

神經群仿生神經網路處理器

Population-based Neuromorphic Spiking Neural Network Processor

組別：B348

指導教授：鄭桂忠 教授

組員姓名：陳楚文、簡毓寬

Abstract

在當今的電腦科學領域中，人工智慧的硬體設計在運算能耗上仍遠不及生物體神經系統。近年來，為了讓硬體設計的效能接近生物體神經系統的能源效率，突波神經網路成為值得關注的發展方向。這些神經模型以神經群以及神經元為基本單位，並通過二值化的突波訊號、網路之稀疏性等特徵，達成降低功耗的目標。

本專題研究以「搭載整數化二次方程累積放電脈衝神經元模型之可調控神經群模型的突波神經網路處理器」論文[1]為基礎，實作了基於脈衝神經網路(SNN)，可調控神經元之種類、連接方式之處理器。

在神經元的設計方面，我們實現了整數式二次方程累積放電脈衝(I-QIF)神經元模型，以及漏電式累積放電脈衝(LIF)神經元模型。此外，我們採用可調控之連接，可實現具有循環(Recurrence)路徑與自我連結，以及前饋(Feed-Forward)之全連接模式，以最佳化在不同應用時的計算靈活性；並透過稀疏分組編碼跳過不必要的連接計算，從而提高了運算的速度。我們還實作了遞減式突觸衰減邏輯，能夠模擬突觸疲勞等生物機制。在神經網路的外部輸入方面，除了有原論文[1]所採用的電流(Current)形式之外部刺激輸入(External input stimulus)，也備有 256 位元之脈衝輸入，能夠接收外部編碼後的脈衝，並能以全連接層(Fully connected layer)，連接到第一神經群，實現具有輸入層、隱藏層、輸出層的多層感知機(MLP)架構。

最後，處理器以 TSMC 90nm 的製程進行合成，其結果可在頻率 125MHz 下進行運算。此外，在本實作專題的應用設計當中，以 MNIST Dataset 為例，將原始灰階圖片降為 16X16，並以 Poisson 編碼，轉為 256 位元之脈衝訊號，實現 256 個輸入神經元、32 個隱藏神經元、10 個輸出神經元的 MLP，並達到 95%之準確度。

Background

近年來，隨著半導體製程的進步，以及數位晶片運算能力的提升，人工智慧(Artificial Intelligence)也隨之快速發展，並受到各個領域的重視。例如，近年來出現的 ChatGPT，就是人工智慧在自然語言處理的機器學習技術下的產物。此外，在其他產業當中，人工智慧也被廣泛地應用，並掀起一股席捲全球的研究熱潮。

在這股熱潮下，各種不同可應用於硬體運算的仿生神經網路模型也相繼被提出，包括：人工神經網路(Artificial Neural Network, ANN)、循環神經網路(Recurrent Neural Network, RNN)、以突波訊號為主的突波神經網路(Spiking Neural Network, SNN)、深度神經網路(Deep Neural Network, DNN)，以及運用卷積運算網路架構的卷積神經網路(Convolutional Neural Network, CNN)。其中，卷積神經網路(Convolutional Neural Network, CNN)具有高度的潛力，被廣泛應用在影像辨識領域中。

然而，在邊緣裝置(Edge Device)的應用中，需要面臨低功耗、高效率等多方面的挑戰；相較於其他神經網路，本實作專題應用之突波神經網路(SNN)，具有以下幾點優勢：

1. 二值化(Binarization)訊號傳遞

突波神經網路中，突波訊號為二值化(Binarization)之數據，僅以 1 與 0 表示載突波事件的發生與否。因此，在全連接層(Fully-connected Layer)的運算中，僅需加法運算，在硬體上能夠大幅節省晶片面積與功耗，並降低延遲。

2. 稀疏性(Sparsity)

突波訊號及神經網路中的權重(Weight)具有稀疏性(Sparsity)，透過稀疏分組編碼(Sparse-group instruction code)處理下，在進行全連接層的運算時，能夠大幅減少所需的時鐘週期(Clock Cycle)數量，使得運算所需時間得以大幅降低。

3. 事件驅動系統 (Event-driven) 特性

累積放電(Integrate-and-Fire)型神經元，只有在神經元膜電位累積超過閾值，才會發出突波訊號；因此，突波神經網路為事件驅動系統 (Event-driven)，相較之下可節省整體功耗。

儘管突波神經網路(SNN)在複雜影像之辨識正確性，難以與卷積神經網路(CNN)相比擬，突波神經網路對運算資源的要求更低，且有事件驅動之特性，因此能實現低功耗與低延遲。在邊緣裝置的領域中，如：微型感測器、無人機等，突波神經網路就具有極大的潛力。

System Architecture

1. 處理器架構

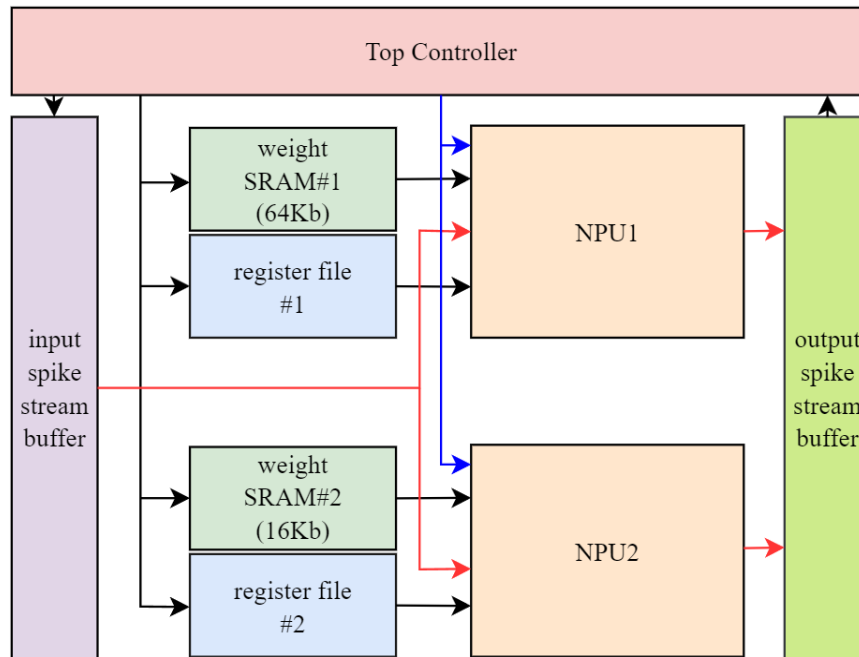


Fig. 1 處理器之方塊圖

1. 輸入與輸出序列暫存器(Input/Output Stream Buffer)

輸入序列暫存器負責接收外部訊號，並利用位址(Address)編碼取得突觸之權重；而輸出序列暫存器在運算結束後輸出每個神經元的突波行為。

2. 主控制器(Top Controller)

主控制器由有限狀態機(FSM)來控制處理器的整體流程，包括外部訊號輸入參數、整體運算流程、時間步(Timestep)的計算與控制。

3. 神經群處理單元(Population Neuromorphic Processing Unit, NPU)

神經群處理單元已於 2-2 節介紹，包括突波輸入暫存(Input spike stream buffer)、後突觸權重累加控制核心(Post-Synaptic Accumulation Core)、突觸邏輯(Synapse)、神經群(Neuron Cluster)、突波輸出暫存(Output spike stream buffer)。

4. 突觸權重記憶體(Weight SRAM)

突觸之權重資訊儲存於 SRAM 當中，而兩個 SRAM 各對應兩個神經群處理單元(NPU1 及 NPU2)，以進行後突觸累加(Post-Synaptic Accumulation)計算。

5. 稀疏編碼暫存器(Register file)

稀疏編碼資訊儲存於暫存器當中，以利神經群處理單元運算時存取。

2. 神經群處理單元(NPU)架構

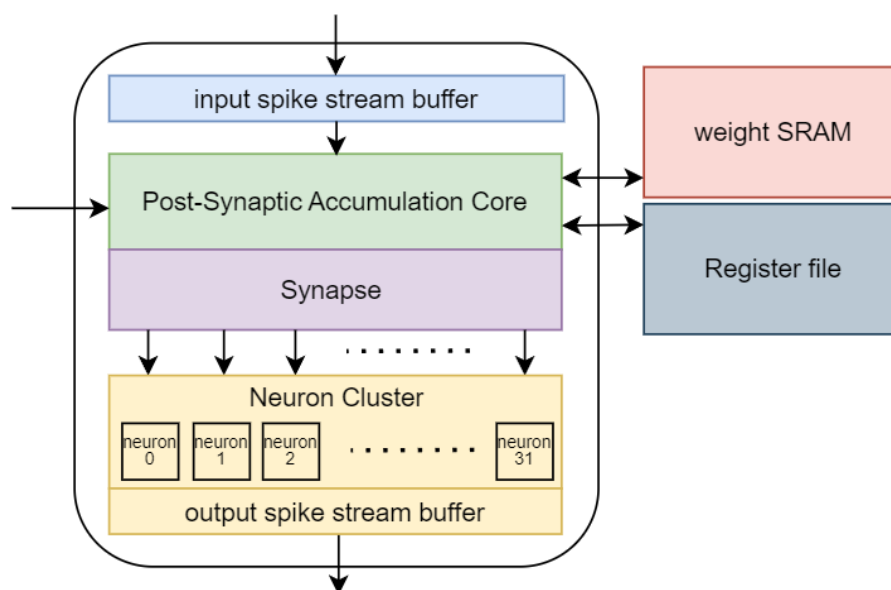


Fig. 2 神經群處理單元之方塊圖

神經群處理單元(NPU)包含了以下的構造:突波輸入暫存(Input spike stream buffer)、後突觸權重累加控制核心(Post-Synaptic Accumulation Core)、突觸邏輯(Synapse)、神經群(Neuron Cluster)、突波輸出暫存(Output spike stream buffer)。其中,突觸邏輯(Synapse)與神經元(Neuron)邏輯為上述所提及,各有 32 份,對應到神經群處理單元(NPU)中的 32 個神經元。此節會重點講解後突觸權重累加控制核心(Post-Synaptic Accumulation Core)。

後突觸權重累加控制核心(Post-Synaptic Accumulation Core)的目的為控制並加速連接層的計算,因突波神經網路(SNN)之權重與突波輸入的分布,皆具有稀疏性的特質,將每個權重讀出來並進行累加相當浪費時間;因此,當權重不需進行累加時(權重為 0 或是對應之突波為 0),此控制核心可以跳過並大幅加速此部分的運算。

此外,欲達到跳過以節省時間的目的,後突觸權重累加控制核心(Post-Synaptic Accumulation Core)需讀取存於暫存器堆(Register file)之稀疏編碼指令。由於 SRAM 之單個位址只能存 8 個突觸權重,因此把每 8 個神經元設定為一組,並以此組為基礎,設定相關稀疏性指令編碼,而指令編碼儲存在 NPU 對應之暫存器堆(Register file)。

3. 神經元模型

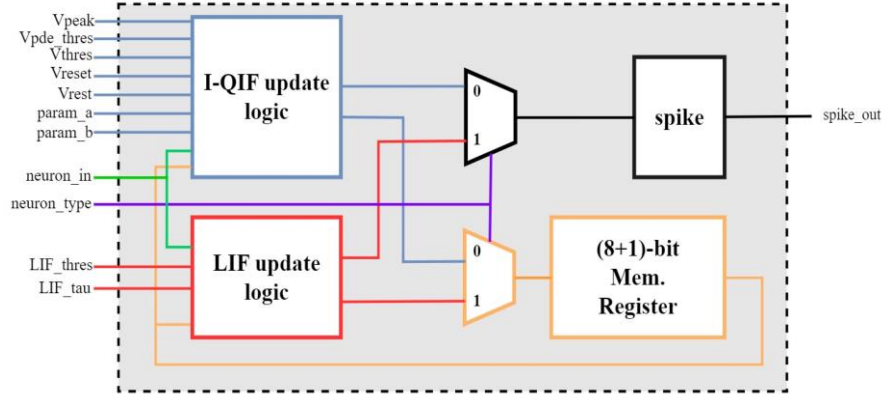


Fig. 3 神經元模組之方塊圖

(1) 整數化二次方程累積放電神經元(I-QIF) [1]

(此部分為參考論文[1]之敘述，並以硬體描述語言實現)

二次方程累積放電脈衝(QIF)神經元膜電位方程式，是一種神經元行為表現之數學模型。其方程式可以透過線性化降低其複雜度，實現於硬體端的整數運算上，並增加運算速度，因而將其應用於此專題中。以下為 QIF 之膜電位方程式：

$$V' = I + a(V - V_{rest})(V - V_{rest}), \text{ if } V \geq V_{peak}, \text{ then } V \rightarrow V_{peak}$$

為應用於硬體端，需將線性化後之公式離散化，如以下式子：

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

I-QIF 神經元電路在一個時間步中，首先根據膜電位之大小，判斷要使用參數 $\{a, V_{rest}\}$ ，又或是 $\{b, V_{thres}\}$ 計算下一時刻膜電位的變化值。接著，在得出膜電位變化值 $V'[t]$ 後，計算出下一個時刻的膜電位。最後判斷下一時刻的膜電位是否超出 V_{peak} ，來決定是否發出突波訊號，及下一時刻更新的膜電位是回到 V_{reset} 抑或是原值。

(2) 漏電式累積放電脈衝(Leaky Integrate-and-Fire, LIF)神經元

(此部分為本實作專題新增)

LIF 模型較為簡單，但亦不失生物特性。神經元模型動態方程如下所示：

1. 膜電位漏電(leakage)計算

此處的 I 為神經元的外部電流輸入， V 則是膜電位， τ 是影響漏電速度的時間常數；依以下的公式計算出下一個時刻的膜電位 V_1 。

$$V_1[t] = \frac{1}{\tau}(I[t-1] + V[t-1])$$

2. 脈衝(spike)輸出判斷

若 V_1 大於設定之 V_{th} ，則輸出脈衝(spike)；反之則不輸出脈衝。

$$spike = \begin{cases} 1, & V_1[t] > V_{th} \\ 0, & else \end{cases}$$

3. 膜電位更新

然而，膜電位範圍為 $0 \sim V_{th}$ 之間。因此，將 V_1 限制(clamp)於此範圍，此即下一時刻更新的膜電位 V 。

$$V[t] = \begin{cases} V_{th}, & V_1[t] > V_{th} \\ V_1[t], & 0 \leq V_1[t] \leq V_{th} \\ 0, & V_1[t] < 0 \end{cases}$$

若將 τ 設為2的幕次方，則僅需加法器、比較器與位移器，而不需乘法器，即可實現神經元中的運算，對硬體資源的使用十分友善，因此在本實作專題的設計中，我們將其額外加入，在不使用太多面積的情況下，增加了應用上的彈性。

4. 突觸(Synapse)設計

(1) 主要架構設計

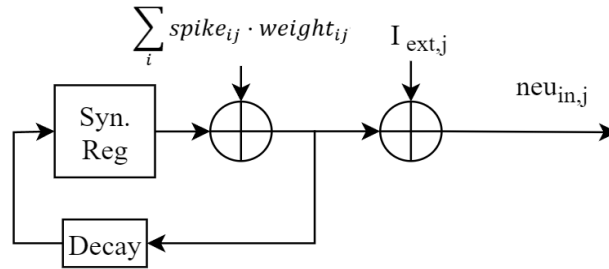


Fig. 4 突觸邏輯之方塊圖

為模仿生物體的突觸行為，於神經元連接間的突觸實作了突觸記憶及衰減邏輯，如以上方塊圖所示。方塊圖中，一開始的輸入為，神經群之輸入突波經全連接(Fully-connection)的運算後，對應神經元的答案輸入。突觸的輸出則為突觸輸入、突觸記憶、及外部電流之總和，輸出至上述神經元硬體模型中的電流 I 。

$$neu_in_j[t] = I_{ext,j} + decay \left(syn_j[t-1] + \sum_i spike_{ij} \cdot weight_{ij} \right)$$

此外，在這之間的突觸記憶，是透過將上一個時間步的輸入及記憶值進行衰減後，再將其暫存起來，成為下一個時間步的記憶值。

$$syn_j[t] = decay \left(syn_j[t-1] + \sum_i spike_{ij} \cdot weight_{ij} \right)$$

Experimental Results

演算法與硬體驗證

我們使用 Python 語言實作演算法，考慮到情況，我們使用多筆不同參數、狀態下的輸入資料，與 Verilog 實作的硬體進行輸出結果之比對；兩者比較後，在每筆資料中，皆得出完全一致之結果，演算法已正確地透過硬體實現。

數位電路合成

使用 Synposys Design Compiler 進行硬體的合成，製程為 TSMC90GUTM，Cycle time 為 8 ns (125MHz)，Total Cell Area 為 393259.1 μm^2 ，Total Power 約為 6.1804 mW。

應用

為展示我們所新增的 MLP 連接模式與 LIF 神經元，我們使用 MNIST 手寫數字資料集先將原始資料轉為 16*16 之圖像，並以 Poisson Encoding 與 flatten，轉為 256-bit 之脈衝(spike)，作為神經網路的 input layer。使用第一神經群處理單元(NPU1)的 32 個神經元作為隱含層(hidden layer)，第二神經群處理單元(NPU2)中的 10 個神經元，作為神經網路之輸出層(output layer)，全部皆使用漏電式累積放電脈衝(LIF)神經元模型；構成 256*32*10 之三層前饋式脈衝神經網路。

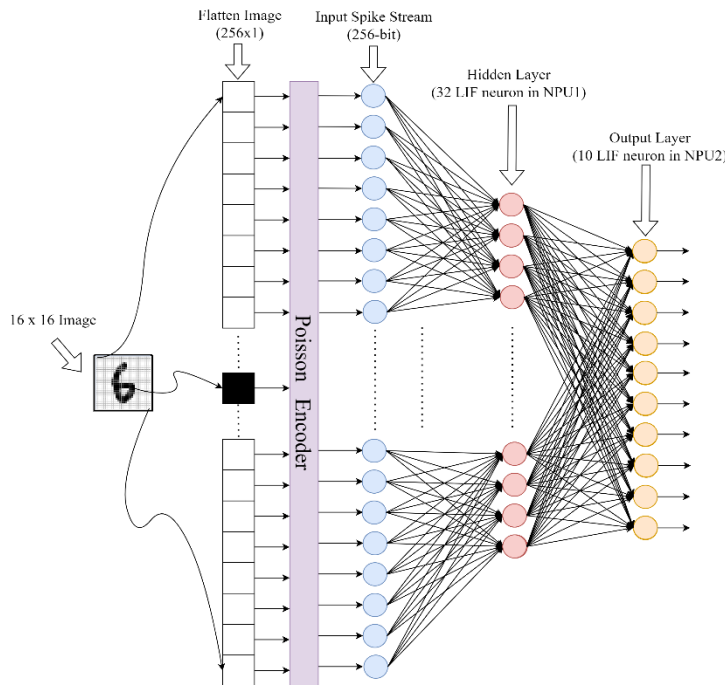


Fig. 5 神經網路架構之示意圖

我們使用 Pytorch 套件，進行神經網路的訓練；在 50 個時間步的運算下，結果可達到 95%的準確度，顯示本處理器在圖形辨識之典型應用中，亦具備可行性。

Conclusion

專題以脈衝神經網路處理器為主軸，著重於模仿生物體的神經網路行為，並藉此獲得生物體能源效益上之優勢。

首先，以可調控之神經群網路架構，透過自定義參數，控制神經網路之連接方式；本實作專題在論文[1]提及的架構外，新增具有連接順序優先性之前饋式神經網路，進一步增加處理器於應用端之靈活性。並且，實作稀疏分組編碼架構，在進行突觸權重累積時，可跳過大量運算，達到減少運算時間之目的。此外，實作突觸記憶與衰減邏輯，可模仿生物體神經之突觸疲勞等生理現象。最後，實作論文[1]中的 I-QIF 神經元電路，透過線性化與整數化，可實現多種突波行為，亦兼顧硬體友善；我們在本實作專題中，新增 LIF 神經元電路，並與 I-QIF 神經元共用部分電路，增加處理器之靈活性。

成功完成此 SNN 處理器的硬體架構設計，並完成硬體的驗證，確認其功能及行為模式皆符合預期。設計完成後，完成硬體合成，合成結果顯示我們的設計能夠讓此處理器操作在 125MHz 的頻率。

我們已 MNIST 手寫數字資料集之辨識，展示本處理器於應用端之能力；在 50 個時間步的運算下，可達到 95% 之辨識準確度，顯示本處理器在典型的神經網路應用中，亦有相當之可行性。

Reference

[1] Z. -W. Yeh et al., "POPPINS: A Population-Based Digital Spiking Neuromorphic Processor with Integer Quadratic Integrate-and-Fire Neurons", 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401426.

心得感想

這次研究帶給我們豐富的學習經驗。首先，我們培養了可於理解文獻內容後歸納出重點的能力，並能實際建構相對應的電路。其次，在大量的實作中，我們逐漸熟習於 Verilog 語言的硬體建構，從一開始甚至寫出無法合成的電路，到最後，已能夠精確地描述電路邏輯，並能提前考慮到面積與時序，選擇適當的架構或加入 pipeline 設計，使電路的表現最佳化。同時，在驗證方面，我們學到如何以其他程式語言輔助，產生大量的測試資料與 Golden Data，確保硬體的行為與演算法一致。

在研究過程中也遭遇了許多挑戰，如：模擬結果與演算法有所差異，或是不熟悉 EDA Tool 的相關設定。我們因而提升了獨立思考、創新和解決未知問題的能力，能夠善用各項資源，找出根本原因，並提出有效的解決方案。最後，也感謝助教提供的相關支援，耐心地指導我們問題解決的方向。