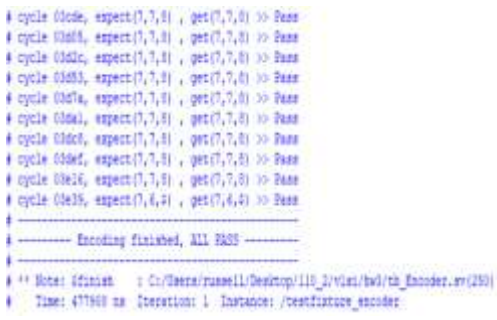
2022 Digital IC Design Homework 3

NAME	陳柏均
Student ID	P76101584

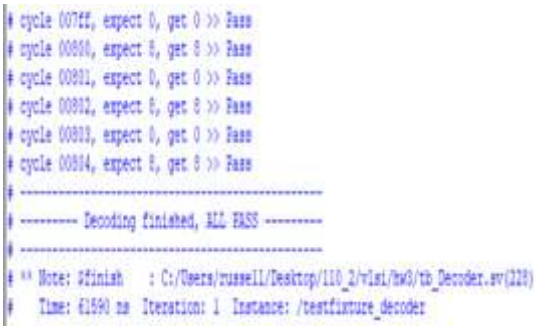
Simulation Result

Functional simulation	Pass (encoder)	Pass (decoder)	Gate-level simulation	Fail (encoder)	Pass (decoder)

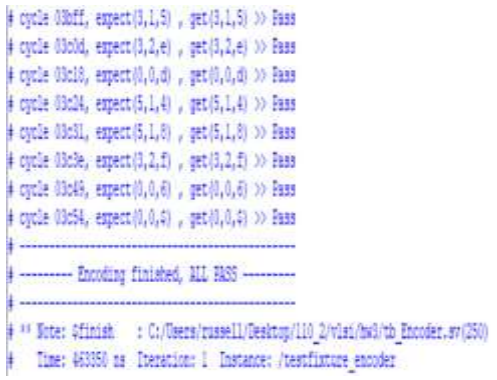
(your pre-sim result) encoder img0



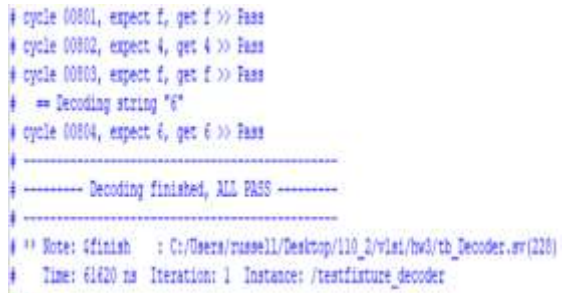
(your post-sim result) decoder img0



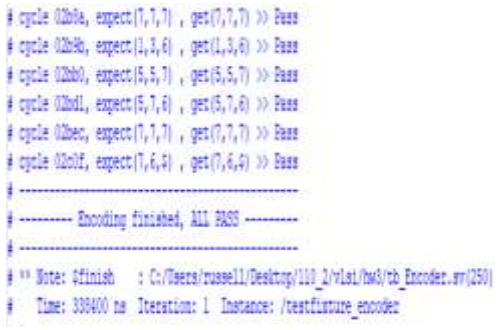
(your pre-sim result) encoder img1



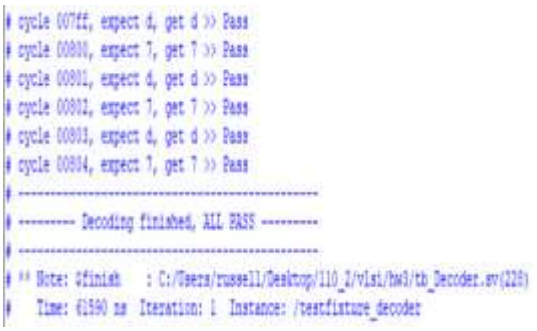
(your post-sim result) decoder img1



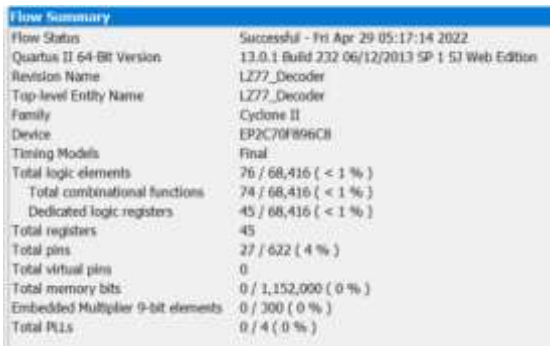
(your pre-sim result) encoder img2



(your post-sim result) decoder img2



Synthesis Result	**encoder**	**decoder**
Total logic elements		76
Total memory bit		0
Embedded multiplier 9-bit element		0

Simulation time img0	simulation time (ns)	61590 ns
Simulation time img1		61620 ns
Simulation time img2	simulation time (ns)	61590 ns
(your flow summary) encoder	(your flow summary) decoder 	

Flow Summary	
Flow Status	Successful - Fri Apr 29 05:17:14 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	76 / 68,416 (< 1 %)
Total combinational functions	74 / 68,416 (< 1 %)
Dedicated logic registers	45 / 68,416 (< 1 %)
Total registers	45
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

Decoder:

1. Assign encode = 0
2. Set the set signal = 1
   If set = 1 => tmp_len = code_len
   Else => tmp_len = tmp_len - 1
3. If tmp_len = 0 => char_nxt = chardata and set = 1
   Else => pos=[(code_pos+1)*4-1]，char_nxt = search_buffer[pos-:4] and set = 0
4. searchbuffer = {search_buffer, char_nxt}
5. tmp_len = tmp_len – 1
6. Repeat the from step2 to step4 until chardata = 8'h24 && tmp_len = 0
7. finish=1

Encoder:

1. Divided into four statements (input, count , output, finish)
2. Input state: If chardata != $ => next state = input state
3. Input state: Else => next state = count state
4. Count state: If offset < 9 => next state = count state
5. Count state: Else => next state = output state
6. Output state: If char_nxt != $ => next state = count state
7. Output state: Else => next state => finish state
8. Finish : Encoder done

- Input state: (finish = 0, valid = 0)

char buffer = {char buffer, chardata}

i=16399, j=16391, length=0, offset = 0

max length = 0, max offset = 0, max j = 16391;

- Count state: (finish = 0, valid = 0)

if Search buffer[i] == look ahead buffer[j]

i = i-8, j = j-8, length = length + 1

Else

if (length > max length)

max length = length, max length = length, max offset = offset, max j = j

i = i-8, j = j-8

else

length = 0, offset = offset + 1, i = 16399 + offset*3, j = 16391

char = char buffer[max pos-: 8];

- output state: (finish = 0, valid = 1)

shift char buffer (max length+1)*8

char_nxt = char, match_len = max length, offset = max offset

i=16399, j=16391, length=0, offset = 0

max length = 0, max offset = 0, max j = 16391;

- finish state: (finish = 1, valid = 0)

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