

Computing-in-Memory with ReRAM

以電阻式記憶體進行內記憶體運算

李明哲

指導老師:張孟凡教授

Abstract

In this AI era, the conventional processors based on the von Neumann structure face challenges. Traditionally, data is transferred between the memory and the processing elements when doing calculation. The data will move across a long distance, which lowers the calculation speed and consumes additional power. However, computing-in-memory (CIM) is able to overcome this issue. CIM allows parallel computing, and performs multiply-and-accumulate (MAC) operation in a short period of time. With such advantages, it therefore overcomes the bottleneck that we face in the von Neumann structure. In this work, I propose a 4 by 4 ReRAM array and peripheral circuits. It performs the MAC operation of 4-b-input and 4-b-weight. By the HSPICE simulation, the sensing margin between the MAC values is 40mV.

Introduction

此篇主要架構提出一 4 by 4 ReRAM array。其能夠執行 4-b-input, 4-b-weight 的 multiply-and-accumulate (MAC) 運算。最多可有 4 條 WL 同時開啟，權重則是被儲存於 array 中。4-b-input 被拆成 2 組 2-b-input 在兩個週期中做運算。在每一個週期中，在 bit line(BL)提升至高電位的狀況下，當 word line (WL)為 HIGH，進而導通 BL 對地路徑時，電阻上有一跨壓產生，進而產生電流。因此，我們以 WL 開啟的時間長度，來代表 input 的大小。每條 BL 之下接有電容，運算後得到的電流會對電容充電，最後，我們去判斷電容上的電壓值以得出相應的 MAC value。

Fig. 1. 為 block diagram。說明了系統架構包含 array 與相配應的周邊電路。

Fig. 2. 為 array 示意圖。運算結果的電流將累加在 BL 上。

Fig. 3. 為模擬波型，圖中可見電容依據不同 input pattern 與運算結果充電。

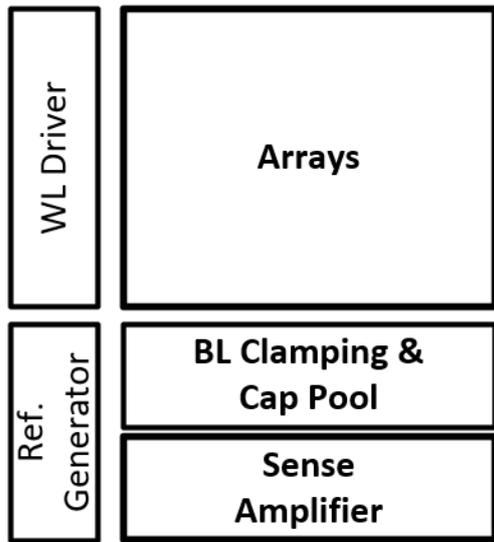


Fig. 1. Block Diagram.

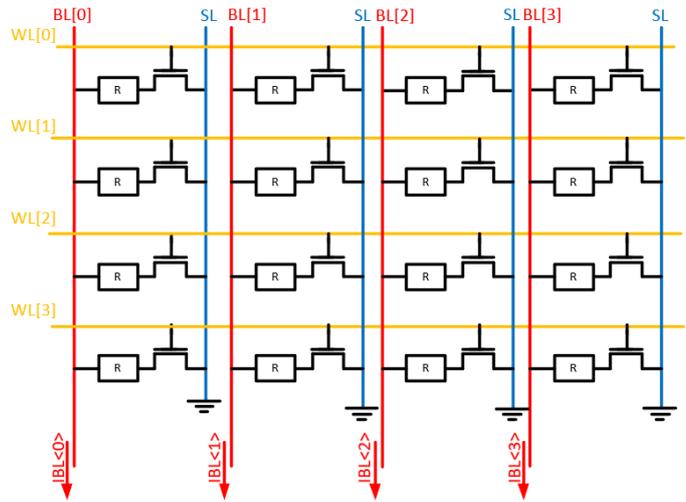


Fig. 2. ReRAM array.



Fig. 3. Simulation waveform

心得感想

這次實作專題，是我第一次建構如此有規模的電路。一開始很單純地想說，只是照著論文上相同的架構重現一遍，應該不會有太大的問題。但實際做起來才發現困難重重。國際上知名的論文往往都是微言大義，許多細節並沒有具體地寫在論文當中，需要自己細心琢磨後才能發現。因此，我往往在閱讀論文、發現不懂的細節、找學長弄懂，這些步驟間循環。最後，建構起整個完整的電路便花上我許多時間。而當建完電路，開始跑模擬後，跑出來的數據卻與論文上的結果有一段差距。這時便是最需要腦力激盪的時刻，必須重頭去分析整個電路是哪裡出了問題，並嘗試提出解決辦法。一路下來，這樣的過程訓練了我分析電路的能力。最後，對於指導教授 張孟帆老師與實驗室學長們這一年下來不辭辛勞的耐心指導，由衷感謝。