2025 Digital IC Design Homework 4

| | | _ 0 _ 0 _ 1 _ 1 _ 1 | | | • | |
|---|-----------|---------------------|--|---|---|--|
| NAME | 王華昀 | | | | | |
| Student ID | N26134308 | | | | | |
| ATCONV Simulation Result | | | | | | |
| Functional Pass simulation | | Pre-Layout | | Dana | | |
| | | Pass | | simulation | Pass | |
| (your functional sim result) | | | | (your pre-sim result) | | |
| S U M M A R Y Congratulations! Layer 0 data have been generated successfully! The result is FAS: Congratulations! Layer 1 data have been generated successfully! The result is FAS: terminate at 46084 cycle *** Mote: Sfinish : Ct/DIC_2025/BM4/StudentID/file/ATCONV/testfixture.sv(224) Time: 2304200 ns Iteration: 0 Instance: /testfixture | | | | S U M M A R Y Congratulations! Layer 0 data have been general congratulations! Layer 1 data have been general congratulations! Layer 1 data have been general congratulations at 46084 cycle "" Note: Sfinish : C:/DIC_2025/HN4/Stude Time: 2304211169 ps Iteration: 0 Instal | erated successfully! The result is PASS!! | |
| System Simulation Result | | | | | | |
| Functional simulation Pass | | | Pre-Layout Pass | | | |
| | | Pass | | simulation | rass | |
| (your functional sim result) | | | | (your pre- | sim result) | |
| SUMMARY Congratulations! Layer 0 data have been generated successfully! The result is is Congratulations! Layer 1 data have been generated successfully! The result is is terminate at 46084 cycle *** Note: offinish : C:/DIC_2025/BM4/StudentID/file/System/testfixture.sv(228) Time: 2304200 ns Iteration: 0 Instance: /testfixture | | | FASS!! Congratulations! Layer 1 data have been generated successfully! The result is FASS!! terminate at 46084 cycle | | | |
| ATCONV Synthesis Result | | | | | | |
| Total logic elements | | | 388 | 388 | | |
| Total memory bits | | | 63 | | | |
| Total registers | | | 124 | | | |
| Embedded multiplier 9-bit | | | 0 | | | |
| elements | | | | | | |
| | | | | | | |

(your flow summary of ATCONV)

Flow Summary

<<Filter>>

Flow Status Successful - Sat May 31 02:18:06 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 388 / 55,856 (< 1 %)

Total registers 63

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 | 483 |
| Total memory bits | 63 |
| Total registers | 124 |
| Embedded multiplier 9-bit | 0 |
| elements | |
| Total Cycle used | 46084 |

(your flow summary of System)

Flow Summary

<<Filter>>

Flow Status Successful - Sat May 31 01:59:31 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 EP4CE55F23C6

Timing Models Final

Total logic elements 483 / 55,856 (< 1 %)

Total registers 63

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

這次的設計在 ATCONV 我每個 cycle 只會進行一次乘加,因此一次 conv 需要 9 個 cycle 才能做完,而這次的作業有分為 layer0 及 layer1,我是使用 FSM 去控現在計算的 layer,會先將 layer0 全部計算完後才會再從 layer0 將數字讀出計算 layer1,主要分為三個狀態 IDLE、LAYER0、LAYER1。當 rst 結束下一個 cycle 狀態會從 IDLE 跳至 LAYER0,但是在 IDLE 時就會將 ROM_rd 及 addr 送出,這樣的用意是來處理 sram 一個 cycle delay 的問題,接著會進入到 LAYER0 開始進行 conv 的計算,主要就是用 cnt 去控制 rom 地址的讀取,一次完整的 conv 要做 9 個 cycle,然後在輸出時會做 bias 的計算以及 relu 的判斷,主要就是把 psum_out[15]拿來判斷是否為負值。直到 conv 中心的 x 及 y 地址會到(0,0)就要進入到 LAYER1 狀態,開始計算 maxpool,同樣是使用 cnt 去進行控制,每一次計算要 5 個 cycle,前面四個 cycle 取值,第五個 cycle 將結果存回 sram2。計算部分全部都用移位取代,所以沒有乘法器。整個 設計不會有兩個同時提出讀取或寫入需求,這樣是因為在加入 system 後就不用處理兩個 master 的問題,我在 ATCONV 的時候就將時序處理好。

加入 system 後我原先以為要強制遵守說明文件的 protocol,也就是模擬 AXI 每次握手到傳輸完成最少要 3 個 cycle 這件事情,並且也要遵守 burst len 的限制,但是看到 moodle 上助教的回答好像不用遵守,再加上這次 master、bus、slave 都是我們自己寫,就會變成其實只要內部控好,把線接一接就好了,這也是為什麼我加上 system 後 cycle 數完全一樣,因為我在 ATCONV 裡有避免掉同時有兩個發出讀取的需求。並且遵守 protocol 會導致在比 PA 時比別人多好幾倍的 cycle,可以理解作業想讓我們練習握手機制,但是如果 mater、bus、slave 都是自己寫並且又沒有去驗 vip 的話就有點沒有意義,或許以後可以改成統一用一樣的 slave,用 tb 去給 ready 訊號之類的方式,並且要握手完成才會把資料送出,才比較好讓大家練習並且比 PA 時也比較公平。