

# Design and Implementation of Hardware Acceleration Architecture for Corner Detection Algorithm Based on PYNQ

組員：電機四 李柏葦、電機四 陳品浩

## 摘要

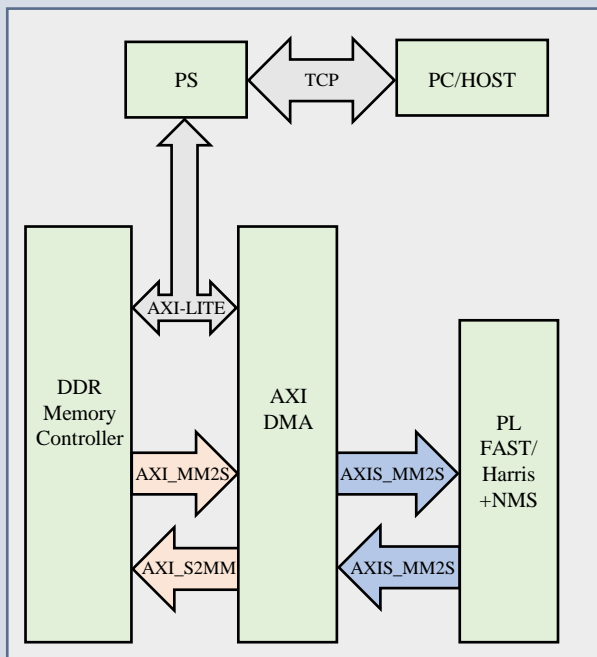
為解決 Visual SLAM 前端特徵提取的計算瓶頸，本專題利用 Xilinx PYNQ-ZU 實作硬體加速。將運算密集的角點偵測移植至 FPGA，完成後再將特徵數據回傳至 PC 進行後端運算。

我們針對不同演算法用了兩種做法：對運算密集的 Harris 演算法採用 HLS 高階合成，對邏輯密集的 FAST 演算法採用 RTL 電路設計。系統透過 PS-PL 協同與 AXI-Stream 高速傳輸，實測 Harris 達 164.7 FPS，FAST 達 131.3 FPS，顯著提升嵌入式系統的即時處理能力。

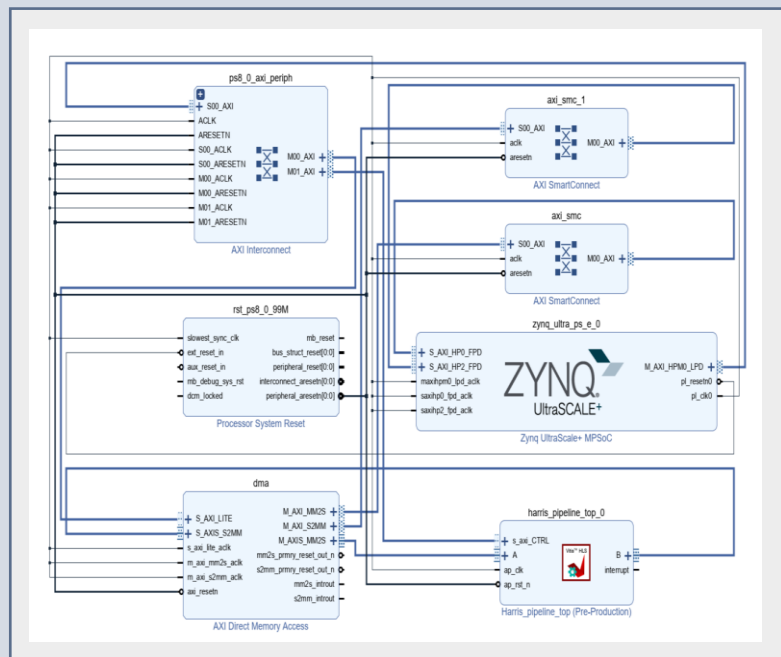
## 硬體加速器設計規格與效能比較

比較項目	Harris 角點偵測	FAST 角點偵測
演算法特性	運算密集 (矩陣運算、梯度計算)	邏輯密集 (條件判斷、決策樹)
設計方法	HLS	RTL
關鍵優化技術	Dataflow 模型、定點數優化	多級 Line Buffer、pipeline NMS
傳輸介面	AXI-Stream + DMA	AXI-Stream + VDMA
運算效能 (FPS)	164.7 FPS	131.3 FPS
加速倍率	較 RPi4 加速 3.94x	較軟體實作加速 4.4x

## 系統架構



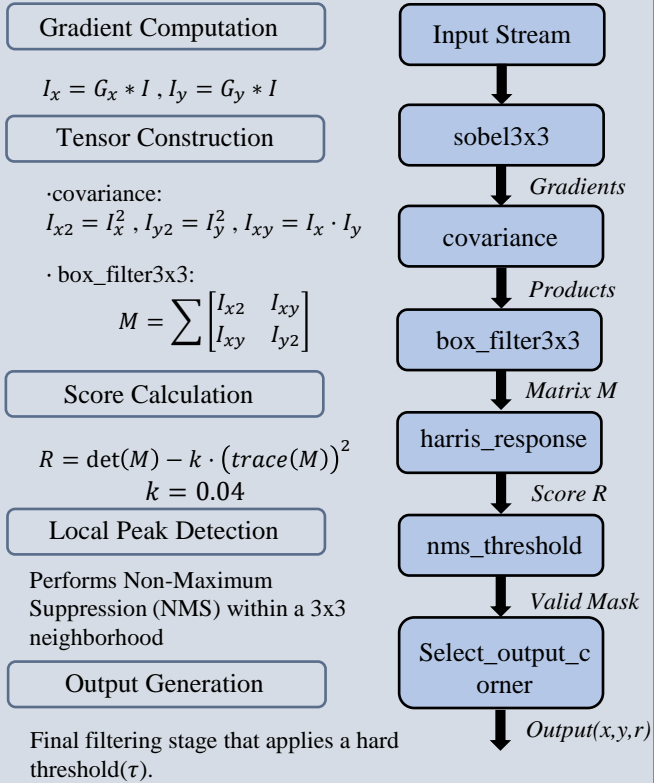
## Block Design



# 基於PYNQ的Harris角點偵測硬體加速加速器

## Design and Implementation of Harris Accelerator

### 演算法架構



### HLS 系統設計策略

1. 平行架構 (Dataflow)：於頂層採用 DATAFLOW，使梯度、張量、NMS 等模組能平行運作，最大化系統吞吐量。

```
#pragma HLS DATAFLOW
```

2. Pipeline II=1：我們設定了 II=1，讓每個 Clock 都能處理一個像素，達到全速運作。

```
#pragma HLS PIPELINE II=1
```

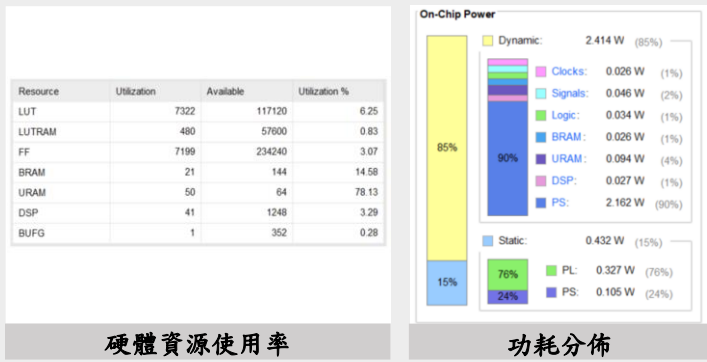
3. 記憶體優化 (Line Buffer & URAM)：為了不讓資料產生背壓 (Backpressure)，我們用 Line Buffer 暫存影像，並把 FIFO 指定存到 URAM 裡面來優化傳輸。

```
#pragma HLS BIND_STORAGE  
variable=stream type=uram
```

4. 定點數優化 (Fixed-Point)：將浮點運算轉為定點數，並使用 64-bit 累加器防止溢位，大幅降低 DSP 資源消耗。

```
typedef ap_int<64> mul_t;
```

### 資源及功耗

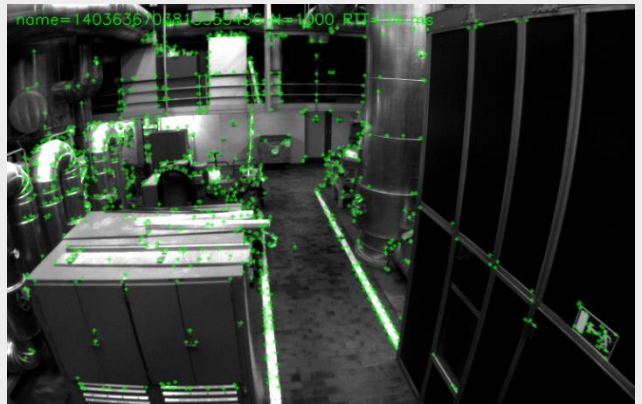


硬體資源使用率

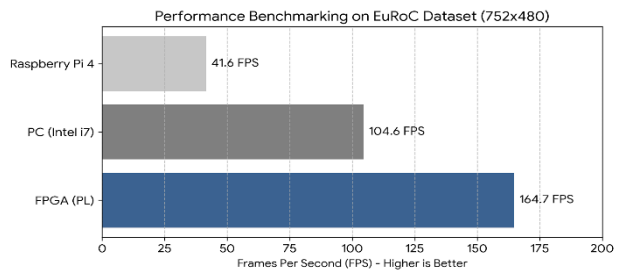
功耗分佈

### 實測結果

實測 EuRoC Dataset (752x480)。



FPGA 即時偵測畫面 (Green Points)



FPGA 加速效益分析

### 結論

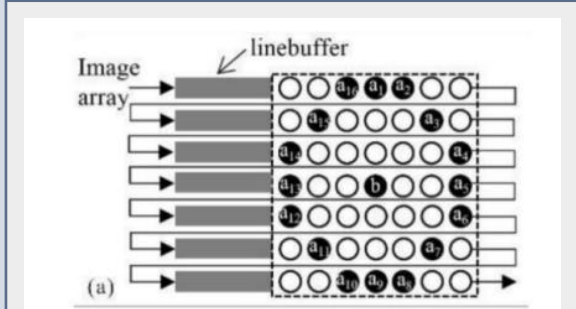
- 透過 Dataflow 架構與 Pipeline 優化，達成 164.7 FPS，較軟體實作達 3.94 倍。
- 採用定點數與 URAM 緩衝策略，維持高準確度 (F1: 0.882) 並將功耗控制於 0.327 W。

# 基於PYNQ的FAST角點偵測硬體加速加速器

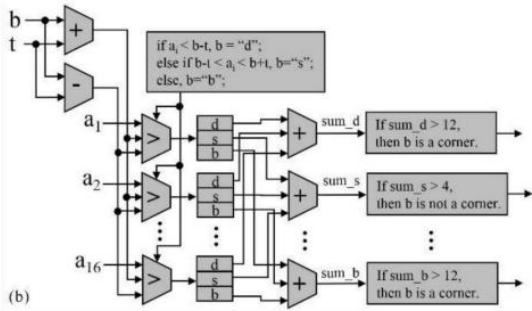
## Design and Implementation of FAST Accelerator

### FAST演算法RTL硬體設計

利用7行 Line Buffer 實現 Row-buffering。



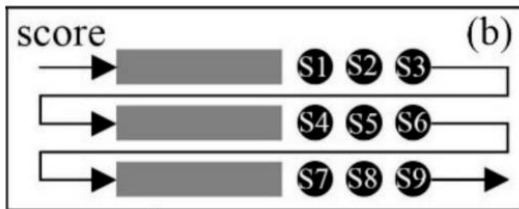
Line Buffer 視窗生成機制



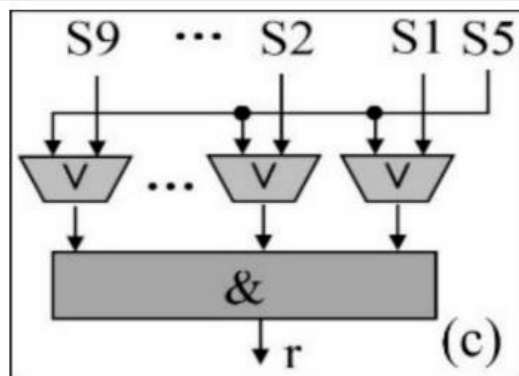
雙閾值並行比較架構

### 3x3 非極大值抑制(NMS)

利用Line Buffer以及比較器樹完成角點篩選。



NMS Line Buffer



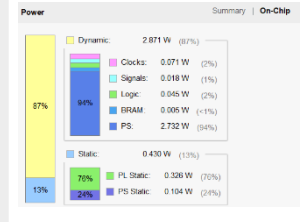
8-鄰域比較器樹 (Comparator Tree)

### 資源及功耗

Utilization Post-Synthesis Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	42498	117120	36.29
LUTRAM	1400	57600	2.43
FF	39715	234240	16.95
BRAM	6.50	144	4.51
BUFG	4	352	1.14

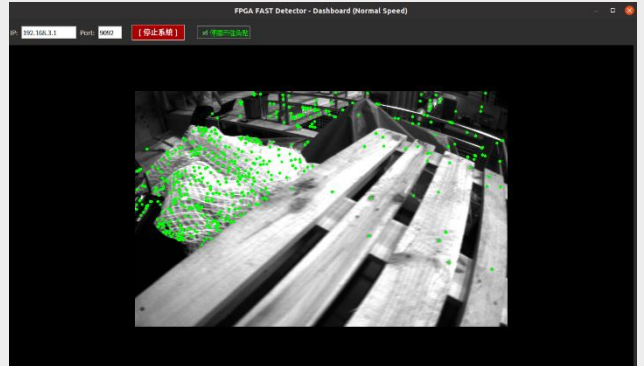
硬體資源使用率



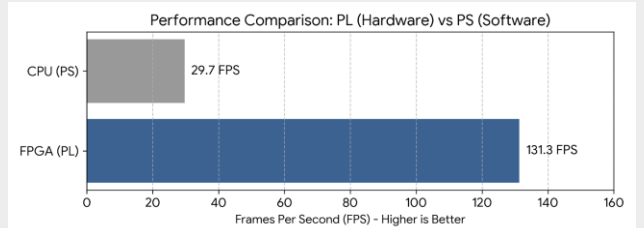
功耗分佈

### 實測結果

實測 EuRoC Dataset (752x480)。



FPGA 即時偵測畫面 (Green Points)



FPGA 加速效益分析

### 結論

- PL 端運算達 131.3 FPS，較軟體實作加速 4.4 倍，滿足 SLAM 即時需求。
- RTL 設計資源佔用低，適合嵌入式邊緣應用。
- 未來將整合 ORB-SLAM 後端至 PL，實現全硬體化。