Overview of the *Carry-Lookahead (CLA) Adder*

Christopher D. Carothers
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, New York U.S.A. 12180-3590

January 18, 2018

1 Equation Dependencies for 64 Bit CLA

Recall that we can express the sum of two numbers as:

$$sum_i = a_i \oplus b_i \oplus c_{i-1} \tag{1}$$

where a_i and b_i are the two inputs numbers we wish to add and c_{i-1} is the carry-in to that bit stage. Also, $\oplus$ is the exclusive-or (XOR) operation which is true only if either a_i or b_i is true but not both. Think of the exclusive-or operation as a parity counter. See `https://en.wikipedia.org/wiki/XOR_gate` for more details.

Now, the above equation is completely dependent on the c_i carry-in array. The driving principle of the CLA adder is to compute the carry-in array first and to leverage as much parallelism (and hardware) as possible.

To accomplish this, we can define the c_i carry-in function as follows:

$$c_i = g_i + p_i c_{i-1} \tag{2}$$

where g_i is the "generate function", which says did we generate a carry in the i^{th} stage and the p_i is the propagate function which says did we propagate a carry in the i^{th} stage assuming the carry-in, c_{i-1}, was positive. This yields the following:

$$g_i = a_i \times b_i \tag{3}$$
$$p_i = a_i + b_i \tag{4}$$
$$c_i = g_i + p_i c_{i-1} \tag{5}$$

Now, using the above recurrence we can find what c_i is for any 4 bit block or "group".

$$
\begin{align}
c_i &= g_i + p_i c_{i-1} \tag{6}\\
c_{i+1} &= g_{i+1} + p_{i+1} c_i \tag{7}\\
c_{i+2} &= g_{i+2} + p_{i+2} c_{i+1} \tag{8}\\
c_{i+3} &= g_{i+3} + p_{i+3} c_{i+2} \tag{9}
\end{align}
$$

Notice, how each of the c_i equations can all be written in terms of the the g, p and c_{i-1}. But, c_{i-1} is really the carry-in for this "group" of 4 bits. So, this means that the carry-in to those group depends on the gc equations, which are:

$$
\begin{align}
gc_j &= gg_j + gp_j gc_{j-1} \tag{10}\\
gc_{j+1} &= gg_{j+1} + gp_{j+1} gc_j \tag{11}\\
gc_{j+2} &= gg_{j+2} + gp_{j+2} gc_{j+1} \tag{12}\\
gc_{j+3} &= gg_{j+3} + gp_{j+3} gc_{j+2} \tag{13}\\
& \tag{14}
\end{align}
$$

$$
\begin{align}
&\mathit{where...} \nonumber\\
gg_j &= g_{i+3} + p_{i+3} g_{i+2} + p_{i+3} p_{i+2} g_{i+1} + p_{i+3} p_{i+2} p_{i+1} g_i \nonumber\\
gp_j &= p_{i+3} p_{i+2} p_{i+1} p_i \tag{15}
\end{align}
$$

Again, notice, how each of the gc_j equations can all be written in terms of the the gg, gp and gc_{j-1}. But, gc_{j-1} is really the carry-in for this "section" of 4 bits. So, this means that the carry-in to those group depends on the sc equations, which are:

$$
\begin{align}
sc_k &= sg_k + sp_k sc_{k-1} \tag{16}\\
sc_{k+1} &= sg_{k+1} + sp_{k+1} sc_k \tag{17}\\
sc_{k+2} &= sg_{k+2} + sp_{k+2} sc_{k+1} \tag{18}\\
sc_{k+3} &= sg_{k+3} + sp_{k+3} sc_{k+2} \tag{19}\\
& \tag{20}
\end{align}
$$

$$
\begin{align}
&\mathit{where...} \nonumber\\
sg_k &= gg_{j+3} + gp_{j+3} gg_{j+2} + gp_{j+3} gp_{j+2} gg_{j+1} + gp_{j+3} gp_{j+2} gp_{j+1} gg_j \nonumber\\
sp_k &= gp_{j+3} gp_{j+2} gp_{j+1} gp_j \tag{21}
\end{align}
$$

2 Steps for Calculation for 64 Bit CLA with 4 Bit Blocks

1. Calculate g_i and p_i for all i.

2. Calculate gg_j and gp_j for all j using g_i and p_i.

3. Calculate sg_k and sp_k for all k using gg_j and gp_j. *Note, it is at this point, we can shift to computing the top-level sectional carries. This is because the number of sections is less than or equal the block size which is 4 bits.*

4. Calculate sc_k using sg_k and sp_k for all k and 0 for sc_{-1}.

5. Calculate gc_j using gg_j, gp_j and correct $sc_k, k = jdiv4$ as sectional carry-in for all j.

6. Calculate c_i using g_i, p_i and correct $gc_j, j = idiv4$ as group carry-in for all i.

7. Calculate sum_i using $a_i \oplus b_i \oplus c_i$ for all i.