

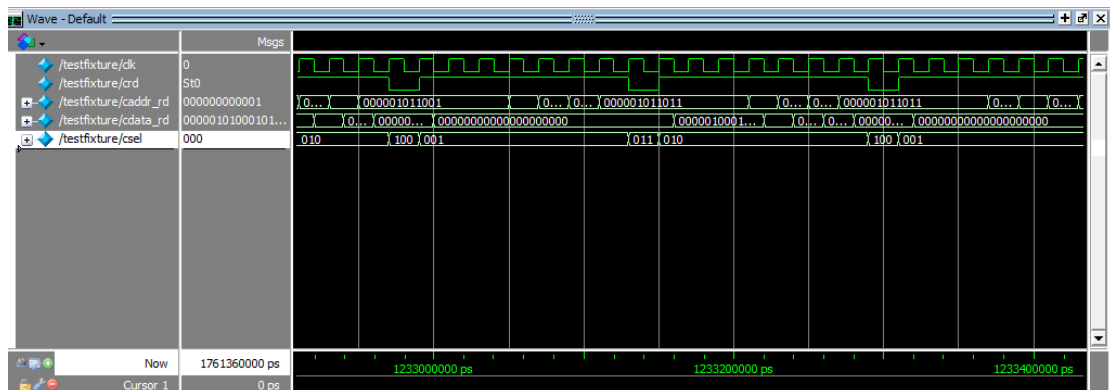
Homework 9

資訊所 P76134082 陳冠言

1. RTL 的模擬結果截圖

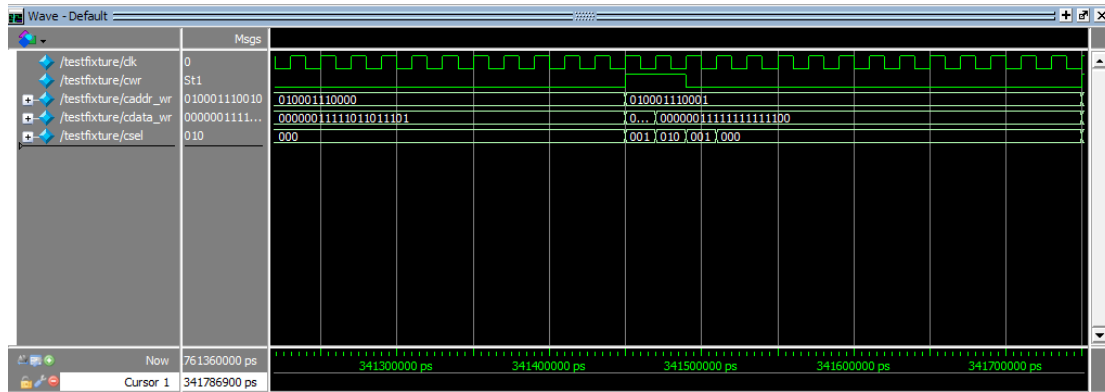
```
VSIM 24> run -all
#-----
#
# START!!! Simulation Start .....
#
#-----
#
# Layer 0 (Convolutional Output) with Kernel 0 is correct !
# Layer 0 (Convolutional Output) with Kernel 1 is correct!
# Layer 1 (Max-pooling Output) with Kernel 0 is correct!
# Layer 1 (Max-pooling Output) with Kernel 1 is correct!
# Layer 2 (Flatten Output) is correct!
#
#-----
#
#----- S U M M A R Y -----
#
# Congratulations! Layer 0 data have been generated successfully! The result is PASS!!
# Congratulations! Layer 1 data have been generated successfully! The result is PASS!!
# Congratulations! Layer 2 data have been generated successfully! The result is PASS!!
#
#-----
#
# ** Note: $finish      : E:/intelFPGA/HW9/testfixture.v(271)
#         Time: 1761360 ns Iteration: 0 Instance: /testfixture
# l
# Break in Module testfixture at E:/intelFPGA/HW9/testfixture.v line 271
```

2. 解釋波型



上圖為輸出資料記憶體之讀取動作時序圖，主要表達從記憶體中讀取卷積運算所需的資料，例如輸入圖像資料、權重和偏置等。

圖中 crd (Convolution Read) 訊號大部分時間為高電位，表示系統正在從記憶體中密集地讀取資料。caddr_rd (Convolution Read Address) 訊號顯示了讀取位址的連續變化，這意味著系統正在循序掃描記憶體以獲取運算所需的數據，例如輸入特徵圖的像素值或卷積核的權重。cdata_rd (Convolution Read Data) 則顯示了從這些位址實際讀取到的數據內容。此波形揭示了卷積運算前期的數據準備工作，系統正按照預設的順序和選擇邏輯，從記憶體中提取進行卷積計算所必需的輸入數據和權重值。

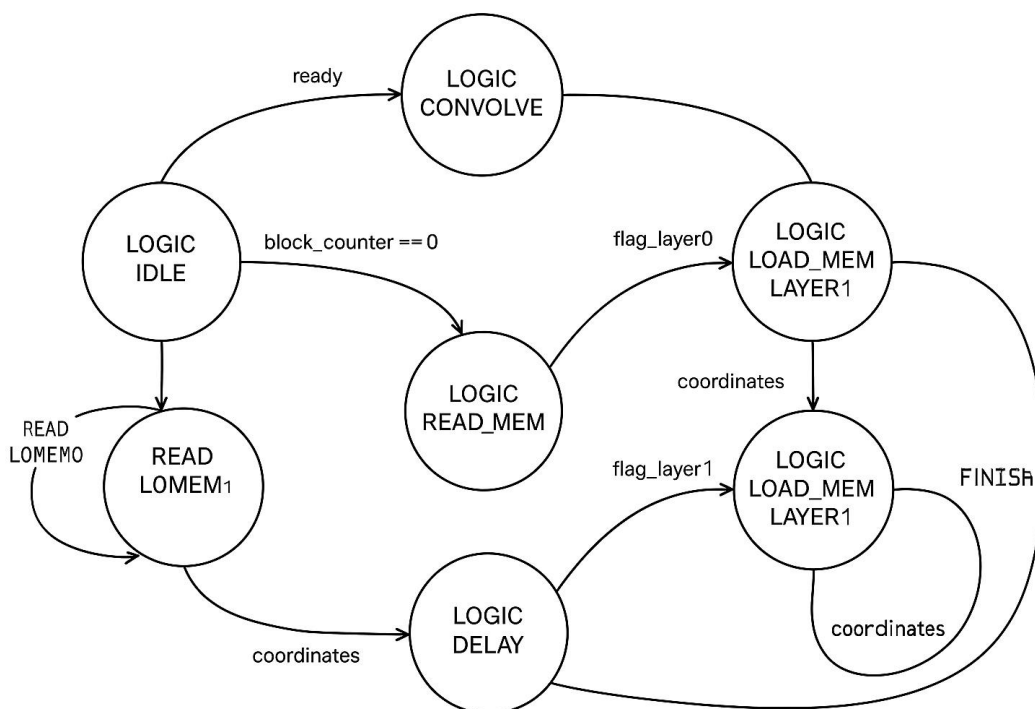


上圖為輸出資料記憶體之寫入動作時序圖，主要表達將運算結果寫入記憶體中，而 `csel` 信號用於控制讀取和寫入操作的具體執行方式。

圖中 `cwr` (Convolution Write) 訊號大部分時間為高電位，表示系統正在將運算結果密集地寫回記憶體。`caddr_wr` (Convolution Write Address) 訊號指示了結果數據被寫入的目標記憶體位址，這些位址的變化反映了輸出特徵圖的數據組織方式。`cdata_wr` (Convolution Write Data) 則顯示了實際寫入的卷積運算結果。

此波形描繪了卷積運算後期的結果儲存過程。系統將計算完成的輸出特徵圖數據，按照 `csel` 的指示（可能關聯到不同的卷積核或層）以及 `caddr_wr` 指定的位址，存入記憶體中。

3. 畫出狀態機



總體運作流程主要是 FSM 從 LOGIC_IDLE 開始，等待啟動信號。

1. **卷積 (Convolution):** 在 LOGIC_CONVOLVE 狀態處理輸入數據。
2. **寫入 Layer 0 結果:** 在 LOGIC_LOAD_MEM_LAYER0 狀態將卷積結果寫入記憶體。
3. **座標更新與流程控制:** LOGIC_DELAY 狀態負責更新當前處理的座標，並根據前一個狀態和座標決定下一步是繼續當前層的處理、進入下一層，還是結束。
4. **讀取記憶體:** LOGIC_READ_MEM 狀態根據 current_read_state 從相應記憶體區塊讀取數據，用於最大池化或扁平化。
5. **寫入 Layer 1 結果:** 在 LOGIC_LOAD_MEM_LAYER1 狀態將最大池化結果寫入記憶體。
6. **寫入 Layer 2 結果:** 在 LOGIC_LOAD_MEM_LAYER2 狀態將扁平化結果寫入記憶體。
7. **完成:** LOGIC_FINISH 狀態表示所有運算已結束。

4. 學到什麼？

透過本次作業，我學到了：

1. **卷積運算的原理和實現：** 深入理解了卷積運算的原理，包括如何使用卷積核對輸入圖像進行處理，以及如何計算卷積運算的結果。
2. **RTL 設計和驗證：** 學習了如何使用 RTL (Verilog) 語言設計一個卷積運算系統，包括狀態機設計、資料流設計和記憶體管理。
3. **ModelSim 的使用：** 掌握了 ModelSim 模擬工具的使用方法，包括如何創建項目、添加文件、編譯程式碼、運行模擬和觀察波形。透過波形分析，可以驗證設計的正確性，並找出潛在的問題。
4. **定點數運算：** 學習了如何使用定點數進行運算，包括如何表示定點數，以及如何進行加法、減法、乘法等運算。
5. **ASIC Flow 的考量：** 瞭解了在 ASIC flow 中，需要考慮硬體資源的限制，以及如何優化設計以滿足時序要求。
6. **除錯技巧：** 學習了如何使用 ModelSim 的除錯工具，例如，斷點、單步執行、資料追蹤等，來定位和解決設計中的問題。