# 神經群仿生神經網路處理器

# Population-based Neuromorphic Spiking Neural Network Processor

組別：B139

指導教授：鄭桂忠 教授

組員姓名：古庭羽、宋明軒、蔡承峻

## ABSTRACT

隨著深度學習快速發展，運算量和功耗都成指數成長，對於邊緣型應用裝置需要專用硬體來降低功耗，脈衝神經網絡可以在降低功耗的情況下，能夠達到與深度學習相近的準確度。脈衝神經網絡（Spiking neural networks) 被譽為第三代人工神經網絡，在設計上較深度學習模型更具仿生特性。

本專題旨在實作神經群仿生神經網路處理器以及其應用，第一部份 SNN 硬體設計，為重做論文[1]之電路，第二部分 SNN 應用，為根據設計出的電路構思其應用，並實測其準確性及效能。硬體部份在設計上使用兩顆神經群架構，使用神經處理單元(NPU)分別為 NPU1 及 NPU2，在神經元的設計上以二次方程累積放電脈衝(QIF) 神經元模型為雛形，將膜電位線性化，設計 I-QIF 神經元，並考量到權重稀疏性，設計稀疏編碼分組，跳過不必要的計算。

突波神經網路處理器在神經元數量限制下，可以應用各式的突波神經網路，在本專題研究的最後將神經群仿生神經網路應用在迷宮尋找最短路徑上，並能達到 100%正確率。此外，可以針對不同應用設計不同神經群連接模式並設定不同權重，例如應用在果蠅視神經避障系統。且處理器經比較後發現優於現今具代表性的 SNN 處理器，證明本研究設計出從仿生角度上，在應用端中是最具效率的神經群突波神經網路處理器。

# INTRODUCTION

## 1. 系統設計

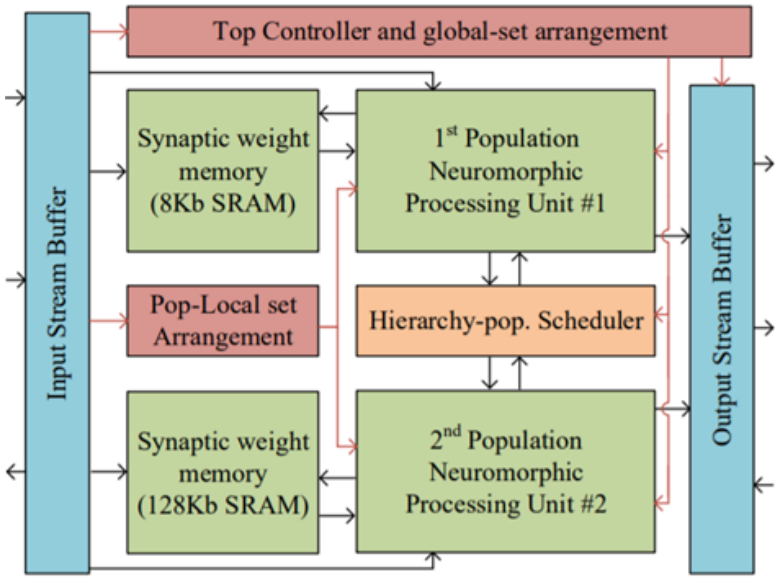


Fig.1 整體電路架構圖[1]

整體的電路架構如 Fig.1，大致上可以分成三個部分，第一部份(藍色)是輸入輸出暫存器，第二部份(紅色)是區域暫存器，第三部分(綠色)是神經處理單元。我們將參數設定以及權重透過外部接角輸入到輸入暫存器，輸入暫存器會透過資料的位址(address)儲存到指定的區域暫存器或 SRAM 內，當資料輸入完畢，主控制器會進入「時間步」階段，在這個階段神經處理單元會被啟動，並透過區域暫存器儲存的資料讀取神經元參數，再進行該時間步的突波確認、神經突觸衰減確認以及神經元膜電位與輸出突波更新，計算完成後會在每個時間步的最後透過輸出暫存器輸出當個時間步的突波結果，並重複執行時間步，直到外部系統結束訊號為 1。

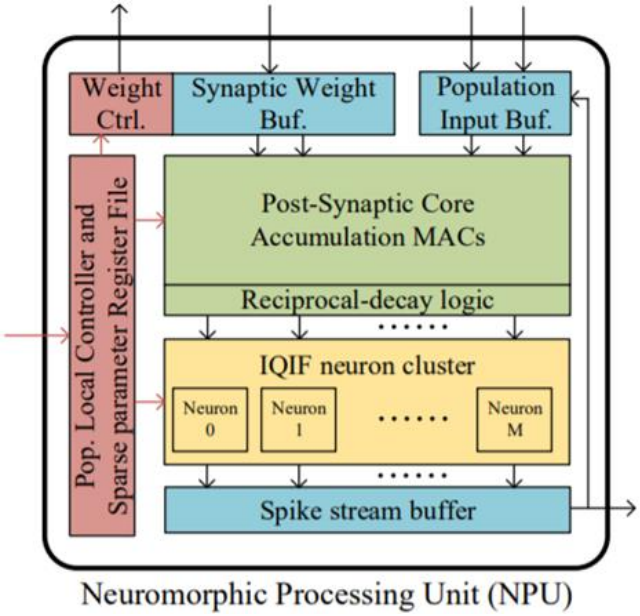


Fig2. 神經處理單元架構圖[1]

當神經處理單元被啟動，會先透過突波解碼器做外部輸入以及網路內部突波確認，神經處理單元控制器會透過突波編碼以及稀疏分組編碼向 SRAM 讀取一組權重，累加到後突觸暫存器，若該組權重全為零，則跳過該組計算節省時間，在設計上模擬因神經疲乏導致的突觸電位衰減，根據當前時間步判斷是否需要做衰減，最後將後突觸膜電位輸入到各個神經元，做神經元膜電位累加，若神經元膜電位超出閥值，則會輸出突波，透過突波暫存器連接到不同神經網路上。

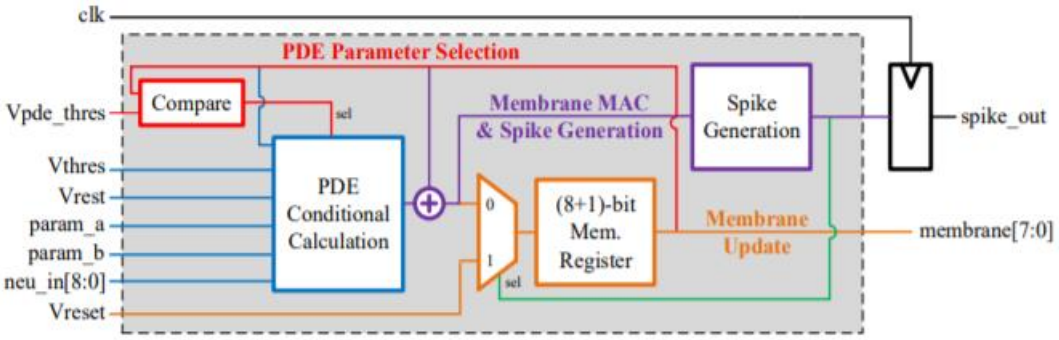


Fig.3 IQIF 神經元電路圖[1]

神經元透過神經參數設定以及後突觸輸入電流計算偏微分方程式，在每個時間步將偏微分方程的運算結果加到膜電位上，若相加膜電位超過閥值則輸出突波並將膜電位設定為預設值，若小於閥值則維持相加後之膜電位並不輸出突波。

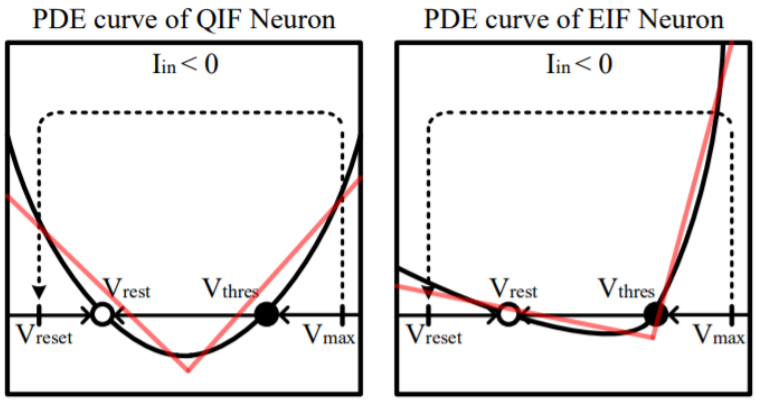


Fig.4 神經元膜電位偏微分方程式線性化[1]

參照 QIF 以及 EIF 神經元膜電位偏微分方程式，其偏微分方程式的趨勢線如 Fig.4 所示，可以透過細胞膜電位中的靜息膜電位$V_{rest}$以及閾值膜電位$V_{thres}$的中間值，並將左右兩邊趨勢線性化，寫成如下式之線性方程。

$$v'[t] = \begin{cases} I[t] + a(v_{rest} - v[t]), & \text{if } v[t] < v_{pde_{th}} \\ I[t] + b(v[t] - v_{thres}), & \text{if } v[t] \geq v_{pde_{th}} \end{cases}, \text{if } v \geq v_{peak}, \text{then } v \rightarrow v_{reset}$$

其中，$a$、$b$、$V_{rest}$、$V_{thres}$、$V_{reset}$、$V_{pde_th}$為可由外部設定之神經元參數，$V_{peak}$為神經元固定參數，I[t]為神經元輸入電流，此方程式即為神經元架構中的 PDE calculation module，於每個時間步，會做一次偏微分方程式以及膜電位的更新。

在後突觸衰減方程中，每隔固定的時間步便需要做衰減，衰減方程為減去其倒數後的值，為降低複雜度，把倒數的值設定成必須為二的次方，使得
倒數可以透過位移器就能做到，同時為了避免衰減消失(decay-vanish)，將式子設計成至少需要扣除$decay_{min}$，使的算式如下所示。

$$syn[t] = syn[t-1] - SEL(syn[t-1] \gg decay\_\alpha, decay_{min})$$

## 2. 實驗結果

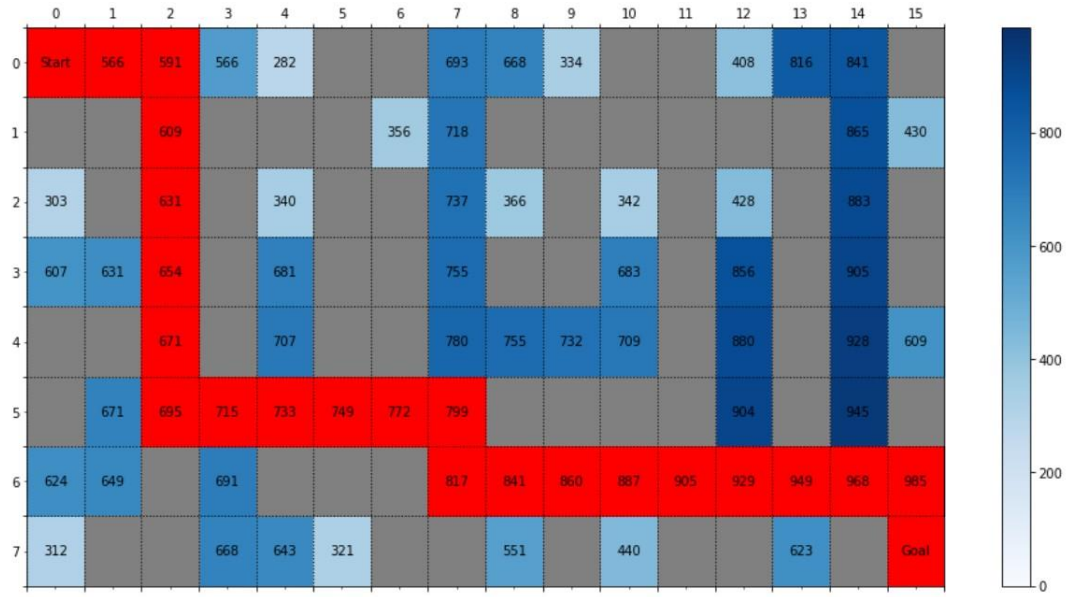


Fig.5 SNN 尋找迷宮最短路徑實驗結果

本專題研究最後透過 spiking neural network processor 尋找迷宮的最短路徑，使用 Jupiter notebook 以及 Python 軟體來模擬硬體行為，並將參數設定如下。

$$
\begin{aligned}
NPU2 \;=\; \{&"num": 128,\\
&"num\_ext\_spike": 32,\\
&"ext\_cur": ext\_cur2,\\
&"global\_neuron": global\_neuron2,\\
&"Vthres": 200,\\
&"Vrest": 100,\\
&"a": 1,\\
&"b": 4,\\
&"Vpde\_th": 180,\\
&"Vreset": 50,\\
&"Vpeak": 240,\\
&"alpha": 2,\\
&"decay\_cnt": 3,\\
&"weight\_addr": "../file/weight2.txt"\}
\end{aligned}
$$

$$global\_neuron2 = \{"ext\_cur": 0,$$
$$"Vthres": 200,$$
$$"Vrest": 100,$$
$$"a": 1,$$
$$"b": 4,$$
$$"Vpde\_th": 180,$$
$$"Vreset": 50,$$
$$"Vpeak": 240,$$
$$"alpha": 2,$$
$$"decay\_cnt": 3\}$$

$global_neuron2$為 NPU2 之全局神經元參數設定。

$$ext\_cur2 = maze\_generator()$$

使用$maze_generator()$來產生隨機地圖。

由於此應用不會使用到 NPU1，因此將 NPU1 關閉，參數設定為 0，只設定 NPU2 的 128 顆神經元，並且 ext_cur 為神經元外部刺激，目的為使神經元被激活或是抑制，抑制神經元為迷宮中的牆壁，激活神經元為迷宮中的終點，因此透過設定 ext_cur 可以產生迷宮地圖，我們使用 maze_generator()這個 function 來產生隨機迷宮地圖，並且轉換成 ext_cur 對應數值設定。

$$SNN = SNN(NPU1, NPU2)$$

最後將參數設定讀到自訂義之 SNN class 裡，將 variable 命名為 SNN。

$$SNN.maze(step = 2000)$$

參數設定完成後跑上圖之 function，並且將 step 設定為 2000，意思是跑 2000 個時間步。迷宮運作原理為持續刺激終點位置的神經元，使之產生向四周擴散的突波，並且抑制牆壁位置的神經元，使之不會受到外界刺激，因此相鄰的神經元不會透過牆壁傳遞突波。從 Fig.5 可以看到每個方格內都有數值介於 0 到 2000，意義是在這 2000 個時間步裡，該方格所對應的神經元產生了幾個突波，數字越高代表神經元產生突波頻率越高，實驗結果證明從起點出發沿著突波頻率最高之相鄰神經元走，可以找到最短路徑，對於任意迷宮地圖皆可達到 100%正確率。

## 3. 結論

在這個專題中我們使用兩顆神經群架構，總共 32 加 128 顆神經元，並且將 SNN 應用在尋找迷宮最短路徑上。若是改變神經群連接模式以及設定不同權重，將可以使用在不同的應用場景下，譬如設計從果蠅視覺神經研究設計出避障決策神經群突波網路，或是應用在手寫辨識、影像辨識。

在電路設計方面，因為在有限的時間內只有設計一個版本的電路，timing 只有壓到 10ns，若是時間允許，我們將會在電路的面積及效能方面做出改善，設計多個版本的電路架構，最後找到最符合效能及成本考量的電路。

總結而言，在應用方面，突波神經處理器具有設計的靈活性，以神經群為一個單位，透過不同神經群的連結模式，可以提高網路複雜度，達到不同應用，在效能方面，突波神經網路可以在相近準確度下達到比深度學習模型更低的功耗，在邊緣型應用裝置可以達到低功耗高準確度的應用，並且因為突波神經網路具有仿生 biomimetic 特性，因此在系統神經科學上可以模擬大腦神經行為，具有科研價值。

# 心得感想

經過一整年專題的訓練，從一開始閱讀學長的論文，跟助教每周討論，並討論各個 module 的功能與可以實作的方法，最後再用硬體描述語言 Verilog 將論文中的各個 module 實做成硬體，並且改良硬體的效能以及面積，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，例如模擬結果和軟體驗證結果不同時，要學會善用手邊的 tool，從眾多 module 跟 signal 間抽絲剝繭，找到問題的源頭，並且想到一個好的 solution 解決問題，並且使用最少的成本。通常從找到問題到解決問題需要花費很多時間與心力，但經過不斷不斷的練習，可以增進解決問題的能力，並且減少 debug 的時間，下次在設計電路前會更加注意，避免犯過的錯誤重複發生。同時隊友跟助教也是很好的老師，多跟他們討論可以激發更多想法，刺激自己不斷成長。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。

# Reference

[1] Zuo-Wei Yeh et al.,"POPPINS : A Population-Based Digital Spiking Neuromorphic Processor with Integer Quadratic Integrate-and-Fire Neurons"