智慧路燈與AI加速器研究

Research of intelligent street light & AI accelerator

Research of intelligent street light & AI accelerator

資訊三甲 洪緯宸

研究動機

- 專題目的為研究AI影像辨識加速晶片架構，增加城市的智慧化，為智慧城市做鋪墊，提供未來自駕車、科技執法、即時回報車禍系統的基礎建設。

解決問題

- 由於市面上影像辨識使用的CPU，反應時間不夠快，且功耗高，因此無法在最關鍵的時刻做出反應，防止車禍發生
- 我希望可以透過設計AI硬體加速器，讓硬體可以更快的對影像做出判斷，並做出相應的即時響應

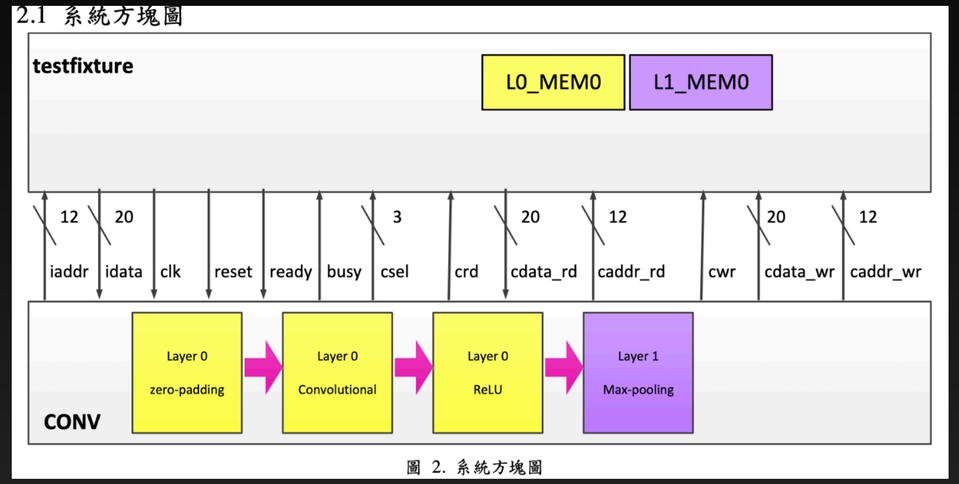
AI 硬體加速方式

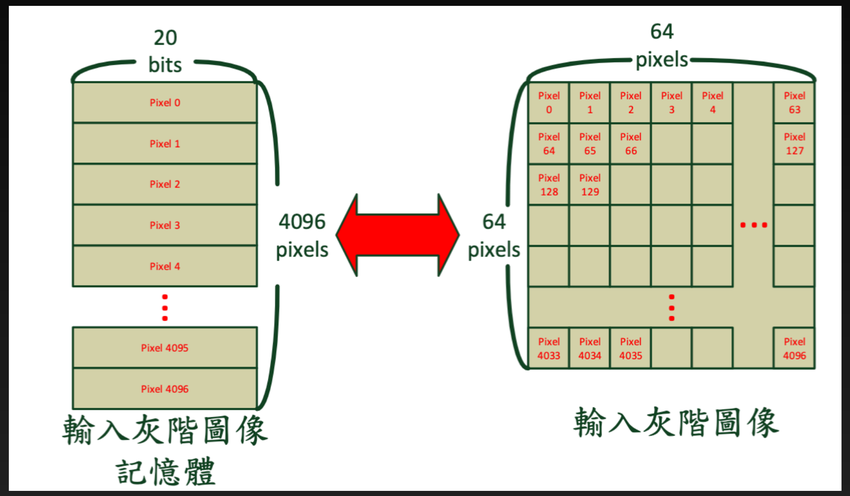
- 硬體加速：將常用到的運算直接寫成硬體做加速
- 指令擴增：針對運算的資料型態做ALU的優化
- 增加運算單元：因為影像辨識需要大量的平行處理，因此大量運算單元可以大幅增加計算的速度

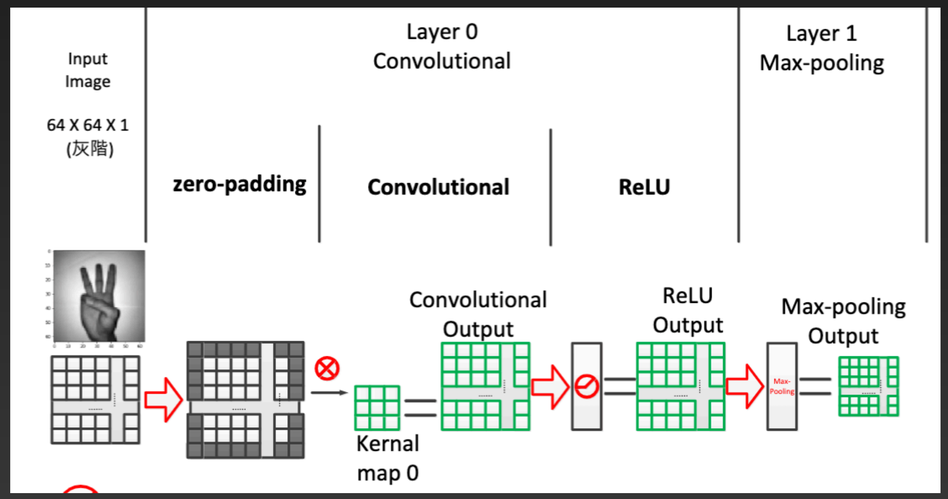
純硬體加速

- 進行複雜與重複性高的工作且要求時間短時，我們會把處理此類工作的運算核心做成硬體
- 優點：增加處理速度 降低功耗 減少指令數 可以做pipeline的客製化
- 缺點：缺乏靈活性 增加CPU面積

Convolution 硬體加速實作







面積 26029.39748

時間 3225902 NS

Rank S 面積 27000以下

指令擴增

- 現今很多AI加速器採用RISC V架構做指令的擴增，目的是將重複性高、複雜、量多的計算，在幾 個指令內處理完，對更大吞吐量，更快的計算做優化

- 優點：更大的靈活性 速度快 減少指令數目

- 缺點：需客製化 需要compiler 一起上下整合

指令擴增實例

- 以AI model常用到的 kernel convolution為例，傳統的方式要將圖像資料一個一個從memory拿出來，跟kernel相乘之後，再依全部相加，並且要重複好幾次，計算以及拿記憶體都是非常花時間的

- 參考晶心科技的vector register file，除了32個32bits的reg 以外，新增32個64bits的vector register file用來存要計算的圖像資料以及運算結果，CPU的擴增指令可以直接對vector register file 做計算及存取，減少了計算及拿記憶體時間

指令擴增實作

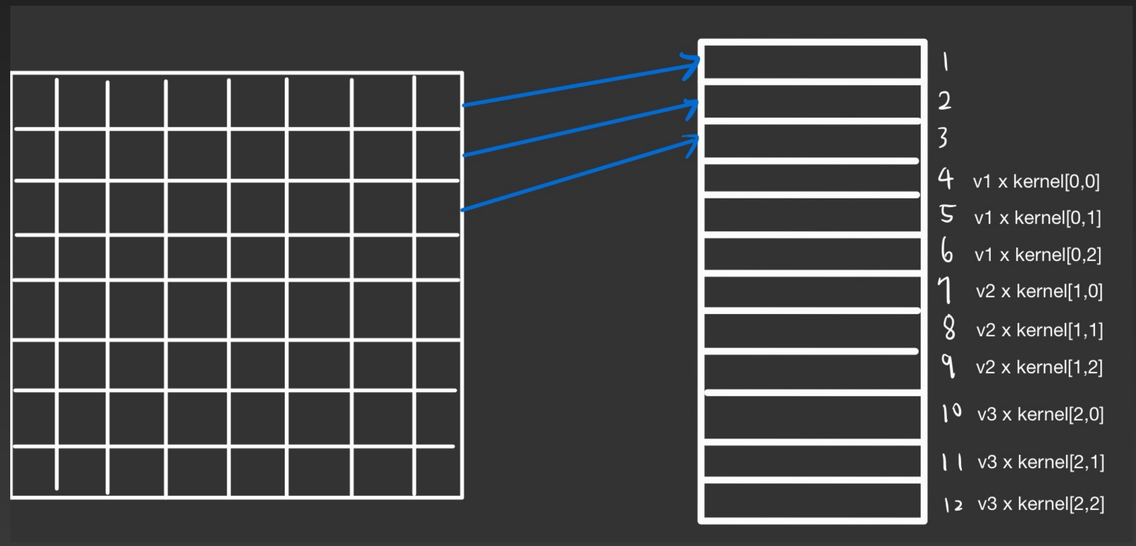
新增指令

- 新增指令：
- Vload : 將64bits memory資料load到vector register file
- vmul_vx：將vecter reg的值乘上reg file的值再存回vector register file
- Kernel_add : 將vector register file指定位置相加起來

指令擴增實作

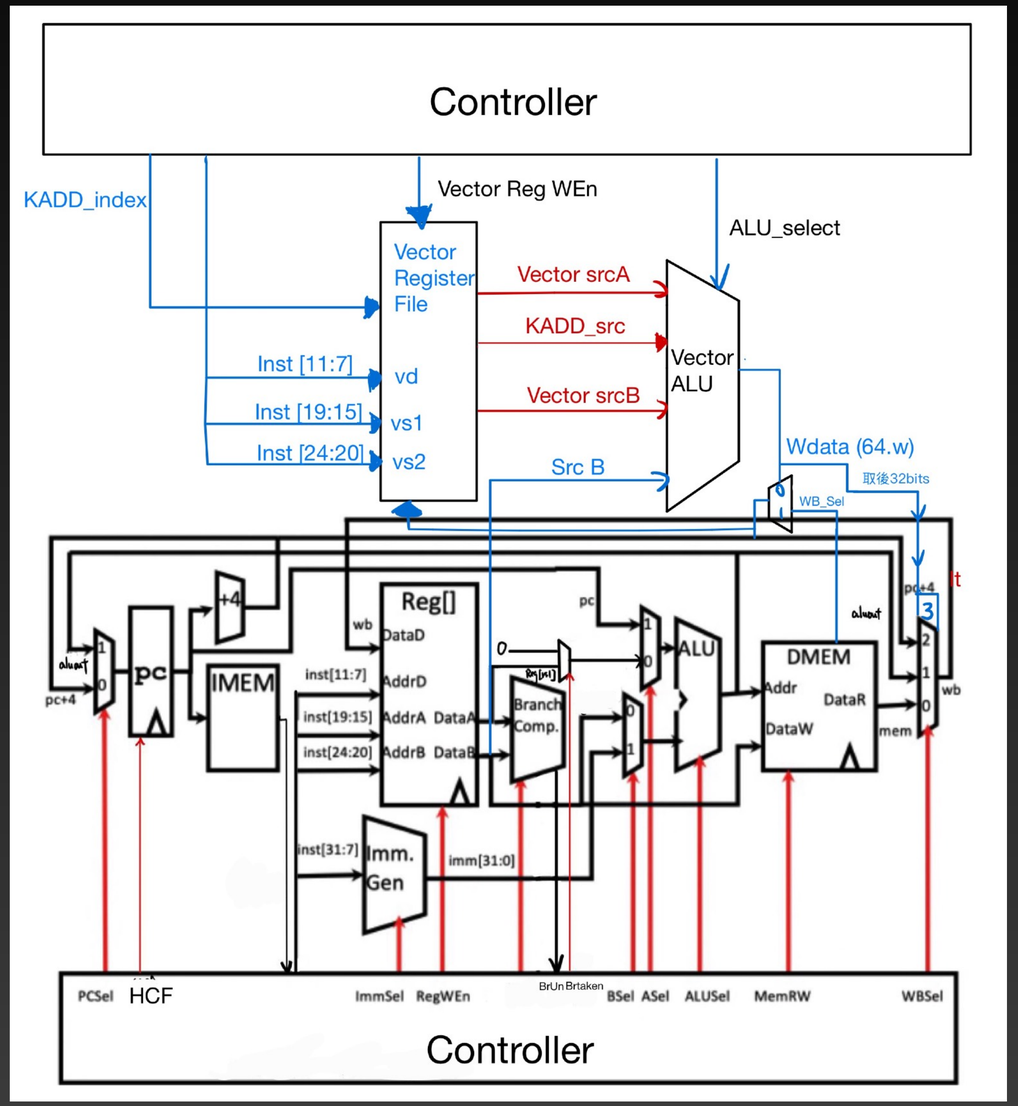
模擬層 以input 8x8 kernel 3x3為例

- 將8x8 input前三行load到vector的v1 v2 v3，再將它們分別乘上kernel，第一層乘kernel第一層，所以會有三個分別存到v4 v5 v6,
- 2 3層做一樣的事情，將第二層圖片乘上第二層kernel存到v7 v8 v9，做完總共有9個乘過kernel的vector，最後我用自己定義的指令Kernel_add將對應的vecter element加起來，連加6次就可以得到一排output kernel,再往下作5次就可以得到完整的output



指令擴增實作

硬體層



基本RISC V CPU所使用的Cycle數

Cycles:	11880
Instrs.retired:	11880
CPI:	1

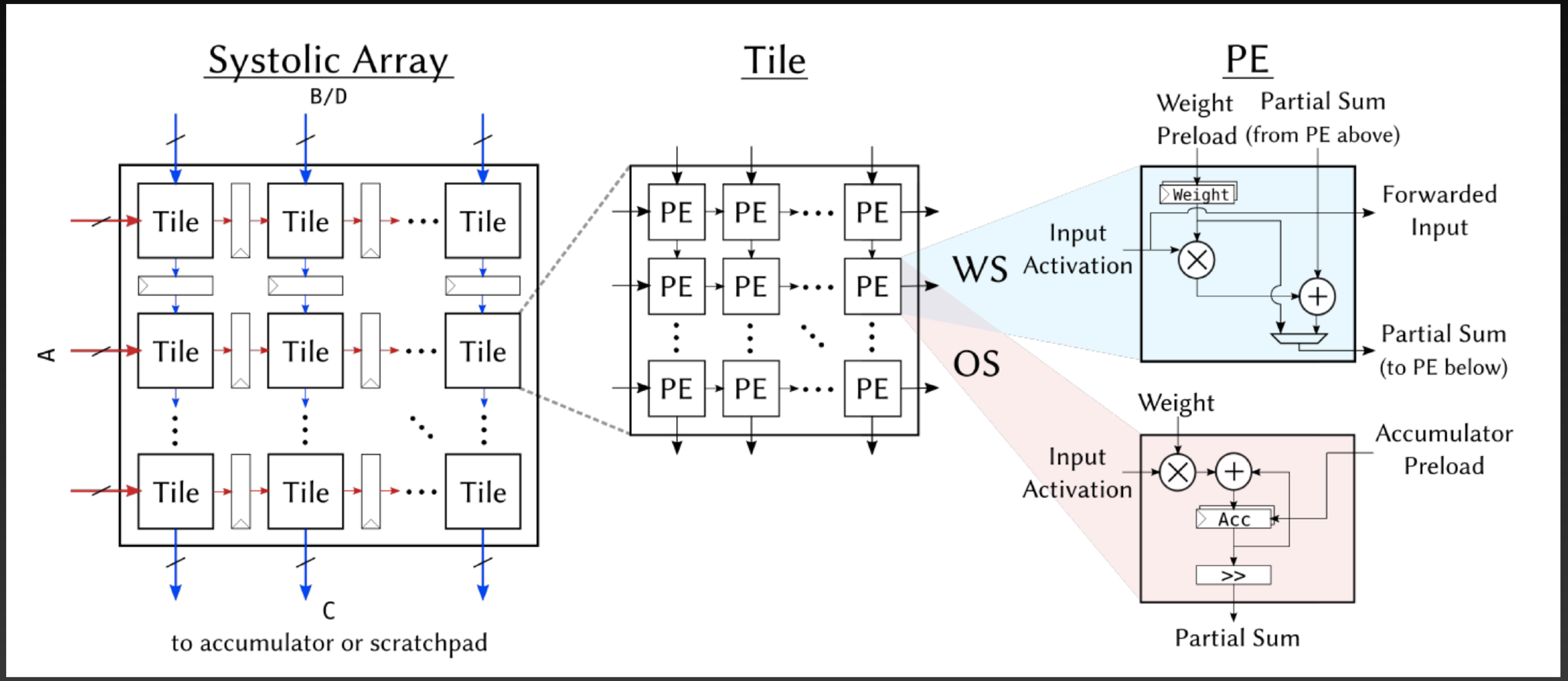
Vector extension CPU所使用的Cycle數

```
Reached Halt and Catch Fire instruction!
inst:    312 pc:    260 src line: 141
```

在相同的CPI與頻率下

Vector extended CPU比一般RISC V CPU快了38倍

增加運算單元
Systolic Array



謝謝教授的聆聽