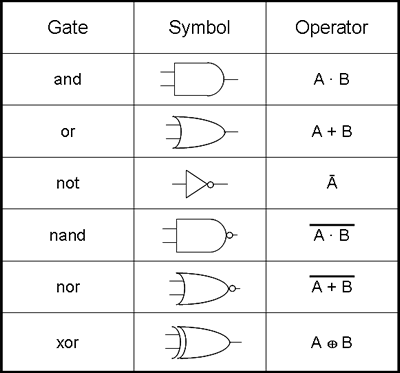
**LOGIC CIRCUITS**

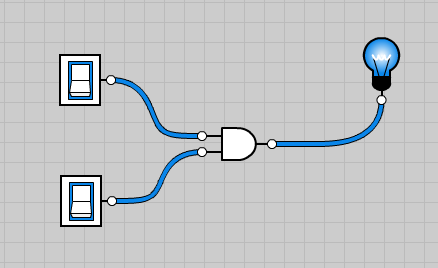
1. Logon to the Logic Gate Simulator Website   
   <http://www.cs.kent.edu/~volkert/F10-10051/notes/logsim.html>

[](http://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&docid=sX-y7kzWAttMnM&tbnid=wlmHz1qAnvSA4M:&ved=0CAUQjRw&url=http://www.ib-computing.com/html/program/topic_4/boolean.html&ei=yQngUcLUEtDu0gXzs4HwBQ&bvm=bv.48705608,d.d2k&psig=AFQjCNHQtkHyah6d0qsifwm7OJCVYHlFgg&ust=1373723451407374)

**Truth table – indicates all the possible outcomes you can get from the circuit**

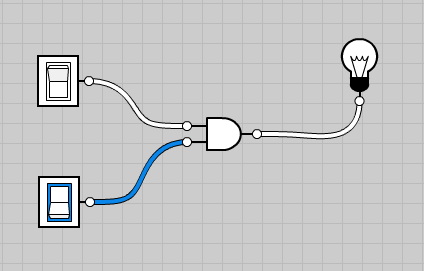
**AND gate - X AND Y 🡪 A.B**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **X and Y** |
| **O** | **O** | **O** |
| **O** | **1** | **O** |
| **1** | **O** | **O** |
| **1** | **1** | **1** |



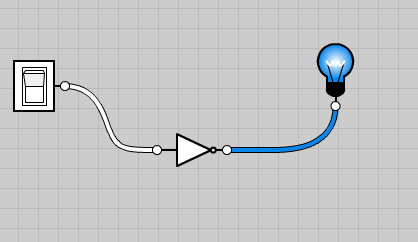
**OR gate – X OR Y 🡪 X+Y**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **Xor Y** |
| **O** | **O** | **O** |
| **O** | **1** | **1** |
| **1** | **O** | **1** |



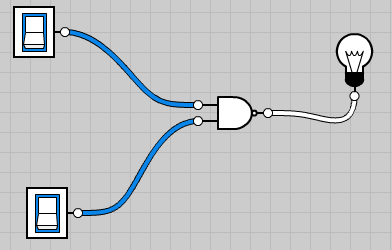
**NOT gate – turns the opposite 🡪 X**

|  |  |
| --- | --- |
| **X** | **NOT X** |
| **O** | **1** |
| **1** | **0** |



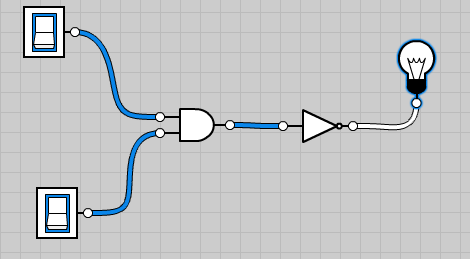
**NAND gate – Not X AND Y 🡪 X.Y**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **NOT(X and Y)** |
| **O** | **O** | **1** |
| **O** | **1** | **1** |
| **1** | **O** | **1** |
| **1** | **1** | **O** |



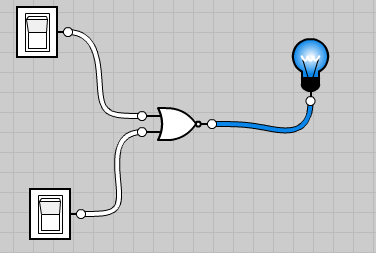
**NOT(X and Y)**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **X and Y** | **Output** |
| **O** | **O** | **O** | **1** |
| **O** | **1** | **O** | **1** |
| **1** | **O** | **O** | **1** |
| **1** | **1** | **1** | **O** |



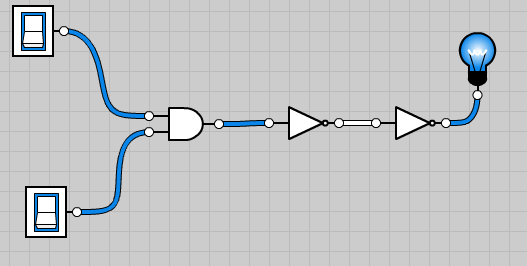
**NOR gate – Not X OR Y 🡪 X+Y**

|  |  |  |
| --- | --- | --- |
| **X** | **Y** | **NOR X or Y** |
| **O** | **O** | **1** |
| **O** | **1** | **O** |
| **1** | **O** | **O** |
| **1** | **1** | **O** |



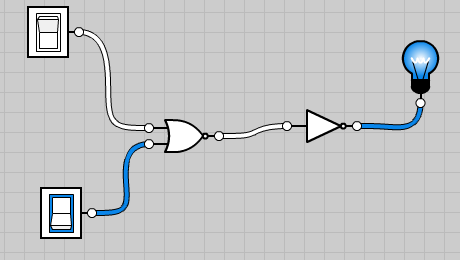
**NOT (NOT (X and Y))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **Y** | **X and Y** | **(NOT (X and Y)** | **NOT(NOT (X and Y))** |
| **O** | **O** | **O** | **1** | **O** |
| **O** | **1** | **O** | **1** | **O** |
| **1** | **O** | **O** | **1** | **O** |
| **1** | **1** | **1** | **O** | **1** |



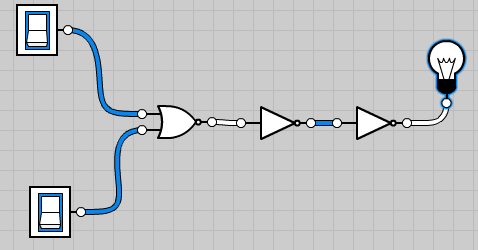
**NOT (X OR Y)**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **X OR Y** | **NOT(X OR Y)** |
| **O** | **O** | **O** | **1** |
| **O** | **1** | **1** | **O** |
| **1** | **O** | **1** | **O** |
| **1** | **1** | **1** | **O** |



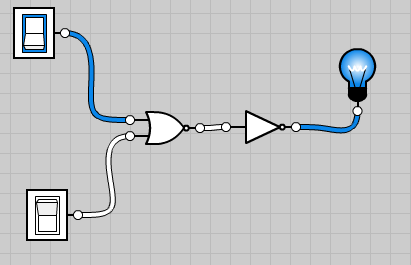
**NOT (NOT(X OR Y)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **Y** | **X OR Y** | **(NOT (X or Y)** | **NOT(NOT (X and Y))** |
| **O** | **O** | **O** | **1** | **O** |
| **O** | **1** | **1** | **O** | **1** |
| **1** | **O** | **1** | **O** | **1** |
| **1** | **1** | **1** | **O** | **1** |



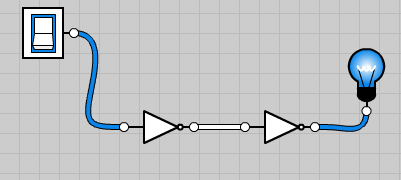
**NOT ( X NOR Y)**

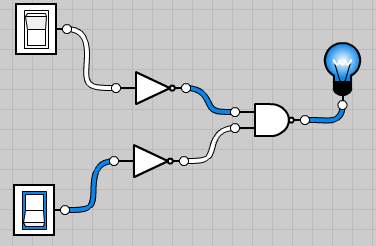
|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **X NOR Y** | **NOT(X OR Y)** |
| **O** | **O** | **1** | **O** |
| **O** | **1** | **O** | **1** |
| **1** | **O** | **O** | **1** |
| **1** | **1** | **O** | **1** |



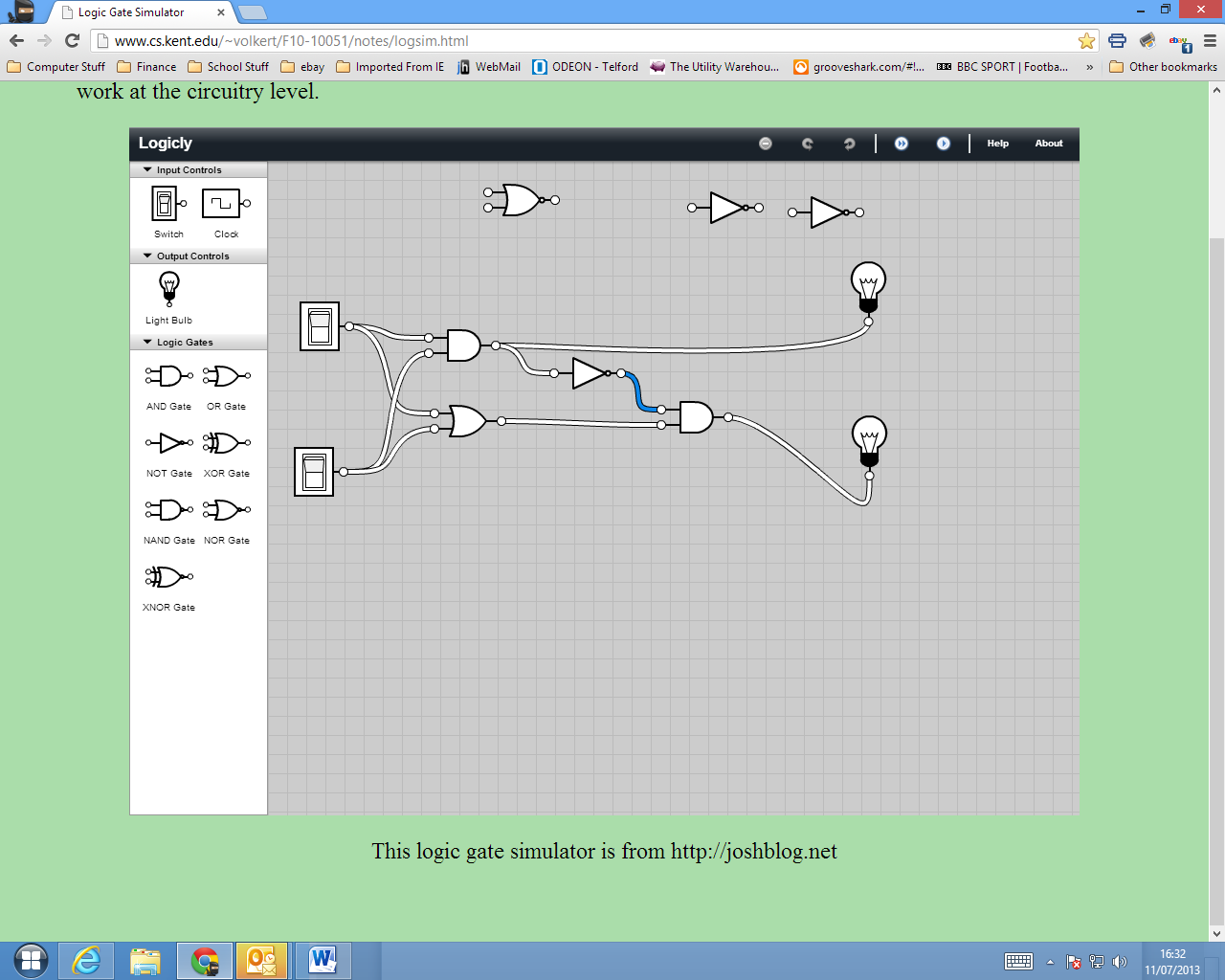
**NOT (NOT X)**

|  |  |  |
| --- | --- | --- |
| **X** | **NOT X** | **NOT(NOT X)** |
| **O** | **1** | **O** |
| **1** | **O** | **1** |



**NOT X NAND NOT Y**

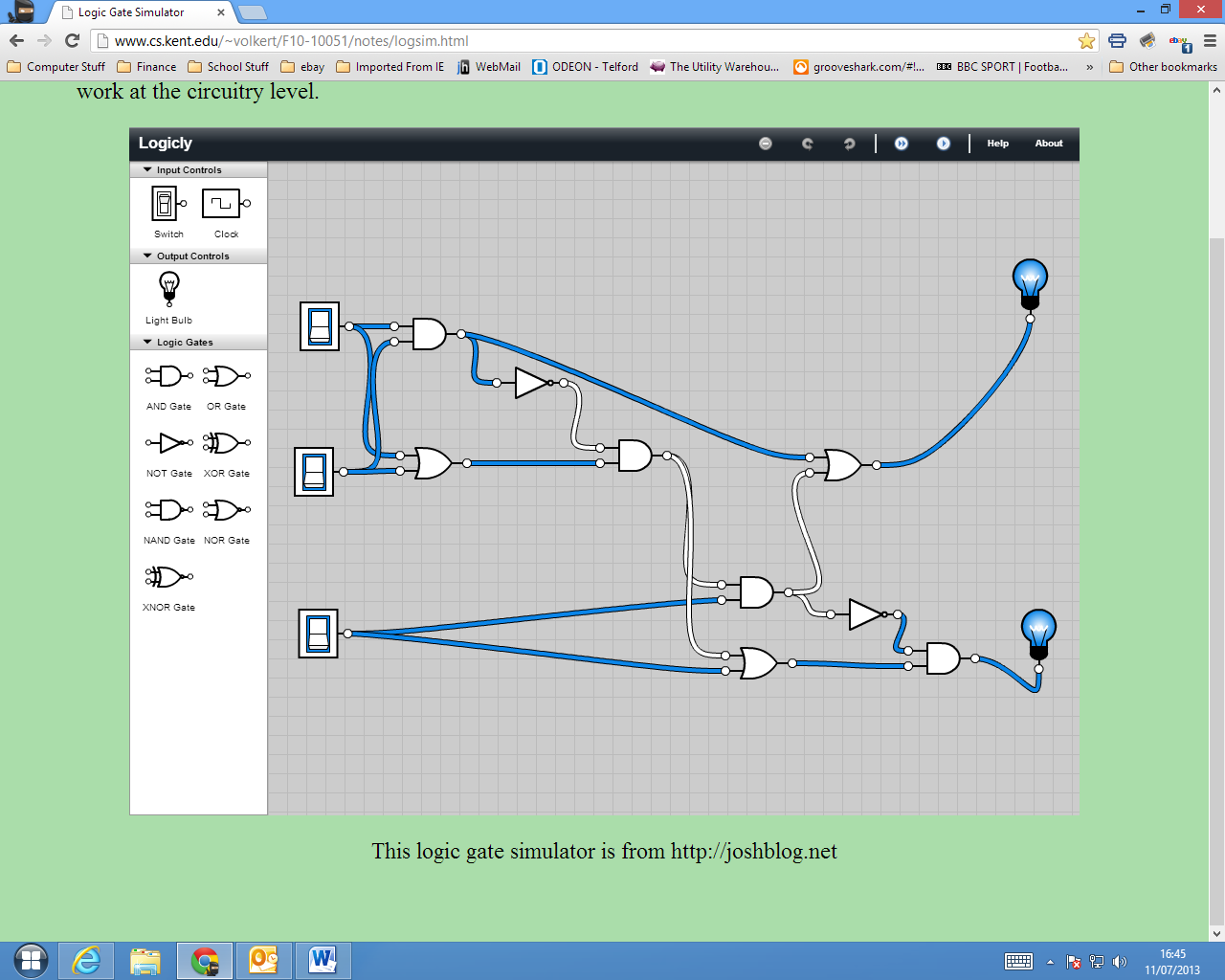
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **Y** | **NOT A** | **NOT B** | **NAND A B** |
| **O** | **O** | **1** | **1** | **O** |
| **O** | **1** | **1** | **O** |  |
| **1** | **O** | **O** | **1** |  |
| **1** | **1** | **O** | **O** | **1** |

**This is a useful circuit. Can you think why?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | Y | X AND Y | NOT (X AND Y) | X OR Y | (NOT (X AND Y) AND (X OR Y) |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |

**This is a binary arithmetic processor. It is a half adder**

**With more switches this would be an accumulator**

* 1. **This is an extremely useful circuit!**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | A | B |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

This is how an accumulator works. It can add any Binary numbers the first two switches deal with the two numbers and the third switch deals with the carried numbers in the case of 1 + 1+ 1

