Logic Gate Investigation

Using this [online logic-gate simulator](http://www.edumedia.rmit.edu.au/emg/gallery/Project/LogicGates/logic_builder.htm), build the following logic-gate circuits and determine the truth table for each one. Then summarize the behaviour of each circuit.

We’ll do the first few together as a class.

**The NOT gate (also called an *inverter*)**

|  |  |  |
| --- | --- | --- |
|  | **A** | **output** |
| **0** |  |
| **1** |  |
| **Description:** | |

**The AND gate**

|  |  |  |  |
| --- | --- | --- | --- |
| ***A AND B* is often written *A* x *B*. Can you see why?** | **A** | **B** | **output** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |
| **Description:** | | |

**Combinations of AND with NOT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **C**  **D** | **A** | **B** | **C** | **D** | **output** |
| **0** | **0** |  |  |  |
| **0** | **1** |  |  |  |
| **1** | **0** |  |  |  |
| **1** | **1** |  |  |  |
| **Why can the light bulb not be turned on?** | | | | |

**The OR gate**

|  |  |  |  |
| --- | --- | --- | --- |
| ***A OR B* is often written *A* + *B*. Can you see why?** | **A** | **B** | **output** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |
| **Description:** | | |

**The NOR gate**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **output** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |
| **Description:** | | |

**The NAND gate**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **output** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |
| **Description:** | | |

**The XOR gate**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **output** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |
| **Description:** | | |

**A circuit for turning a street lamp on**

**A set of street lamps in the city is on an automatic circuit. The circuit is designed to turn the lamps on if:**

* **EITHER a manual switch is flipped by a technician**
* **OR**
  + **It’s late enough in the day (as determined by a timer) AND It’s dark enough outside (as determined by a light-sensor)**

**Part of the truth table for the circuit has been filled in for you. Fill in the rest of it.**

**You may build the circuit in the simulator to help you if you wish, but you should be able to fill in the table just by thinking it through.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **E**  **D** | **Inputs** | | | **Outputs** | | |
| **A** | **B** | **C** | **D** | **E** | **Street lights** |
| **0** | **0** | **0** | **1** |  |  |
| **0** | **0** | **1** | **0** |  |  |
| **0** | **1** | **0** | **1** |  |  |
| **0** | **1** | **1** | **0** |  |  |
| **1** |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | | | | | |

**Using logic gates to add two binary bits**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **0** | **0** | **1** | **1**  **1** |  |
| **B** | **+ 0** | **+ 1** | **+ 0** | **+ 1** |  |
|  | **0** | **1** | **1** | **10** |  |
| **Write** | **0** | **1** | **1** | **0** | **Is there a logic-gate that fits this pattern?** |
| **Carry** | **0** | **0** | **0** | **1** | **Is there a logic-gate that fits this pattern?** |

**Use your answers in the table above to design a simple circuit that has this truth table.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | **Outputs** | |
| **A** | **B** | **C (write)** | **D**  **(carry)** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** |

**This circuit is called a *half-adder circuit*.**

**Build your circuit in the online simulator. Test it on all values of A and B to make sure that the written bits and carried bits are correct. Then copy and paste a screen-capture of your diagram in the box below.**

**Please crop out everything except the circuit itself.**

|  |
| --- |
|  |

**But what if there’s a carry from the previous column?**

**That’s where the full-adder comes in!**

**A *1-bit* *full-adder circuit* adds three binary bits:**

* **Two bits from the column we’re now adding**
* **The carry-bit that came from the previous column**

**Example**

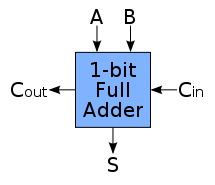
**Suppose we’re adding the binary numbers 11 and 01**

**1 1 🡨 carries**

**11**

**+ 01**

**100**

**When we’re adding up the 2nd column, we need a circuit that will add not only the 1 and the 0,   
but also the 1 that was carried over from the 1st column.**

**In the diagram to the right,**

**A represents the 1 from the 2nd column**

**B represents the 0 from the 2nd column**

**Cin represents the 1 carried over from the 1st column,**

**S represents the 0 that’s written below the 2nd column**

**Cout represents the 1 carried over to the 3rd column**

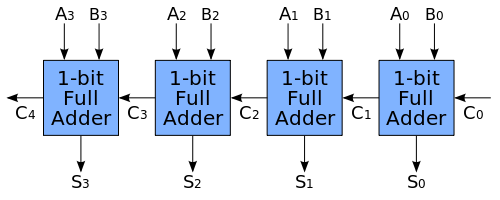
**We’ll see in class how to build a 1-bit full adder circuit using logic gates.**

**A circuit for adding multi-digit numbers**

**If A is a binary number with bits A3A2A1A0,**

**and B is a binary number with bits B3B2B1B0**

**then, A + B can be computed using the circuit below.**



**Half  
Adder**

**The final answer is the string of digits C4S3S2S1S0**

**The CPUs in your laptops and phones contain adder circuits like this one, except with dozens of bits.**