EE 2720

Handout #22.
• **Ripple Carry Adders**

Consider two n-bit numbers

\[ A = a_{n-1} a_{n-2} \ldots a_1 a_0 \]
\[ B = b_{n-1} b_{n-2} \ldots b_1 b_0 \]

One circuit that can add them is the **ripple carry adder**. An n-bit ripple carry adder is shown in figure 1.

![Diagram of ripple carry adder](image)

**Fig. 1**: An n-bit ripple carry adder.

In the above fig.1, FA means full adder. By now you know what a full adder is and how to design it (look at HW#6).

Also, in the above fig.1, \( s_0, s_1, \ldots, s_{n-2}, s_{n-1} \) are the sum bits, while \( c_0, c_1, \ldots, c_{n-2}, c_{n-1} \) are the carry-out bits.

\( c_{-1} = c_{in} \) is the overall carry-in to the n-bit ripple carry adder, while Cout is the overall carry-out.

You can use the n-bit ripple carry adder.
of Fig. 1 to perform the addition \( A + B \) or the subtraction \( A - B = A + \text{two's complement of} \ B = A + \text{one's complement of} \ B + 1 \). If you want to perform \( A + B \), then you set the \( \text{cin} \) to \( \text{cin} = 0 \); if you want to perform \( A - B \), then you must use \( b_{n-1}, b_{n-2}, \ldots, b_1, b_0 \) instead of \( b_{n-1}, b_{n-2}, \ldots, b_1, b_0 \) and \( \text{cin} = 1 \).

Propagation Delay: The propagation delay (worst-case delay) through the \( n \)-bit ripple carry adder of Fig. 1 is \( D_{\text{ripple}} = n \times D_{\text{FA}} \), where \( D_{\text{FA}} \) is the worst-case propagation delay through a full adder. As a comment, what I have to say here is that the ripple carry adder, although it is a very simple design, it is also very slow. Its propagation delay is directly proportional to the length of the numbers that are added; you double the length, the delay gets double, etc...

A faster, but more expensive adder design is the Carry Lookahead Adder not to be presented here in EE 2720. You will learn about it in EE 3755 if you take it. I teach this course.

Cascading (connecting) ripple carry adders.

Example 1: Using two 4-bit ripple carry
adders, design an 8-bit ripple carry adder.

Answer: Figure 2 below shows the design:

\[ \begin{aligned}
    &a_7 a_6 a_5 a_4 b_7 b_6 b_5 b_4 \\
    \text{4-bit ripple carry adder} \\
    &s_7 s_6 s_5 s_4 \\
    \text{C} \rightarrow \\
    \text{Cout} \\
\end{aligned} \]

\[ \begin{aligned}
    &a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 \\
    \text{4-bit ripple carry adder} \\
    &s_3 s_2 s_1 s_0 \\
    \text{C} \rightarrow \\
    \text{Cin} \\
\end{aligned} \]

Fig. 2: An 8-bit ripple carry adder constructed from two 4-bit ripple carry adders. Here, the two numbers to be added are \(A = a_7 a_6 \cdots a_1 a_0\) and \(B = b_7 b_6 \cdots b_1 b_0\).

Note: The above topic on ripple carry adders is optional in your textbook, but I thought it would be a good idea presenting it. Anyways, I also do this in EE 3755.