2023 Digital IC Design Homework 4

NAME	陸啟倫
Student ID	N26114950

Simulation Result

Functional simulation	100	Gate-level simulation	100

```
---------- 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!!

terminate at      46085 cycle
-----------------------------------------------------

** Note: $finish    : C:/2023DIC/HW4/testfixture.v(178)
   Time: 2304250 ns  Iteration: 0  Instance: /testfixture
```

```
---------- 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!!

terminate at      46085 cycle
-----------------------------------------------------

** Note: $finish    : C:/2023DIC/HW4/testfixture.v(178)
   Time: 2304259339 ps  Iteration: 0  Instance: /testfixture
```

Synthesis Result

Total logic elements	564
Total memory bits	0
Embedded multiplier 9-bit elements	0
Total cycle used	46085

Flow Status	Successful - Tue May 16 15:25:40 2023
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	564 / 55,856 (1 %)
Total registers	151
Total pins	82 / 325 (25 %)
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

這次實作的是 Convolution 電路，我透過一個 5-state 的狀態機來控制此電路。當電路被初始化時，則會先進入到 Initial 狀態，接著當接收到 ready 訊號後，則會在下個 clk 進入到 Get_kernel 狀態，在此狀態會和 IMAGE MEM 拿取圖片的 pixel 數值，並存到 kernel 暫存器中。當存取 9 個 pixel value 後，則會進入到 Conv_Relu 狀態，進行 Convolution 和 Relu 運算並同時寫回 layer0 MEM 中。我在電路中沒使用到乘法器，考量到 Weight 都是 2 的倍數，可以使用 bit 位移解決運算。在下個 clk 時，則會回到 Get_kernel 狀態反覆一樣的動作，直到 caddr_wr=4095 時就會進入到 Get_pool 狀態，在此狀態下會和 layer0 MEM 索取資訊，並同時進行 Max pooling 運算。每完成一次運算，就會進入到 Pool_round 狀態進行 round 運算，並同時將結果寫回 layer1 MEM 中，並持續 Get_pool 和 Pool_round 的狀態循環，直到運算完所有數值再回到初始狀態。

*Scoring = (Total logic elements + Total memory bits + 9*Embedded multipliers 9-bit elements) X Total cycle used*

*** Total logic elements must not exceed 1000.**