Sustainable and Energy-Efficient in-Memory Computing Accelerator

（永續高效的記憶體內運算加速器）

指導教授：林英超
專題成員：余祥任、簡裕倉、高桉逸

開發工具：Timeloop, Accelergy, HSPICE, python 3.12
測試環境：Ubuntu 22.04

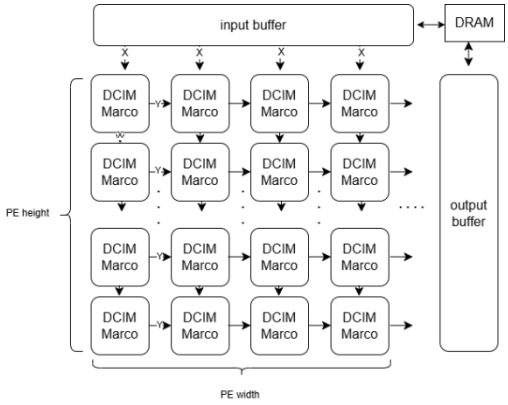
一、簡介：

隨著 AI 所需算力提升，專用運算加速器應運而生，而 AI 運算加速器中的大量數據搬移所造成的效能瓶頸日益嚴重；同時 AI 加速器在製造以及運行的過程中也會造成大量的碳排放。為了減少數據搬移與碳排放問題，我們採用數位記憶體內運算（Digital Compute-in-Memory, DCIM），將計算單元直接整合至記憶體中，**有效減少數據搬移**。並採用最佳化演算法，不斷修改硬體參數與映射策略，對**運行時間與碳排放進行最佳化**，來實現兼顧效能與低碳排放的記憶體內運算加速器設計。

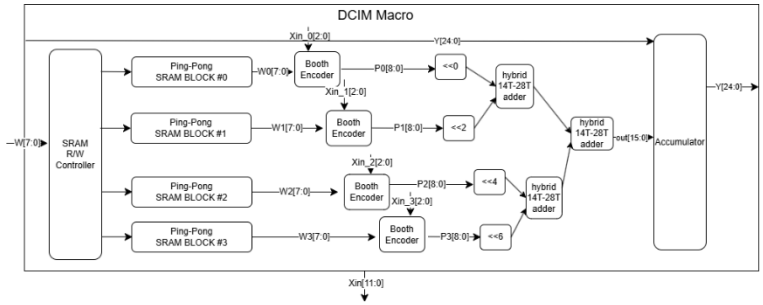
二、研究方法：

本研究設計出基於 DCIM Macro 的加速器架構（圖一），將傳統 AI 加速器中的處理單元（PE）改為 DCIM Macro，來降低數據搬移所造成的高延遲與高能耗，並結合 Ping Pong SRAM、混合 14-Transistor (14T)、28-Transistor (28T)加法器及 Booth 乘法器來優化 DCIM Macro（圖二），提升計算效能並減少晶片面積，**同時降低製造與運行過程中的碳排放**。由於 DCIM Macro 難以透過硬體描述語言進行模擬，因此我們使用 HSPICE 電路分析軟體來模擬 DCIM Macro 設計，取得 DCIM Macro（後用 PE 代稱）能耗與面積數據，提供給 Accelergy[1]統計系統的面積與能耗，之後利用文獻 [2] 所提出的 GAMMA 平台中使用到的基因演算法，協助我們在硬體架構與 AI 的神經網路之間尋找最佳的映射策略，並根據找尋到的策略，使用 Timeloop[3] 評估系統的運行時間。最後根據模擬出的總能耗，搭配文獻[4]所提出的碳排放評估模型（Architecture Carbon Modeling Tool, ACT），利用其提出的碳強度（Carbon Intensity Use）參數（表一）以及加速器運行和製造階段的碳排放公式，去推算整體的碳排放，並針對碳排放與運行時間的數據來調整硬體參數（如 PE Height 跟 PE Width）回傳給 Accelergy 與 Timeloop 進行迭代，**在效能與碳排放間達到最佳點**。

執行完之後，會產生我們所設計加速器的各個評估指標，包含 Cycles、Energy、能量延遲乘積（Energy Delay Product, EDP）和碳延遲乘積（Carbon Delay Product, CDP）等的統計結果。將神經網路各個 layer 的結果總和起來，利用 Cycles 與 Energy 轉換成的 Inference/s 與 Inference/J，與透過 ACT 估算出加速器的碳排放，一起和 Eyeriss[5]做比較。



圖一：記憶體內運算加速器系統架構圖

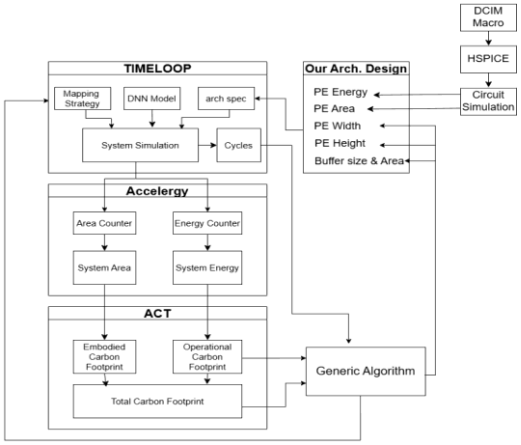


圖二： DCIM Macro 內部架構

為了去估算整體的碳排放，我們將使用以下公式:

- 運行階段碳排放之計算：$OP_{CF} = CI_{use} \times$ Energy（CI_{use} 為使用階段的碳強度、Energy 為在硬體上運行工作負載所消耗的能量）
- 單一晶片製造階段碳排放：$E_{SoC} =$ Area $\times$ Energy（Area 為硬體面積、CPA為每單位硬體面積製造所產生的碳排放）

- 每秒的推論次數: Inference/s ＝ 1/ Total Execution Time
- 每焦耳推論次數：Inference/J ＝1/ Total Energy of All Layers



圖三： 整體架構圖

Table 1: Input parameters in ACT architectural carbon model.

Parameter	Description	Range
T	App. execution time	From SW profiling
LT	HW lifetime	1-10 years
N_r	Number of ICs	From HW design
K_r	IC packaging footprint	0.15 kg CO_2
A	IC Area	From HW design (cm^2)
p	Process node	3-28 nm
MPA	Procure materials	~0.50kg CO_2 per cm^2
EPA	Fab energy	0.8-3.5 kWh per cm^2
CI_{use}	HW CO_2 intensity	30-700 g CO_2 per kWh
CI_{fab}	Fab CO_2 intensity	30-700 g CO_2 per kWh
GPA	GHG from fab	0.1-0.5 kg CO_2 per cm^2
Y	Fab yield	0-1
CPA	CO_2 from fab	0.1-0.4 kg CO_2 per cm^2
E_{DRAM}	DRAM embodied CO_2	0-0.6 kg CO_2 per GB
E_{SSD}	SSD embodied CO_2	0-0.03 kg CO_2 per GB
E_{HDD}	HDD embodied CO_2	0-0.12 kg CO_2 per GB

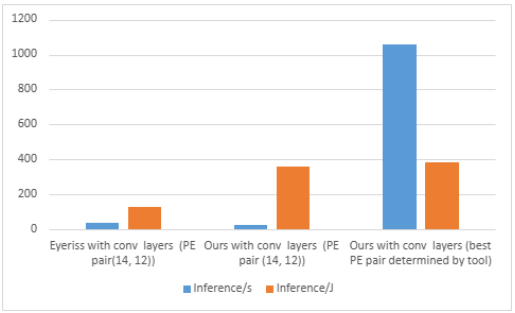
表一：文獻[4]的碳排放模型輸入參數

三、實驗結果：

在表二中，PE pair 為 PE 二維陣列的 row 與 column 的大小，可以明顯看出，在相同 PE 數量下，我們的設計比 Eyeriss 的碳排放更少，而不限制 PE 的組合，碳排放也會比 Eyeriss 少 3.6x，用 CDP、EDP 綜合考量碳排放、能耗與效能，也會比 Eyeriss 相對來的好。而圖四為在運行 Alexnet 下，我們設計的加速器與 Eyeriss 的結果比較圖，在相同 PE 下，雖然 Inference/s 我們比 Eyeriss 差 0.77x，但在 Inference/J 上，卻比 Eyeriss 好 2.87x，當我們持續提升 PE，也可以看見 Inference/s 有明顯的提升，說明我們的設計在效能方面是具有成長性的。

Accelerator	Operational CO_2 (ug) / inference	Embodied CO_2(g)/ inference	EDP (J*cycles)	CDP (g*cycles)
Eyeriss [5] (PE pair(14, 12))	1.56	50	46183	$288*10^6$
Ours (PE pair(14, 12))	0.5421	1.127	21014	$8.5*10^6$
Ours (best PE pair Selected by tool)	0.5415	13.91	1480	$7.4*10^6$

表二：各項加速器的評估指標



圖四：各項加速器運行相同 layers 的比較圖

四、參考資料：

[1] Y. N. Wu, J. S. Emer, and V. Sze, “Accelergy: An architecture-level energy estimation methodology for accelerator designs,” in ICCAD 2019

[2] S.-C. Kao et al., 2020. GAMMA: automating the HW mapping of DNN models on accelerators via genetic algorithm. In ICCAD 2020

[3] A. Parashar et al., “Timeloop: A systematic approach to DNN accelerator evaluation,” in ISPASS 2019, pp. 304–315. DOI:https://doi.org/10.1109/ISPASS.2019.00042

[4] U. Gupta et al., "ACT: Designing Sustainable Computer Systems With An Architectural Carbon Modeling Tool," in ISCA 2022, pp. 784-794, 2022

[5] Y.-H. Chen et al., “Eyeriss: An energy efficient reconfigurable accelerator for deep convolutional neural net works,” in IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, Jan. 2017