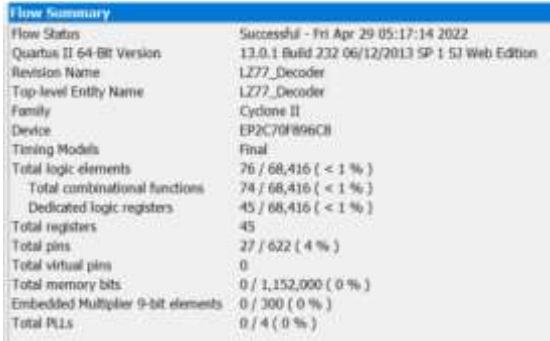


## 2022 Digital IC Design Homework 3

NAME	陳柏均				
Student ID	P76101584				
<b>Simulation Result</b>					
Functional simulation	Pass (encoder)	Pass (decoder)	Gate-level simulation	Fail (encoder)	Pass (decoder)
(your pre-sim result) encoder img0 <pre># cycle 000e, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 0010, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 001c, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 0025, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 002a, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 003a, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 0040, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 004d, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 0056, expect(7,7,8) , get(7,7,8) &gt;&gt; Pass # cycle 0063, expect(7,6,8) , get(7,6,8) &gt;&gt; Pass # # ----- Encoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Encoder.sv(250) # Time: 477908 ns Iteration: 1 Instance: /testfixture_encoder</pre>			(your post-sim result) decoder img0 <pre># cycle 007ff, expect 0, get 0 &gt;&gt; Pass # cycle 00800, expect 8, get 8 &gt;&gt; Pass # cycle 00801, expect 0, get 0 &gt;&gt; Pass # cycle 00802, expect 8, get 8 &gt;&gt; Pass # cycle 00803, expect 0, get 0 &gt;&gt; Pass # cycle 00804, expect 8, get 8 &gt;&gt; Pass # # ----- Decoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Decoder.sv(228) # Time: 61590 ns Iteration: 1 Instance: /testfixture_decoder</pre>		
(your pre-sim result) encoder img1 <pre># cycle 00bfff, expect(3,1,5) , get(3,1,5) &gt;&gt; Pass # cycle 00c0d, expect(3,2,e) , get(3,2,e) &gt;&gt; Pass # cycle 00c18, expect(0,0,d) , get(0,0,d) &gt;&gt; Pass # cycle 00c24, expect(5,1,4) , get(5,1,4) &gt;&gt; Pass # cycle 00c31, expect(5,1,8) , get(5,1,8) &gt;&gt; Pass # cycle 00c3e, expect(3,2,f) , get(3,2,f) &gt;&gt; Pass # cycle 00c45, expect(0,0,6) , get(0,0,6) &gt;&gt; Pass # cycle 00c54, expect(0,0,4) , get(0,0,4) &gt;&gt; Pass # # ----- Encoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Encoder.sv(250) # Time: 463350 ns Iteration: 1 Instance: /testfixture_encoder</pre>			(your post-sim result) decoder img1 <pre># cycle 00801, expect f, get f &gt;&gt; Pass # cycle 00802, expect 4, get 4 &gt;&gt; Pass # cycle 00803, expect f, get f &gt;&gt; Pass # == Decoding string "6" # cycle 00804, expect 4, get 6 &gt;&gt; Pass # # ----- Decoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Decoder.sv(228) # Time: 61620 ns Iteration: 1 Instance: /testfixture_decoder</pre>		
(your pre-sim result) encoder img2 <pre># cycle 02b0a, expect(7,7,7) , get(7,7,7) &gt;&gt; Pass # cycle 02b3b, expect(1,3,6) , get(1,3,6) &gt;&gt; Pass # cycle 02b6d, expect(5,5,7) , get(5,5,7) &gt;&gt; Pass # cycle 02bd1, expect(5,7,6) , get(5,7,6) &gt;&gt; Pass # cycle 02bec, expect(7,7,7) , get(7,7,7) &gt;&gt; Pass # cycle 02c0f, expect(7,6,5) , get(7,6,5) &gt;&gt; Pass # # ----- Encoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Encoder.sv(250) # Time: 338400 ns Iteration: 1 Instance: /testfixture_encoder</pre>			(your post-sim result) decoder img2 <pre># cycle 007ff, expect d, get d &gt;&gt; Pass # cycle 00800, expect 7, get 7 &gt;&gt; Pass # cycle 00801, expect d, get d &gt;&gt; Pass # cycle 00802, expect 7, get 7 &gt;&gt; Pass # cycle 00803, expect d, get d &gt;&gt; Pass # cycle 00804, expect 7, get 7 &gt;&gt; Pass # # ----- Decoding finished, ALL PASS ----- # ** Note: \$finish    : C:/Users/russell/Desktop/110_2/vlsi/tw3/tb_Decoder.sv(228) # Time: 61590 ns Iteration: 1 Instance: /testfixture_decoder</pre>		
<b>Synthesis Result</b>			<b>encoder                      decoder</b>		
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	<div>(your flow summary) decoder</div> 	

### 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 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)*