Q-learning演算法硬體加速器設計與實現

“Design and Implementation of Hardware Accelerator for Q-learning Algorithm“

指導教授:朱元三 教授　　專題生:王得安、羅介佑

摘要

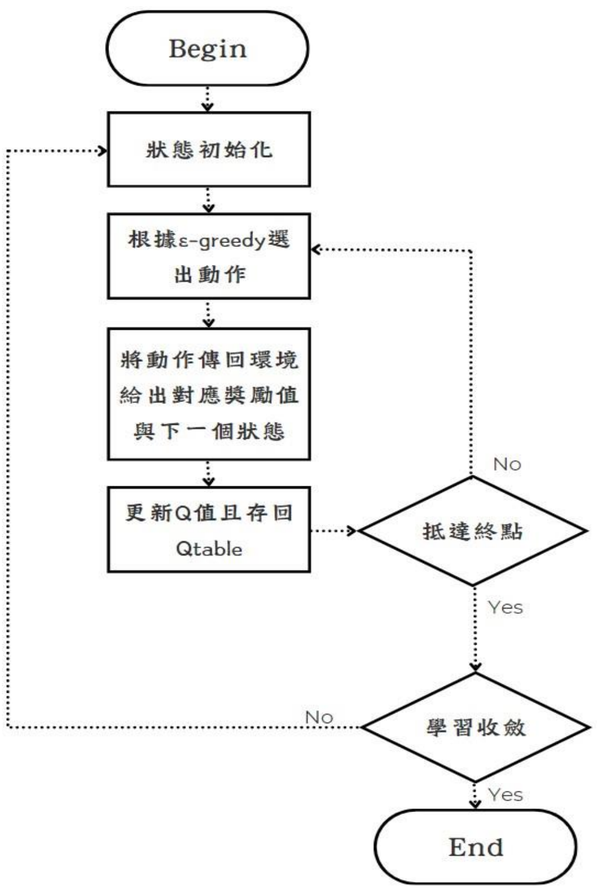
隨著無人機導航、自動駕駛及自動倉儲管理系統的快速發展，這些應用場景中無人系統需要解決不同環境下的複雜決策問題。強化學習（尤其是Q-learning）因其自我學習決策的特性而廣泛應用於此類無人系統中，能夠通過試誤學習獲得最優策略。然而，隨著應用場景的多樣化及計算需求的增加，如何提升Q-learning的運算速度與降低功耗成為一大挑戰。

本研究基於Q-learning設計了一款2D路徑搜尋加速器，透過硬體架構的設計來優化Q-learning算法，以加速其在複雜環境中的計算效率。透過模擬測試，我們驗證了該硬體加速器在不同障礙物配置的環境中，相較於傳統軟體實現達到了顯著的速度提升及功耗降低。此成果證明了我們的加速器在無人機導航及自動化管理系統等領域的應用潛力。

演算法介紹

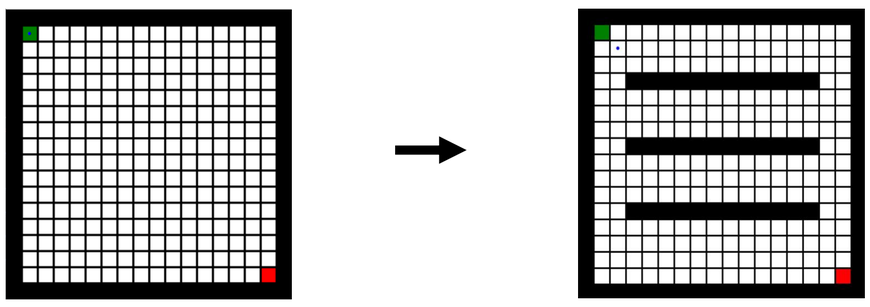
Q-learning 運算流程：

Q-learning 演算法的核心在於透過反覆的學習過程，不斷更新狀態-動作對應的 Q 值，最終建立最佳的行動策略。在演算法的初始階段，所有可能的狀態與動作對應的 Q 值被設為零。隨著代理（agent）在環境中移動並選擇動作，它會根據 ε-greedy 策略，以一定機率選擇當前 Q 值最高的動作（利用）或隨機探索（探索）。代理在執行動作後獲得即時回饋(reward)，同時根據 Q-learning 的更新公式調整 Q table的數值。這一公式將新的回報信息融入原 Q 值中，進而提升策略的精確度。此過程重複進行，每次更新使 Q table的數值逐步收斂。隨著時間推移，代理不再需要探索而是基於學到的 Q 值選擇最佳路徑，從而實現最優化的路徑搜尋。



模擬環境

16*16的二維空間來作為學習環境，其中一格為一次行動的距離，一格為一種狀態。代理人的任務是在此環境中探索出一條最短的路徑來行動，起點與終點為空間中最長距離的兩個點，左上角及右下角，並且加入障礙物測試其對環境複雜度的影響。



硬體設計

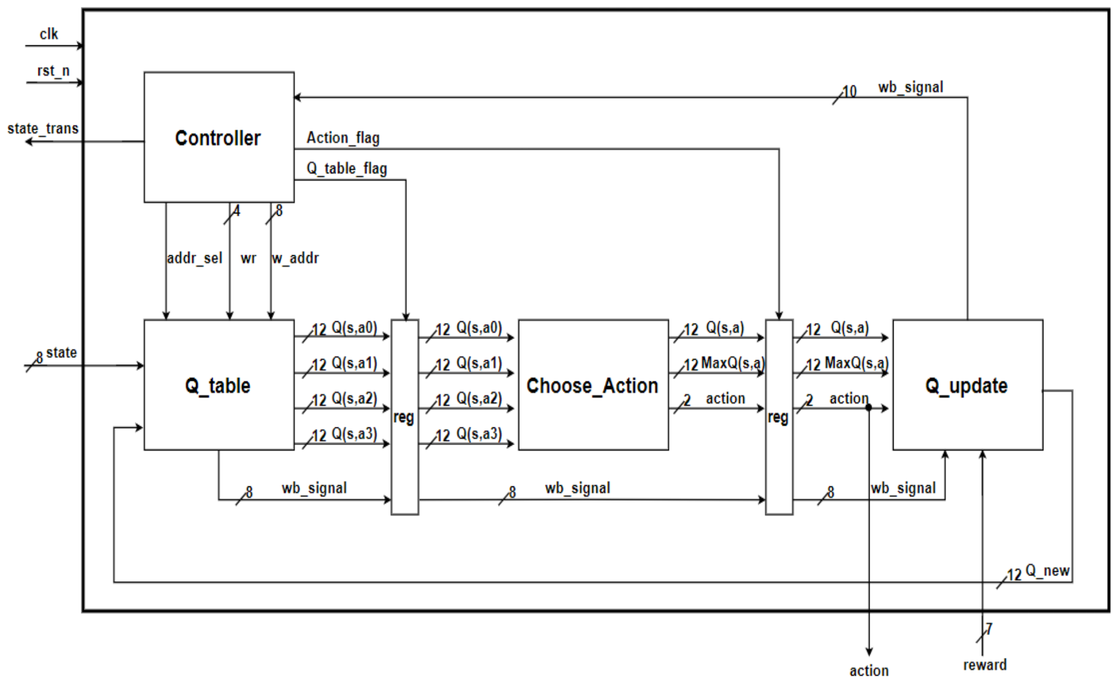
整個系統包含數個關鍵模組，以下為系統架構與流程概述:

Q-table：主要用於存儲每個狀態-動作對應的 Q 值，這是 Q-learning 算法中的核心數據結構。

Choose Action ：根據 ε-greedy 策略，選擇當前狀態的最佳行動。該模組通過比對 Q 值來選擇最大 Q 值的動作，並在隨機情況下進行探索。

Q-update ：實現 Q-learning 更新公式的核心計算單元，負責根據回饋值與學習參數更新 Q 值。

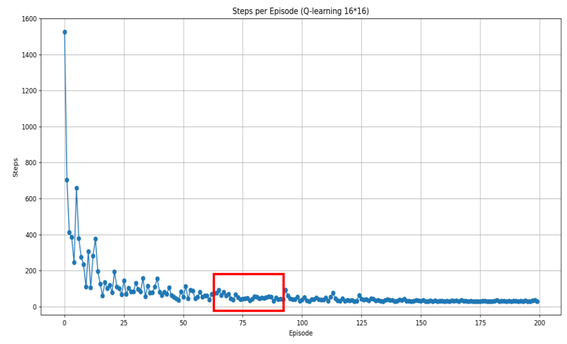
Controller：控制 Q 值的存取及各模組間的數據傳遞，協調硬體整體運算流程。

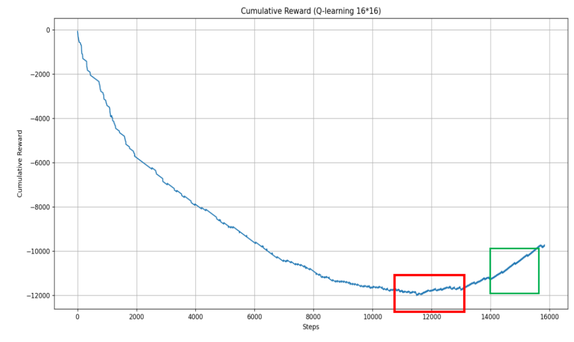


實驗結果

左圖為每回合行走步數圖，橫軸為回合，縱軸為每回合到達終點步數，圖中紅框處為收斂位置，約在60至70回合左右。

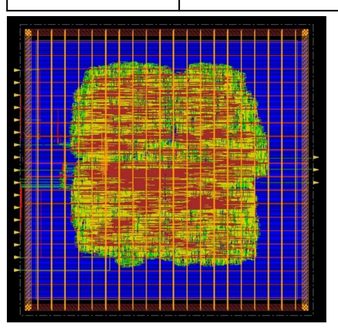
右圖為當前步數之獎勵累加圖，橫軸為步數，縱軸為累積獎勵值，圖中紅框處為收斂開始位置，綠框便是已經收斂。





下表以每回合收斂步數觀察對於不同模擬環境，比較軟體和硬體分別執行最佳路徑的結果。

實驗結果比較				
	軟體		硬體	
	收斂回合數	執行時間	收斂回合數	執行時間
無障礙物迷宮	100	0.38s	95	0.933ms
障礙物一條(橫)	110	0.48s	85	0.920ms
障礙物二條(橫)	105	0.42s	102	0.997ms
障礙物三條(橫)	85	0.46s	75	0.990ms



Q-learning 加速器晶片規格	
Technology	TSMC U18
Period	25(ns)
Frequency	40(Mhz)
Power	46.614(mW)
Area	1124381.2 (um^2)
Core VDD	1.62V

結論

從實驗結果來觀察，雖然比較結果顯示，在收斂的回合數上，硬體與軟體沒有絕對相關性，因為資料以定點小數的表示方式來進行運算，在每次 barrel shifter 進行位移運算時都會流失精度，且以 LFSR 設計的隨機性不如軟體，但以實驗到達目標點 200 回合的運算時間來比較，硬體的學習效率還是具有顯著的優勢。