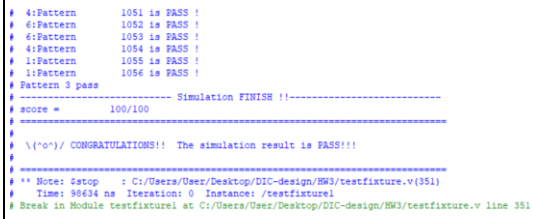
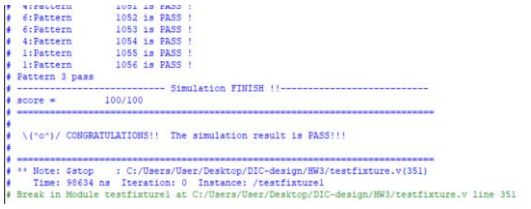
2024 Digital IC Design
Homework 3: matrix multiplier

NAME	陳育政
Student ID	E24094198

Simulation Result

Functional simulation	100	Gate-level simulation	100	Clock width	17 ns	Gate-level simulation time	98634 ns





Synthesis Result

Total logic elements	448
Total memory bit	0
Embedded multiplier 9-bit element	1

Flow Status	Successful - Tue May 07 18:28:05 2024
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	MM
Top-level Entity Name	MM
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	448 / 55,856 (< 1 %)
Total registers	321
Total pins	36 / 325 (11 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	1 / 308 (< 1 %)
Total PLLs	0 / 4 (0 %)

Description of your design

我的設計是用分成四個 states 的 FSM 完成，state 分別為 MAT1_READ、MAT2_READ、MULTIPLY 及 OUTPUT。並且針對 matrix1 和 matrix2，我是直接預設 2 個 8-bit，固定大小為 4*4 的 2D array(我認為如果用變動大小的 matrix size 可能會導致在 synthesis 時有問題，因為就合成角度而言，要去合成一個會變動大小的硬體感覺並不合理)，並用 row_1、row_2、col_1 及 col_2 控制 in_data 要存到 matrix 的哪個位置，因此 MAT1_READ 和 MAT2_READ

的 2 個 states 寫法其實很相似。

至於 invalid matrix 會發生的原因，我有想到兩個可能性。第一個是 matrix 不同 row 輸入的 column 數不一，例如: matrix1 的第一個 row 有 3 個 components，但是第二個 row 卻變成有 2 個或 4 個 components，則代表不是一個正常的 matrix，至於第二個原因則是 matrix1 的 column 數和 matrix2 的 row 數不一致，導致兩個 matrix 無法相乘。關於第一個 invalid matrix 的偵測，我是多用一個 MAT_COL_SIZE 紀錄 matrix 的第一個 row 的 column size，每當遇到 col_end signal 被觸發時，就根據對應 matrix 檢查 col_1 或 col_2 和 MAT_COL_SIZE 的大小是否相等，若不相等就代表兩個 row 的 column size 並不一致，則把 VALID_MATRIX signal 設為 0。而第二個 invalid matrix 的偵測，則是直接在 MULTIPLY state 檢查 col_1 是否等於 row_2 即可。

再者，我發現助教提供的 Scoring formula 計算方式，multiplier 的加權比重有達到 9，因此為了降低這部分的消耗，我特別設計只用 1 個乘法器的 module。假設 matrix1*matrix2=RESULT，在 MULTIPLY state 時，我會每次計算 matrix1 的一個 cell 和 matrix2 的一個 cell 相乘的結果值，並加到 mat_result 這個 register 內，當我們乘完後，跳到 OUTPUT state，把 mat_result 的值輸出到 out_data(mat_result 就是對應 RESULT 的其中一個 cell 的值)。

最後，在最初的模擬時，我發現 in_data 這個訊號，會有延遲一個 clock 才有正確新的資料輸入，舉例來說: 假設上一輪最後 in_data 的值是 17，則當我要進到下一輪時，會把 busy 設回 0，但是接下來的第一個 in_data 仍會是 17。 因此為了避免這樣的誤讀錯誤資料，我多設一個 wait_one_cycle 訊號，讓 module 在讀下一輪輸入前，多等一個 cycle。

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (Total cycle used*clock width)*