Recall the we determined two separate sets of Equations for describing a Full-ADDER:

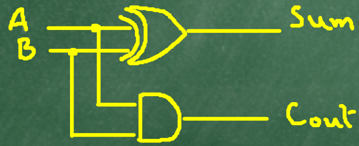
if $C_{in} = 0$ : $Sum = a \oplus b$
$C_{out} = a \cdot b$

and

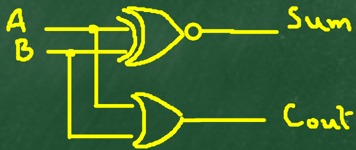
if $C_{in} = 1$ : $Sum = a \odot b$
$C_{out} = a + b$

This gives us 2 distinct circuits, each depending on the value of Cin.

So, if $C_{in} = 0$, we get




and if $C_{in} = 1$, we get



Let's think about this for a moment...

In this form, Can we Say that Cin Is being added to anything?

From our discussion of the half-adder, We Concluded that the half-Sum $S = A \oplus B$. (this ignored Cin, essentially $C_{in}$ was treated as 0)

Now, from the Full-adder Truth Table, we also conclude that the half-Sum, when $C_{in} = 1$ is $A \odot B$.

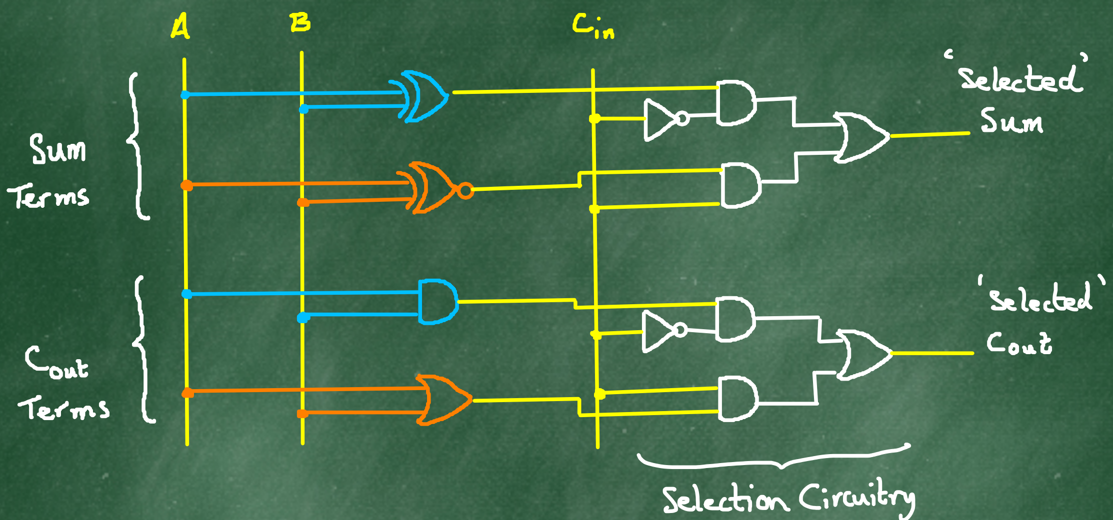
therefore, the value of Cin Is reflected in the half-Sum, and Indeed In Cout Calculation.

Our challenge is to Combine the set of two equations for the Sum and for the Cout Into one Single Circuit.

We Could do this by creating a Circuit that Contains both Sets of equations and Selecting, or choosing, one or the other, to be the final output, depending on the value of Cin.

# Consider the following Circuit

| $C_{in}$ | 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 |




| $C_{in}$ | 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 |

$C_{in} = 0$

$$Sum = A \oplus B$$

$$C_{out} = A \cdot B$$

$C_{in} = 1$

$$Sum = \overline{A \oplus B} = A \odot B$$

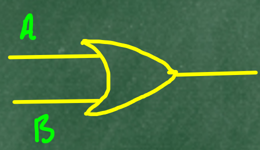
$$C_{out} = A + B$$

The operation of the Selection Circuitry Is based on the following two observations:

## OBSERVATION 1

When an Input to an OR-gate = 0, output will be the Same as the other Input. We Say that the output follows that Input

| a | b | a+b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

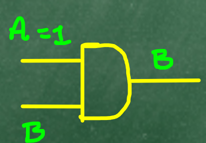



If, by construction, I Can ensure that a Certain Input is 0, I Know the output will follow the other Input.

## OBSERVATION 2

When an Input to an AND-gate = 1, the output follows the other Input:

| a | b | a.b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

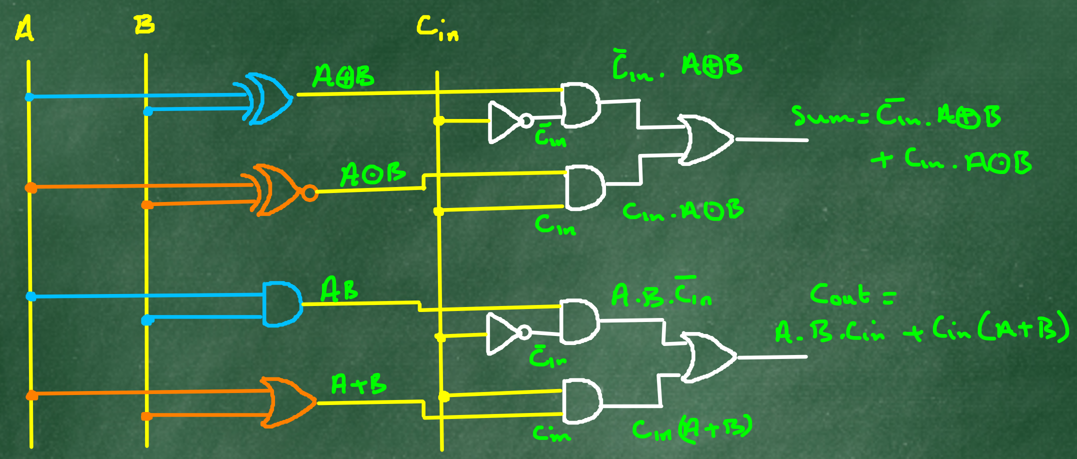



If I Know, by Construction, that The Input, A, = 1. I am ensured that the output will follow B.

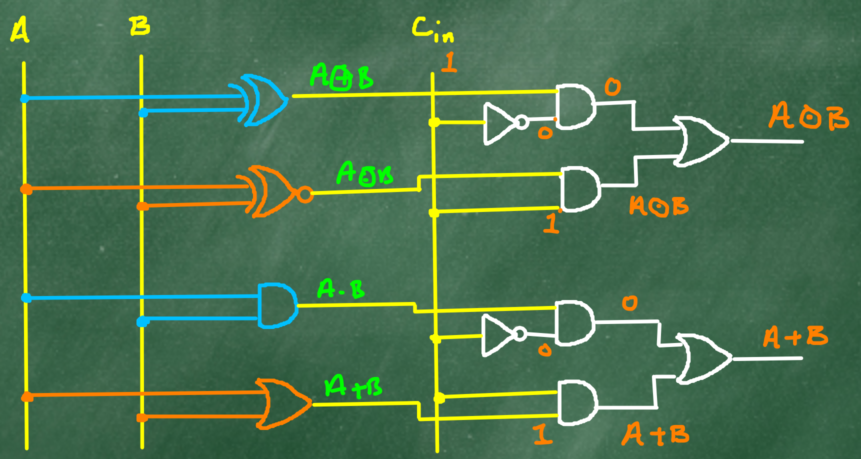
If I set A = 0, the output will be = 0 regardless of the value of B

In both Cases, we Can explicitly Choose one Input and manipulate it to Control how the output behaves with respect to the other Input.

With this knowledge, we can trace the logic through our circuit:




if $C_{in} = 1$, for example, we get the following:




Try it yourself: trace the logic through if $C_{in} = 0$.

In our Full-adder Implementation, we are using the value of $C_{in}$ to Create one of two Possible pathways through a Circuit.

The ability to choose different pathways through a Circuit, depending on the value present at a Particular Point in the circuit at a Particular time, is of fundamental Importance In designing Programmable machines.