2024 Digital IC Design Homework II

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | E24094198 | | |
| Student ID | 陳育政 | | |
| **Functional Simulation Result** | | | |
| **FIFO Pass** | | **LIFO Pass** | **CIPU Pass** |
| **Stage 1** | | | |
|  | | | |
| **Stage 2** | | | |
|  | | | |
| **Stage 3** | | | |
|  | | | |
| **Description of your design** | | | |
| 原先我以為FIFO和LIFO的資料輸入是有先後順序的，因此最初計劃設計一個FSM控制狀態，但是在檢視助教提供的testbench後，我發現FIFO和LIFO的資料是同時餵入。考量到FIFO和LIFO的資料處理方式不太一樣，我最後把FIFO和LIFO的處理拆分成2個獨立的FSM，而LIFO的資料處理又會比FIFO更繁瑣一點(多了valid\_thing/done\_thing的訊號)，於是相較於FIFO的4個states，我在設計LIFO的FSM將其分成7個states，便於控制訊號。我的FIFO FSM根據教授上課的講義所說的，分成3個部分，分別為Sequential Circuit的currState register及Combinational Circuit的nextState logic/Output logic。至於LIFO FSM的設計，原先我也打算設計成3個部分，但是輸出結果一直會有問題(應該是訊號的控制有誤差)，所以最後我是設計成只有currState register和nextState logic共2個部分，這個部份是我日後可以再寫的更標準的。有關於第二部分和第三部分的luggage處理，我是分別用Luggage和Remain\_Luggage存行李資料，Luggage只有存當次LIFO的行李，每次LIFO輸出完畢，舊的行李資料就會被覆蓋掉。而Remain\_Luggage則是存所有行李，並用remain\_luggage\_count計算目前所有行李的總數，當有pop-out luggage行為時，就根據thing\_num的值把相對應數量的行李從Remain\_Luggage的尾巴pop-out，而remain\_luggage\_count也減去相對應數字，代表移除行李。 | | | |