計畫目標：

第一年計畫的研究核心在於發展節省功率消耗的新架構，包括快取記憶體、匯流排以及其他在資料路徑上的元件。我們將在面積成本、功率消耗以及效能觀點的選擇上來處理這些問題。除了首要減少功率消耗之外，我們還會對晶片上的快取進行面積最佳化。這裡的快取包含了兩個部分：標籤陣列與資料陣列。同樣地，從效能觀點來說，存取時間應該被盡可能地縮短來提高整個處理器的效能，其原因在於較長的存取時間通常會導致較慢的處理器時脈週期，以及較長的管線層級。我們將會分割整個快取記憶體來達到降低存取時間。

II. 研究內容與原計畫相符程度

完全符合

III. 預期目標達成情況與綜合自評

本年度之研究成果主要在於低功率快取架構上的改進，藉由所提出的雙層濾器架構，有效地減少非必要的快取存取動作，以達到降低功率消到的目的。在台大資工系賴飛羆教授教授的帶領與督導下，本研究進行的進度與原先計畫的進度完全一致，而研究成果更發表於國際知名的期刊及會議上，顯見本計畫之執行相當成功。除低功率快取架構外，本計畫亦針對低功率匯流排進行相關研究，研究成果已投稿 IEEE ICCAD 2004，唯會議審議委員之審查結果尚未發表，因此在本年度的期中報告中不特別針對此部分進行說明，待第二年之期中報告再予以詳細說明之。綜觀本年度之計畫執行情形與研究成果，非常滿意。

IV. 學術成果發表

(1) Yen-Jen Chang, Shanq-Jang Ruan, and Feipei Lai, "Design and Analysis of Low-Power Cache Using Two-level Filter Scheme," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, No. 4, pp. 568-580, August 2003.

(2) Yen-Jen Chang, Chia-Lin Yang, and Feipei Lai, "Value-conscious cache:simple technique for reducing cache access power," in Proc. of Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 16-21, Feb. 2004

附錄(一)　國際合作計畫出國人員心得報告

說明：

本計畫執行人員預計於 93 **年** 7 **月前往加拿大進行訪問，出國人員之心得報告俟完成交流訪問後，於申請出國經費時一併繳交。**

預定出訪人員：

1. **賴飛羆 教授 (計畫主持人，台大資訊工程系暨研究所教授)**
2. **蔡坤霖 (博士班研究生)**
3. **鍾玉芳 (博士班研究生)**