# Adding Bits With Logic Gates (cont.)

## Example:

1 1 1 0 1 1 0 1  
01 11 10 01 11 10 10 00

cout-> 1 0 |1011011 <-sum

## Truth Table for a Full Adder

We want to create a logical circuit to implement a full adder. Lets start with the truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin | a | b | sum | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Later we will see a mechanism for creating equations and circuits from truth tables.

For now, we are going to create a circuit by intuition and pattern-matching.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin | a | b | sum | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Consider the two blocks above.

In the first block Cin = 0, in the second Cin = 1.

In the first block, sum = aXORb and Cout = a.b, which is the same as the half adder.

In the second block, sum = aXNORb and Cout = a+b.

## Rewriting

If Cin = 0:

sum = aXORb, Cout = a.b

If Cin = 1:

sum = aXNORb, Cout = a+b

The following circuit implements this: (get picture)

a and b connected to each circuit listed above appropriately.

Those circuits connected to Cin by 2 multiplexors.

## Multiplexor:

One input to an AND can control the gate. If it's 0, the gate outputs a 0. If it's 1 the gate outputs the other input to it.

By putting an OR after two ANDs, where one input to one AND is Cin and one input to the other AND is Cin', the value of Cin controls whether the result is taken from the first AND or the second AND.