


2025 Digital IC Design Homework 4

NAME	陳冠言		
Student ID	P76134082		
ATCONV Simulation Result			
Functional simulation	Pass	Pre-Layout simulation	Pass
<pre># ----- SUMMARY ----- # # Congratulations! Layer 0 data have been generated successfully! The result is PASS!! # # Congratulations! Layer 1 data have been generated successfully! The result is PASS!! # # terminate at 51206 cycle # ----- # # ** Note: \$finish : D:/intelFPGA/DIC/HW4/testfixture.sv(224) # Time: 2560300 ns Iteration: 0 Instance: /testfixture</pre>		<pre># ----- SUMMARY ----- # # Congratulations! Layer 0 data have been generated successfully! The result is PASS!! # # Congratulations! Layer 1 data have been generated successfully! The result is PASS!! # # terminate at 51206 cycle # ----- # # ** Note: \$finish : D:/intelFPGA/DIC/HW4_1/testfixture.sv(224) # Time: 2560307525 ps Iteration: 0 Instance: /testfixture</pre>	
System Simulation Result			
Functional simulation	Pass	Pre-Layout simulation	Pass
<pre># ----- SUMMARY ----- # # Congratulations! Layer 0 data have been generated successfully! The result is PASS!! # # Congratulations! Layer 1 data have been generated successfully! The result is PASS!! # # terminate at 143365 cycle # ----- # # ** Note: \$finish : D:/intelFPGA/DIC/HW4_2/testfixture.sv(228) # Time: 7168250 ns Iteration: 0 Instance: /testfixture # 1 # Break in Module testfixture at D:/intelFPGA/DIC/HW4_2/testfixture.sv line 228</pre>		<pre># ----- SUMMARY ----- # # Congratulations! Layer 0 data have been generated successfully! The result is PASS!! # # Congratulations! Layer 1 data have been generated successfully! The result is PASS!! # # terminate at 143365 cycle # ----- # # ** Note: \$finish : D:/intelFPGA/DIC/HW4_3/testfixture.sv(228) # Time: 7168258217 ps Iteration: 0 Instance: /testfixture # 1 # Break in Module testfixture at D:/intelFPGA/DIC/HW4_3/testfixture.sv line 228</pre>	
ATCONV Synthesis Result			
Total logic elements	461 / 55,856 (< 1 %)		
Total memory bits	0 / 2,396,160 (0 %)		
Total registers	138		
Embedded multiplier 9-bit elements	0 / 308 (0 %)		
Total Cycle used	51206 cycle		

Flow Summary	
 <<Filter>>	
Flow Status	Successful - Thu May 08 23:03:23 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	ATCONV
Top-level Entity Name	ATCONV
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	461 / 55,856 (< 1 %)
Total registers	138
Total pins	124 / 325 (38 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	0 / 308 (0 %)
Total PLLs	0 / 4 (0 %)

System Synthesis Result	
Total logic elements	813 / 55,856 (1 %)
Total memory bits	0 / 2,396,160 (0 %)
Total registers	375
Embedded multiplier 9-bit elements	0 / 308 (0 %)
Total Cycle used	143365 cycle
Flow Summary	
<<Filter>>	
Flow Status	Successful - Wed May 21 22:58:44 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	top
Top-level Entity Name	top
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	813 / 55,856 (1 %)
Total registers	375
Total pins	124 / 325 (38 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	0 / 308 (0 %)
Total PLLs	0 / 4 (0 %)
Description of your design	

1. ATCONV Design Specifications:

本設計包含一個主要的狀態機控制系統，能夠處理 64x64 輸入圖像，並通過稀疏卷積和最大池化操作產生 32x32 的輸出。設計採用了模組化架構，由幾個關鍵部分組成：狀態控制器、卷積計算單元、地址生成器及記憶體介面管理器。

主要狀態機

- INIT：初始化狀態
- ATCONV_9PIXELS：執行 9 像素的稀疏卷積
- LAYER0_WRITERELU：將結果寫入 Layer 0 並應用 ReLU 激活
- MAXPOOL_4PIXELS：對 Layer 0 輸出執行 2x2 最大池化
- LAYER1_WRITECEILING：將向上取整後的池化結果寫入 Layer 1
- FINISH：完成處理並發出 done 信號 1

處理流程

Layer 0 處理：

1. 從 ROM 讀取當前中心點及其擴張卷積核覆蓋的 8 個鄰域像素
2. 計算卷積加權和並加上偏置
3. 應用 ReLU 激活函數（負值變為零）
4. 將結果寫入 Layer 0 記憶體 1

Layer 1 處理：

1. 從 Layer 0 記憶體讀取 2x2 窗口的 4 個像素
2. 找出最大值
3. 對最大值進行向上取整（小數部分有任何位為 1 時，整數部分加 1）
4. 將結果寫入 Layer 1 記憶體

記憶體控制

ROM 控制：

- 使用 ROM_rd 信號控制 ROM 讀取操作
- 通過 iaddr 生成適當的地址以讀取所需像素 1

Layer 0/1 記憶體控制：

- 使用 layer0_ceb/layer1_ceb 控制記憶體啟用
- 使用 layer0_web/layer1_web 控制讀寫模式
- 通過 layer0_A/layer1_A 指定地址
- 將計算結果通過 layer0_D/layer1_D 寫入記憶體

2. System Design Specifications:

本設計是一個基於硬體加速的空洞卷積（Atrous Convolution）神經網路處理器，主要用於影像處理。系統由四個主要模塊組成：ATCONV_Wrapper、BUS、ROM_Wrapper 以及 SRAM_Wrapper。

ATCONV_Wrapper

這是系統的核心處理，實現了空洞卷積、ReLU 激活以及最大池化操作：

1. 擴張卷積實現：

- 使用 3x3 卷積核與擴張率(atrous rate)為 2
- 採用 9 個固定權重（KERNEL_0 到 KERNEL_8）和一個偏置值（KERNEL_BIAS）
- 輸入影像大小為 64x64 像素
- 設計包含邊界處理機制，超出範圍時會自動進行鉗位操作

2. 量化與計算：

- 使用 16 位元定點數學運算
- 卷積結果經過 ReLU 非線性激活函數處理
- 池化操作採用 2x2 窗口，步長為 2
- 池化結果進行適當的量化與四捨五入處理

3. 狀態機控制：

- 設計包含 11 個狀態，從初始化到完成
- 包括 ROM 讀取、卷積計算、記憶體寫入/讀取、池化操作等狀態
- 當處理完所有數據後，會觸發 done 信號表示完成

BUS

匯流排負責連接 ATCONV_Wrapper 與 ROM 和 SRAM：

1. 主從架構：

- 一個主設備（Master）連接三個從設備（Slave）
- 使用 ID 識別不同的從設備：ID_0 對應 ROM，ID_1 對應 SRAM0，ID_2 對應 SRAM1

2. 數據路由：

- 根據 ID 標識，自動將地址、數據和控制信號路由到正確的從設備
- 處理讀寫操作的握手機制
- 支持突發傳輸操作

ROM_Wrapper

這提供對唯讀記憶體訪問 port，用於儲存輸入影像數據：

1. 狀態控制：
 - 實現簡單的狀態機，處理等待命令和讀取數據狀態
 - 管理突發讀取的計數和控制
2. 讀取操作：
 - 接收來自匯流排的讀取請求
 - 按照指定地址提取數據

SRAM_Wrapper

為靜態隨機存取記憶體提供接口，用於存儲中間結果和最終輸出：

1. 多功能狀態機：
 - 包含待命、讀取操作和寫入操作三個狀態
 - 根據匯流排請求切換適當的狀態
2. 讀寫控制：
 - 處理數據讀取和寫入操作
 - 管理記憶體地址和數據流
 - 支持突發傳輸

數據流程

1. 輸入階段：
 - 從 ROM 讀取輸入影像數據
 - 根據擴張卷積的需求，按特定模式訪問數據點
2. 卷積階段：
 - 將 9 個輸入數據與卷積核權重相乘並累加
 - 添加偏置值並進行移位操作（相當於除以 16）
 - 應用 ReLU 激活函數（負值置零）
3. 中間存儲：
 - 將卷積結果寫入 SRAM0
 - 完成整個影像的卷積操作
4. 池化階段：
 - 從 SRAM0 讀取卷積結果
 - 執行 2x2 最大池化操作
 - 進行適當的四捨五入處理
5. 輸出階段：
 - 將池化結果寫入 SRAM
 - 當所有數據處理完成時，觸發 done 信號