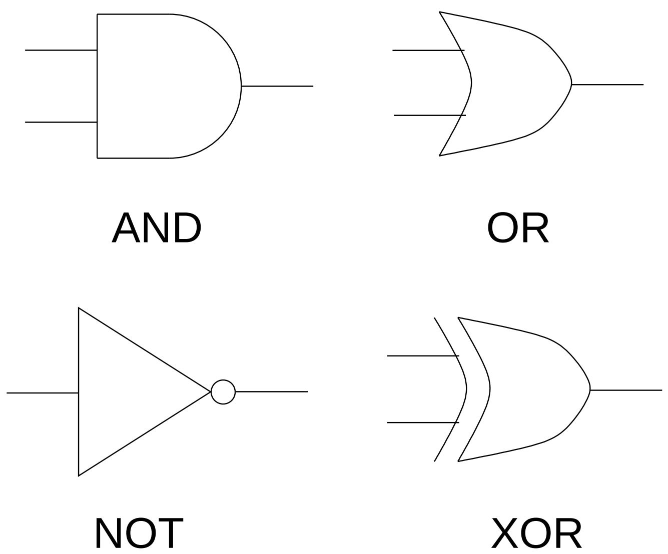
15-251: GTI

Guide to Drawing Circuits **February 18, 2003**

The purpose of this document is to introduce you to the mechanics of drawing circuit diagrams. This should help you make your diagrams neat and clear, and thus understandable. This is necessary for you to think about and verify your design, and for us to grade it.

1 Gates

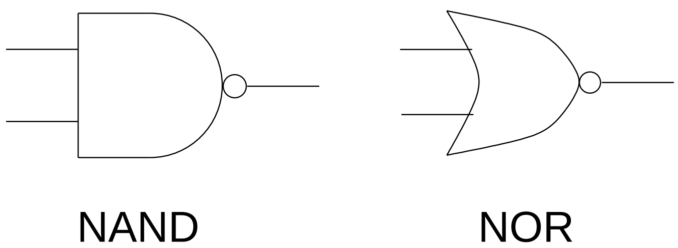
Basic gates are drawn like this:



The two wires going into the left side of each gate are the inputs, the wire coming out of the right is the output. Gates and simple circuits are conventionally drawn facing to the right like this, but in more complex circuit diagrams they can be drawn facing in any direction; the shape of the circuit will indicate the direction of signal flow along the wire. (The "signal" is the boolean value on the wire—true or false, 1 or 0.)

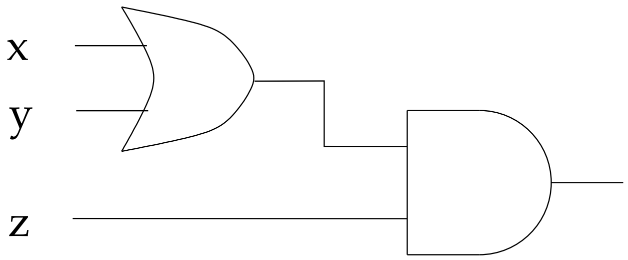
Note that the only major difference between AND and OR is the shape of the back. Thus be sure to always make the backs of your ORs concave and the backs of your ANDs flat.

That little circle on the nose of the NOT gate is called a "bubble", and in general it means that the signal on the wire is negated. Thus by adding a bubble to the front of the AND and OR gates, we obtain their opposites:

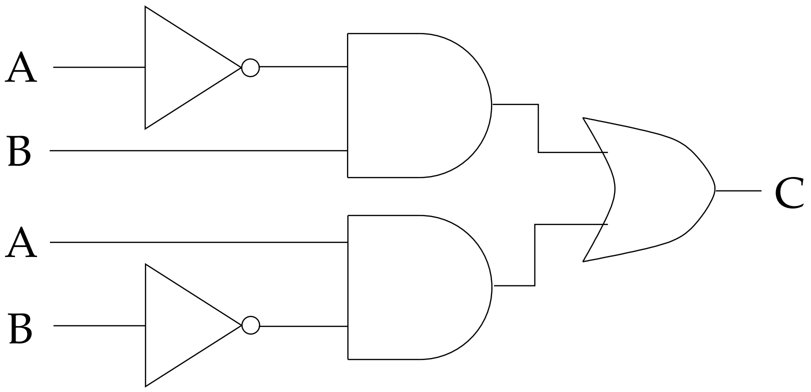


Don't use just a bubble on a wire to negate a value, though; it's much clearer to use a NOT gate.

Gates are combined by connecting the output of one gate into the input of another:



This (unlabelled) output of this circuit is (x OR y) AND z. Here's a larger circuit:

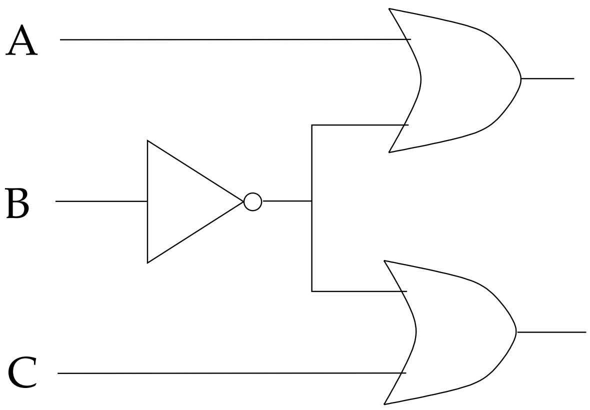


The output, C, is ((NOT A) AND B) OR (A AND (NOT B)). Note how the inputs A and B were used multiple times.

2 Signal flow

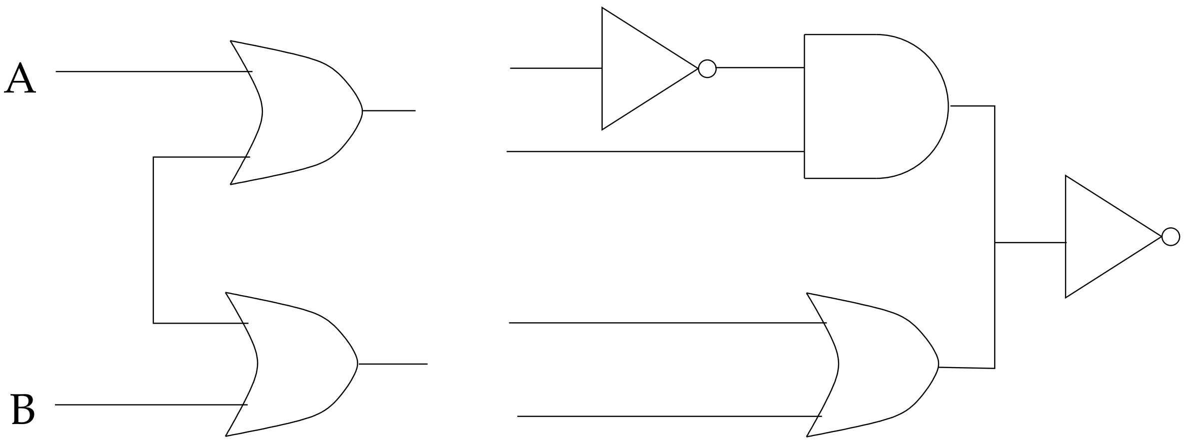
The signal in a circuit flows along the wires, and through each gate from its input to its output. This means that a gate is constantly looking at its inputs, computing the appropriate value, and putting that value on its output. We say that the gate *drives* its output.

If you wish to use the output of a gate as an input multiple times, you may split the wire:



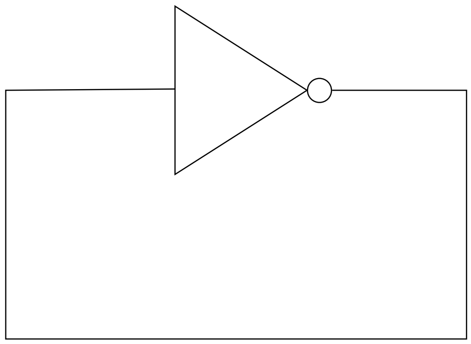
This circuit computes two values: (A OR (NOT B)), and ((NOT B) OR C).

Be careful, though: Each wire must be driven by exactly one gate. You can't have an input that comes from nowhere, or a wire which is connected to the output of two gates.[1] Thus, the following circuits are illegal:

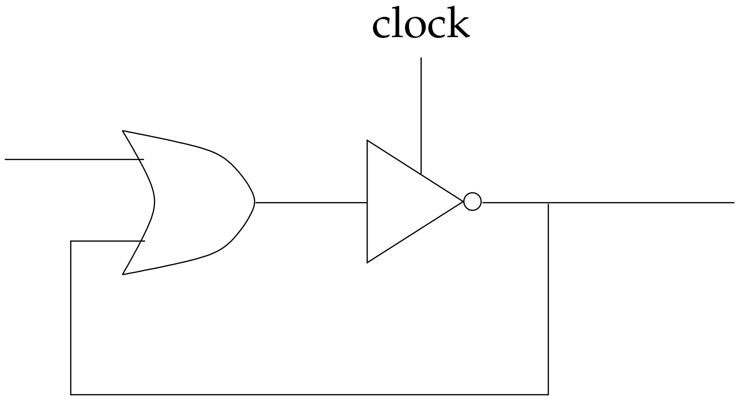


no input! combined output!

You must also be careful about loops in your circuits. You *can* feed the output of a circuit back into a prior spot in the circuit, but it must be done properly. Specifically, there must be a clocked gate in any loop. The clock will stop the signal from zipping around the loop as fast as it can. So the following is bad:



(What would happen in such a circuit?) But this is OK:

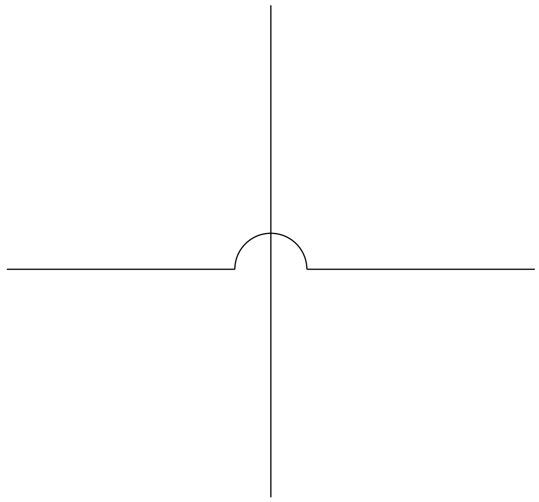


[1]If you do these things in real physical hardware, nasty things happen, like short circuits.

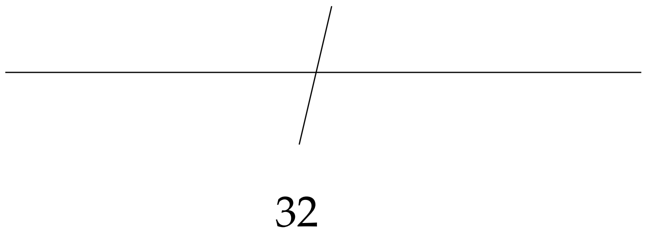
3 Wires

As you've probably noticed, wires are conventially drawn as always travelling either horizontally or vertically. This just helps to make the diagrams neater.

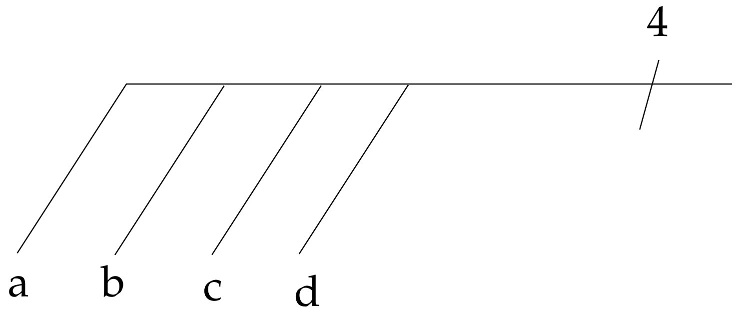
When your circuits get complex and involve many gates, you will inevitably need to have two unconnected wires cross on your diagram. To avoid confusion, draw one of the wires "jumping" over the other:

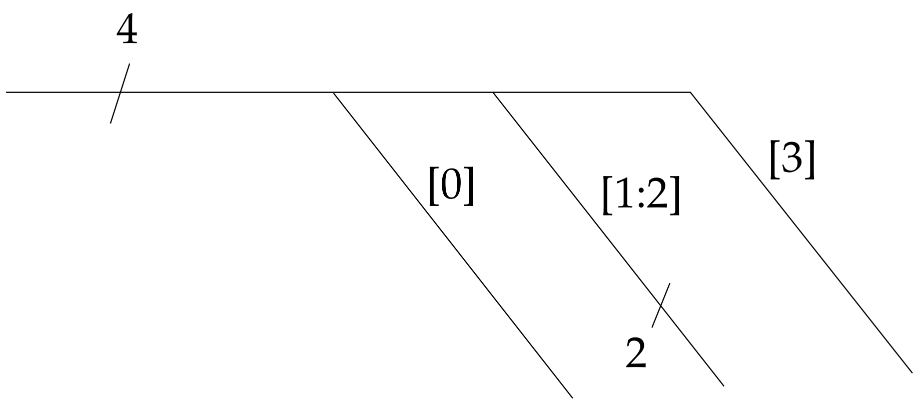


Also, while designing your processor, you will want to deal with inputs to circuits which are multiple bits, which means multiple wires. A 32-bit number in a computer is carried on 32 separate wires. This is drawn as a "bundle":



Joining and splitting wires is denoted thus:



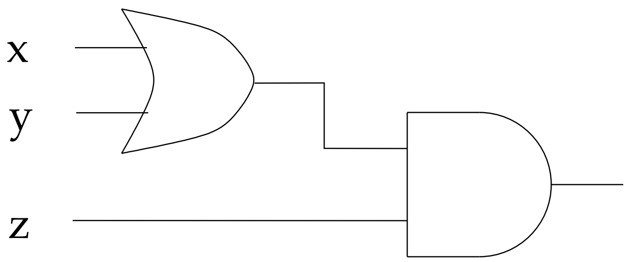


In the second figure, a bundle of 4 bits is split into two single bits (bits 0 and 3 of the bundle), and a bundle of 2 (consisting of bits 1 and 2).

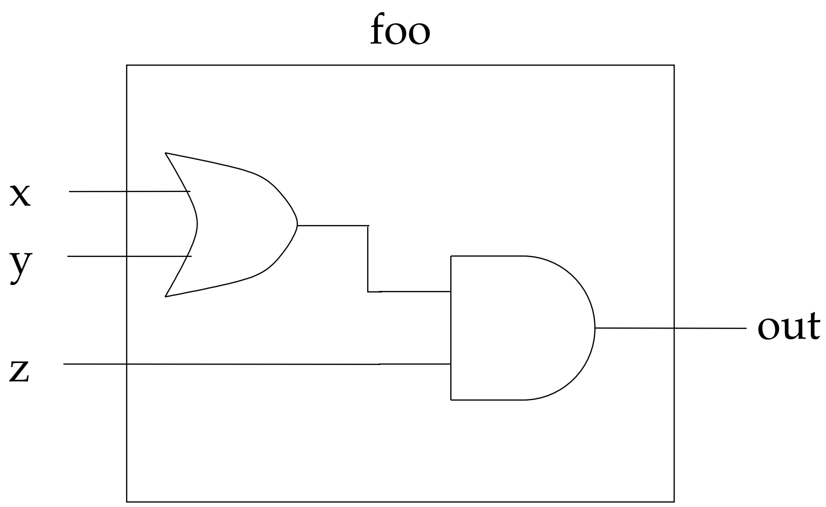
4 Modules

A key technique in building the processor (or any complex engineering design) is *modularity*. Once you have designed a circuit with a specific purpose and function, you may make that circuit into a module with certain inputs and outputs. From that point on, you need only draw the module as a labelled box, rather than drawing the entire circuit again each time. This will save you effort and paper, but more importantly it will clear away *mental* clutter and make the diagram easier to comprehend.

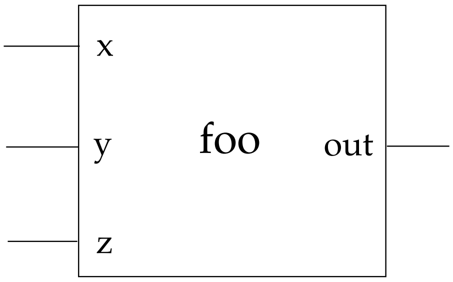
Suppose you wanted to make a module out of the first circuit we saw above:



You would draw a box around it and give it a name (as well as naming all its inputs and outputs):



From then on you would draw it as follows:



You would put it in circuits just like a basic gate.

5 Software options

You can draw circuits in Microsoft Word or use an X Windows drawing program called `xfig`. Check out the course webpage at `http://www.discretemath.com` for information on drawing circuits in these programs. There are some software packages designed specifically for drawing circuit diagrams (Visio for Windows is great), but we don't know of any available in the cluster and cannot offer documentation for those packages.