國立清華大學 電機工程學系

實作專題研究成果摘要

Implementation of Dual-Split-Control 6T SRAM-Based Computing-in-Memory for DNN Edge Processors

應用於深度學習神經網路邊緣處理器之雙重分割控制靜態隨機存取記憶體內運算

專題領域：系統組
組　　別：A92
指導教授：張孟凡 教授
組員姓名：黃冠穎、劉俊廷

研究期間：109 年　7 月 1 日至 110 年 5 月底止，計 10 個月

中文摘要

靜態隨機存取記憶體是一種揮發性的記憶體用來在積體電路中的內嵌式記憶體部分。靜態隨機存取記憶體贏過其他種型態的記憶體是因為他的速度較快且相容於其他處理器。對於記憶體記憶體內運算 (CIM)，記憶體不再只是儲存資料，還能在記憶體內執行簡單的運算，執行完後，再將資料傳到中央處理器，而非只是搬資料至中央處理器後才做運算，如此能減少資料搬移所造成性能和功率的損失。然而，一些靜態隨機存取記憶體記憶體內運算面臨一些挑戰，如面積、表現、能源效率與產生不同資料型態的模式與電晶體表現限制。記憶體內運算是一種 AI 加速器架構，運算上是將記憶體內的資料，先在記憶體內先計算過，如此一來打破傳統透過處理器進行分析的馮紐曼型架構。雖然處理器的運算速度遠快於記憶體內的讀寫速度，但處理器內的資料處理速度仍會受記憶體傳輸頻寬所限制以及傳輸距離，因為資料在 DRAM 或硬碟上，會需要很長的傳輸距離，而影響運算速度及功耗，相比下，記憶體內運算可以使用低功耗、高效率的方式，在終端裝置上進行影像或語音辨識的能力。

此專題報告中的兩種分割電壓控制下靜態隨機存取記憶體的記憶體內運算支援兩種神經網絡模型分別為 XNOR 神經網絡和修正後的二進制神經網絡 MBNN。在此專題報告中，我們主要聚焦在修正後二進制神經網絡模型。為了實現較小的偏差電壓、更短的讀取時間、更低的能源消耗、更小的晶片面積、穩健的操作以及更高的能效，我們使用的靜態隨機存取記憶體內的記憶體運算並使用標準形態下靜態隨機存取記憶體分割的字元線，其中包括動態輸入感知參考生成（DIARG）結構，與演算法有關的非對稱控制（ADAC）結構和共模不敏感小偏移電壓模式感測放大器（CMI-VSA）以及兩種分割電壓控制的靜態隨機存取記憶體（DSC SRAM）結構。在 25°C 以及典型的 N 型氧化物－半導體場效電晶體和典型的 P 型氧化物－半導體場效電晶體的記憶體運算模式之下，我們應用的共模不敏感小偏移電壓模式感測放大器（CMI-VSA）量測得的最小偏差電壓達到了 36 毫伏特。

為了減少整體靜態隨機存取記憶體的面積，我們使用一些概念來實現兩種分割電壓控制下靜態隨機存取記憶體的布局。並為兩種分割電壓控制下靜態隨機存取記憶體的記憶體運算畫出 64 * 64 靜態隨機存取記憶體陣列製作的佈局，結果顯示比起之前標準型態的 6 個電晶體的靜態隨機存取記憶體能縮小 0.12 倍的總體面積。

Abstract

Static Random Access Memory (SRAM) is a volatile memory (NVM) used as embedded memory in an integrated circuit. Over other types of memory its speed is fast such that it is compatible with processors. For computing-in-memory structure (CIM), the memory is no longer just for storing data, but can also perform simple calculations in the memory. After the computing execution, the data is transferred to the central processing unit. Not only just moving the data to the central processing unit to do calculations. By this way, it can reduce the performance and power loss caused by data movement. However, some difficulties that computing-in-memory (CIM) SRAM faces practical challenges in terms of area overhead, performance, energy efficiency, and yield against variations in data patterns and transistor performance. CIM is a kind of AI accelerator architecture. The actual calculation process is directly performed calculations in the memory through the data in the memory.

The actual analysis results will transmit to the processor, thereby breaking Von Neumann Architecture which completely analyzes all the data through the processor. Even if the computing speed of the processor is much faster than the memory read and write, the data processing speed will still be limited by the memory transmission bandwidth and transmit distance. Because the data is on DRAM or hard disk, it will need a long transmission distance, which will affect the calculation speed and power consumption. In contrast, the ability of in-memory operations to perform image or voice recognition on terminal devices with low power consumption and high efficiency.

This proposed DSC SRAM-CIM unit-macro supports two neural network models: an XNOR neural network (XNORNN) and a modified binary neural network (MBNN). In this project we mainly focus on the modified binary neural network (MBNN) model. To achieve low offset voltage, fast access time, lower power consumption, compact area, robust operations, and high energy-efficiency, our proposed SRAM-CIM uses a split-word-line compact-rule 6T SRAM, including a dynamic input-aware reference generation (DIARG) scheme, an algorithm-dependent asymmetric control (ADAC) scheme, a common-mode-insensitive small offset voltage-mode sensing amplifier (CMI-VSA), and a dual-split-controlled SRAM (DSC) scheme. The measured minimum offset of the voltage mode sensing amplifier (VSA) in our work reached 36mV in typical NMOS and typical PMOS (TT corner) CIM mode at 25°C.

In order to further reduce the footprint area, we also considered using some structure to implement the DSC6T SRAM layout. Using DSC6T SRAM CIM ARRAY layout the footprint area is 12% lower than that of a standard SRAM.

Index

Report Index

Figure Index

I. Preface

For deep-neural-network (DNN) processors, which are commonly used in artificial intelligence processors, product-sum operations mainly dominate the overall computation workload, while movement and storage of large volumes of data is also required. Thus, DNN processors are more likely to be implemented to artificial-intelligence(AI) devices that require low-power consumption, low-cost and fast inference. Owing to the above-mentioned restrictions, usually binary DNN are used since they can reduce computation as well as hardware costs, making it possible to be used for artificial intelligence computing.

However, the memory bottleneck problem that conventional digital all solutions cannot solve still exists. Computing-in-memory (CIM) methods address these problems by enabling parallel computing, reducing the number of memory accesses, and suppressing intermediate data, since CIM structure allows for the data processing to be within the memory.

When we first encountered Computing-in-Memory (Processing-in-Memory) Circuits for Deep Learning, AI chips and other memory Integrated Circuits (SRAM, STT-MRAM, ReRAM, PCM, eFlash, 3D-NAND), we hoped that they could spark novel ideas and implementation methods. These thoughts inspired us to combine them while also using some structure for implementation to address the challenges that conventional SRAM models face, mainly concerning area overhead and energy efficiency. The methods and implementations that we adopted will be thoroughly mentioned and explained in the following sections.

II. Introduction

A. Principle analysis

In this project we solely implement the modified binary neural network (MBNN) mode, and the binarized function for it is as follows:

$$x^b = Sign(x) = \begin{cases} 1 & if\ x \geq 0, \\ 0 & otherwise \end{cases}$$

As for the binarized function for binary weights in MBNN, the function is:

$$w^b = Sign(w) = \begin{cases} +1 & if\ w \geq 0, \\ -1 & otherwise \end{cases}$$

where x^b is the binarized activation, and w^b is the binarized weight.

Fig. 1 shows the architecture and basic waveform of the MBNN SRAM-CIM structure. The inputs for the MBNN-based CIM operations are either “1” or “0” as mentioned above. This structure uses the algorithm-dependent asymmetric control (ADAC) scheme by changing the value of AF (1 or 0) to activate either the left sending mode or the right sensing mode. When AF = 1, if an input (IN[i]) is 1, then its WLL (WLL[i]) is “1” and its WLR (WLR[i]) is “0”. When the input is “0”, then both WLL

and WLR equal to “0”. When the input-weight-product result (IWP) of an MBNN operation is “+1”, the DSC6T cell generates a charge current on the BLL, and when the IWP is “-1” then it generates a discharge current. If IWP = 0, then the DSC6T cell does not generate any cell current to BLL. Since AF = 1, then we activate the left sensing mode, WLR = 0, thus BLR is disconnected from BLL to remain as a floating state. BLL then represents the total number of IWP results associated with each MBNN operation on activated DSC6T cells. Finally, the MBNN count can be then digitized by sensing V_{BLL}.

Since the ADAC scheme specifies whether to use only WLL-BLL or WLR-BLR for sensing, ADAC + DSC6T consumes less I_{BL} and power is reduced compared to conventional 6T cells due to less parasitic load on WLL/WLR (1 transistor per cell), less I_{BL} on the selected BL, and no I_{BL} from the opposite BL which is not selected.

We use the algorithm dependent asymmetric control (ADAC) scheme to reduce power consumption by only activating one WL for each operation. The ADAC scheme combined with the split-WL feature of DSC6T cells reduces BL current and power consumption. This can be explained by a reduction in parasitic load on activated WLs (one transistor per cell), a reduction in BL current on the selected BL, and a lack of BL current from unselected BLs.

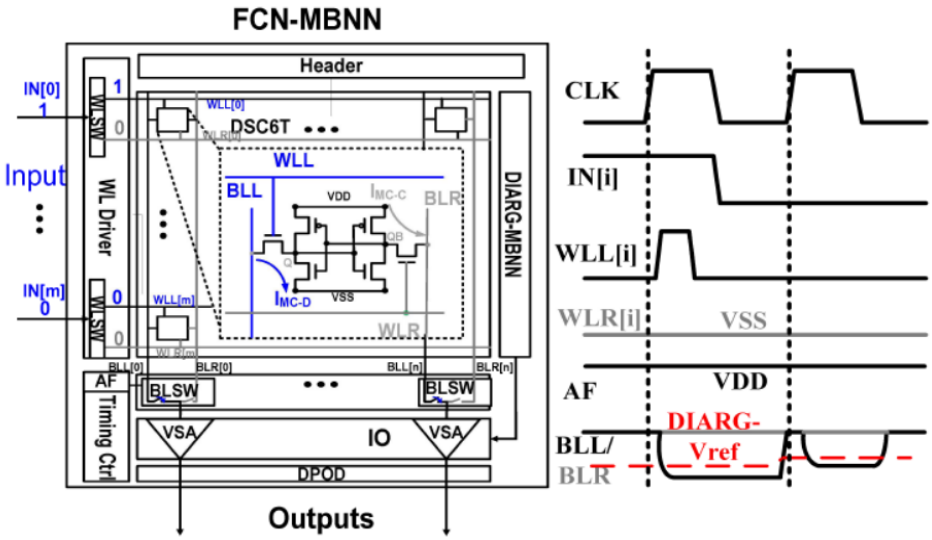

Fig. 1. Structure and waveform of the MBNN SRAM-CIM[1]

B. System design

The section above described the overall structure and operation of the whole circuit. In this section, we will focus on each scheme and segment of the MBNN SRAM-CIM, giving a more detailed description of its operations.

1) DSC6T SRAM

Fig. 2 presents a schematic of a dual-split-control (DSC) 6T SRAM cell. The footprint area of this DSC6T cell is the same as that of the compact 6T SRAM cell but with split wordlines (SWL: WLL and WLR), and split VDD lines (CVDD1 and

CVDD2). This DSC6T cell achieves compact cell area and low VDD_{min} through the use of split wordlines, VSS, VDD . By lower one side VDD, DSC SRAM has lower power consumption compared to standard SRAM.

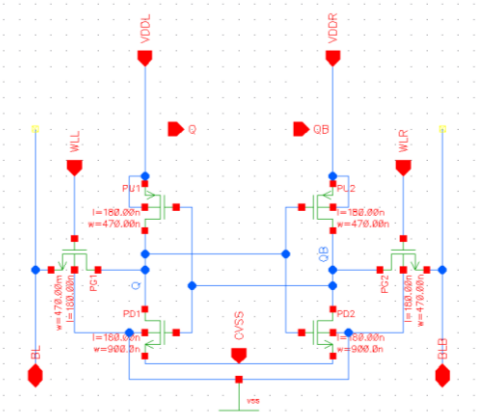

Fig. 2. DSC6T SRAM Schematic and Waveform

Fig. 2 also presents the waveform of the DSC6T cell during read/write operations. For normal write and read operation, which is the same as those of conventional 6T SRAM cells, WLL and WLR are short concurrently. Lower power consumption was achieved by using the split-VDD depending on the value of CVDD1 and CVDD2.

2) CMI-VSA

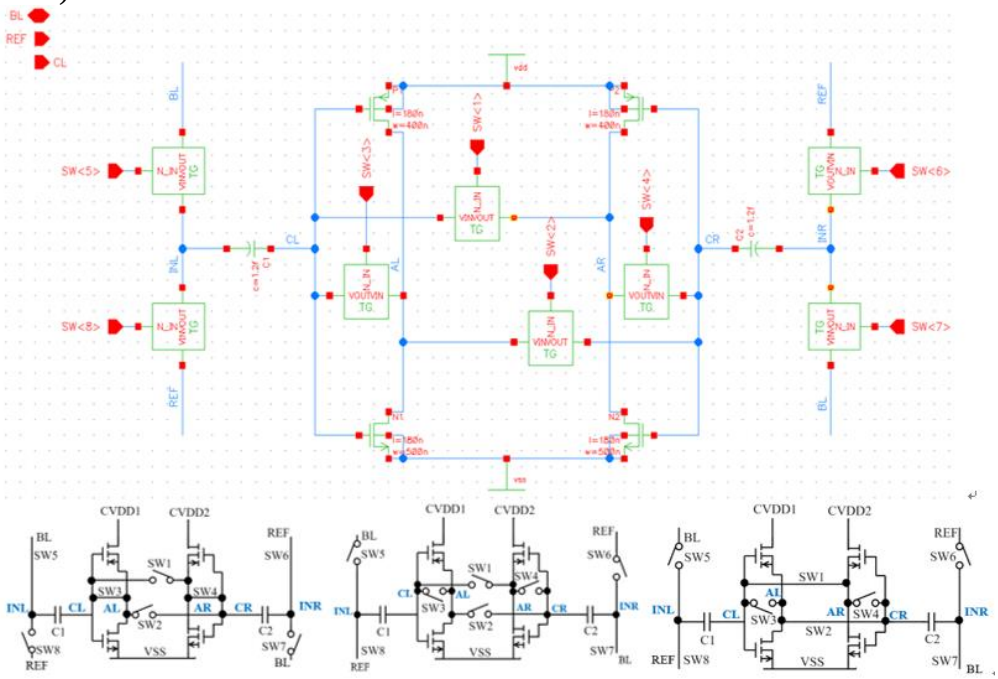

Fig. 3. CMI-VSA Schematic and Three Phases

A sense amplifier is a read circuitry that is used when data is read from the memory, and senses the low power signals from a bitline that represents a data bit (either 1 or 0) stored in a memory cell, and amplifies the small voltage swing to recognizable logic levels so that data can be interpreted properly by logic.

The common-mode-insensitive small-offset voltage-mode sense amplifier (CMI-VSA) as shown in Fig. 3 provides tolerance for a small BL signal margin against wide

VBL common-mode range. High power consumption on BL and BLB with multiple activated WLs and insufficient signal margin against input-offset of the sense amplifier for robust read operations. The CMI-VSA overcomes these issues.

The common-mode-insensitive small-offset voltage-mode sense amplifier employs three phases (PH1 − PH3) for sensing operations. By using different states of switches (SW1 − SW8), we have different performance. In standby mode, SW1 = SW2 = ON and SW3 = SW4 = OFF, while the CMI-VSA latches the previous result at its internal nodes. During sensing operations, WL signals are triggered to develop the VBL in the SRAM array and VREF in the DIARG scheme. For a given BL developing time, CMI-VSA is enabled to implement the three phases. In PH1 (voltage development), SW3 = SW4 = ON and SW1 = SW2 = OFF to force the two inverters into an auto-zero state. This biases the CL and CR nodes at their respective trigger points.

In PH2 (pre-amplification), setting SW1 to SW4 = OFF, puts $V_{CL}/V_{CR}/V_{AL}/V_{AR}$ in a floating state. Setting SW5 = OFF and SW8 = ON switches V_{INL} from V_{BL} to V_{REF} and then couples ($V_{BL} - V_{REF}$) to V_{CL} through C1, such that $V_{CL} = V_{TRP-L} - (V_{BL} - V_{REF})$. Setting SW6 = OFF and SW7 = ON switches V_{INR} from V_{REF} to V_{BL} and then couples ($V_{REF} - V_{BL}$) to V_{CR} through C2, such that $V_{CR} = V_{TRP-R} + (V_{BL} - V_{REF})$. This increases the voltage difference (V_{INV}) between V_{CL} and V_{CR} to $2 \times (V_{BL} - V_{REF})$.

In PH3 (amplification), setting SW1 = SW2 = ON enables the two inverters to amplify the inverter voltage difference in order to generate a full swing for V_{CL} and V_{CR}.

In the end, the waveform that we made is quite similar to the IEEE paper. We even made a better sense amplifier compared to conventional sense amplifiers which can achieve an offset of only 36mV and still function correctly when tested on the 1024 Monte Carlo test.

3) DIARG

The dynamic input-aware reference generation (DIARG) scheme that comprises two reference columns of fixed-zero replica memory-cells (F0RC), a WL-combiner (WLCB), and a reference-WL-tuner (RWLT). The DIARG scheme for MBNN generates an appropriate reference voltage V_{REF} based on the number of input = 1. When *m* WLs are activated, *m* corresponding WLCBs are enabled. Then, *m* number of I_{MC-C} are generated on the BLL and *m* number of I_{MC-D} are generated on the BLR. By this means, the reference voltage is generated based on the number of input = 1 values.

4) WL Driver

For the WL driver, initially we tried to use a 6 to 64 decoder, but later we decided to let multiple input WL open at the same time, since in each operation multiple WLs can be activated at once to compute the input-weight product. In order to let multiple input WL open at the same time we use eight 3 to 8 decoders to achieve our goal.

5) WLSW

WL-selections switch (WLSW), which activates left-sensing mode if AF = 1 by asserting only the WLLs of the selected rows, when all WLRs are disconnected. Similarly, when AF = 0, it activates the right-sensing mode, and asserts the WLRs of the selected rows, while all WLLs are disconnected. This way, we could achieve the algorithm-dependent asymmetric control (ADAC) scheme from the WL side.

6) SRAM Array

The 4K SRAM array is required to store a huge amount of intermediate data. In this simulation, we use a 64 by 64 SRAM array, namely a 4Kb SRAM, which also contains a header for each BLL and BLR, and a BLSW and VSA for each column. When using a conventional 6T SRAM array, both of the pass-gates (PGL and PGR) were simultaneously activated by the same word-line, such that BLL and BLR both consumed current for product-sum operations. In other words, we cause unnecessary waste of power resources, thus lowering the energy efficiency. To overcome waste of power, we use DSC6T SRAMs with only one pass-gate turned on, reducing the average current and power consumption of the DSC6T SRAM-CIM by 46.5%, compared to a conventional 6T SRAM array. Later, we design the SRAM layout and will be shown later.

7) BLSW

Each of the BLLs is connected to its corresponding VSA when AF = 1 via a bit-line selection switch (BLSW), whereas BLR = VDD is isolated from VSA. Then, VSA detects V_{BLL} and directs its output (SAOUT).

8) WLCB

The WL-combiner (WLCB) where *n* inputs are activated, *n* corresponding WL-combiners are enabled. So we can view it as computing how many inputs are open.

C. DSC6T Structure Layout

We use the unidirectional method to draw the layout of the SRAM cell, which lets the poly-gate go horizontal. In this way, a similar process variation is achieved, and the layout is also presented as a rectangle. By presenting it this way, it makes it easy to arrange the cells. The contacts on the four sides of the layout can be shared with other cells on the top, bottom, left, and right side to reduce layout area.

We put dummy cells around the 16 by 16 SRAM cell array. The SRAM array, with some dummy cells which are roughly the same as an SRAM cell. However, the dummy cell's Q is connected to VDD and QB is connected to VSS to avoid floating voltage. If we let all originally BL and BLB in layer-1 moved to layer 2 and use the 3D via to connect layer-1 and layer-2. The area of the SRAM without putting BL and BLB is 5440 μm^2; and the one putting all metal two (BL and BLB) is about only 4780μm^2. The 6T SRAM bit cell has 12% footprint area advantages over the 6T 2D SRAM bit cell.

D. Write Margin

Set the initial values of Q and QB as Q=1 and QB=0, and set BL from VDD to 0. When the voltage values of Q and QB are equal, then BL voltage is the write margin.

When PN ratio size becomes bigger, then the standard 6T SRAM WM comparison against DSC6T SRAM WM becomes bigger too. According to the measurement, standard SRAMs are 55% harder to write compared to the SRAMs using DSC.

E. Power Consumption

Define VDDA is the voltage between CVDD1-CVDD2, and with a bigger P/N and W/L ratio, we achieve 63% and 56% of power consumption compared to standard SRAM, respectively. Using DSC scheme that one side VDD is lower than other side without affect SRAM function, it capable of lowering overall SRAM power consumption. This can be seen in Fig.4.

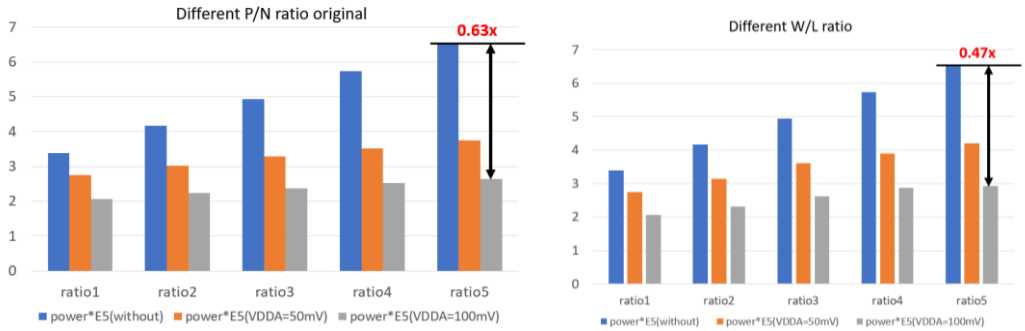

Fig. 4. Power Consumption of DSC6T SRAM with different Ratios

F. Hold Static Noise Margin

Disconnect the circuit that is originally connected to the two inverters, add the voltage source and sweep the voltage from 0 to VDD. Set WL to 0 to measure the hold static noise margin (SNM), and wait for the measured Q and QB to flip, then the voltage source is hold SNM. The higher the value, the stronger the noise resistance is.

Using 1024 Monte Carlo, we find that in all corners at 25°C, the best case is slow NMOS and fast PMOS (SF) corner, which has a minimum of 328mV and a maximum of 394mV. Because PMOS uses the smallest size, a faster PMOS helps to increase its strength, making the strength of PMOS and NMOS equivalent, which is better for the overall circuit stability. However, the worst case is fast NMOS and slow PMOS (FS) corner, which has a minimum of 293mV and a maximum of 368mV. Because stronger NMOS and weaker PMOS make the overall circuit stability worse and also make the anti-noise ability worse.

G. Read SNM

Disconnect the circuit that is originally connected to the two inverters, and add the voltage source and sweep the voltage from 0 to VDD. Set WL to 1 to measure the read static noise margin (RSNM), and wait for the measured Q and QB to flip, then the voltage source is RSNM.

Using 1024 Monte Carlo, we find that in all corners at 25°C. The best case is SF corner, which has a minimum of 125mV and a maximum of 217mV. The worst case is FS corner, which has a minimum of 43mV and a maximum of 136mV. Because the stronger NMOS and weaker PMOS make the overall circuit stability and the anti-noise ability worse. The reason for the worst and best RSNM is the same as HSNM, and RSNM has WL open, so BL and BLB may affect the internal stability of SRAM, resulting in the SNM in reading being much lower than the SNM in holding.

H. Sense Amplifier Comparison

The sense amplifier that we implemented in this work is better than conventional ones in terms of the offset voltage. As mentioned earlier, the CMI-VSA can work with only an offset voltage of only 36mV and output successfully, while conventional ones might need an offset voltage of 4x to 5x of the CMI-VSA. In Fig. 17, we test out both sense amplifiers with an offset voltage of 36mV and run 1024 times Monte Carlo simulation. For the conventional sense amplifier, many simulations do not sense correctly, whereas the CMI-VSA does not produce any error.

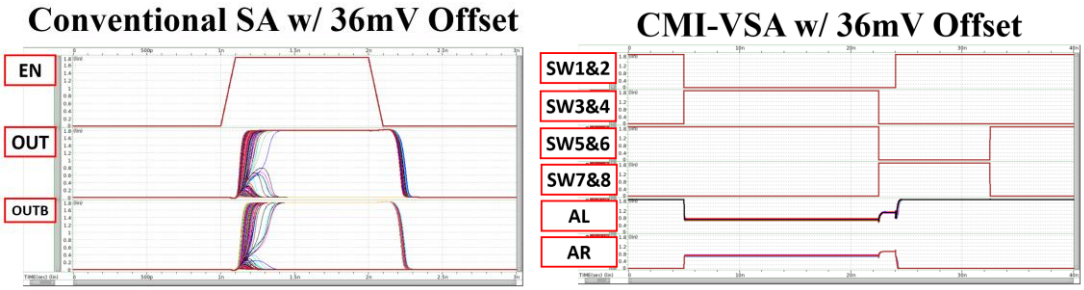

Fig. 5. Results of conventional SA and CMI-VSA with offset voltage of 36mV

心得感想

經過一整年孟凡教授專題扎實的訓練，從一開始的文獻閱讀與上台簡報，向大家彙整重點，並討論各領域文獻的優缺點與可解決問題的方向。最後再模擬文獻中所提及的電路，以此為基本架構，再自己創新電路架構，以達到改善文獻中的缺點或達到更好的電路表現。

在這次的獨立研究中，學習到了許多寶貴的經驗，例如在有限的時間內，增加文獻閱讀的速度與技巧，並快速理解各篇文獻報告中的重點，熟悉建構與模擬電路的工具並減少除錯的時間，同時培養訓練獨立思考與創新與解決前所未有的問題的能力，然而，研究過程中遇到各種挑戰與問題，很多都需要進行取捨，如調大靜態隨機存取記憶體的電容可能導致 bump 的 waveform 越大，但放大器的運作可以在更小的偏差電壓（offset voltage）下運作，這就是設計電路架構所面臨的取捨 (tradeoff) ， 我們在設

計中也經過了多次修正電路，並更改些周圍（peripheral）的電路架構與開關（switch）的設計與較複雜的電路放大器（sense amplifier）架構，與隊友和 mentor 的指導下，最終得到超乎預期的結果–雖然製成不同，但只比文獻中提及的 20 到 50 mV 下還要大一些的偏差電壓，讓放大器能有更好的放大效果和創新的兩個 VDD，來節省整體 64*64 靜態隨機存取記憶體功率的架構。儘管在這長時間的獨立研究過程中遇到不少難題與挫折，但克服這些問題後學到的也更多，著實感謝這段時光的自我淬鍊。