

Discrete component logic

Written by Hans Summers

Monday, 31 August 2009 21:33 - Last Updated Saturday, 09 January 2010 00:15

I often use discrete component implementations of logic gates in my projects, often consisting of AND and OR gates made from diodes and resistors. I "invented" or "discovered" how to make these gates when I was about 14 and have been happily using them ever since.

The reason I use these gates is often to save space: if all I need is one AND gate, for example to reset a counter, then I can easily implement it with 2 diodes and a resistor. It saves using a whole additional IC such as a 74HC08 containing four AND gates, of which I only need one. At other times, it can be convenient to be able to build a logic gate with as many inputs as I like. For example, a 7 input OR-gate is easily implemented just by adding more diodes to the basic 2-input gate. I might not have the IC's I would need in my junkbox, so I'd have to order them and wait for the delivery to arrive. But I'll always have resistors and diodes, so this is sometimes another reason for using logic gates made from discrete components.

There are disadvantages to these very simple logic gates too. The most noticeable is that the output current ability of the gates is low. It is determined by the resistor used, and by the current handling ability of the signals driving the gate inputs. If the discrete logic gate output is driving a high impedance IC input such as are found on the modern 74HC-series IC's, there is no problem. The 0.7V voltage drop across the diodes means that you cannot cascade the simple discrete component logic gates, because you may easily violate the input voltage logic level specifications of the IC's following the gate: the voltage will fall below what is recognised as a "1" for example. Finally, the discrete component implementations of logic gates might be slower than their "proper" IC equivalents. I do not currently know how much of a problem this is likely to be.

Now I'll try to explain the various types of gates which I commonly use.

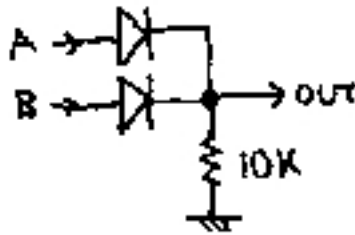
OR gate

Discrete component logic

Written by Hans Summers

Monday, 31 August 2009 21:33 - Last Updated Saturday, 09 January 2010 00:15

OR

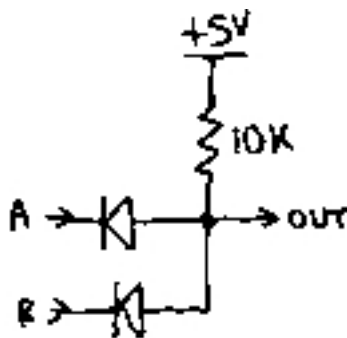


A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

~~AND gate~~

AND gate

AND



A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

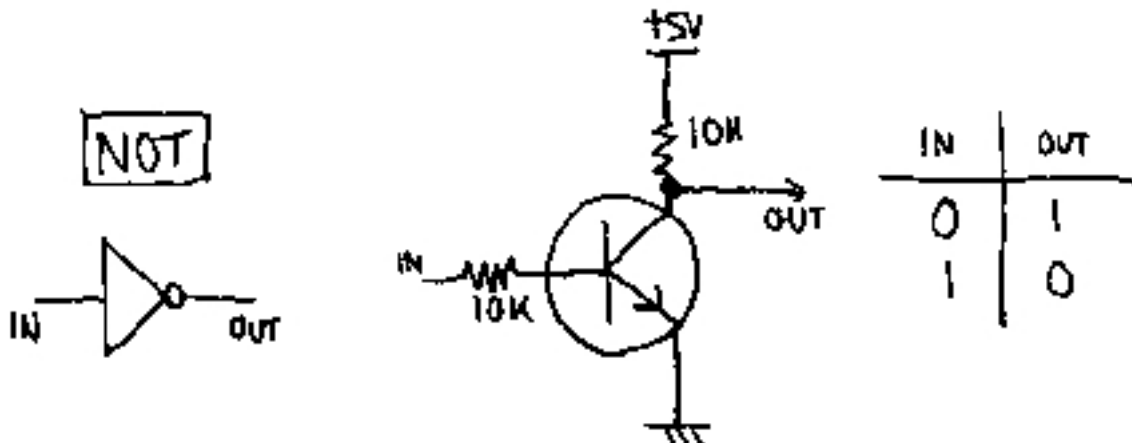
Now look at my AND gate. It's similar to the OR gate except that the diodes point in the other direction, and the resistor goes to +5V not ground. The output of an AND gate is "1" only if BOTH the inputs are "1". In my diode-resistor implementation, if either input is "low" voltage (logic "0") then the diode will conduct and the output is effectively shorted to ground. If both of the input voltages are "high" (logic "1") then neither of the diodes will conduct, so the output is not shorted to ground: it remains at +5V (logic "1") via the 10K resistor. This gives the desired result. Note that again, due to the silicon junction voltage, the actual "low" output voltage is 0.7V higher than the "low" input voltage.

NOT gate (inverter)

Discrete component logic

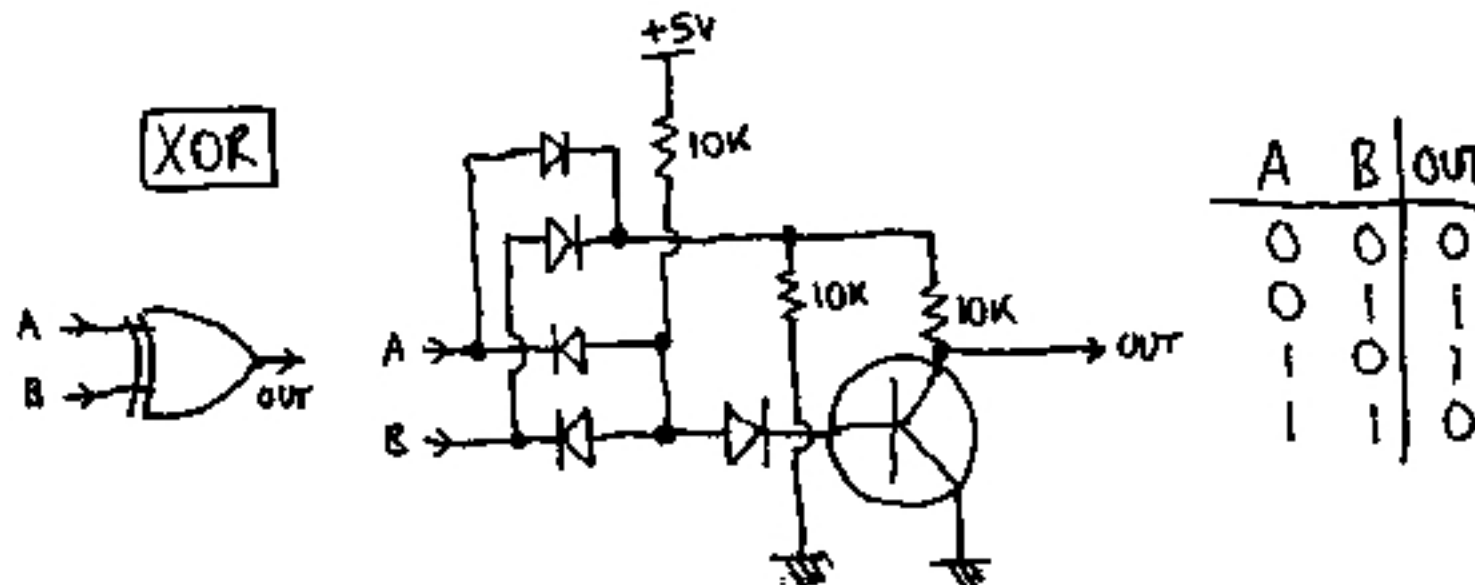
Written by Hans Summers

Monday, 31 August 2009 21:33 - Last Updated Saturday, 09 January 2010 00:15



NAND and NOR gates

These are easy: just use AND or OR followed by a NOT (inverter). In the case of the NAND gate, the transistor's base resistor can be omitted since the maximum current is already limited by the 10K resistor in the diode-resistor AND gate. The base resistor is still needed in a NOR gate.



XOR gate

This one is a little more complicated. If we start off imagining an OR gate, that works for the first three of the possible four states in the truth table. But not the final one, where both inputs are "1". So we need a way of forcing the output to zero. I accomplished this using a transistor as a switch, with its base driven by an AND gate on the two inputs. So that when both inputs are "1", the AND gate will activate the transistor, which will force the output to zero. So the XOR gate can be considered as an OR gate, plus an AND gate, plus a switch to zero the output. The additional diode in the base lead of the transistor is to create an additional 0.7V drop, without it

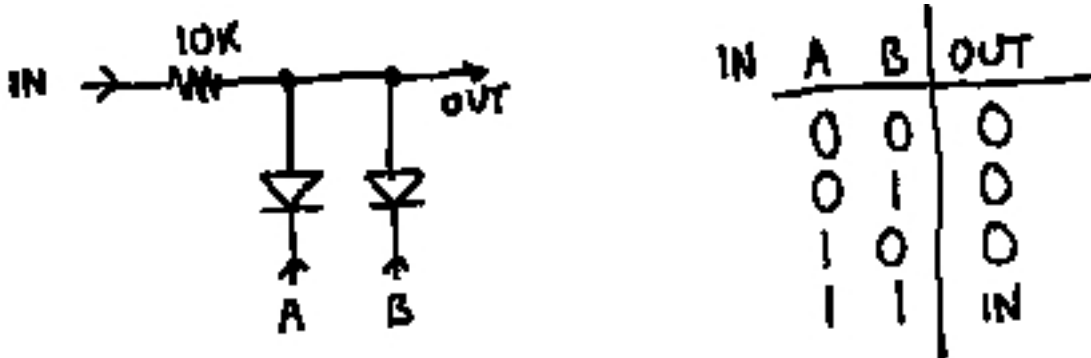
Discrete component logic

Written by Hans Summers

Monday, 31 August 2009 21:33 - Last Updated Saturday, 09 January 2010 00:15

the 0.7V of the AND gate output (even when one of the inputs is zero) would be enough to put the transistor into partial conduction.

Signal gate



I like this one for gating incoming pulse trains. I use this circuit sometimes for frequency counter inputs for example. In the example to the right, the pulse is only allowed from IN to OUT when both the A and B inputs are "1". It's really just another kind of AND gate I suppose, effectively having three inputs. But saves one diode compared to an AND-gate implementation as above. (Well, even a 2-input AND gate or OR gate can be reduced to a single resistor and a single diode, in that case - but somehow I prefer the more balanced approach with two diodes and a resistor). In the diagram to the right, the diodes could be reversed and the output would then be forced to "1" if either of A or B were a "1". This could be useful to drive a negative edge triggered counter clock input, for example.

Other notes

1. Gates with more than two inputs: Easy! Just add more diodes, one for each input!
2. The 10K resistor: I have shown 10K for the purposes of illustration only. In reality resistance value needs to be chosen depending on the input resistance of the following circuit that needs to be driven by the output. It needs to be relatively small in comparison. If the gate is driving a single 4000-series CMOS or 74HC-series logic IC input, then the resistor could easily be made upwards of 100K or even 1M, depending on the application.

XNOR gate

Many thanks to Harry Brunt, who took the time to write to me, with this interesting idea for an XNOR gate (like an XOR gate followed by an inverter). Harry writes:

"I've just read your interesting article about making logic gates from diodes. I wanted to built

Discrete component logic

Written by Hans Summers

Monday, 31 August 2009 21:33 - Last Updated Saturday, 09 January 2010 00:15

a discrete XNOR gate using a minimum number of components (to avoid having to buy a quad XNOR IC for the sake of one gate). After messing around a bit with your XOR gate design, I found that by reversing the polarity of all diodes, substituting a PNP for the NPN transistor and swapping the earth for +V and vice versa, I obtained an XNOR gate - in fact the schematic is just a mirror image of the XOR gate. I've only tried it using SPICE so far, but I can't see why it wouldn't work in practice. I think making an XNOR gate using a single transistor is pretty neat!"

The truth table for XNOR is:

A	B	Output	-----	0	0
---	---	--------	-------	---	---