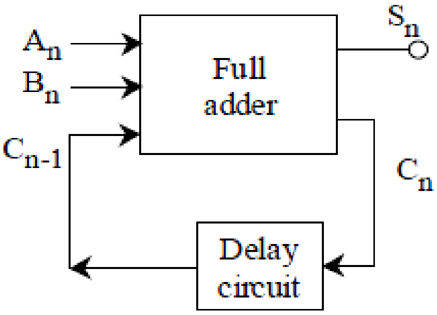
PARALLEL BINARY ADDER

SERIAL BINARY ADDER

The **serial binary adder** is a digital circuit that performs binary addition bit by bit. The serial full adder has three single bit inputs for the numbers to be added and the carry in. There are two single bit outputs for the sum and carry out. The carry in signal is the previously calculated carry out signal. The addition is performed by adding each bit, lowest to highest, once each clock cycle.



Serial binary Adder

PARALLEL BINARY ADDER

The adders discussed in the previous section have been limited to adding single-digit binary numbers and carries. The largest sum that can be obtained using a full adder is 11_2. Parallel adders let us add multiple-digit numbers. If we place full adders in parallel, we can add two- or four-digit numbers or any other size desired. Below figure uses STANDARD SYMBOLS to show a parallel adder capable of adding two, two-digit binary numbers. In previous discussions we have depicted circuits with individual logic gates shown. Standard symbols (blocks) allow us to analyze circuits with inputs and outputs only. One standard symbol may actually contain many and various types of gates and circuits. The addend would be input on the A inputs (A_2 = MSD, A_1 = LSD), and the augend input on the B inputs (B_2 = MSD, B_1 = LSD). For this explanation we will assume there is no input to C_0 (carry from a previous circuit).

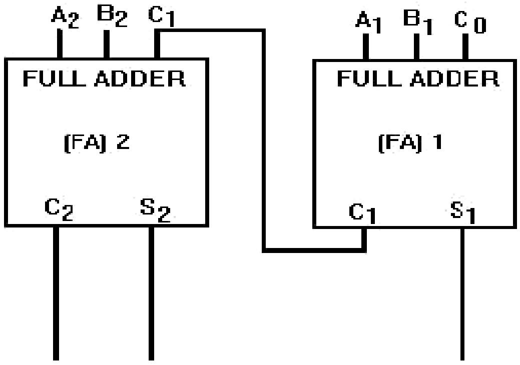


Figure: Parallel binary adder**.**

Now let's add some two-digit numbers. To add 10_2 (addend) and 01_2 (augend), assume there are numbers at the appropriate inputs. The addend inputs will be 1 on A_2 and 0 on A_1. The augend inputs will be 0 on B_2 and 1 on B_1. Working from right to left, as we do in normal addition, let's calculate the outputs of each full adder. With A_1 at 0 and B_1 at 1, the output of adder 1 will be a sum (S_1) of 1 with no carry (C_1). Since A_2 is 1 and B_2 is 0, we have a sum (S_2) of 1 with no carry (C_2) from adder 1. To determine the sum, read the outputs (C_2, S_2, and S_1) from left to right. In this case, $C_2 = 0$, $S_2 = 1$, and $S_1 = 1$. The sum, then, of 10_2 and 01_2 is 011_2 or 11_2. To add 11_2 and 01_2, assume one number is applied to A_1 and A_2, and the other to B_1 and B_2, as shown in figure 3-10. Adder 1 produces a sum (S_1) of 0 and a carry (C_1) of 1. Adder 2 gives us a sum (S_2).

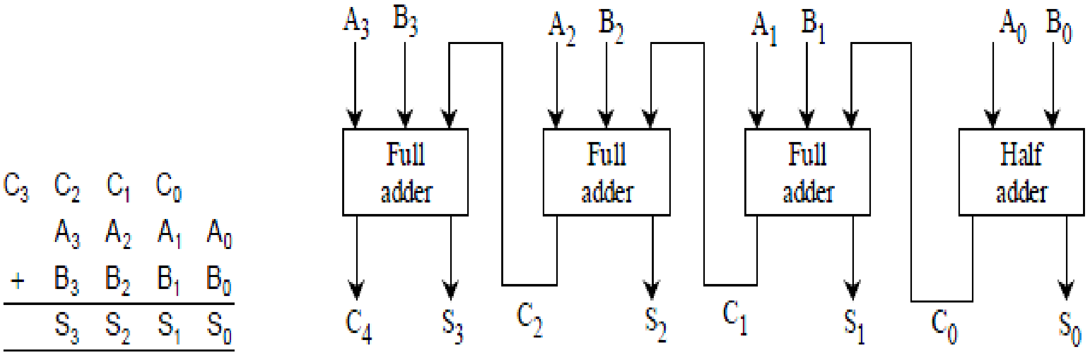


Fig : 4-bit full adder

At the time of adding the LSBs; only two bits have to added and hence a half adder is sufficient but at higher significant positions, a full adder is required to add the bits of A and B along with the carry from the previous position. Hence, each full adder adds the n_{th} bits with the carry from the $(n - 1)_{th}$ bit.