

2022 Digital IC Design Homework 3

NAME		何立洋			
Student ID		Q36101066			
Simulation Result					
Functional simulation	Pass (encoder)	Pass (decoder)	Gate-level simulation	Pass (encoder)	Pass(decoder)
<pre>cycle 112e0, expect(7,7,8) , get(7,7,8) >> Pass cycle 112ee, expect(7,7,8) , get(7,7,8) >> Pass cycle 11305, expect(7,6,4) , get(7,6,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 2112360 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>			<pre>cycle 112ee, expect(7,7,8) , get(7,7,8) >> Pass cycle 11305, expect(7,6,4) , get(7,6,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 2112360 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>		
<pre>cycle 00801, expect 0, get 0 >> Pass cycle 00802, expect 8, get 8 >> Pass cycle 00803, expect 0, get 0 >> Pass cycle 00804, expect 8, get 8 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61620 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>			<pre>cycle 00803, expect 8, get 8 >> Pass cycle 00804, expect 0, get 0 >> Pass cycle 00805, expect 8, get 8 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61650 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>		
<pre>cycle 104b5, expect(5,1,0) , get(5,1,0) >> Pass cycle 104f1, expect(3,2,f) , get(3,2,f) >> Pass cycle 10538, expect(0,0,6) , get(0,0,6) >> Pass cycle 1057d, expect(0,0,4) , get(0,0,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 2008260 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>			<pre>cycle 104b5, expect(5,1,0) , get(5,1,0) >> Pass cycle 104f1, expect(3,2,f) , get(3,2,f) >> Pass cycle 10538, expect(0,0,6) , get(0,0,6) >> Pass cycle 1057d, expect(0,0,4) , get(0,0,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 2008260 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>		
<pre>cycle 00803, expect f, get f >> Pass == Decoding string "6" cycle 00804, expect 6, get 6 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61620 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>			<pre>cycle 00804, expect f, get f >> Pass == Decoding string "6" cycle 00805, expect 6, get 6 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61650 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>		
<pre>cycle 04aa8, expect(5,7,6) , get(5,7,6) >> Pass cycle 04ab6, expect(7,7,7) , get(7,7,7) >> Pass cycle 04acd, expect(7,6,4) , get(7,6,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 574680 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>			<pre>cycle 04aa8, expect(5,7,6) , get(5,7,6) >> Pass cycle 04ab6, expect(7,7,7) , get(7,7,7) >> Pass cycle 04acd, expect(7,6,4) , get(7,6,4) >> Pass ----- Encoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/tb_Encoder.sv(250) Time: 574680 ns Iteration: 1 Instance: /testfixture_encoder 1</pre>		
<pre>cycle 00802, expect 7, get 7 >> Pass cycle 00803, expect d, get d >> Pass cycle 00804, expect 7, get 7 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61620 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>			<pre>cycle 00801, expect 7, get 7 >> Pass cycle 00802, expect d, get d >> Pass cycle 00803, expect 7, get 7 >> Pass cycle 00804, expect d, get d >> Pass cycle 00805, expect 7, get 7 >> Pass ----- Decoding finished, ALL PASS ----- ** Note: \$finish : C:/Users/HLY/Desktop/Dic/hw3/hw3_decoder/tb_Decoder.sv(228) Time: 61650 ns Iteration: 1 Instance: /testfixture_decoder 1</pre>		
Synthesis Result			encoder		decoder
Total logic elements			21569/68416		96/68416
Total memory bit			0/1152000		0/1152000
Embedded multiplier 9-bit element			0/300		0/300

Simulation time img0	574710(ns)	61650(ns)																																		
Simulation time img1	574710(ns)	61650(ns)																																		
Simulation time img2	574710(ns)	61650(ns)																																		
<table><tr><td>Flow Status</td><td>Successful - Fri Apr 29 01:40:09 2022</td></tr><tr><td>Quartus II 64-Bit Version</td><td>13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition</td></tr><tr><td>Revision Name</td><td>LZ77_Encoder</td></tr><tr><td>Top-level Entity Name</td><td>LZ77_Encoder</td></tr><tr><td>Family</td><td>Cyclone II</td></tr><tr><td>Device</td><td>EP2C70F896C8</td></tr><tr><td>Timing Models</td><td>Final</td></tr><tr><td>Total logic elements</td><td>21,569 / 68,416 (32 %)</td></tr><tr><td> Total combinational functions</td><td>21,299 / 68,416 (31 %)</td></tr><tr><td> Dedicated logic registers</td><td>16,701 / 68,416 (24 %)</td></tr><tr><td>Total registers</td><td>16701</td></tr><tr><td>Total pins</td><td>28 / 622 (5 %)</td></tr><tr><td>Total virtual pins</td><td>0</td></tr><tr><td>Total memory bits</td><td>0 / 1,152,000 (0 %)</td></tr><tr><td>Embedded Multiplier 9-bit elements</td><td>0 / 300 (0 %)</td></tr><tr><td>Total PLLs</td><td>0 / 4 (0 %)</td></tr></table>			Flow Status	Successful - Fri Apr 29 01:40:09 2022	Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition	Revision Name	LZ77_Encoder	Top-level Entity Name	LZ77_Encoder	Family	Cyclone II	Device	EP2C70F896C8	Timing Models	Final	Total logic elements	21,569 / 68,416 (32 %)	Total combinational functions	21,299 / 68,416 (31 %)	Dedicated logic registers	16,701 / 68,416 (24 %)	Total registers	16701	Total pins	28 / 622 (5 %)	Total virtual pins	0	Total memory bits	0 / 1,152,000 (0 %)	Embedded Multiplier 9-bit elements	0 / 300 (0 %)	Total PLLs	0 / 4 (0 %)		
Flow Status	Successful - Fri Apr 29 01:40:09 2022																																			
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition																																			
Revision Name	LZ77_Encoder																																			
Top-level Entity Name	LZ77_Encoder																																			
Family	Cyclone II																																			
Device	EP2C70F896C8																																			
Timing Models	Final																																			
Total logic elements	21,569 / 68,416 (32 %)																																			
Total combinational functions	21,299 / 68,416 (31 %)																																			
Dedicated logic registers	16,701 / 68,416 (24 %)																																			
Total registers	16701																																			
Total pins	28 / 622 (5 %)																																			
Total virtual pins	0																																			
Total memory bits	0 / 1,152,000 (0 %)																																			
Embedded Multiplier 9-bit elements	0 / 300 (0 %)																																			
Total PLLs	0 / 4 (0 %)																																			
<table><tr><td colspan="2">Flow Summary</td></tr><tr><td>Flow Status</td><td>Successful - Thu Apr 28 22:53:35 2022</td></tr><tr><td>Quartus II 64-Bit Version</td><td>13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition</td></tr><tr><td>Revision Name</td><td>LZ77_Decoder</td></tr><tr><td>Top-level Entity Name</td><td>LZ77_Decoder</td></tr><tr><td>Family</td><td>Cyclone II</td></tr><tr><td>Device</td><td>EP2C70F896C8</td></tr><tr><td>Timing Models</td><td>Final</td></tr><tr><td>Total logic elements</td><td>96 / 68,416 (< 1 %)</td></tr><tr><td> Total combinational functions</td><td>66 / 68,416 (< 1 %)</td></tr><tr><td> Dedicated logic registers</td><td>81 / 68,416 (< 1 %)</td></tr><tr><td>Total registers</td><td>81</td></tr><tr><td>Total pins</td><td>27 / 622 (4 %)</td></tr><tr><td>Total virtual pins</td><td>0</td></tr><tr><td>Total memory bits</td><td>0 / 1,152,000 (0 %)</td></tr><tr><td>Embedded Multiplier 9-bit elements</td><td>0 / 300 (0 %)</td></tr><tr><td>Total PLLs</td><td>0 / 4 (0 %)</td></tr></table>			Flow Summary		Flow Status	Successful - Thu Apr 28 22:53:35 2022	Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition	Revision Name	LZ77_Decoder	Top-level Entity Name	LZ77_Decoder	Family	Cyclone II	Device	EP2C70F896C8	Timing Models	Final	Total logic elements	96 / 68,416 (< 1 %)	Total combinational functions	66 / 68,416 (< 1 %)	Dedicated logic registers	81 / 68,416 (< 1 %)	Total registers	81	Total pins	27 / 622 (4 %)	Total virtual pins	0	Total memory bits	0 / 1,152,000 (0 %)	Embedded Multiplier 9-bit elements	0 / 300 (0 %)	Total PLLs	0 / 4 (0 %)
Flow Summary																																				
Flow Status	Successful - Thu Apr 28 22:53:35 2022																																			
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition																																			
Revision Name	LZ77_Decoder																																			
Top-level Entity Name	LZ77_Decoder																																			
Family	Cyclone II																																			
Device	EP2C70F896C8																																			
Timing Models	Final																																			
Total logic elements	96 / 68,416 (< 1 %)																																			
Total combinational functions	66 / 68,416 (< 1 %)																																			
Dedicated logic registers	81 / 68,416 (< 1 %)																																			
Total registers	81																																			
Total pins	27 / 622 (4 %)																																			
Total virtual pins	0																																			
Total memory bits	0 / 1,152,000 (0 %)																																			
Embedded Multiplier 9-bit elements	0 / 300 (0 %)																																			
Total PLLs	0 / 4 (0 %)																																			

Description of your design

簡介:

Encoder:

我的 state 狀態有 4 個，s0 用在 reset 的狀態，s1 是存 chardata，s2 是將 datamem 存入”sh”、”l_a”、”s_l”，s3 則是”l_a”和”s_l”的匹配。

1.儲存

Encoder 方面我用了 datamem 存 chardata 的資料後，使用三個暫存器儲存資料，分別是”sh”、”l_a”、”s_l”，其中”s_l”代表 search_buffer+look-ahead_buffer。儲存都是用 for 迴圈複製電路。

2.匹配

我取”l_a”和”s_l”的對應位元做比較，並根據對應位元匹配的結果輸出。
”t[7]=s_l[16-k]===l_a[7]”、”if(t[7]&t[6]&t[5]&t[4]&t[3]&t[2]&t[1])”。

Decoder:

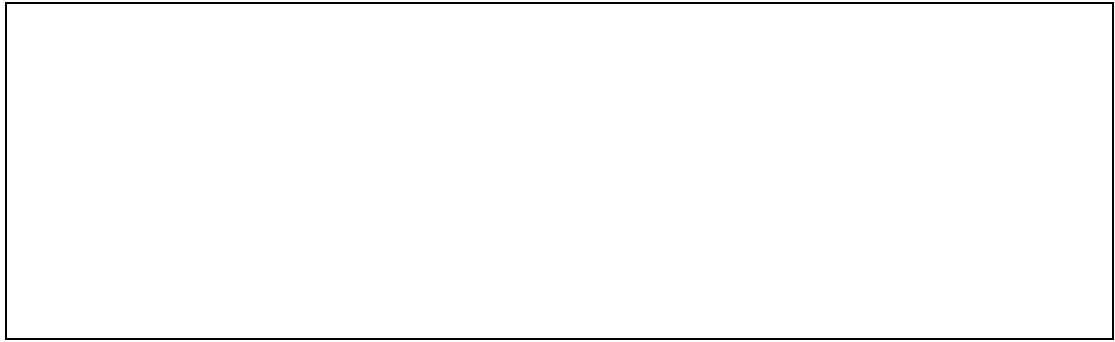
除了第一筆資料，我使用 code_len_count、code_pos_count 判斷，位元儲存的順序、次數、位置。

感想:

第三次作業難度比第二次高很多，雖然很有挑戰性，卻也很費力。如果時間充足，我覺得完成度應該會更高。但還是希望下次作業出簡單點，或給多點時間。尤其這次作業和期中考重疊，讓我這段時間每天都必須忙到晚上。

雖然我的說明很簡潔，但其實現在的做法是做過數次的改才成功的，我現在還記得我第一次把”l_a”和”s_l”的對應位元的所有結果寫出來，光這一步我就改了六次。而 Decoder 雖然和 Encoder 簡單不少，但兩者曾經因為我誤解說明，而讓我繞了不少路。

對這次作業的看法。我從這次作業學到很多東西，但也覺得不善，我目前會使用的語法不多，經驗也不足，面對複雜的作業很容易出錯。



*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element)*