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 8T-SRAM with Digital Sparsity Optimization Architecture

65奈米類比式靜態隨機存取記憶體內運算與數位稀疏性優化架構之整合

專題領域：系統組

組　　別：A407

指導教授：張孟凡

組員姓名：俞芷妍、張勻馨

研究期間：2023 年 7 月 1 日至 2024 年 5 月 1日止，共 10 個月

ABSTRACT

The rise of big data technology and AI chips has heightened the need for faster computing. To address the limitations of traditional Von Neumann architecture, Computing in Memory (CIM) emerges, enabling both data storage and processing within memory. This reduces data transmission delays between memory and CPU, enhancing computing efficiency and energy savings.

Our primary focus is SRAM-CIM. Through implementation and simulation, we compared Analog CIM and Digital CIM, finding Analog CIM (8T SRAM array with charge sharing) to exhibit superior energy efficiency. We refined charge sharing circuit architecture to enhance the linearity of in-memory computation results and reduce power by 8%. Additionally, we devised a RWL pulse generator with delay cells to integrate Analog CIM into digital data flow.

Digital Bitline-Transpose CIM boasts superior spatial mapping and accuracy. We introduced sparsity optimization to allow CIM to skip computation for input values of 0, thus accelerating system operations.

Throughout the project, we utilized the 65nm process for implementation and simulation. To merge Analog CIM's energy efficiency with Digital CIM's input-first processing, we combined modified Analog 8T SRAM CIM with Sparsity Optimization. This resulted in a 46% increase in energy efficiency and accelerated computation.

摘要

隨著大數據技術與 AI 晶片的發展，對電腦運算速度的需求越來越高，爲了克服傳統的馮紐曼架構（Von Neumann architecture）的限制，記憶體內運算電路（Computing in Memory, CIM）的出現讓記憶體不再只是儲存資料，而能在內存存儲器中執行運算，以減少數據在內存和中央處理器之間的傳輸次數和延遲，從而提高運算效率和節省能源。

本次專題中，我們主要的研究主題爲 SRAM-CIM。透過實作模擬論文中的 Analog CIM 和 Digital CIM，我們觀察到論文 [1] 中的 Analog CIM (8T SRAM array with charge sharing) 有較佳的 Energy Efficiency，而爲了使記憶體內運算後的結果更加線性，我們調整 charge sharing 的電路架構進行模擬，與未調整前相比線性程度提高且降低 8% 功耗。此外爲了與數位部分整合，我們利用 delay cell 製作了 RWL 脈衝產生器。Digital CIM 採用了論文 [2] 中的 Bitline-Transpose CIM，具有較好的 spatial mapping 能力且具有高度精確性。此外，我們額外設計了 sparsity optimization，讓 CIM 在遇到 input 爲 0 的情況能夠跳過此 input 的乘法與累加過程，加速系統運算。

我們在本次專題中使用 65 奈米製程進行實作與模擬。透過優化 Charge Sharing 電路與增加 Sparsity Optimization 結構, 我們善用 Analog CIM 較佳的 Energy Efficiency 優勢與 Digital CIM 對 input 先行處理的機制。不但 CIM 運算速度加快，同時 Energy Efficiency 有 46% 的提升。

I. Experimental Principles and Methods

本次專題我們模擬實作了 Analog CIM 和 Digital CIM，並分析彼此間的異同並進行改良。在 Analog CIM 中，多條 Word-line(WL) 訊號會做爲多個 input，記憶體單元則儲存 weight，兩者的組合會決定 Bit-line(BL) 是否放電，最終 BL 的電壓變化表示累加的乘法結果。Analog CIM 的架構我們參考論文 [1] 中的 8T-SRAM CIM，並針對模擬結果的線性度去改良電路設計。Digital CIM 中，SRAM 儲存的 Weight 會直接和對應的 input 進行乘法運算後，並於 digital domian 進行累加。實作部分參考了論文 [2] 中的 Bitline-Transpose CIM，進行模擬並針對 input sparsity 進行改良。最後我們嘗試將兩種 CIM 的優勢合併，設計出新的 CIM 電路。

1.1、Experimental Methods

針對 CIM 結構的研究與改良，我們主要透過以下四個步驟進行專題實作:

- Analog CIM 的模擬實作
- Digital CIM 的模擬實作
- 模擬結果的比較與整合
- 實作中的改良

1.2、Analog CIM with 8T SRAM cell array and charge sharing method

類比式記憶體內運算的架構參考了資料 [1]，以下說明運作原理。

1.2.1、Introduction

類比式的 SRAM 記憶體內運算，主要是透過控制 Word-line (WL) 的開啓或關閉，與 SRAM 儲存的值反應，造成 Bit-line(BL) 上不同的電壓下降值。只有當儲存在 SRAM 中的 weight 和輸入進 WL 的 input 同時爲 1 時，這個單元才會產生電流讓 BL 電壓下降，此時 BL 上的電壓變化會代表 1-bit weight 和 1-bit input 的相乘結果。若在一條 BL 上同時使用多條 WL 的運算的話，每個單元有各自的 weight 和 input pulse 組合，產生電流後 BL 上的電壓變化便可以代表所有 cell 的相乘累加結果。

1.2.2、Implementation

在類比式內運算的部分，我們參考了論文 [1] 中的電路架構與運算原理，本次實作出的電路架構圖如圖 1，包含 4x4 8T SRAM Cell Array、計算-補償電容區。綜合考量運算正確性與吞吐量後，此架構在一個操作週期中可實現 4 個 2-bit input 與 4-bit weight 相乘的記憶體內運算總合，並輸出 4-bit ADC output。2-bit input 以數位的方式決定了送入 Read word-line (RWL) 的脈衝數量，input=2’b01 的時候給 1 個 RWL 脈衝，input=2’b11 的時候給 3 個連續的 RWL 脈衝，以此類推。4-bit weight 會在運算前寫入相鄰的四顆 SRAM cell 中，並在記憶體內運算時取樣在不同的電容上 (從 MSB 到 LSB 的電容值比例爲 8:4:2:1)，最後將四顆電容相連、進行電荷分享 (charge sharing)，得到考慮過權重的 MAC 結果。內運算結束後，計算電容上的電壓代表了累加 4 個 2x4b 的運算結果。

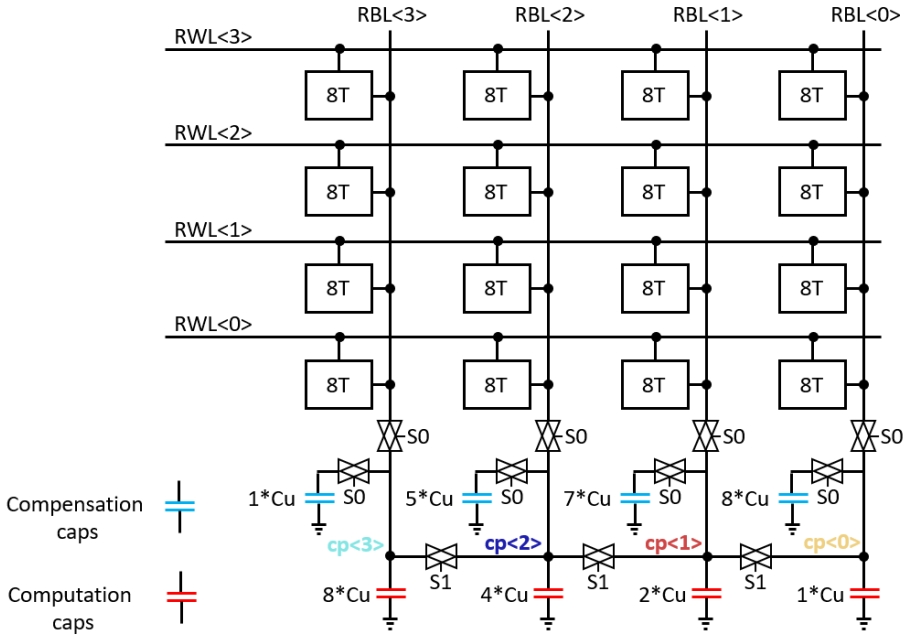

圖 1. Analog 8T-SRAM CIM 電路圖

1.2.3、Operation principle

圖 2 爲內運算時各訊號的波形圖，總共分成 RBL 預充電、RBL 電壓取樣、電荷分享三個時期。首先是 RBL 預充電時期，RBL、計算電容、補償電容都被充電到 VDD。接著進入到 RBL 電壓取樣時期，對於每一條 RWL，我們都會根據其 2-bit input 給相對應數量的 RWL 脈衝，而 RBL 就會有相應的電壓下降，此時爲了避免非線性的情況，我們希望每條 RBL 負載相同，而這就是要加入補償電容的原因。電壓取樣完成後就會進入電荷分享時期，此時需要將計算電容與補償電容分開，並且把四個計算電容相接、做電荷分享。我們最後得到的電壓下降量，會與累加四組 2-bit input 和 4-bit weight 相乘的 MAC 值成正比。

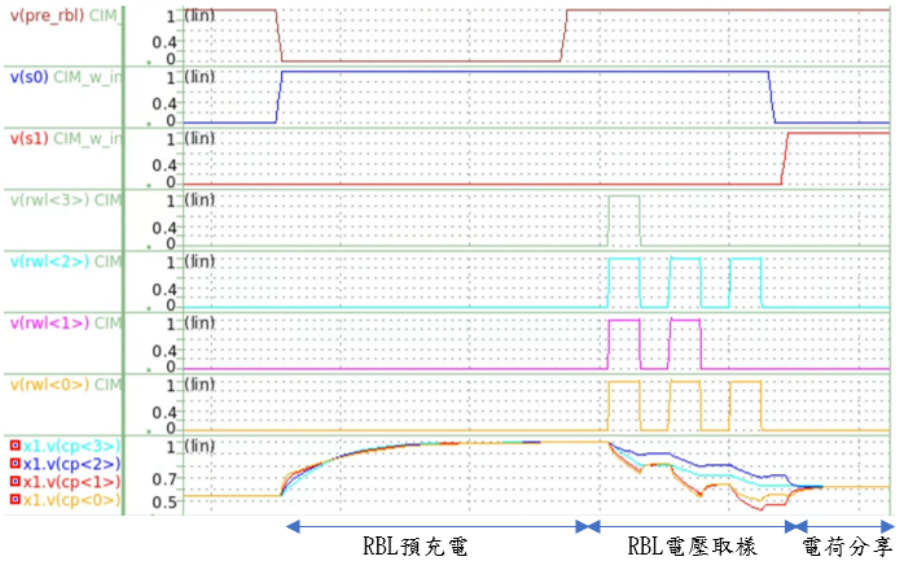

圖 2. 內運算範例與控制訊號的演示

1.3、Digital Bitline-Transpose CIM

1.3.1、Introduction

Digital CIM 的結構我們參考了論文 [2] 中的 Bitline-Transpose CIM(BLT-CIM) 設計，如圖 3。這顆 CIM 是做爲 Transformer CIM 設計的，特徵爲利用水平布局的 bitline 取代 wordline 來接收 input 訊號。BLT-CIM 的結構大致可以被分爲三塊：CIM controller、SRAM-CIM、Macro accumulator。

- CIM controller 具有 3b 的控制訊號表示當前工作狀態是 weight writing 或是 input feeding。
- 16 × 256 SRAM-CIM array 爲完全數位的記憶體內邏輯運算，主要是以 6T SRAM 構成以求最高的 weight 儲存密度，並由 4T NOR 作乘加運算並連接到 Accumulator。
- Macro Accumulator 會將 bit-serial input 相乘完的資料進行 shift-accumulation。

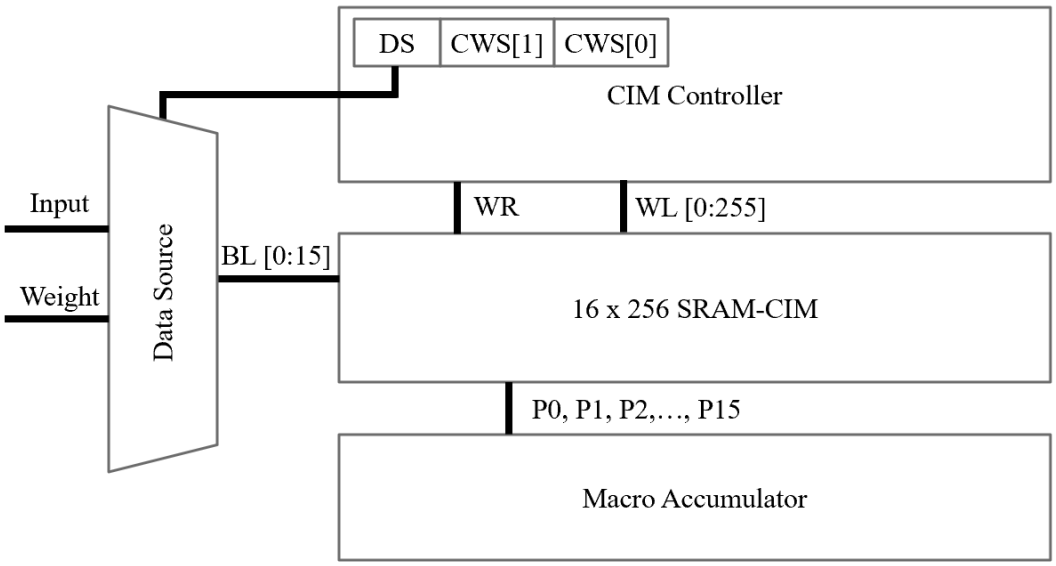

圖 3. Bitline-Transpose CIM 電路模擬實作的 Block Diagram

1.3.2、Modification

(a) Sparsity Optimization

考慮到 transformer 會進行大量的運算，當 input 爲 0 時，所有 partial sums 以至於 accumulator 的累加在這個 block cycle 裡都會是 0，但中間卻仍然需要經過 cycle 的運算過程。因此我們在實作完基礎結構的模擬後，決定額外設計電路邏輯：在偵測到 16-inputs 爲 0 時，CIM 會跳過這組運算，直接運算下一組 input 訊號。

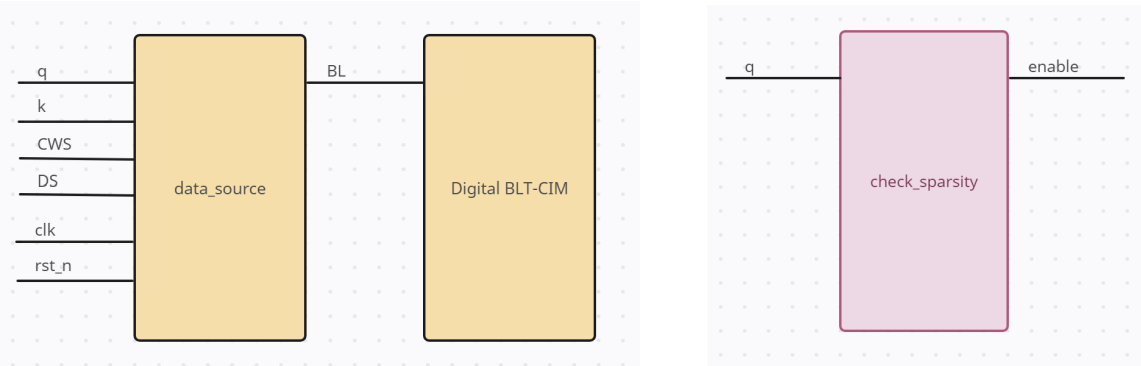

圖 4. Sparsity Optimization 架構圖

II. Experimental results and analysis

2.1、Analog CIM with 8T SRAM cell array and charge sharing method

模擬結果以及改良電路的過程如下：

2.1.1、Simulation of Original Circuit

由於我們是使用 RBL 的電壓來代表 MAC 運算結果，每條 RBL 放電的速度必須相等，爲此我們需要讓每一個 RBL 上的負載平衡。

在我們的第一版設計中，爲了讓 RBL sampling 時每一個 RBL 上的負載相同，我們統一了所有開關 (transmission gate, TG) 的大小，而計算電容與補償電容的比例則參考了論文 [1] 以每條 RBL 上電容總和爲 9*Cu 去實作，但模擬後發現 Conn 節點電壓與 MAC 值不成線性關係。針對波形分析後，我們發現在固定 input 的情況下，weight 每個位元的 RBL 電壓下降速度不一。這是因爲內運算時 RBL 在做 RC 充放電，而每條 RBL 看到的負載由補償電容/記算電容的組合方式決定，當電容值相差越小 (RBL<2> 的情況)，RBL 電壓下降越多。由於 RBL 彼此間的負載差異，Conn 節點電壓與 MAC 值不成線性關係，我們必須找出新的解決方法。

2.1.2、Read-bitline RC balence and optimization

(a) Tuning switch size

針對負載不平衡的現象，我們選擇去微調每個 RBL 上，連接 RBL 與計算-補償電容區的開關大小。模擬結果顯示，在固定 input 的情況下，weight 每個位元的 RBL 電壓下降速度確實相同。但是模擬出的電荷分享後節點電壓與 MAC 值仍然不成線性關係，這是由於 CIM 最後階段總共有四個計算電容、八個 TG 在參與電荷分享，而非理想的 TG 會影響電荷分享結果，所以結果仍非線性。

(b) Tuning compensation capacitor

從上述兩個設計中，可觀察到非理想性來自於不平衡的負載和電荷分享，因此我們決定針對每條 RBL 微調補償電容值，讓兩個非理想性相互抵銷。當只開一條 RWL，且 input=1 時，模擬出的 Conn 節點電壓與 MAC 值呈現性關係，相關係數高達 0.9995，顯示我們確實可以根據電荷分享後的電壓，回推出 MAC 值。

不過，當我們增加 RWL input 的數量，會發現開不同 RWL 數量會產生不同的 V(Conn)-MAC 做圖斜率，而理論上不論開幾條 RWL，斜率應該都要相同。我們推測是因爲電晶體的電流大小有極限，開 n 條 RWL 時，RBL 上的電流大小不會剛好是開 1 條 RWL 時的 n 倍，所以開越多條 RWL 時，拉力會比預期還更小，V(Conn)-MAC 斜率值也會更小。

(c) Optimize switch placement and switch number

上述方法最大的阻礙是開不同數量的 RWL 時，無法回歸到同一條 V(Conn)-MAC 線上，所以線性度差、無法讓 ADC 判斷成正確的數值。而在模擬 2.1.2(a) 的時候，我們發現在不影響運算功能的情況下，開關的位置也會影響負載。最後我們決定在維持相同功能下減少開關的數量，降低開關在平衡負載以及電荷分享時的影響。實驗結果顯示在維持線性度下，我們提高了一次可以開的 RWL 數量，也就是累加的數量到四個，以及 input 最多可以到 2-bit。

2.1.3、Comparison

在模擬電路時，我們用三個指標去判斷電路的可行性。第一項是 Power，測量方法是開啓四條 RWL、每個 RWL 都給 3 個脈衝、weight 都存 4’b1111，模擬此情況下的功耗。第二項是 Coefficient of determination，我們同時開啓四條 RWL，模擬脈衝數量爲 1、2、3 個的時候，weight 儲存 0 到 15 的情況，紀錄電荷分享後的電壓與 MAC 值的數值。對電壓-MAC 圖做線性回歸，當 coefficient of determination 越大，表示內運算的結果越線性，運算後的電壓越可以被 ADC 判斷成正確的值。最後一項是 Regression intercept，我們知道當運算結果等於 0 的時候，理論上 RBL 不會有電壓下降，因此 MAC=0 時的理想電壓應爲 1 伏特。所以當回歸線的截距越靠近 1，表示此電路的結果會越接近理論。

從模擬結果可看見，減少開關數量、改變電容平衡負載等方法，皆可提升相關係數，其中又以微調補償電容和改變開關數量的方法最好。但是從 Power 來看，改變開關數量的方法可以在維持與微調電容差不多的線性度時，又有較小的功耗，此外它也是回歸線截距最接近 1 的電路。綜上所述，我們選定以改變開關數量做爲最終方案，與未調整前的電路相比，提高線性程度的同時，也節省了 8% 的 Power。

	No RC balence	RC balence by tuning switch size	RC balence by tuning compensation caps	Optimize switch placement
Power (W)	1.83E-0.5	1.61E-05	2.05E-5	1.69E-05
Coefficient of determination	0.9947	0.9958	0.9965	0.9961
Regression intercept	0.9722	0.9752	0.9805	0.9853

表 1. 比較各版本電路的表現

2.2、Bitline-Transpose CIM

電路的模擬是經過 RTL simulation，Synthesis simulation，以及 Gate simulation 得出實驗結果。此專題中我們透過 Verilog 建構以上電路，並經過幾次修改與校正以獲得較佳的 Area 和 Timing 表現。

論文 [2] 原先的電路設計是讓 testbench 直接對 Bitline 給 random 值，而我們額外設計的 Sparsity Optimization 爲了偵測 input 是否爲 0，我們需要先有儲存好的 data source，讓其經過檢測的 module 後決定是否要進入 CIM 運算，當確定 Input 不爲 0 時才會送到 bitline。當偵測到某 Input 爲 0，系統會跳過這組運算，直接開始下一組 Input 的運算，節省了一個 block cycle 的運算時間。

加上 data source 後的面積爲 176376.0041 (um^2)，雖然比起未加 Sparsity Optimization 的電路增加了 8.3%，但 data source 作爲儲存資料的記憶體本身並不屬於 BLT-CIM，檢測 sparsity 的模組實際上只有 1b 的 register。在 clock cycle 爲 6ns 的條件下，slack 爲 1.86，能耗爲 26.199 (uW)。

2.3、Comparison

爲了比較 Analog CIM 和 Digital CIM 的異同，我們修改了 Digital CIM(with Sparsity Optimization) 的電路，讓 2.2 的 BLT-CIM 的 specification 和 2.1 改良後的 Analog 8T-SRAM CIM 一致，也就是 2 bit input 和 4x4 的 SRAM cell，並比較了兩者的 Energy Efficiency。

Energy Efficiency 我們採用以下公式：

$$\textbf{Energy Efficiency}(\textbf{TOPS}/\textbf{W}) = \frac{\textbf{operations}}{\textbf{time} \times \textbf{power}} \tag{1}$$

在 clock cycle 爲 10ns，跑分別跑一組與三組 input 乘加運算的條件下，我們的模擬得到以下的結果：

	總 cylce 數	Operations	Power(W)	Energy Efficiency(TOPs/W)
Analog 8T-SRAM CIM	4	(4 + 3) x 3	4.56 x 10^{-4}	1.15
Digital BLT-CIM	13	(4 + 3) x 3	1.54 x 10^{-4}	1.05

表 2. Analog CIM 與 Digital CIM 的比較

從上表我們可以看到 Analog 的 8T-SRAM CIM 之 Energy Efficiency 較 Digital CIM 來得佳。Accuracy 因爲 Digital 會是 100% no loss 的狀態，所以會是 Digital 的 BLT-CIM 來得佳，且 Digital 的 CIM 也具有較好演算能力，能針對 input 等作處理後再進行運算。

III. Improved implementation and results

3.1、Circuit Design

經過 Analog CIM 和 Digital CIM 的實作模擬與比較後，發現其各有優勢，而我們希望能藉由合併兩者的優勢，使電路有更好的表現。考慮到 Analog CIM 的 Energy efficiency 較高，我們計畫採用 Analog-CIM 的 8T-SRAM CIM 與 charge sharing 架構繼續進行研究，期望在節省能耗的同時達到較好的精確性，同時透過組合 Digital CIM 中的 Sparsity Optimization 機制加速運算的時間。

(a) 8T SRAM array with RWL pulse generator

Bitline-Transpose CIM 數位電路經過 sparsity 優化後，會傳送 4 個 2-bit input 給類比區的 8T-SRAM CIM 陣列，接著這些 input 在一個 clock cycle 內轉換成每個 RWL 上的 pulse 數量、進行內運算得到類比電壓結果。

透過一系列的邏輯閘，Input 與 clock 訊號會被轉換成 RWL Pulse。我們首先將數位電路送過來的 clock 訊號送進 delay cell 延遲一次，再把延遲前後的 clock 訊號送進 AND Gate，兩者重疊的部分就會形成比 clock 半周期還短的 pulse，我們稱此訊號爲 pulse1。而一次 cycle 內我們需要最多三個 (input=2'b11 時) 不重疊的 pulse，所以 pulse1 會再被延遲一次和兩次，形成 pulse2 和 pulse3 訊號。2-bit input 會決定三個 enable 訊號的值 (EN1=Input<1>|Input<0>，EN2=Input<1>，EN3=Input<1>&Input<0>)，這三個 enable 訊號再去控制 pulse1、pulse2、pulse3 是否送入 RWL。

(b) Sparsity optimization

這部分的結構與 1.3.2(a) 大致相同，僅對 presicion 等做微調。在偵測到 2-bit input 爲 0 時，CIM 會跳過這組運算，直接讓 CIM 運算下一組 input 訊號。Block Diagram 如下：

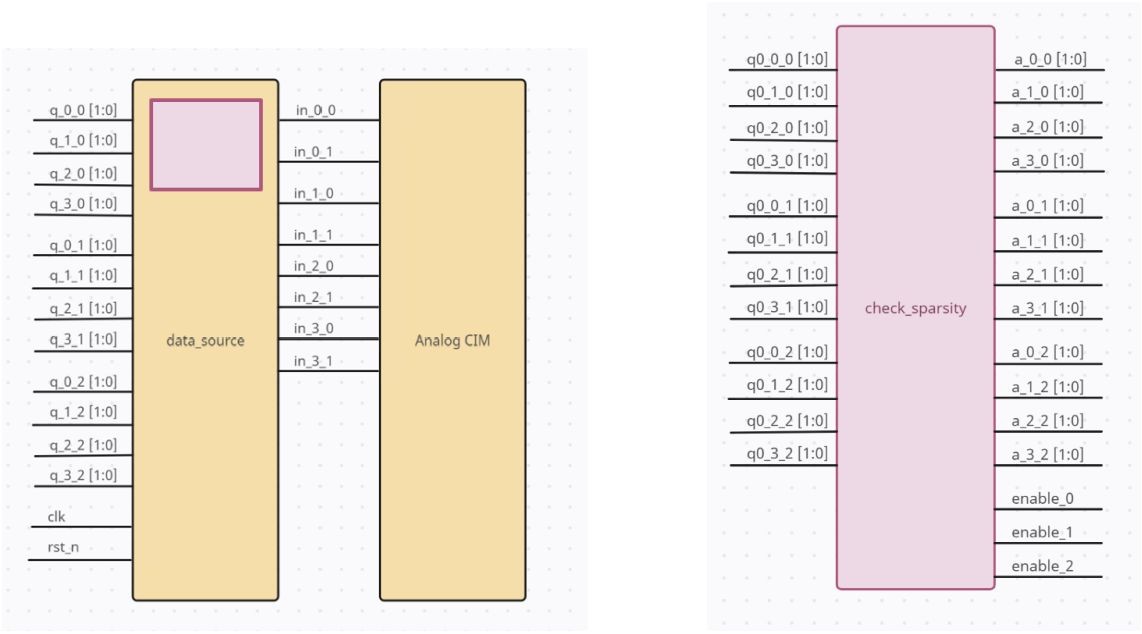

圖 5. 合併電路的 Block Diagram

3.2、Experimental results

合併後的實驗結果如下表 3，由於使用 Sparsity Optimization，我們可以跳過全零 input 的組合，而少了一組 input 的運算。總 cycle 數的下降令運算速度變快，同時 Energy Efficiency 也有 46% 的提升。

	總 cylce 數	Operations	Power(W)	Energy Efficiency(TOPS/W)
Previous	4	(4 + 3) x 3	4.56 x 10^{-4}	1.15
This work	3	(4 + 3) x 3	4.14 x 10^{-4}	1.69

表 3. Analog 8T-SRAM CIM 合併 Sparsity Optimization 前、後的比較

IV. Conclusion

1. 在第一部分的模擬實作中，我們觀察到 Analog 的 8T-SRAM CIM 有較佳的 energy efficiency，透過改良也具有較佳的線性表現而有較好的準確性。而 Digital 的 BLT-CIM 有非常好的準確性，也具有 spatial mapping 能力。除此之外，Sparsity Optimization 也可以有效加快 CIM 的運算速度。

2. 合併改良後的 Analog 8T SRAM CIM 與 Sparsity Optimization 結構，可以觀察到 CIM 的運算速度加快，同時令 Energy Efficiency 有 46% 的提升

V. Reference

[1] Qing Dong, Mahmut E. Sinangil, Burak Erbagci, Dar Sun, Win-San Khwa, Hung-Jen Liao, Yih Wang, Jonathan Chang, “A 351TOPS/W and 372.4GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine-Learning Applications”, ISSCC2020.

[2] F. Tu et al., ”A 28nm 15.59μJ/Token Full-Digital Bitline-Transpose CIM-Based Sparse Transformer Accelerator with Pipeline/Parallel Reconfigurable Modes,” 2022 IEEE International Solid-State Circuits Conference (ISSCC)

[3] Mahmut E. Sinangil, Burak Erbagci, Rawan Naous, Kerem Akarvardar, Dar Sun, Win-San Khwa, Hung-Jen Liao, Yih Wang, Jonathan Chang, “A 7-nm Compute-in-Memory SRAM Macro Supporting Multi-Bit Input, Weight and Output and Achieving 351 TOPS/W and 372.4 GOPS”, JSSC, VOL. 56, NO. 1, pp. 188-198, JANUARY 2021.

VI. Review and Reflection

在這一年的專題訓練中，我們經過了 CAD TOOL 的練習、文獻的閱讀到實際模擬與改良架構，中間的過程不僅僅需要許多的關於記憶體的知識，並且對於電路的操作更要熟悉。

從選題開始，我們分析了數篇論文、尋找有興趣的方向，最後在類比與數位領域中選擇以兩篇論文爲基礎去發展。開始研究之後，光是模擬出論文中的架構就花了很多時間跟精力，接著還要想出整合兩篇架構的橋樑，以及根據可行性去取捨，最後再提出新的構想，過程非常紮實。在本次專題報告中，我們在一次次嘗試中改善架構、提高了類比運算的線性程度，也持續精簡 code，提高數位電路的 performance，最後還成功整合了兩個電路，成就感十足。

特別感謝我們的指導學長姐，謝謝他們總是不厭其煩回答我們的問題，也經常與我們約時間開會討論研究方法、研究方向，更給予我們許多實作上的意見與優化的建議。同時，非常感謝張孟凡教授在過程中給予我們許多啓發，讓我們多去主動問問題，以及了解業界潮流狀況。在這一年的專題研究中，我們眞的收穫良多。除此之外，在專題研究的過程中，我們也遇到了很多問題，而這些碰壁的經驗也讓我們認知到自己的不足之處，更期許未來能掌握更多的知識!