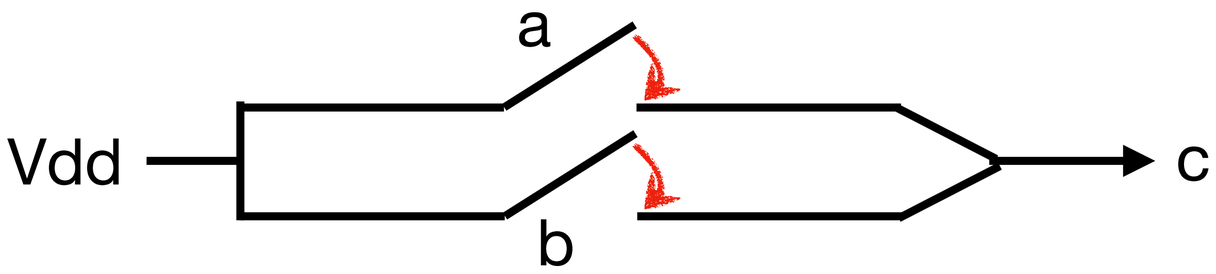
Adder Design

AND and OR Gates

AND Gate

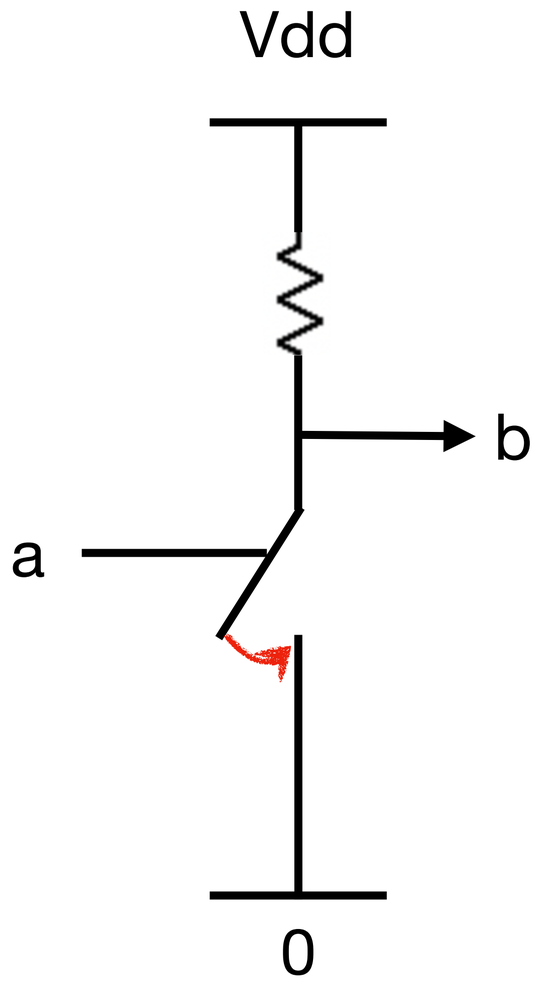
a	b	c
0	0	0
0	1	0
1	0	0
1	1	1



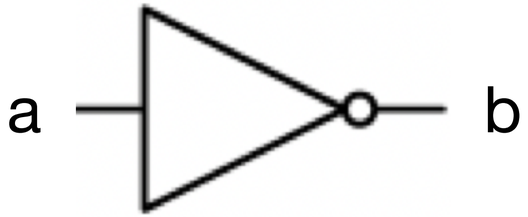
OR Gate

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

Inverter - NOT Gate

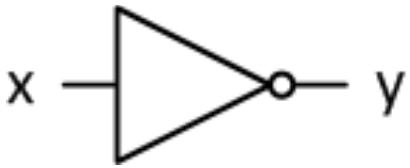


NOT Gate

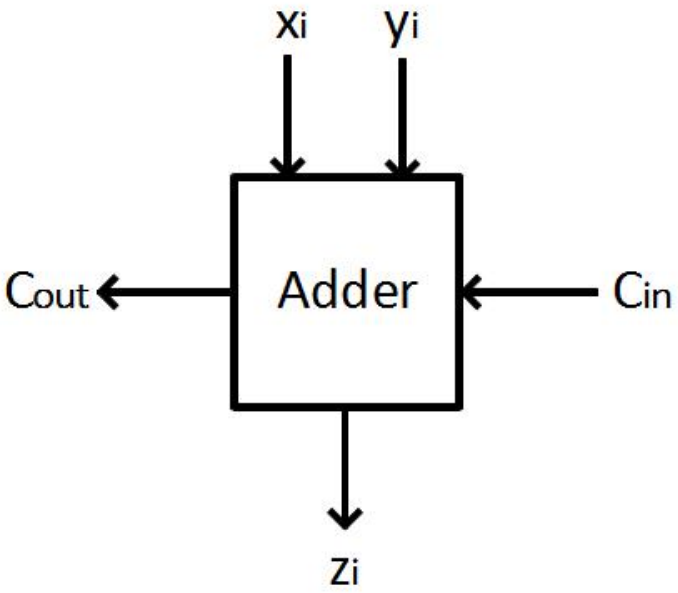


a	b
0	1
1	0

Logic Gates

<table>
<tr><th>Type</th><th>Representation</th><th>Truth Table</th></tr>
<tr><td>AND</td><td> 
</td><td>
<table>
<tr><th>x</th><th>y</th><th>z</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>OR</td><td> 
</td><td>
<table>
<tr><th>x</th><th>y</th><th>z</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>NOT</td><td> 
</td><td>
<table>
<tr><th>x</th><th>y</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

A Full Adder

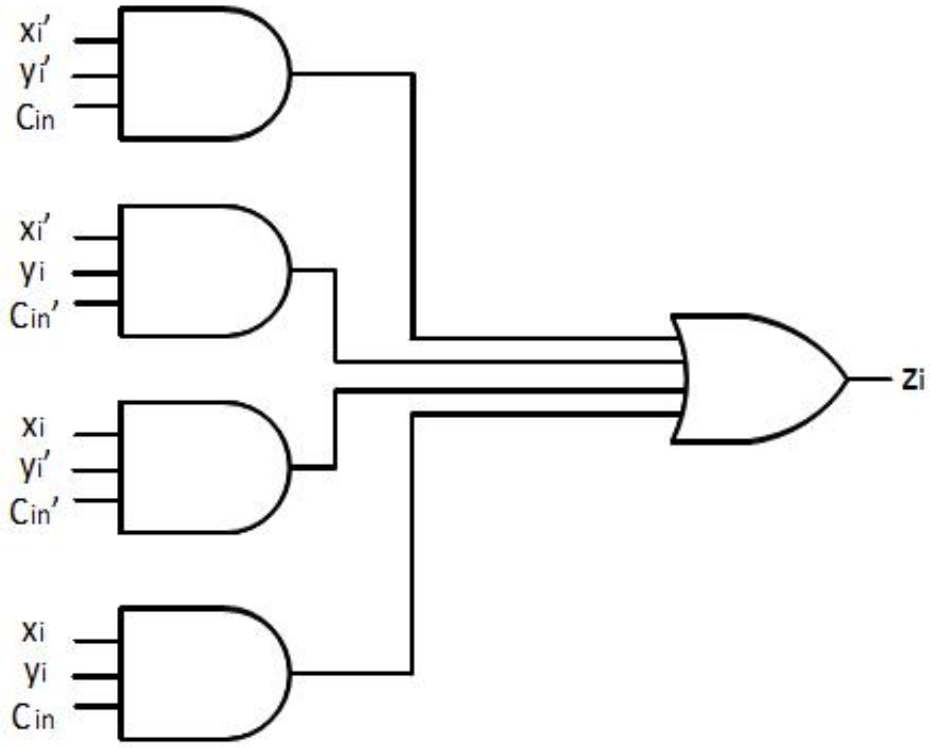


Inputs			Outputs	
x_i	y_i	C_{in}	C_{out}	z_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Implementation 1

- $z_i = x_i' y_i' c_{in} + x_i' y_i c_{in}' + x_i y_i' c_{in}' + x_i y_i c_{in}$
- Implementation:

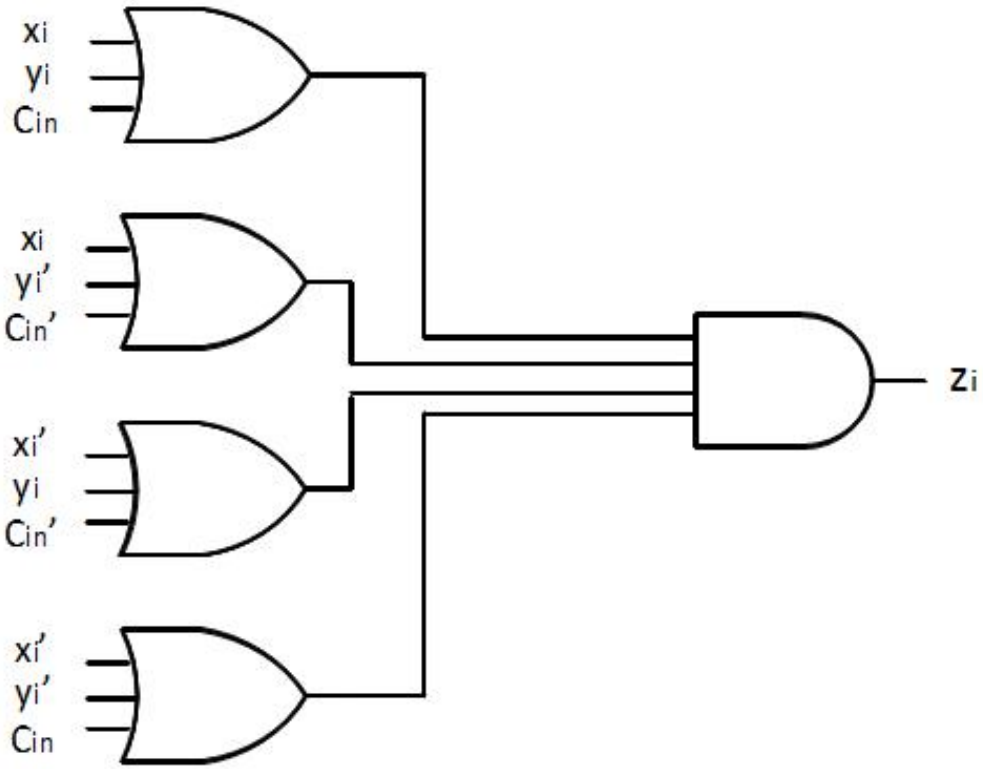
Inputs			Outputs	
x_i	y_i	C_{in}	C_{out}	z_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Implementation 2

- $z_i = (x_i+y_i+c_{in})(x_i+y_i'+c_{in}')(x_i'+y_i+c_{in}')(x_i'+y_i'+c_{in})$
- Implementation:

Inputs			Outputs	
x_i	y_i	C_{in}	C_{out}	z_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



An Example

$$\begin{array}{r} 1001000 \\ +1100011 \\ \hline 10101011 \end{array}$$

Implementation: