

# 電機系專題製作

## 主題：實現偵測執行緒並行度之管理核心

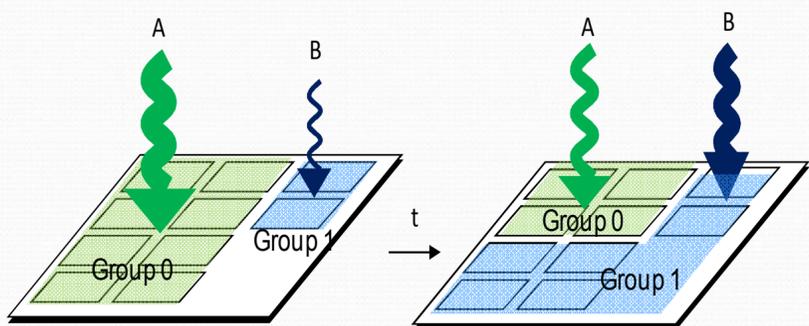
參展人員：趙姝戎

### 相關說明：

單晶片多核心處理器(Chip Multiprocessor)已成為現今處理器設計的主流。傳統單晶片多核心系統中，單晶片多核心處理器能透過內部的單一處理器核心架構來探勘指令層級並行度(Instruction Level Parallelism, ILP)，並且能透過多顆處理器的並行運算來探勘執行緒層級並行度(Thread Level Parallelism, TLP)。然而傳統單晶片多核心架構必須在硬體設計規劃之初，在高單一執行緒效能與高生產量做取舍，無法動態的調整指令層級並行度與執行緒層級並行度的探勘能力，造成了目前單晶片多核心處理器面對未來多變的應用程式類型處理上的效率不彰。

專題所提出的系統架構可在多條執行緒執行時，根據執行緒並行度(Instruction Level Parallelism, ILP)的大小，重新分配使用處理器核心的資源，以增加處理器核心的使用率。

以圖(一)為例，執行緒依照ILP的大小分配處理器資源，線條越粗表示ILP越高，分配的核心也越多。可用處理器資源總共有10顆，執行緒A的Group 0初始有8顆處理器、執行緒B的Group 1初始有2顆處理器，執行一段時間(t)後，偵測執行緒A和B的並行度，假設A下降、B上升，則釋放執行緒A的部分處理器核心，閒置的核心就能群組到Group 1使用，達到降低Group 0的核心浪費率的目的。



圖(一) 根據執行緒並行度重新分配核心動作示意圖

### 系統架構：

專題所使用的架構基礎名為超多純量(Hyperscalar)架構，此架構是一種單晶片多核心的微處理器系統架構，能動態群組多顆處理器核心為一個運算能力較高之超純量核心，而重新組態的特性讓多核心處理架構擁有高度彈性，當執行緒層級並行度低時，透過多核心共同運行而提高單一執行緒效能，反之則透過多核心獨立運作提供高生產量。為了使處理器使用效率提升，將動態偵測執行緒的ILP變化，使得系統可依照指令並行度高低來群組或釋放處理器群。

### 設計理念：

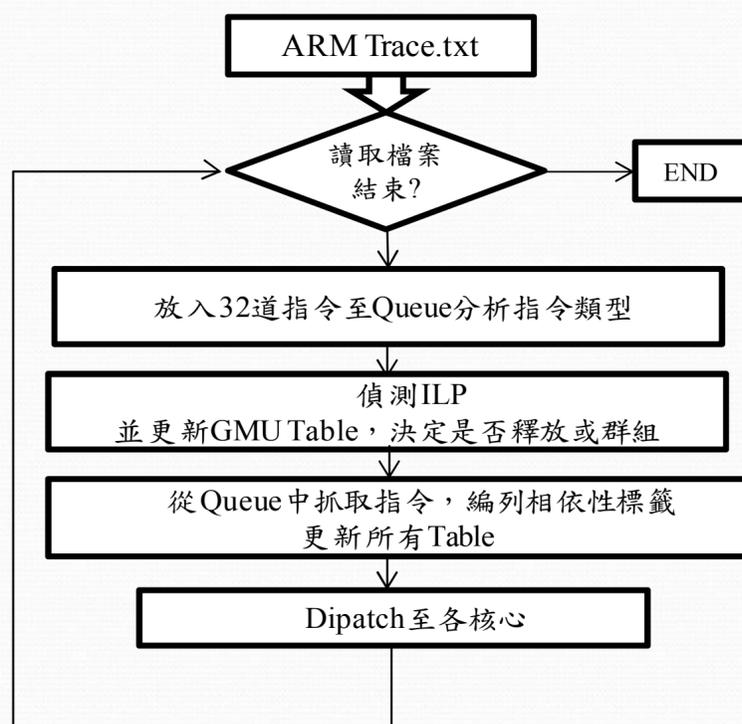
設計理念一：依據執行緒並行度來釋放或群組核心數量。

加入判斷執行緒並行度機制，並使用新增的兩道指令CRM(Core Register Move)和ReIC(Release Core)釋放處理器核心。CRM指令能將資料從要被釋放掉的核心搬移到此群組的其他核心中，以確保釋放核心後群組內的資料正確性；ReIC指令表示將此顆核心釋放，當此指令到達WB stage會發送release訊號給群組管理核心單元，告知核心已完全清空且資料全數轉移完成。群組管理核心單元派發這兩道指令後，系統即可依據執行緒並行度來釋放或群組核心數量。

設計理念二：使多核心能以分工的方式執行單一執行緒。

為了提高單一執行緒效能，必須將指令編列相依性標籤以及table，使多核心能以分工的方式執行單一執行緒，提升效能。

結合上述兩點，圖(二)為執行之流程圖。



圖(二) 模擬器架構之程式碼流程圖

### 期望達到的結果：

使用的效能評估程式是MediaBench suite，MiBench suite是著重於嵌入式系統的效能評估程式，希望在加入管理核心單元的處理核心架構會比原架構高。