Lecture 5: Physical Realisation of Logic Gates

The purpose of this lecture is to present a set of models which show how logic gates are made, and how they behave in practice. We will set the scene by discussing what is meant by a physical model. It is important to realise that our understanding of the laws of nature is just an approximation. We can see this by considering any branch of physics. For example, by means of Newton's laws we can explain to a high degree of accuracy the orbits of the planets in the solar system. Until the end of the last century it was widely believed that Newton's laws were correct, since they fitted data exactly to the precision that it could be measured. However, when more accurate means of measuring the planetary motion were found, it was discovered that the orbits deviated from the predictions made from Newton's laws by very small amounts that couldn't be attributed to experimental error. Hence it had to be concluded that Newton's laws were only approximate, and a more complex and accurate theory was required. This was provided by quantum mechanics, but particle physics again demonstrated that this was also only approximate and worse still that it was not possible to make complete measurements. It remains a conjecture as to whether there really are laws of physics and whether we could discover them. However, the approximate theories are still of use since they provide models of the real world that we can use in design. In many instances we are interested in the simplest model that we can find that encapsulates the physical behaviour of the system we are designing. For example, it would clearly be ludicrous to use quantum mechanics if we were designing a car, since Newtonian mechanics will be more accurate than we need. Further, for the bearing parts of a machine, which are machined accurately and lubricated, we would choose a simple mechanical model in which friction did not appear. Thus, we see that we pick a model to suit the purpose and accuracy required by our task, and we will now investigate how this is done for logic design.

The components found in logic gates are of three kinds: resistors, capacitors and transistors. For our simplest models we will ignore the capacitors. The resistors will be modelled by Ohm's law, which states that the electric current flowing in the resistor is related to the voltage difference across the ends by the equation V= IR. (This of course is only an approximate model by is sufficiently accurate for logic circuit design at any level of detail). We will start with a very simple model of the transistor which is shown in diagram 5.1. It has three terminals marked G S and D, and it obeys the following rules:

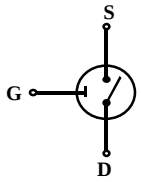

Diagram 5.1
A Simple transistor model

1 There is no connection between G and D or G and S
2 If the voltage between G and D (Vgd) is less than 2 volts the switch is open and there is no connection between D and S.
3 If the voltage Vgd is greater than 2 volts the switch is closed and D is connected directly to S.

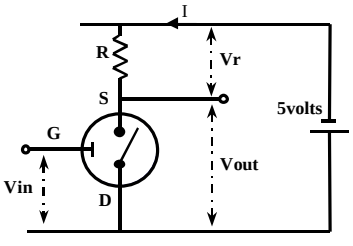

Diagram 5.2: An invertor circuit

Although this is a highly simplified model it contains the sufficient detail to explain how the gates are constructed to compute the correct logic. An invertor is shown in Diagram 5.2, and we shall consider how it works. Suppose that the input voltage (which in this case is the same as our voltage Vgd in the rules above) is set to zero. The switch remains open, and so there is nothing connected to the S terminal. It is therefore not possible for any current to flow through the resistor, and the voltage drop across it is therefore zero by ohm's law. Thus the output voltage is the same as the battery voltage, that is 5v. Now suppose that a voltage of 5 volts was applied to the input (Vin=Vgd=5v). The switch now closes, and there is a voltage of 5V across the resistor. Current flows, and the voltage at the output becomes zero, since it is connected through the transistor to the 0v line. Hence, if we think of 5v as logic state 1 and 0v as logic state 0, we can see that the circuit is an invertor. Diagram 5.3 shows how the invertor is simply adapted to make *nand* and *nor* gates. In the case of the nand gate, it is clear that both transistor switches must be closed before the output voltage is connected to the 0v battery terminal, and therefore becomes 0. For the nor gate either transistor can make the connection. Clearly we can construct *and* and *or* gates by adding an invertor stage after the *nand* and *nor* gates respectively. The *exclusive or* gate can be constructed from four *nands*, though in practice there is a simpler implementation which we will not discuss.

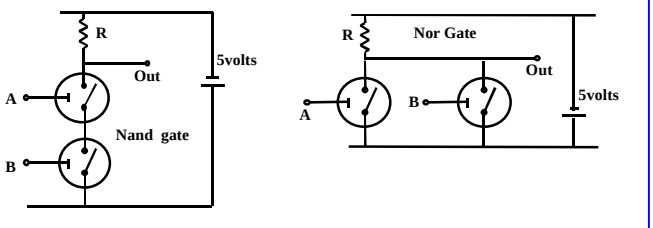

Diagram 5.3
Nand and Nor gate implementations

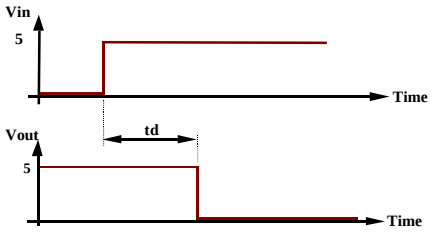

Diagram 5.4: Transistor time delay

Although the transistor model we are using can model all the logic behaviour correctly, we need to make it more accurate for practical design, and the first new feature we will add is its time delay which we denote *td*. We can think of this as the length of time it takes to close the switch. This can be expressed best in the form of a timing diagram which is shown in Diagram 5.4. Consideration of the time delay is important since we need to ensure that the correct logic values are present on the input of any circuit at the correct time. This is the synchronisation problem, and occurs in large circuits where the inputs to a final stage of processing are computed using different numbers of gates. An artificial example of this is shown in Diagram 5.5 in which a fragment of a circuit is shown. Suppose that A and B both change state at the same time as shown in the timing diagram. Logically the output should not change, however, because of the transistor delay time it takes longer for the signal from A to reach the output than the for the signal from B. Hence, while waiting for the change from A to arrive, the output goes to logic 1 for a period of td. Thus, a false logic level is introduced momentarily into the circuit. These false logic states are referred to as spikes since they are normally very short, and one of the most difficult parts of digital design is to ensure that spikes do not occur, or that they do not disturb the functionality.

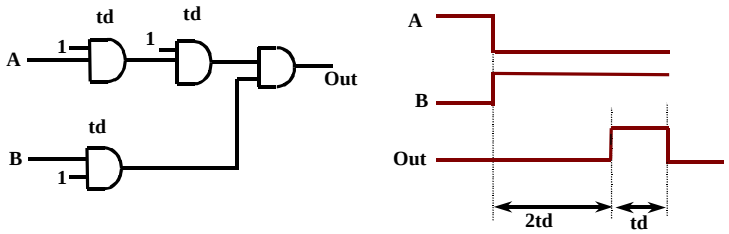

Diagram 5.5: The synchronisation problem

The next refinement that we make to our transistor model takes account of the fact that the switch is not perfect. In fact it is better modelled as a variable resistor rather than a switch, and that its relation to the gate voltage is as shown in Diagram 5.6. Here it will be seen that the change in resistance is gradual, and there is a band during which the transistor is said to be changing state. This is in contrast to the simpler switch model where the change of state was instantaneous. To complete the picture we must make one further refinement which is to add a capacitor between the G terminal and the D terminal. The model of the transistor could now be depicted by the symbol shown in Diagram 5.7, but to make drawings simpler, we will adopt the normal symbol also shown in 5.7. We now have a fairly accurate model of the NMos silicon transistor. The names given to the terminals are the gate (G), the Source (S) and the Drain (D). The effect of the capacitor is to prevent the voltage on the gate changing instantaneously. To see this we need to introduce a model of the behaviour of a capacitor. This is governed by the equation:

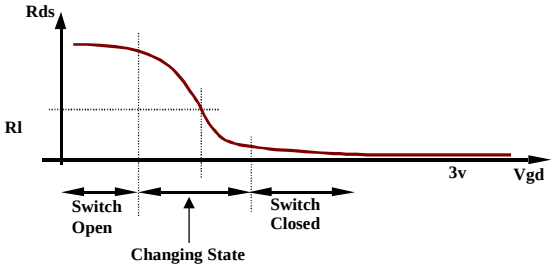

Diagram 5.6: Transistor characteristics

I= C (dV/dt) (C is a constant called the capacitance)

Now we consider a practical way of feeding an input signal to the transistor, for example from another transistor circuit. Diagram 5.8. shows a transistor connected to a capacitor which we can think of as modelling the input of a second transistor. To make the analysis simple we will choose the switch model for the first transistor, so while A is at logic 1, the voltage across the capacitor is zero. If we denote it as V we have the following equations:

5 - V = IR (Ohm's law modelling the resistors behaviour)

V = 5 - IR

V = 5 - RC (dV/dt) (eliminate I using the capacitor law above)

Re arrange and integrate

$\int dV/(5-V) = \int (1/RC)\, dt$

- log (5-V) = t/RC + K (since we take V=0 at t=0 it follows that K = -log(5))

5-V = exp(-t/RC + log(5)) = exp(-t/RC) exp(log(5)) = 5 exp(-t/RC)

V = 5(1- exp(-t/RC))

This yields the output voltage shown in the timing diagram of Diagram 5.8. Notice that the voltage will never reach 5v, and that the bigger the value of C the longer the time taken for V to rise. The second thing that we now must note is that since no voltage ever changes instantaneously, there will be a period of time when it is not possible to determine whether the transistor has changed or not. In other words the gate will not be at either logic 0 or at logic 1, it will be in a non-deterministic state. It is principally the input capacitance that is responsible for the time delay *td* that we introduced earlier on.

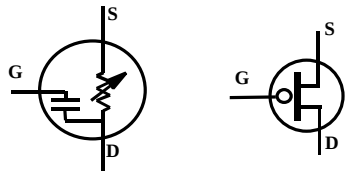

Diagram 5.7: The NMOS transistor

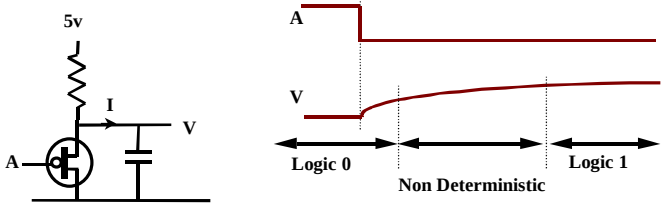

Diagram 5.8: Effect of a capacitor

We can now describe with greater accuracy what happens when an invertor changes state. Since in our more accurate model there is a variable resistor between the source and drain, the circuit can be viewed as a potential divider. In practice the resistor used will be chosen to be in the middle of the transistor range, so that it is small compared with the largest resistance of the transistor, and big by comparison with the smallest resistance. This is shown as Rl, in Diagram 5.6. Since the resistance of the transistor is never infinite, and never zero the output voltage is never either zero of 5 volts, but in practice ranges between about 0.2 and about 3.5 volts, with a non deterministic band stretching from roughly 0.5 to 1.5 volts. Thus, in practice our waveforms will look like the picture shown in Diagram 5.9. One last feature should be observed, and this is that there is a second capacitor which goes between the gate and the source. (It is in fact a symmetric device, so we can swap the source and drain connections.) The addition of the second capacitor does not change the behaviour except when the gate is not connected to anything when it will drift up towards 5v. Thus the default input is logic 1, and not logic 0 as we might suppose. This means that all inputs to a circuit that we require to be 0v must be physically connected to the 0v terminal of the power supply.

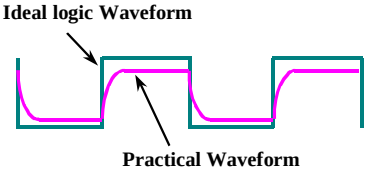

Diagram 5.9
Waveforms with rise and fall times