國立清華大學 電機工程學系
實作專題研究成果摘要

A 65nm In-Memory Computing Macro with Enhanced Channel-Wise Adder Tree Architecture

65奈米製程記憶體內運算巨集強化通道加法樹架構之設計與整合

專題領域：系統領域

組　　別：A525

指導教授：張孟凡

組員姓名：林柮杉、陳科錡

研究期間：113年7月1日至114年5月1日止，共10個月

Abstract

Static random-access memory (SRAM) offers fast access speed, low power consumption, and excellent compatibility with standard CMOS processes. These characteristics make it particularly suitable for emerging computing-in-memory (CIM) architectures, which aim to address the performance bottlenecks of traditional Von Neumann systems caused by frequent data transfers between memory and processing units.

In a CIM system, memory and computation are integrated into a single platform, allowing basic operations to be executed directly within the CIM macro. This reduces the need to transfer data back and forth, resulting in improved performance and lower energy consumption. By applying CIM techniques to SRAM-based designs, our project aims to explore an efficient architecture that not only accelerates data processing but also reduces system-level power usage. This approach is especially promising for applications like AI inference on edge devices, where both speed and energy efficiency are critical.

In our Special Topic, we implemented the High-bit Full-precision Multiply Cell (HFMC) architecture as described in [1], and constructed the Double-Bit 6T SRAM (DBcell) structure to support multiply-and-accumulate (MAC) operations. In our implementation, weights are delivered through the DB-SRAM using dual wordlines (WL), while inputs are applied through LBL and LBLB lines. To optimize the MAC operation, we designed and improved the channel-wise adder tree (CAT) located beneath the HFMC array. Our enhanced CAT structure improves the accumulation efficiency and overall compute performance, resulting in a better figure of merit (FoM).

摘要

靜態隨機存取記憶體(SRAM)有著高速存取、低功耗特性以及與標準 CMOS 製程的高度相容性。這些特性使其特別適合應用於記憶體內運算 (Computing-in-Memory, CIM) 架構中。CIM 架構的主要目的為解決傳統馮·諾依曼 (Von Neumann) 架構，記憶體與處理器之間頻繁資料傳輸所造成的效能瓶頸。

在 CIM 系統中，記憶體與計算功能被整合於同一平台，使得基本運算可以直接在 CIM macro 內執行，不需要頻繁搬動資料，這能夠有效提升系統效能並降低能耗。我們這次的專題便是以 SRAM 為基礎，結合 CIM 之架構，設計出一種兼具高能源效率與低功耗的運算架構，期望能夠提升資料處理速度並降低整體系統功耗。在 AI 邊緣運算等對運算速度與能源效率要求極高的應用中，這種架構展現出極大的潛力。

在本專題中，我們實作了文獻 [1] 中所提出的 High-bit Full-precision Multiply Cell (HFMC，高位元精度乘法單元) 架構，並完成了雙位元 6T SRAM (Double-Bit cell, DBcell) 的架構，以支援乘加 (MAC) 運算。在我們的設計中，權重值由 DB-SRAM 經由雙字線 (WL) 傳送，輸入資料則透過 LBL 與 LBLB 線路注入至乘法單元。為優化化乘加運算流程，我們設計並改良了位於 HFMC 陣列下方的 channel-wise adder tree (通道式加法樹)，藉此提升累加效率與整體運算效能，進而有效改善整體效能指標 (Figure of Merit, FoM)。

目錄

1. 研究背景與動機

隨著人工智慧與邊緣運算技術的蓬勃發展，系統對於即時資料處理與能源效率的需求日益提升。然而，傳統馮·諾依曼架構中，記憶體與處理器間頻繁的資料搬移，已成為系統效能與功耗的主要瓶頸。為解決此問題，近年來興起的記憶體內運算 (Computing-In-Memory, CIM) 架構提供了一種將資料儲存與運算整合的方式，有效降低資料搬移的次數，提升系統整體的運算效率與能源使用效率。

在眾多記憶體技術中，靜態隨機存取記憶體 (SRAM) 具備高速、低功耗與 CMOS 製程高度相容等特性，使其成為 CIM 應用中的理想選擇。特別是在邊緣 AI 裝置 (如語音辨識、影像處理等) 中，如何設計高效能且低功耗的 SRAM-CIM 架構，成為推動技術實用化的重要課題。

2. 研究目的

本篇本研究目標為實作出基於 65nm CMOS 製程的 SRAM-based CIM Macro，並針對乘加運算 (MAC) 核心的效能進行優化，主要目的包括：

1. 實現高位元精度乘法運算電路：
    - 採用 HFMC (High-bit Full-precision Multiply Cell) 結構，支援高效能資料乘法運算。
2. 建出雙位元 SRAM 結構 (DB-SRAM)：
    - 透過雙字線 (Dual Wordline) 設計，提升權重輸出能力並支援複雜的資料通道設計。
3. 設計並優化通道式加法樹 (Channel-wise Adder Tree)：
    - 提升乘加結果的累加效率與資料路徑的吞吐量。
4. 提升系統整體 Figure of Merit (FoM)：
    - 透過結構與電路層級的最佳化，兼顧運算效能、面積與功耗，滿足 AI 邊緣應用對能源效率與反應速度的高標準。

3. 分析與方法

3.1. Operation of CIM Macro

本篇專題所實作之 CIM Macro，目的在於降低傳統馮·諾依曼架構中記憶體與運算單元間頻繁資料搬移所造成的延遲與功耗。透過將儲存與運算功能整合於同一硬體平台中，CIM 架構可在記憶體內直接執行乘加運算 (Multiply-and-Accumulate, MAC)，提升系統效能並降低能量消耗。

本架構整體可劃分為三個核心架構，分別為：

1. 雙位元 6T SRAM (Double-Bit cell, DBcell) 陣列
2. 高位元精度乘法單元 (High-bit Full-precision Multiply Cell, HFMC)
3. 通道式加法器樹 (Channel-wise Adder Tree, CAT)

以下將說明三個架構之間的運作流程與資料是如何輸送與操作。

3.1.1. 雙位元 6T SRAM (Double-Bit 6T SRAM, DB-SRAM) 陣列

此架構為經改良過後之雙位元 6T SRAM 結構，主要用於權重儲存與接收輸入，每個儲存單元中含有兩組可獨立控制的 6T bitcell，可儲存雙位元權重，並以兩條獨立字線 (WL / RWL) 進行控制。這使得每個儲存單元可在單一操作週期內提供 1-bit 或 2-bit 權重值，支援多種位元精度的運算需求。而輸入資料 (Activation) 則透過 LBL (Local Bit Line) 與其反相 LBLB 傳入至下一級之 HFMC 運算單元。

3.1.2. 高位元精度乘法單元 (High-bit Full-precision Multiply Cell, HFMC)

DB-SRAM 所儲存的權重與透過 LBL / LBLB 傳入的輸入資料將共同進入 HFMC。每個 HFMC 支援： $2b\ Input \times 1b\ Weight$ 運算

HFMC 內部以邏輯運算 (如 NAND gate) 的方式來實現乘積計算，輸出為 partial products，接著會將此 partial products 送至下一級之 adder tree 進行後續加總。

3.1.3. 通道式加法器樹 (Channel-wise Adder Tree, CAT)

所有 HFMC 單元的輸出會傳至對應的 Channel-wise Adder Tree (CAT)。此架構可對整個輸入通道中的值進行累加，輸出端最終會產生一個對應通道的累積乘加值（MACV），供後續數位處理或神經網路層使用。

3.2. DBcell & HFMC

Double-Bit SRAM 與 HFMC 整合架構說明

為實現在記憶體中進行浮點與整數乘加操作，我們參考論文[1]中以雙位元儲存單元 (DBcell) 為基礎，結合高位元精度乘法單元 (HFMC) 的記憶體內運算 (CIM) 架構。這個架構的設計核心在於透過小面積的雙位元 SRAM cell 搭配一組乘法邏輯，達成在記憶體陣列中直接執行乘加運算。

3.2.1 DBcell 架構

DBcell 採用兩組 6T-SRAM 為基礎的位元儲存單元，以分離字線結構 (Split Wordline) 搭配控制訊號 WL, RWL 與 HWL，實現高效率的雙位元資料儲存與讀寫切換。在記憶體模式 (Memory Mode) 中，HWL 為高電位，此時資料可經由 GBL (Global Bit Line) 與 LBL (Local Bit Line) 之間的通路進行寫入；而在運算模式下，HWL 拉為低電位，切斷 GBL 與 LBL 的連結，避免資料路徑干擾。

此外，透過控制 RWL_{2N} 與 WL_{2N+1}，可選擇性的從 LBL / LBLB 上讀取 DBcell 中兩組 bitcell 儲存的位元權重 (WB_{2M}, WB_{2M+1})，實現對 2b 權重的讀取與運算，為後續 HFMC 操作提供權重。

3.2.2 HFMC 架構

HFMC (High-bit Full-precision Multiply Cell) 位於 DBcell 下方，為一組支援高精度運算的乘法單元，其功能是進行兩組 $2b\ Input \times 1b\ Weight$ 全精度乘法的運算，而其輸出結果為兩組 Partial Product (PP)，後續會將此結果傳至加法器樹進行累加。

而 HFMC 的乘法實現為純數位邏輯，他是透過：

1. 將 IN 轉換為 LBL/LBLB 的電壓訊號
2. 將儲存在 DBcell 的 2-bit W 資料讀出，並轉換為乘法控制訊號
3. 在 HFMC 中與 IN 資料進行對應的位元邏輯運算(如 OR / NOR)

每個 HFMC 單元可同時支援兩組乘法通道，對應兩組獨立輸入與權重資料，達成更高的資料處理吞吐量。

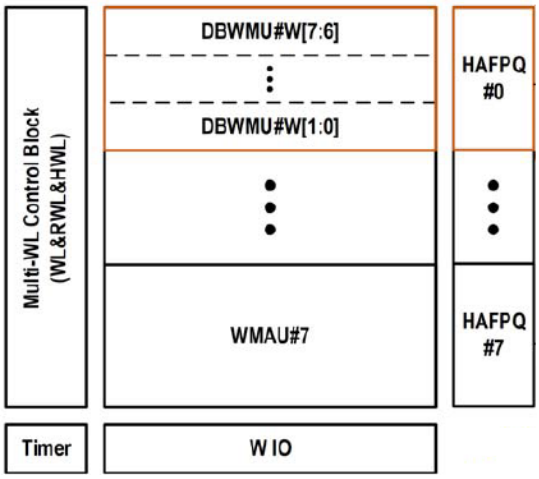

Fig. 1 參考論文提供之 CIM 架構 [1]

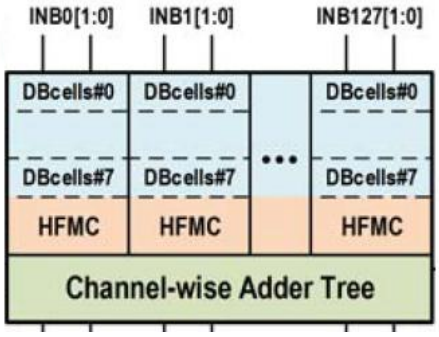

Fig. 2 參考論文之運算電路架構示意圖 [1]

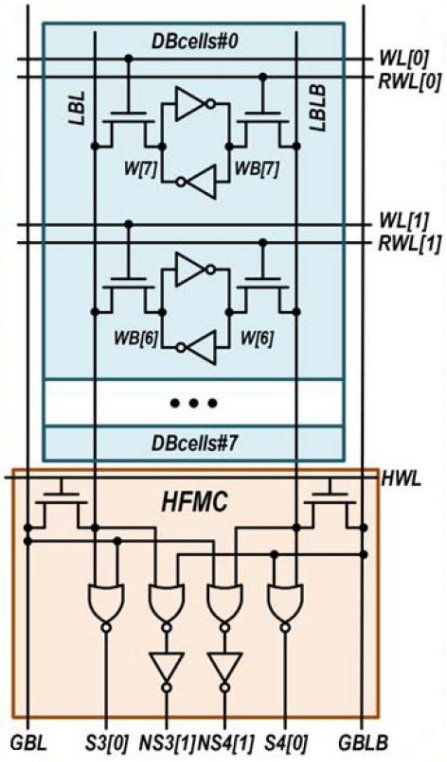

Fig. 3 DBcell & HFMC 之電路 [1]

3.3 Channel-wise Adder Tree Design and Optimization

在數位型的記憶體內運算 (Compute-in-Memory, CIM) 架構中，通道加法器樹 (Channel-wise Adder Tree, CAT) 扮演著關鍵角色，負責彙總來自多筆乘法操作的結果，特別是在處理卷積神經網路 (CNN) 時，系統通常會同時處理大量輸入與權重配對，而加法器樹正是這些資料彙總的主要瓶頸。

CAT 設計中的主要挑戰之一是其可擴展性問題。當平行通道數量增加 (例如 64 或 128 channels) 時，加總所需的階層級數會呈對數成長，而整體加法器數量則呈指數增加。這將導致面積急遽上升、功耗增加，並產生顯著的訊號延遲，尤其在進位訊號傳遞的過程，延遲增加的現象會變得更加明顯。

此外，由於加法器樹需要在每個運算週期內處理大量的部分和 (Partial Sum)，切換非常頻繁，使其成為動態功耗的來源。在某些設計中，CAT 的功耗甚至可佔據整體 CIM Macro 超過 30%。因此，如何在不犧牲訊號準確度與穩定性的前提下，降低電晶體數量、靜態與動態功耗，以及訊號延遲，成為設計優化的重點。

為了克服上述問題，加法器樹的設計需要考慮下列幾項優化策略：

- 選擇不同的加法器架構，以取得在功耗、延遲與面積之間最佳化的折衷；
- 採用交錯式混合加法器架構 (如 14T/28T 或 PTL/28T 混用) 來提升整體能效；

3.4 Comparison of Adders: 28T, 14T, and PTL

我提升記憶體內運算系統中加法器陣列的能效與面積效率，本專題針對三種全加法器架構進行深入探討與模擬比較並改良，分別為 28T (傳統 CMOS 靜態邏輯加法器)、14T (採用 GDI 技術的簡化架構) 與 PTL (傳輸閘邏輯)。這些架構各有優劣，適用於不同的設計需求與使用場景。

3.4.1 28T FULL ADDER

28T 全加法器是目前標準邏輯庫中廣泛使用的設計之一，其主要優勢在於穩定性高、邏輯準確性佳。該架構採用傳統靜態 CMOS 邏輯閘構成，具備完整邏輯擺幅與良好的抗雜訊能力。其中最大的特點在於具備強大的輸出驅動能力，即使在不同輸入組合下，輸出仍可穩定達到 VDD 或 VSS，確保訊號在傳遞過程中維持穩定電位，適合應用於加法器樹的後段關鍵層級，以維持整體電路穩定。

然而，其主要缺點為 面積大與功耗高。由於 28T 架構使用全補邏輯設計 (complementary CMOS)，需要更多電晶體數目來實現功能，因此在晶片佈局上佔用更多空間，且靜態功耗相對偏高。儘管如此，若應用情境對穩定性與驅動力要求高，28T 架構仍是具可靠性的選擇。

3.4.2 14 T FULL ADDER

14T 全加法器是一種透過 Gate Diffusion Input (GDI) 技術所實現的輕量化加法器架構，目的是在不影響功能的情況下降低所需電晶體數目，達到減少面積與功耗的目標。由於使用 GDI 而非傳統 CMOS 邏輯閘，14T 架構的佈局更為緊湊，動態功耗也相對降低。

但該架構存在幾個潛在缺點。首先是閾值電壓降 (threshold voltage drop) 問題。由於部分內部節點在運作過程中無法達到完整電壓擺幅，常卡在 $V_{DD} - V_{thn}$ 或 $V_{SS} + V_{thp}$，導致邏輯不穩定，甚至使後段緩衝器出現短路電流，造成靜態功耗增加。

其次是部分邏輯路徑驅動能力不足，特別是在輸出端經過多顆串接的 MOSFET 時，會造成臨界路徑延遲增加。例如，當所有輸入皆為 0 時，輸出 Sum 可能透過一條偏弱的導通路徑被驅動，在大型加法器樹中會特別明顯地影響延遲表現。

儘管如此，14T 架構仍適合應用於非關鍵運算路徑，例如加法器樹的前段或重複性高的區塊，在不影響邏輯正確的前提下達到節省功耗與面積的目的。

3.4.3 PTL FULL ADDER

PTL 全加法器透過傳輸閘 (transmission gate) 邏輯實現，進一步簡化邏輯結構並追求更好的功耗與延遲表現。本專題所採用的 PTL 架構中，我們使用並聯式 XOR 閘來同時產生 XOR 與 XNOR，省去傳統設計中所需的反向器，從而降低電路複雜度與靜態功耗。同時，透過將原本集中於“$A \oplus B$”節點的負載重新分配至“$A \oplus B$”與“A·B”兩條平行路徑，使得每條路徑的寄生電容顯著下降，進一步縮短了最壞情況延遲。然而，該設計也存在某些電壓擺幅不完整的問題，進而導致後續電路產生不穩定行為。

3.5 Interleaved Hybrid Adder Design Configuration

在 Full Adder 設計中，面積、穩定性、功耗與延遲之間存在著基本的取捨。以 GDI (Gate Diffusion Input) 邏輯實現的 8T 加法器就是此類權衡的代表之一。相較於傳統的 28T CMOS 加法器，8T 架構可大幅減少電晶體數量與面積，並降低動態功耗，但同時也面臨擺幅不足、驅動能力不足與閾值電壓降 (threshold drop) 等問題。

為改善上述缺點，改良版的 14T 加法器採用了 全擺幅 GDI (Full-Swing GDI, FS-GDI) 結構，有效解決 GDI 原始電路中存在的擺幅問題與穩定性不足，雖然因此略微增加了面積，但相較於 8T 加法器而言，14T 在保持面積小型化的同時，大幅提升了邏輯穩定性與靜態功耗表現，並仍保有相較 28T 架構更小的硬體資源需求。

然而，GDI 結構的問題之一臨界路徑驅動力不足，在加法器樹中若連續串接多個 14T 加法器，仍會造成訊號衰減與延遲加劇的問題。為此，我們在本研究中導入了一種 交錯式混合加法器樹架構 (Interleaved Hybrid Adder Tree)。

如 Fig. 6 所示，此架構於加法器樹中交錯配置 28T 與 14T 加法器。透過將具有強驅動力的 28T 加法器與 14T 加法器交錯排列，可補償 GDI 結構在輸出端的不足，避免不完整邏輯電平在多級串接中持續惡化。這種配置有效打斷訊號衰減鏈 (degradation chain)，大幅提升整體運算的穩定性與正確性。

雖然與純 14T 加法器樹相比，交錯式結構會略為增加面積，但與全 28T 架構相較仍具顯著面積優勢，同時能維持相近的延遲表現與訊號品質。因此，該交錯式混合架

構在功耗、面積與效能三者之間取得了最具平衡性的設計折衷方案。

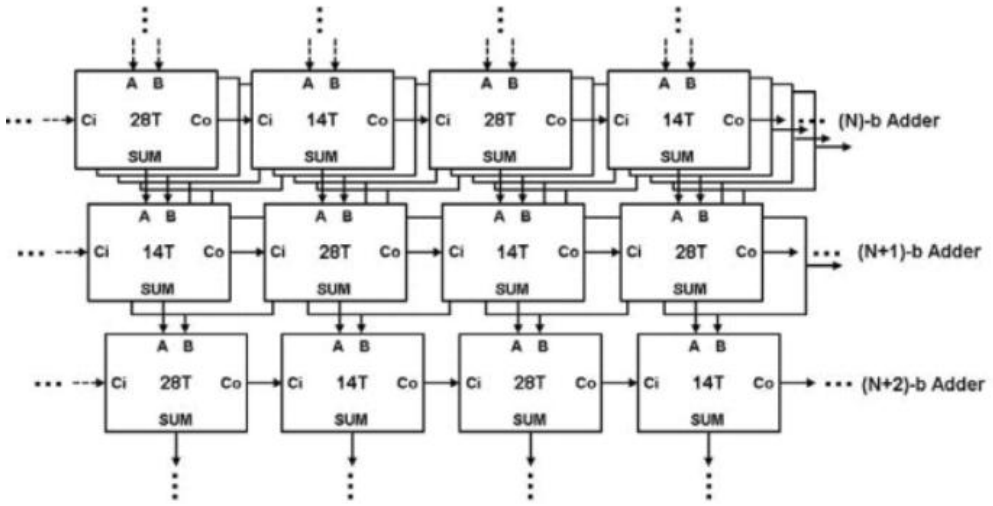

Fig. 4 Interleaved Hybrid Adder Tree [2]

4. Experimental Results

4.1 MAC 運算之結果：

根據 HFMC 之真值表：

$\overline{Weight}$	$\overline{Input}$		$Input \times Weight$
LBL / LBLB	IN[1:0]	GBLB & GBL	Out0[1:0] / Out1[1:0]
0	11	00	11
	10	01	10
	01	10	01
	00	11	00
1	X	X	00

Table. 1 Truth table of HFMC

我們可知當 $Weight = 1$、$IN = 11_{(2)}$時，對應的乘積輸出值為 3。為了驗證我們所設計的 HFMC & Channel-wise Adder Tree 架構是否正確地完成乘加 (MAC) 運算，我們進行了以下模擬：

設定整個運算矩陣為 $1\,row \times 32\,column$，並將所有 column 上的 weight 設定為 1，同時所有輸入 IN 亦統一設為 $11_{(2)}$。根據 HFMC 的邏輯，每一個 cell 將輸出乘積 3。因此，經由 Channel-wise Adder Tree 累加後，整列的 MAC 結果 (MACV) 理應為：$MACV = 3 + 3 + \cdots + 3 = 3 \times 32 = 96$

實際模擬後，我們觀察到最終累加結果確實為 96，驗證了系統在此測試條件

下的功能正確性。此結果也顯示 Channel-wise Adder Tree 架構成功地將 32 個乘積值準確地加總，達成整體 MAC 運算的目標。

4.2 不同 CAT 架構之比較：

為了深入比較不同加法器架構在實際電路層級下的效能差異，本專題針對 28T、14T & 28T hybrid 與 PTL & 28T hybrid 三種加法器樹進行模擬與分析，並在多組 pattern 條件下，觀察其延遲(delay)與功耗(power)的表現。我們使用 HSPICE 模擬，搭配 65 奈米 CMOS 製程模型，進行電路層級模擬，以獲得精確的時序與能耗數據。為量化比較各加法器的整體效能，我們定義一項綜合評估指標 $FoM = (Figure\ of\ Merit) = 1 / (Delay \times Power)$，此指標能反映單位能耗下的計算速度與效率。FoM 數值越高，代表該架構在能效與速度之間的表現越佳。

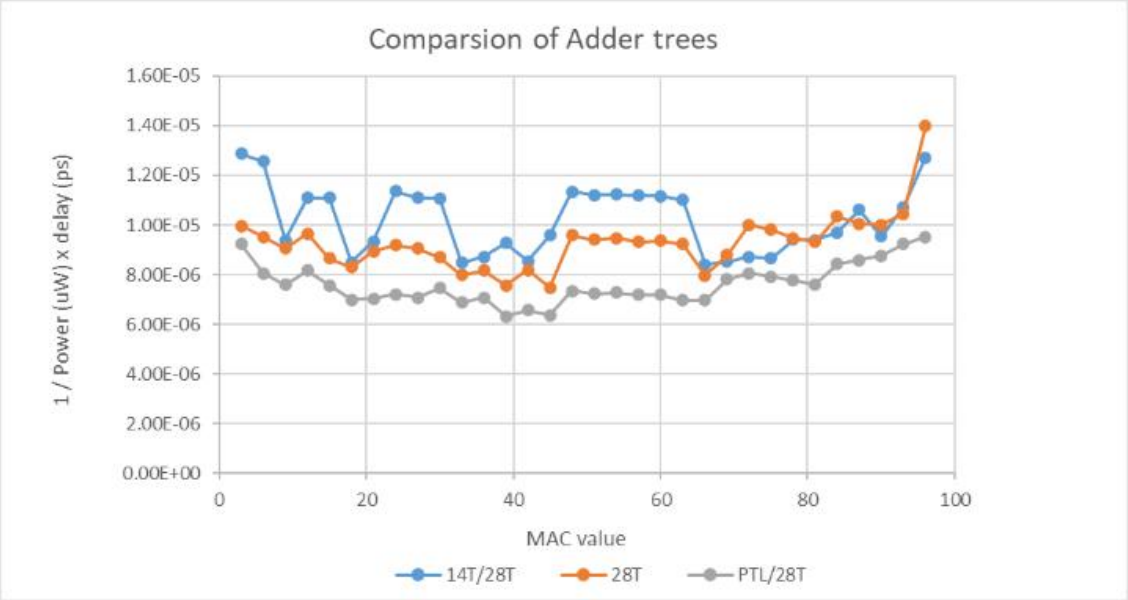

Fig. 11 FOM of three adder tree structure at different MAC value

從 Fig. 11可以觀察到

1. **28T 架構**的 FOM 整體趨勢平穩，位於中間水準，其在功耗與延遲上表現穩定，但並未在任何區段達到最優能效。這反映出其結構穩定性高、變異小，但功耗偏高限制其能效上限。
2. **14T/28T 混合架構**在多數 MAC value 區段中表現較佳，FOM 整體略高於 28T，特別是在 MAC 值位於 40 至 70 區間時有顯著上升。這顯示 GDI 架構在前段加總中確實降低了整體功耗，且透過與 28T 交錯補償輸出，使訊號穩定性得以維持。

3. **PTL/28T 混合架構**由於 PTL 結構驅動力弱，可能造成延遲上升與短路電流發生，使功耗增加，導致 FOM 明顯下滑。此外，由於 PTL 電路對製程波動與佈局寄生參數較為敏感，若應用於大型陣列中，需特別注意訊號完整性與佈局對稱性，以避免不對稱路徑導致邏輯錯誤。

綜合比較，14T/28T 架構在整體表現最佳。純 28T 架構則雖穩定，但整體效能表現不突出。這進一步驗證混合式架構的優勢與交錯配置策略的有效性。若進一步考慮硬體面積，可發現 14T/28T 與 PTL/28T 架構均明顯優於純 28T 加法器樹。由於 14T 與 PTL 加法器在單元面積上均遠小於 28T，整體面積可顯著縮減。此特性對於高密度 layout 或邊緣運算尤為重要，因此在綜合能效與面積後，14T/28T 架構在功耗、面積與效能三者之間取得了最好的折衷表現。

5. Conclusion

本在本專題中，我們實作了一個 SRAM-CIM Macro，採用交錯式混合加法器架構，將 14T 與 28T 全加法器依邏輯層級交錯配置於通道加法器樹 (Channel-wise Adder Tree)中。為了解決 GDI 架構之 14T 加法器在驅動能力不足與訊號擺幅衰減上的問題，我們在關鍵位置插入具強驅動力的 28T CMOS 加法器，以提升訊號穩定性與傳播品質，同時避免大幅增加電路面積。

儘管不同加法器設計在功耗、面積與延遲方面各有取捨，我們所做的交錯式架構成功在這三者之間取得平衡。14T 加法器提供良好的面積與能效表現，而 28T 加法器則確保邏輯訊號具備完整擺幅與穩定性。

模擬結果驗證此架構相較於全 28T 或全 14T 加法器樹，能達成更佳的能效與更低的延遲。本專題所完成的交錯式加法器樹在功耗、延遲與面積三者之間取得最佳平衡，適用於具多通道輸入的整數型神經網路。

6. 心得感想

回顧這次的專題研究，從最初進入張孟凡老師實驗室的那一刻開始，我們就深刻感受到「Compute-in-Memory（CIM）」不僅僅是一個嶄新的研究主題，更是一個橫跨電路設計、記憶體結構與神經網路運算等多重領域的挑戰。對我們這些僅具備基本電子學背景的學生來說，面對這樣龐大的知識體系，無疑是一場艱難的起點。

在老師與學長姐的引導下，我們從閱讀文獻做起，輪流上台報告各篇關鍵論文，從模糊的認知逐步建立起對 CIM 架構與 SRAM、RRAM 等記憶體型態的基本理解。透過這段時間的訓練，我們不僅熟悉了 paper 中常見的架構圖與模擬方法，更學會如何從一篇篇研究中提取有價值的設計觀點，這也為後續的實作打下紮實的基礎。

我們也修習了積體電路設計導論課程，接觸 SPICE 模擬語法、電路佈局與參數分析等工具。在寒假期間，我完成了 使用 FinFET 製程實現的 layout 設計。考慮排版、金屬層配置、DRC/LVS 驗證等整個後端流程。儘管 FinFET 製程的限制與規則複雜，經常遇到 layout routing 壓力過大或繞線失敗等挑戰，但在不斷修改與請教下，我最終成功完成 layout，這對我來說是個里程。

到了下學期，我們聚焦於一篇論文中所提出的加法器架構，並選擇進行改良與優化。我們比較了三種加法器樹結構（28T、14T/28T、PTL/28T），設計了交錯式混合架構來提升能效，同時使用 HSPICE 進行電路模擬，並以 FOM 指標作為比較基準。期間我們也遇到許多挑戰，像是 sizing 不當導致邏輯錯誤、電壓擺幅不足造成訊號失真，或是輸出位元無法正常收斂，但在學長姊們的指導下，我們一一排除錯誤，也更深入理解 CMOS 電路設計中看似簡單卻影響重大的細節。

這次專題讓我們從一個學生視角，真正進入到研究與工程實作的世界。我們學會了如何從「看懂」電路，到「設計」電路，並最終「驗證」電路是否真的能運作。這些經驗不僅提升了我們對電路設計的信心，也讓我們更加理解跨領域整合與團隊合作的價值。

最後，衷心感謝張孟凡老師提供寶貴的研究機會，並感謝每位實驗室的學長姐在過程中給予我們耐心的指導與技術上的協助。這段經歷，對我們來說不只是完成一個專題，更是走進半導體與記憶體運算世界的重要一步。

7. Reference

[1] A. Guo et al., "A 28nm 64-kb 31.6-TFLOPS/W Digital-Domain Floating-Point-Computing-Unit and Double-Bit 6T-SRAM Computing-in-Memory Macro for Floating-Point CNNs," 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023, pp. 128-130, doi: 10.1109/ISSCC42615.2023.10067260.

[2] Y. -D. Chih et al., "16.4 An 89TOPS/W and 16.3TOPS/mm2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," 2021 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2021, pp. 252-254, doi: 10.1109/ISSCC42613.2021.9365766.

[3] Yin, N., Pan, W., Yu, Y., Tang, C., & Yu, Z. (2023). Low-Power Pass-Transistor Logic-Based Full Adder and 8-Bit Multiplier. Electronics, 12(15), 3209. https://doi.org/10.3390/electronics12153209