**2020 Digital IC Design**

**Homework 4: LZ77 Encoder**

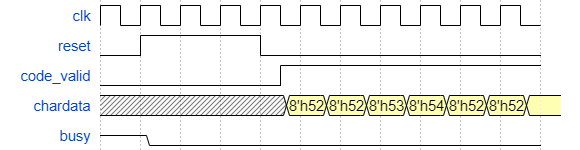
1. **Introduction**
2. **Design Specifications**
   1. **Block overview**
   2. **I/O Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | I/O | Width | Description |
| clk | I | 1 | 系統時脈訊號。本系統為同步於時脈正緣之同步設計 |
| reset | I | 1 | **高**位準"非"同步(active high asynchronous)之系統重置信號 |
| code\_valid | I | 1 | 當code\_valid為high，表示tb輸出的code\_pos、code\_len、chardata為有效值。 |
| code\_pos | I | 4 | 進行解碼時，代表匹配字串起始位置的offset。 |
| code\_len | I | 4 | 進行解碼時，代表匹配字串的長度。 |
| chardata | I | 8 | 進行編碼時，代表輸入的待編碼字元；  進行解碼時則代表匹配字串後的下一個字元。 |
| valid | O | 1 | 輸出有效訊號，當valid為high時，tb會檢測系統輸出是否正確。 |
| encode | O | 1 | 當encode訊號為high，表示系統進行編碼；當encode訊號為low，表示系統進行解碼。 |
| busy | O | 1 | 當系統讀取tb輸入之待編碼字串完畢後，需將busy訊號拉為high，表示系統正進行編碼與解碼驗證。解碼結束後，系統需將busy設定為low，以通知tb準備下一筆待編碼字串。 |
| offset | O | 4 | 進行編碼時，代表匹配字串起始字元距離待編碼字元的距離。 |
| match\_len | O | 4 | 進行編碼時，代表匹配字串長度。 |
| char\_nxt | O | 8 | 進行編碼時，代表匹配字串後的下一個待編碼字元；進行解碼時，代表解碼出的字元。 |

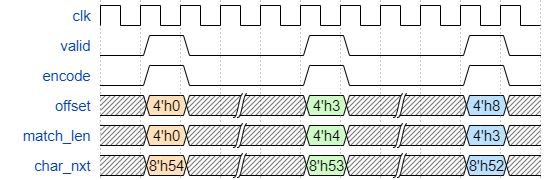
* 1. **Function Description**

本系統的搜尋緩衝區大小為9個字元，前看緩衝區大小為8個字元，且輸入之待編碼字串長度不超過30個字元。當tb端接收到系統的busy訊號為low時，便會將code\_valid拉為high，並且在每個cycle輸入一個待編碼字元，待編碼字串的結尾以字元’E’代表，當系統讀取字串結束後，需將busy訊號拉高為high，開始進行編碼動作。每完成一段字串的編碼，系統需將valid訊號拉高為high，此時tb會立即驗證系統輸出的offset、match\_len、char\_nxt是否正確。當編碼結束後，系統需進行解碼驗證，tb端會將先前解碼的結果依序由code\_pos、code\_len、chardata輸入，系統每解碼出一個字元，便將valid設為high，並將解碼出的字元由char\_nxt輸出，tb端會立即判斷解碼出的字元正確與否，待一段字串解碼完成後，tb便會再輸入下一組code，而如果所有字元皆已解碼完成，系統便可將busy訊號設為low，等待tb輸入下一組待編碼字串。

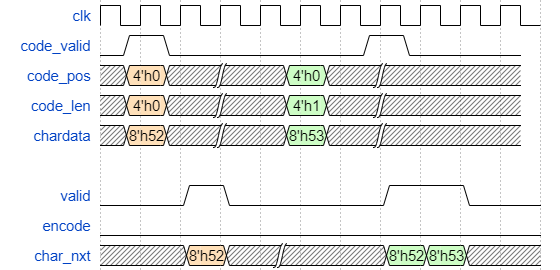
下圖為系統字串輸入時序規格圖，tb會在每個clk cycle負緣給予下一個待編碼字元。



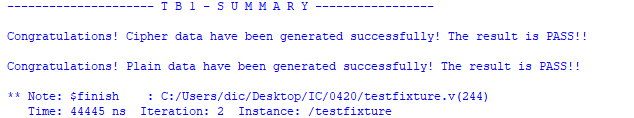
下圖為編碼時序規格圖，將valid拉高為high，同時將encode訊號也拉高為high以代表系統正進行編碼，tb端接收到valid訊號為high時便會驗證輸出的offset、match\_len與char\_nxt。

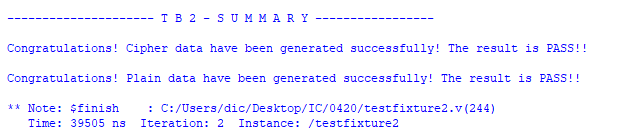


下圖為解碼時序規格圖，tb端輸入編碼時會將code\_valid設為high，系統讀取code\_pos、code\_len、chardata後便可進行解碼，每當解碼出一個字元，系統需將valid拉高為high，encode訊號設置為low代表進行解碼，並將解碼出的字元由char\_nxt輸出，tb會立即驗證解碼出的字元是否正確。



1. **Scoring**
   1. **Functional Part [60%, 兩份TB各30%]**





* 1. **Gate Level Part [20%, 兩份TB各10%]**

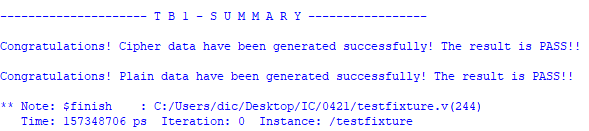
3.2.1 Synthesis

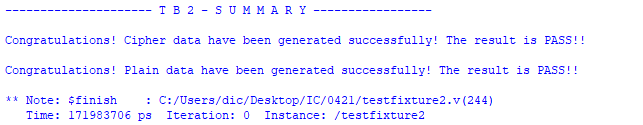
Your code should be synthesizable. After synthesizing in Quartus, the file named *CS.vo* and CS.sdo will be obtained.

**Device : Cyclone II EP2C70F896C8**

3.2.2 Simulation

All of the result should be generated correctly using **RC4.vo** and **RC4.sdo**, and you will get the following message in ModleSim simulation. (There should be no setup or hold time violations.)



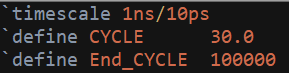


* 1. **Performance Part [20%]**

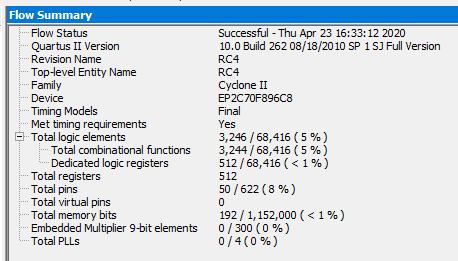
[20%] 完成 3.2 之要求並記錄合成後的 logic element 數量，以及模擬時能通過的最小 cycle 與模擬時間。評分標準為

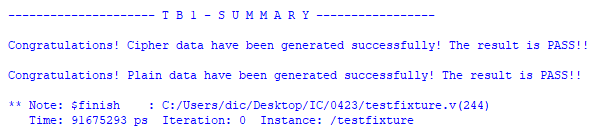
(Total logic elements + total memory bit + 9\*embedded multiplier 9-bit element)×(longest gate-level simulation time in ns)

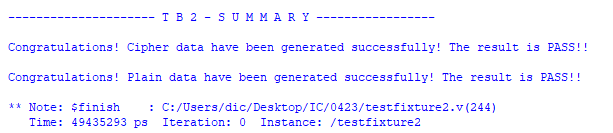
\*越低越好\*



可根據自己的設計調整CYCLE







1. **Submission**

You should classify your files into three directories and compressed to .zip format. The naming rule is HW4\_studentID\_name.zip.

|  |  |
| --- | --- |
|  | RTL category |
| \*.v | All of your Verilog RTL code |
|  | Gate-Level category |
| \*.vo | Gate-Level netlist generated by Quartus |
| \*.sdo | SDF timing information generated by Quartus |
|  | Documentary category |
| \*.pdf | The report file of your design (in pdf). |

4.1 Report file

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible, and the summary results and minimum CYCLE in post-sim are necessary.

4.2 Please submit your .zip file to folder HW4 in the moodle.

Deadline: 2020-05-19 23:55

1. If you have any problem, please contact by the TA by email

f74044088@gs.ncku.edu.tw 倪祺婷

weiting84610@gmail.com 陳威廷