FREQUENTLY ASKED QUESTIONS

April 11, 2017

Content Questions

How do you know the most "efficient" way to build a logic gate, and does it matter?

Usually you want to perform an operation with the fewest gates possible, because it's usually cheaper and simpler (and usually simpler means more robust). However a design might depend on what you are trying to optimize... this might be speed or power consumption or whatever. You might use more gates to avoid logic races, or allow use of some particular available chip, or fit things in physical space, or whatever.

What's the difference between OR and exclusive OR?

A logical OR, $Q = A + B$, gives Q true when either one, or both, of A or B are true; so Q is only false when both A and B are false.

In contrast, exclusive OR, $Q = A \oplus B$, gives Q true when *exactly one*, but not both, of A and B are true. This operation is a bit more like our everyday understanding of the word "or" (as in, do you want carrots or celery?)

Write the truth tables to see the difference!

Are XOR gates produced separately or do you always have to build it out of other gates?

Actually you can get dedicated XOR gates in chip form, or you can make them out of transistors.

What causes time delay in logic gates?

It can take some time for the on-vs-off signal to propagate through a multiple transistor circuits inside a logic gate, because it takes some time for transistors to "turn on" or off when the "control" voltage or current at the gate or base changes. Stray capacitance matters, since it can require some charging up or charging down time to reach an appropriate voltage to turn on or off a transistor.

How does a clock get rid of glitches?

A clock signal can help to avoid glitches by requiring that all logic transitions in the circuit happen *synchronously*, i.e., at the times of a clock transitions. Then you don't get little overlaps in logic signals when they arrive a bit early or late due to variations in gate delays. We'll see more examples of clock use soon.

How do you make a clock signal?

There are a number of ways of making periodic signals ("oscillators"). For clock signals used in digital circuits, the periodic signal is made into a square waves. The simplest way to make a periodic signal is to use an AC generator. You can use an LC circuit as well. A modern way is to use a piezoelectric crystal which vibrates at a particular resonant frequency– these can give very precise frequency signals. Here's more info on oscillators: https://en.wikipedia.org/wiki/Electronic_oscillator.

What is the fastest switch time? What limits it?

The fastest digital speeds (for a clock in a synchronous circuit) are something like THz (10^{12} Hz), which corresponds to ∼picosecond transitions. (More normal digital electronics clock speeds are GHz, so have ∼nanosecond transitions.)

The switch time for a particular gate is limited by stray capacitance (and possibly other non-ideal performance of the transistor components). For a synchronous circuit with a continuously-maintained rate of transitions, I think the main limitation is heat dissipation. The higher the clock speed and rate of transitions in the circuit, the more power is dissipated and the hotter the circuit runs (and solid state components tend to be quite sensitive to temperature).

For the TTL NOR gate, won't the 5 V drop across the resistor always be the same, regardless of $A + B$, making Q always the same?

No— if the transistors are both off, there will be no current drawn from the supply. Then there's no current going through the resistor at the transistors' collectors, so the output will be at $V_{cc} = 5\ V$, i.e., the output will be "true".

If either or both transistors are on, current through the resistor causes a drop across it, and output is low ("false").

What would happen if different logic families are combined? e.g. TTL and NIM signals.

Combining different logic signals doesn't usually work very well because the meanings of particular voltages (i.e., whether some voltage level means 1 or 0) differs. If you want to combine signals from chips belonging to different families, usually you have to build some kind of converter (which will introduce a delay, typically).

Why with NIM or ECL is the negative voltage level chosen to be true?

This is because they operate with negative voltage supplies, which is done for several reasons, mostly related to suppression of noise: see this link.

How does an ASIC work in terms of logic circuits?

There are actually many varieties of and methods for making these. They are often made of *gate arrays*, which are made up of blocks that perform different logic functions. One designs in software some circuit that performs a desired function, and then the circuit is implemented by selection of gate configuration (by making, or destroying connections) at manufacture time. Here's a Wikipedia article.