# Lab 1

## Source Code

### 1. `cla_gl` source code

```
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] g, p;
  wire c1, c2;
  // my naming is bad, sorry~~

  // c1 = g0 | (p0 & c0)
  OR xb0(p[0], A[0], B[0]);
  AND ab0(g[0], A[0], B[0]);
  AND pc_0(cp0, p[0], C0);
  OR c_1(c1, g[0], cp0);
  FA fa0(tmpc0, S[0], A[0], B[0], C0);

  // c2 = g1 | (g0 & p1) | (c0 & p0 & p1)
  OR xb1(p[1], A[1], B[1]);
  AND ab1(g[1], A[1], B[1]);
  AND gp_1(gp1, g[0], p[1]);
  AND4 cpp_1(cpp1, 1, C0, p[0], p[1]);
  OR4 c_2(c2, g[1], gp1, cpp1, 0);
  FA fa1(tmpc1, S[1], A[1], B[1], c1);

  // c3 = g2 | (g1 & p2) | (g0 & p1 & p2) | (c0 & p0 & p1 & p2)
  OR xb2(p[2], A[2], B[2]);
  AND ab2(g[2], A[2], B[2]);
  AND gp_2(gp2, g[1], p[2]);
  AND4 gpp_2(gpp2, g[0], p[1], p[2], 1);
  AND4 cppp_2(cppp2, C0, p[0], p[1], p[2]);
  OR4 c_3(C3, g[2], gp2, gpp2, cppp2);
  FA fa2(tmpc2, S[2], A[2], B[2], c2);
endmodule
```

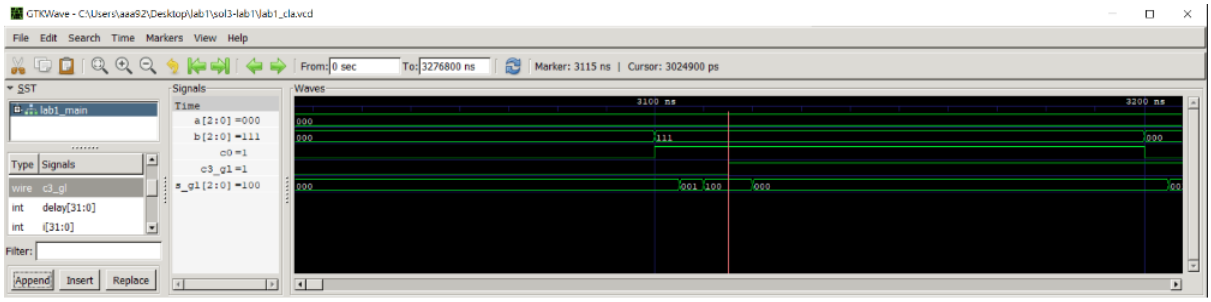
### 2. `rca_gl` source code

```
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
    wire C1, C2;
    FA fa0(C1, S[0], A[0], B[0], C0);
    FA fa1(C2, S[1], A[1], B[1], C1);
    FA fa2(C3, S[2], A[2], B[2], C2);
endmodule
```

# Waveform

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




# Propagation Delays

1. Find the maximum propagation delay of `rca_gl` and one of the corresponding input transitions.

   23 ticks. From 000 + 000 + 0 to 000 + 111 + 1.

2. Find the maximum propagation delay of `cla_gl` and one of the corresponding input transitions.

   20 ticks. From 000 + 000 + 0 to 000 + 011 + 1.

# Some Derivation

> 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 $n$.

$C_n$ can be derived as follows.

$$
\begin{aligned}
C_1 &= G_0 + P_0 C_0 \\
C_2 &= G_1 + G_0 P_1 + C_0 P_0 P_1 \\
C_3 &= G_2 + G_1 P_2 + G_0 P_1 P_2 + C_0 P_0 P_1 P_2 \\
&\vdots \\
C_n &= \sum_{i=0}^{n-1} \left( G_i \prod_{j=i+1}^{n-1} P_j \right) + C_0 \prod_{j=0}^{n-1} P_j
\end{aligned}
$$

The summation of $n+1$ terms and the multiplication is of at most $n+1$ terms, which implies $2\lceil \log_2(n+1) \rceil$ levels. Consider one level for $P_i$ or $G_i$, # of level is $2\lceil \log_2(n+1) \rceil + 1$.

On the other hand, $S_{n-1}$ is $C_{n-1} \oplus (A_{n-1} \oplus B_{n-1})$, which implies $2\lceil \log_2 n \rceil + 2$ levels.

Combined, $\max\{2\lceil \log_2(n+1) \rceil + 1, 2\lceil \log_2 n \rceil + 2\}$ is needed.