

Carry Lookahead Adder

Ruben Van den Bossche

Sam Verboven

Addition

$$\begin{array}{r} 00101110 \\ + 00100111 \\ \hline \end{array}$$

Addition

$$\begin{array}{r} 10111\ 00 \\ 00101110 \\ + \ 00100111 \\ \hline 01010101 \end{array}$$

Addition

- 1-bit addition
 - Sum output is 1 when exactly one input is 1 or all three inputs are 1
 - CarryOut output is 1 when at least two inputs are 1
- Carry ripples through addition
 - Hardware executes in parallel
 - Slow!

Faster addition

- Determine CarryIn's sooner
 - Faster
 - More hardware needed
- Two levels of abstraction
 - Blocks of 4-bit adder
 - Within each block the carry in's are calculated before the addition is performed

Propagate and generate

$$\begin{array}{r} 00101110 \\ + 00100111 \\ \hline \end{array}$$

Propagate and generate

$$\begin{array}{r} 0010 \\ + 0010 \\ \hline \end{array} \qquad \begin{array}{r} 1110 \\ + 0111 \\ \hline \end{array}$$

Propagate and generate

$$\begin{array}{r} \text{????} \\ \text{0010} \\ + \text{0010} \\ \hline \end{array} \qquad \begin{array}{r} \text{????} \\ \text{1110} \\ + \text{0111} \\ \hline \end{array}$$

Propagate and generate

- Generate:

$$g_i = a_i \cdot b_i$$

$$C_4 C_3 C_2 C_1 C_0$$

- Propagate:

$$p_i = a_i + b_i$$

$$a_3 a_2 a_1 a_0$$

$$+ b_3 b_2 b_1 b_0$$

- CarryIn:

$$C_{i+1} = g_i + p_i \cdot C_i$$

$$S_3 S_2 S_1 S_0$$

Propagate and generate

- $C_1 = g_0 + (p_0 \cdot c_0)$

$$\begin{aligned} C_2 &= g_1 + (p_1 \cdot g_0) \\ &\quad + (p_1 \cdot p_0 \cdot c_0) \end{aligned}$$

$$\begin{aligned} C_3 &= g_2 + (p_2 \cdot g_1) \\ &\quad + (p_2 \cdot p_1 \cdot g_0) \\ &\quad + (p_2 \cdot p_1 \cdot p_0 \cdot c_0) \end{aligned}$$

$$\begin{aligned} C_4 &= g_3 + (p_3 \cdot g_2) \\ &\quad + (p_3 \cdot p_2 \cdot g_1) \\ &\quad + (p_3 \cdot p_2 \cdot p_1 \cdot g_0) \\ &\quad + (p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0) \end{aligned}$$

$C_4 C_3 C_2 C_1 C_0$

$a_3 a_2 a_1 a_0$

$+ b_3 b_2 b_1 b_0$

$S_3 S_2 S_1 S_0$

Propagate and generate

$$\begin{array}{r} 0010 \\ + 0010 \\ \hline \end{array} \qquad \begin{array}{r} 1110 \\ + 0111 \\ \hline \end{array}$$

Propagate and generate

$$\begin{array}{r} & 11100 \\ & 1110 \\ \hline 0010 & + 0111 \\ \hline \end{array}$$
$$\begin{array}{r} + 0010 \\ \hline \end{array}$$

Propagate and generate

$$\begin{array}{r} & \textcircled{1}1100 \\ \xleftarrow{\hspace{1cm}} & \\ \begin{array}{r} 0010 \\ + 0010 \\ \hline \end{array} & \begin{array}{r} 1110 \\ + 0111 \\ \hline 0101 \end{array} \end{array}$$

Propagate and generate

$$\begin{array}{r} 0101 \\ 0010 \\ + 0010 \\ \hline 0101 \end{array} \quad \begin{array}{r} 11100 \\ 1110 \\ + 0111 \\ \hline 0101 \end{array}$$

The diagram illustrates the propagation and generation of carries in binary addition. The first column shows the addition of 0101 and 0010, resulting in 0101. The second column shows the addition of 0010 and 1110, resulting in 0101. A circled '1' at the top of the second column indicates a carry being generated. An arrow points from this circled '1' to the first '0' in the result 0101, indicating that this carry is being propagated to the next column.

Superpropagates and -generates

- Calculate c_4 of each 4-bit carry-lookahead adder
- Superpropagate P_i and supergenerate G_i
 - $P_0 = p_3 \cdot p_2 \cdot p_1 \cdot p_0$
 - $G_0 = g_3 + (p_3 \cdot g_2) + (p_3 \cdot p_2 \cdot g_1) + (p_3 \cdot p_2 \cdot p_1 \cdot g_0)$
- Calculate C_i
 - $C_1 = G_0 + (P_0 \cdot C_0)$
 - $C_2 = G_1 + (P_1 \cdot G_0) + (P_1 \cdot P_0 \cdot C_0)$
 - ...

8-bit Carry Lookahead Adder

