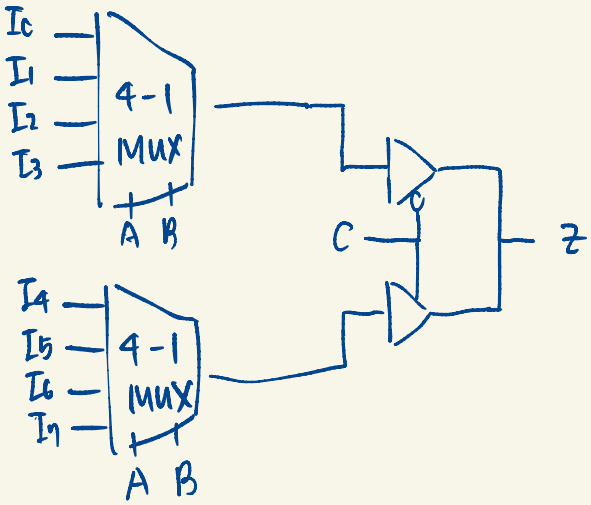
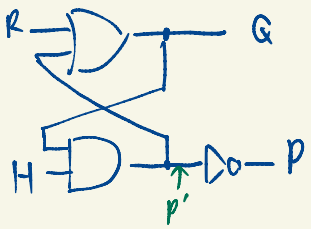
1.



2.

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

3. (a)



∵ output of OR gate $Q = R + p'$

$\Rightarrow Q = R + p' \equiv R + (Q')' = R + Q$ (∵ $p = Q'$)

$\Rightarrow R = 0 \quad \cdots ①$

∵ output of AND gate $p' = QH$

∵ $Q = R + p'$

$\Rightarrow p' = (R + p')H$

$\Rightarrow Q = (R + Q)H = RH + QH = QH$

∵ $Q = QH \Rightarrow H = 1 \quad \cdots ②$

∵ ① & ② $\Rightarrow R = 0, H = 1$ ✓

(b).

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

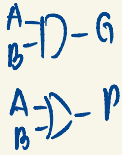
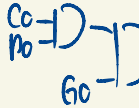
when $R = 1$ & $H = 0$: $Q = 1$, $p' = 0 \Rightarrow p = 1$

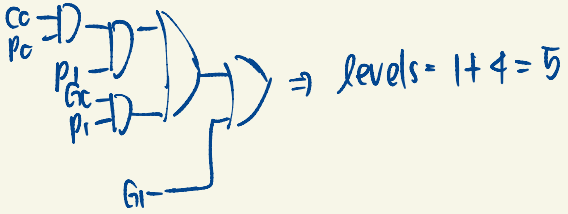
∵ $p = Q' \Rightarrow$ not allowed

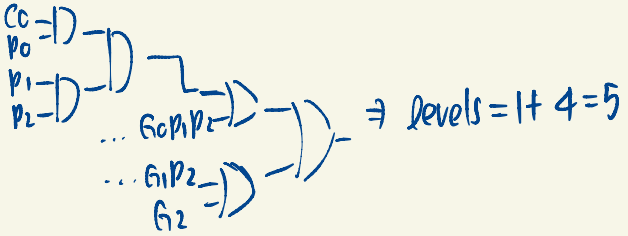
Q \ RH	00	01	11	10
0	0	0	1	not allowed
1	0	1	1	not allowed

$\Rightarrow Q^+ = RH + QH$

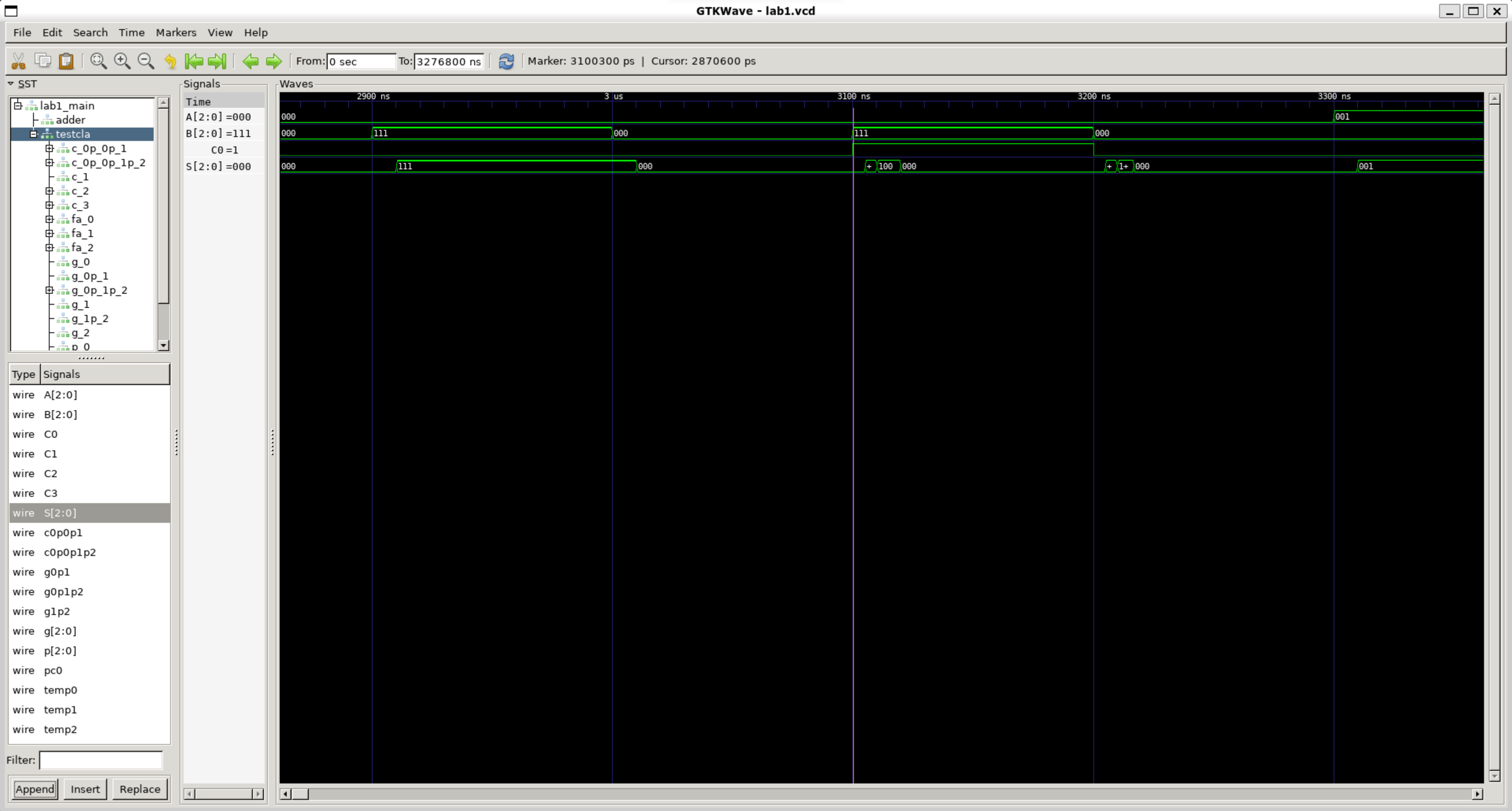
7.

1 bit:  +  ⇒ levels = 1 + 2 = 3

2 bit: -11- +  ⇒ levels = 1 + 4 = 5

3 bit: -11- +  ⇒ levels = 1 + 4 = 5

⇒ # of levels $F(n) = \underset{\text{for pg}}{1} + \underset{\text{and \& or gate}}{2} \cdot (\lceil \log_2(n+1) \rceil)$ ✗✗


GTKWave - lab1.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 3276800 ns
Marker: 3100300 ps | Cursor: 2870600 ps
SST
lab1_main
adder
testcla
c_0p_0p_1
c_0p_0p_1p_2
c_1
c_2
c_3
fa_0
fa_1
fa_2
g_0
g_0p_1
g_0p_1p_2
g_1
g_1p_2
g_2
p_0
Type Signals
wire A[2:0]
wire B[2:0]
wire C0
wire C1
wire C2
wire C3
wire S[2:0]
wire c0p0p1
wire c0p0p1p2
wire g0p1
wire g0p1p2
wire g1p2
wire g[2:0]
wire p[2:0]
wire pc0
wire temp0
wire temp1
wire temp2
Filter:
Append
Insert
Replace
Signals
Time
A[2:0]=000
B[2:0]=111
C0=1
S[2:0]=000
Waves
2900 ns
3 us
3100 ns
3200 ns
3300 ns
000
111
000
111
000
001
000
111
000
100
000
1+
000
001

4.

(1).

```
module rca_gl(
    output C3,       // carry output
    output[2:0] S,   // sum
    input[2:0] A, B, // operands
    input C0         // carry input
    );

    // TODO:: Implement gate-level RCA
    FA fa_0(C1, S[0], A[0], B[0], C0);
    FA fa_1(C2, S[1], A[1], B[1], C1);
    FA fa_2(C3, S[2], A[2], B[2], C2);
endmodule
```

(2).

```
module cla_gl(
    output C3,       // carry output
    output[2:0] S,   // sum
    input[2:0] A, B, // operands
    input C0         // carry input
    );

    // TODO:: Implement gate-level CLA
    wire [2:0] p, g;
    //0: first bit
    AND g_0(g[0], A[0], B[0]);
    OR p_0(p[0], A[0], B[0]);
    AND pc_0(pc0, p[0], C0);
    OR c_1(C1, g[0], pc0);
    FA fa_0(temp0, S[0], A[0], B[0], C0);

    //1: second bit
    AND g_1(g[1], A[1], B[1]);
    OR p_1(p[1], A[1], B[1]);
    AND4 c_0p_0p_1(c0p0p1, C0, p[0], p[1], 1'b1);
    AND g_0p_1(g0p1, g[0], p[1]);
    OR4 c_2(C2, g[1], g0p1, c0p0p1, 1'b0);
    FA fa_1(temp1, S[1], A[1], B[1], C1);

    //2: third bit
    AND g_2(g[2], A[2], B[2]);
```

```
    OR p_2(p[2], A[2], B[2]);
    AND4 c_0p_0p_1p_2(c0p0p1p2, C0, p[0], p[1], p[2]);
    AND4 g_0p_1p_2(g0p1p2, g[0], p[1], p[2], 1'b1);
    AND g_1p_2(g1p2, g[1], p[2]);
    OR4 c_3(C3, g[2], g1p2, g0p1p2, c0p0p1p2);
    FA fa_2(temp2, S[2], A[2], B[2], C2);

endmodule
```

6. (a)Maximum propogation delay for rca_gl (ripple carry adder)

```
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$ iverilog -o lab1_compiled.vvp lab1.v
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$ vvp lab1_compiled.vvp
VCD info: dumpfile lab1.vcd opened for output.
The maximum delay is 23 ticks on transition 000+000+0 --> 000+111+1
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$
```

(b) Maximum propogation delay for cla_gl (carry lookahead adder)

```
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$ iverilog -o lab1_compiled.vvp lab1.v
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$ vvp lab1_compiled.vvp
VCD info: dumpfile lab1.vcd opened for output.
The maximum delay is 20 ticks on transition 000+000+0 --> 000+011+1
justin@DESKTOP-DSHM2G6:/mnt/c/Users/justi/OneDrive/Desktop/LAB-01$
```