Carry look-ahead adder

The ripple-carry adder, its limiting factor is the time it takes to propagate the carry. The carry look-ahead adder solves this problem by calculating the carry signals in advance, based on the input signals. The result is a reduced carry propagation time.

To be able to understand how the carry look-ahead adder works, we have to manipulate the Boolean expression dealing with the full adder. The Propagate $P$ and generate $G$ in a full-adder, is given as:

\[
\begin{align*}
P_i &= A_i \oplus B_i \quad \text{Carry propagate} \\
G_i &= A_iB_i \quad \text{Carry generate}
\end{align*}
\]

Notice that both propagate and generate signals depend only on the input bits and thus will be valid after one gate delay.

The new expressions for the output sum and the carryout are given by:

\[
\begin{align*}
S_i &= P_i \oplus C_{i-1} \\
C_{i+1} &= G_i + P_iC_i
\end{align*}
\]

These equations show that a carry signal will be generated in two cases:

1) if both bits $A_i$ and $B_i$ are 1

2) if either $A_i$ or $B_i$ is 1 and the carry-in $C_i$ is 1.

Let's apply these equations for a 4-bit adder:

\[
\begin{align*}
C_1 &= G_0 + P_0C_0 \\
C_2 &= G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0 \\
C_3 &= G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0 \\
C_4 &= G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0
\end{align*}
\]
These expressions show that C2, C3 and C4 do not depend on its previous carry-in. Therefore C4 does not need to wait for C3 to propagate. As soon as C0 is computed, C4 can reach steady state. The same is also true for C2 and C3.

The general expression is

\[ C_{i+1} = G_i + P_iG_{i-1} + P_iP_{i-1}G_{i-2} + \ldots + P_iP_{i-1}\ldots P_2P_1G_0 + P_iP_{i-1}\ldots P_1P_0C_0. \]

This is a two level circuit. In CMOS however the delay of the function is nonlinearly dependent on its fan-in. Therefore large fanin gates are not practical.

Carry look-ahead adder’s structure can be divided into three parts: the propagate/generate generator Fig.1, the sum generator Fig. 2 and the carry generator Fig. 3.

---

**Fig. 1 Propagate /Generate generator**

**Figure 2: Sum Generator**
Figure 4 shows the carry generator needed to add four bits numbers. To make the carry generator from 4 bits to n bits, we need only add AND gates and inputs for the OR gate. The largest AND gate in the carry section has always n+1 inputs and the number of AND gates requirements is n. Therefore the design of a 16 bits adder needs the last carry generator section to have 16 AND gates, where the biggest AND gate has 17 inputs. Also the OR gate in this section needs 17 inputs.

![Look-Ahead Carry generator](image)

**Fig. 3 Look-Ahead Carry generator**

The size and fan-in of the gates needed to implement the Carry-Look-ahead adder is usually limited to four, so 4-bit Carry-Look ahead adder is designed as a block. The 4-bit Carry Look Ahead adder block diagram is shown in Fig.4. The delay of such circuit is 4 levels of logic.
Figure 4: 4-bit Carry Look Ahead Adder