

Homework 3 + Lab 1

● Graded

Student

陳姍安

Total Points

84 / 90 pts

Question 1

Question 1

8 / 8 pts

+ 8 pts Correct

+ 6 pts A minor mistake

+ 4 pts A major mistake or two minor mistakes

+ 2 pts Some reasonable effort

+ 0 pts Totally wrong or empty

Question 2

Question 2

8 / 8 pts

+ 8 pts Correct

+ 7 pts 1 mistake

+ 6 pts 2 mistakes

+ 5 pts 3 mistakes

+ 4 pts 4 mistakes

+ 1 pt 7 mistakes

+ 0 pts Totally wrong

Question 3

Question 3

16 / 16 pts

✓ + 8 pts 3.1. Correct

+ 6 pts 3.1. A minor mistake

+ 4 pts 3.1. A major mistake or two minor mistakes

+ 2 pts 3.1. Some reasonable effort

+ 0 pts 3.1. Totally wrong or empty

✓ + 8 pts 3.2. Correct

+ 6 pts 3.2. A minor mistake

+ 4 pts 3.2. A major mistake or two minor mistakes

+ 2 pts 3.2. Some reasonable effort

+ 0 pts 3.2. Totally wrong or empty

Question 4

Question 4

28 / 28 pts

✓ + 28 pts Seems to work

+ 16 pts Only CLA seems to work

+ 12 pts Only RCA seems to work

+ 0 pts Empty or seems not to work (need demo?)

- 4 pts Using gates(modules) that not in gate.v, adder.v

- 4 pts Using non-gate-level implementation

Question 5

Question 5

8 / 12 pts

+ 12 pts Correct

✓ + 8 pts A minor mistake

+ 4 pts A major mistake

+ 0 pts Totally wrong or empty

not displaying corresponding change in the output

Question 6

Question 6

Resolved 6 / 8 pts

+ 8 pts Correct

+ 8 pts maximum delay correct, but transition is not default answer

✓ + 6 pts Almost correct

+ 4 pts Partially correct

+ 2 pts Some reasonable effort

+ 0 pts Totally wrong or empty

C Regrade Request

Submitted on: Apr 29

想詢問這題的錯誤是哪裡呢

你好，你的cla實作方式並沒有達到carry-lookahead的效果，根據第7題所計算出的結果，我們最長會需要 $2\log(n)$ level的gate，但你的實作方法最長會需要 n level，所以delay會變大。

Reviewed on: Apr 30

Question 7

Question 7

10 / 10 pts

✓ + 10 pts Correct ($2 \log_2 n$)

+ 8 pts Almost correct

+ 6 pts One minor mistake ($\log_2 n$)

+ 4 pts One major mistake or two minor mistakes

+ 2 pts Some reasonable effort

+ 0 pts Totally wrong or empty

No questions assigned to the following page.

CSIE 2344: Homework 3

Due April 29 (Monday) at Noon

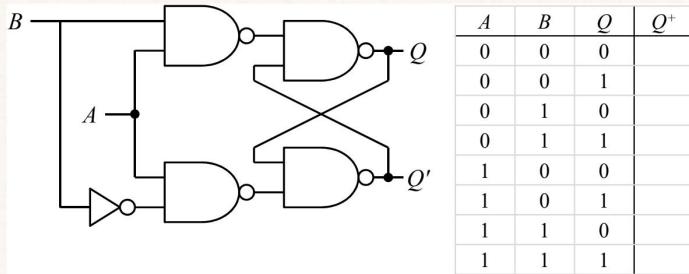
There are 90 points in total. Points will be deducted if no appropriate intermediate step is provided.
When you submit your homework on Gradescope, please select the corresponding page(s) of each question.

1 MUXes and Three-State Buffers (8pts)

Show how to make an 8-to-1 MUX using two 4-to-1 MUXes, two three-state buffers, and one inverter.

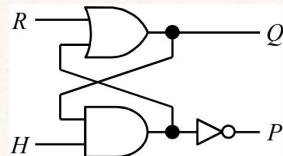
2 Latch I (8pts)

Given the latch as below, complete the following truth table.



3 Latch II (16pts)

A latch can be constructed from an OR gate, an AND gate, and an inverter connected as below.



1. What restriction must be placed on R and H so that P will always equal Q' (under steady-state conditions)?
2. Construct a next-state table and derive the minimum characteristic (next-state) equation for the latch.

No questions assigned to the following page.

4 Lab 1: Source Code (28pts)

1. (16pts) Print out your `cla_g1` source code and attach it with your Homework 3.
2. (12pts) Print out your `rca_g1` source code and attach it with your Homework 3.

We may ask you to demo in the future. Points will be deducted if your demo fails. **You can only demo with the source code same as that you submit.**

5 Lab 1: Waveform (12pts)

Show the waveform of `cla_g1` on input transition from $000 + 000 + 0$ to $000 + 111 + 1$. You should select all the input and output signals of `cla_g1` module.

6 Lab 1: Propagation Delays (8pts)

1. (4pts) Find the maximum propagation delay of `rca_g1` and one of the corresponding input transitions.
2. (4pts) Find the maximum propagation delay of `cla_g1` and one of the corresponding input transitions.

7 Lab 1: Some Derivation (10pts)

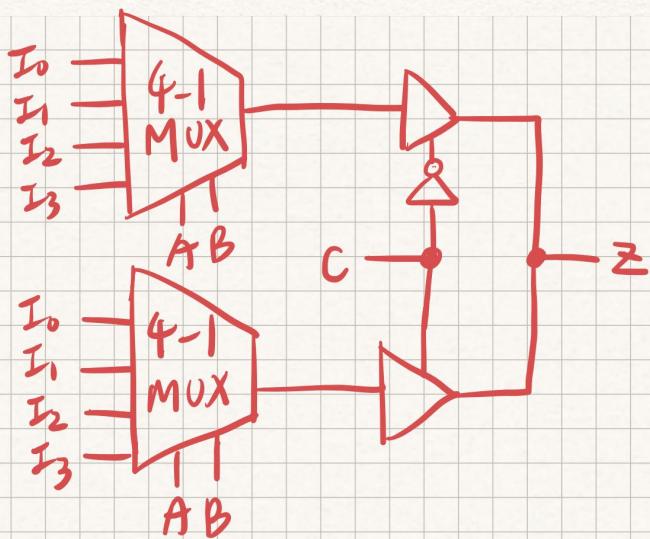
Assume that only 2-input gates are used. Derive the number of levels needed in an n -bit carry-lookahead adder as a function of n .

Questions assigned to the following page: [1](#) and [2](#)

1 MUXes and Three-State Buffers (8pts)

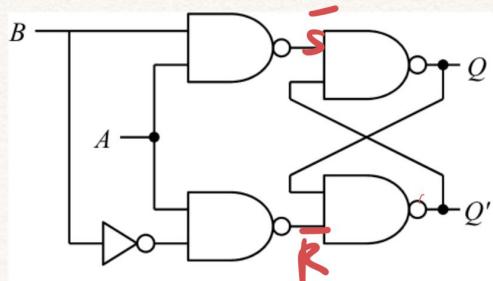
Show how to make an 8-to-1 MUX using two 4-to-1 MUXes, two three-state buffers, and one inverter.

Ans:



2 Latch I (8pts)

Given the latch as below, complete the following truth table.



A	B	Q	Q^+
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Ans:

A	B	\bar{S}	\bar{R}	Q	Q^+
0	0	1	1	0	0
0	0	1	1	1	1
0	1	1	1	0	0
0	1	1	1	1	1
1	0	1	0	0	0
1	0	1	0	1	0
1	1	0	1	0	1
1	1	0	1	1	1

$\bar{S} = (AB)' = A' + B'$

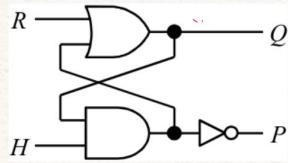
$\bar{R} = (AB')' = A' + B$

\bar{S}	\bar{R}	Q	Q^+	
1	1	0	0	Unchanged
1	1	1	1	
1	0	0	0	Reset to 0
1	0	1	0	
0	1	0	1	Set to 1
0	1	1	1	
0	0	0	X	Inputs Not Allowed
0	0	1	X	

Question assigned to the following page: [3](#)

3 Latch II (16pts)

A latch can be constructed from an OR gate, an AND gate, and an inverter connected as below.



- What restriction must be placed on R and H so that P will always equal Q' (under steady-state conditions)?
- Construct a next-state table and derive the minimum characteristic (next-state) equation for the latch.

Ans: 1. According all the cases listed in the table, $R=1$ and $H=0$ cannot occur at the same time. Therefore, the restriction is that $R=1$ and $H=0$.

time	R	H	Q	Q+	P	
t1	0	0	0	0	1	v
t2	0	0	0	0	1	v
t1	0	0	1	0	1	x
t2	0	0	0	0	1	v
t1	0	1	0	0	1	v
t2	0	1	0	0	1	v
t1	0	1	1	1	0	v
t2	0	1	1	1	0	v
t1	1	0	0	1	1	v
t2	1	0	1	1	1	x
t1	1	0	1	1	1	x
t2	1	0	1	1	1	x
t1	1	1	0	1	1	v
t2	1	1	1	1	0	v
t1	1	1	1	1	0	v
t2	1	1	1	1	0	v

2. By the k-map,

		00	01	11	10	
		Q\H	0	0	1	X
Q		0	0	1	1	X
	0	0	0	1	0	0
	1	0	1	1	0	X

$$Q^+ = R + H \bar{Q}$$

R	H	Q	Q ⁺
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	X
1	1	0	0
1	1	1	1

Question assigned to the following page: [4](#)

4 Lab 1: Source Code (28pts)

1. (16pts) Print out your `cla.gl` source code and attach it with your Homework 3.

2. (12pts) Print out your `rca.gl` source code and attach it with your Homework 3.

We may ask you to demo in the future. Points will be deducted if your demo fails. **You can only demo with the source code same as that you submit.**

1.

```
52 // carry-lookahead adder, gate level modeling
53 module cla_gl(
54     output C3,          // carry output
55     output[2:0] S,    // sum
56     input[2:0] A, B, // operands
57     input C0          // carry input
58 );
59
60 // Implement your code here.
61 wire [3:0] C;
62 wire [2:0] G, P, M;
63
64 AND g0(G[0], A[0], B[0]);
65 AND g1(G[1], A[1], B[1]);
66 AND g2(G[2], A[2], B[2]);
67 OR p0(P[0], A[0], B[0]);
68 OR p1(P[1], A[1], B[1]);
69 OR p2(P[2], A[2], B[2]);
70
71 assign C[0]=C0;
72 assign C3=C[3];
73
74 //assign C[1] = G[0] | (P[0] & C[0]);
75 AND p0c0(M[0], P[0], C[0]);
76 OR c1(C[1], G[0], M[0]);
77
78 //assign C[2] = G[1] | (P[1] & C[1]);
79 AND p1c1(M[1], P[1], C[1]);
80 OR c2(C[2], G[1], M[1]);
81
82 //assign C[3] = G[2] | (P[2] & C[2]);
83 AND p2c2(M[2], P[2], C[2]);
84 OR c3(C[3], G[2], M[2]);
85
86 FA fa0(, S[0], A[0], B[0], C[0]);
87 FA fa1(, S[1], A[1], B[1], C[1]);
88 FA fa2(, S[2], A[2], B[2], C[2]);
89
90 endmodule
```

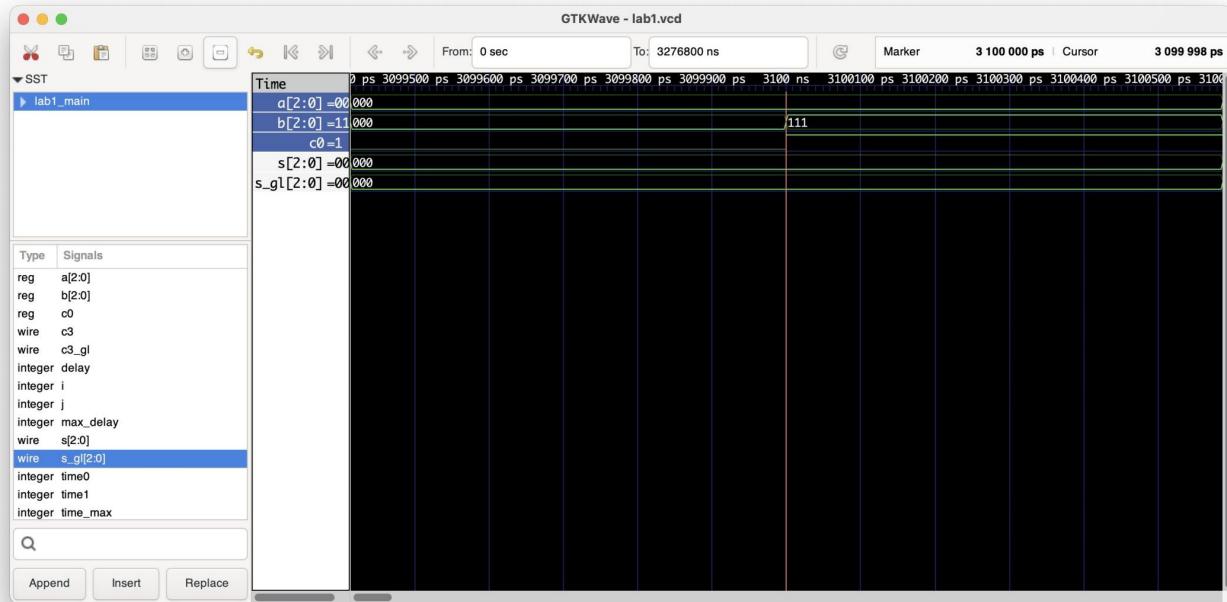
2.

```
35 // ripple-carry adder, gate level modeling
36 module rca_gl(
37     output C3,          // carry output
38     output[2:0] S,    // sum
39     input[2:0] A, B, // operands
40     input C0          // carry input
41 );
42 wire[3:0] c;
43
44 assign c[0] = C0;
45 assign C3 = c[3];
46
47 FA fa0(c[1], S[0], A[0], B[0], c[0]);
48 FA fa1(c[2], S[1], A[1], B[1], c[1]);
49 FA fa2(c[3], S[2], A[2], B[2], c[2]);
50 endmodule
```

Questions assigned to the following page: [5](#) and [6](#)

5 Lab 1: Waveform (12pts)

Show the waveform of `cla_g1` on input transition from $000 + 000 + 0$ to $000 + 111 + 1$. You should select all the input and output signals of `cla_g1` module.



6 Lab 1: Propagation Delays (8pts)

1. (4pts) Find the maximum propagation delay of `rca_g1` and one of the corresponding input transitions.
2. (4pts) Find the maximum propagation delay of `cla_g1` and one of the corresponding input transitions.

1. 23 ticks , $000+000+0 \rightarrow 000+111+1$

The maximum delay is 23 ticks on transition $000+000+0 \rightarrow 000+111+1$
lab1.v:70: \$finish called at 3276800000 (1ps)
3276800 / 000 / 111 / 1 / 1000 / 1111

2. 21 ticks , $000+000+0 \rightarrow 000+111+1$

The maximum delay is 21 ticks on transition $000+000+0 \rightarrow 000+111+1$
lab1.v:70: \$finish called at 3276800000 (1ps)
3276800 / 000 / 111 / 1 / 1000 / 1111

Question assigned to the following page: [7](#)

7 Lab 1: Some Derivation (10pts)

Assume that only 2-input gates are used. Derive the number of levels needed in an n -bit carry-lookahead adder as a function of n .

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

...

$$C_n = \sum_{i=0}^{n-1} \left(G_i \prod_{j=i+1}^{n-1} p_j \right) + C_0 \prod_{j=0}^{n-1} p_j$$

The calculation involves the sum of $n+1$ terms and the product of at most $n+1$ terms, leading to $2\lceil \log_2(n+1) \rceil$ levels of logic.

Adding an additional level for either the propagate (P_i) or generate (G_i) terms, the total number of levels becomes

$$2\lceil \log_2(n+1) \rceil + 1.$$

Question assigned to the following page: [7](#)

In another consideration, the calculation for S_{n-1} involves an exclusive OR (XOR) operation $C_{n-1} \oplus (A_{n-1} \oplus B_{n-1})$ which introduces an additional $2\lceil \log_2 n \rceil + 2$ levels. To sum up, it needs $\max \{ 2\lceil \log_2 (n+1) \rceil + 1, 2\lceil \log_2 n \rceil + 2 \}$ levels.