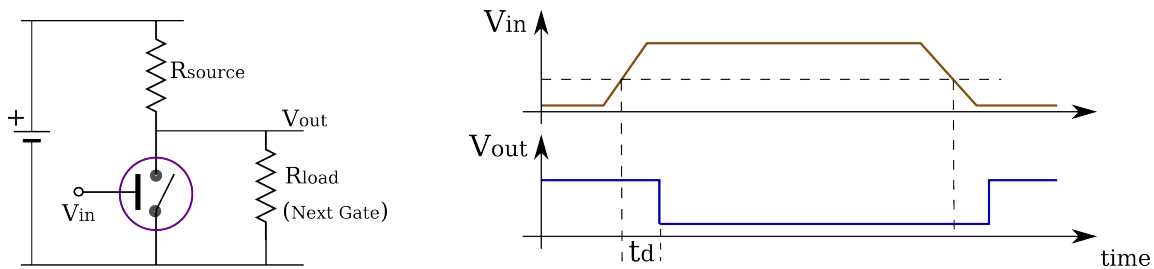


Lecture 6: Time-Dependent Behaviour of Digital Circuits

Two rather different quasi-physical models of an inverter gate were discussed in the previous lecture. The first one was a simple delay model. This model uses a perfect switch (on/off) which connects the output to ground (zero volts) when closed. When the switch is open the output goes high and settles at a positive voltage (the exact value depends on the resistors R_{source} and R_{load}). There is a time delay of t_d between the input voltage attaining a threshold value and the switch operating. Thus the time dependent behaviour in response to a *ramp-up / ramp-down* input voltage is a perfect digital waveform slightly delayed from the input. This is clearly a simplification, but captures the essence of the circuit's behaviour.



The second model of the inverter gate used a variable resistor instead of a switch. In this model the output does not change instantaneously like a switch; V_{out} plotted as a function of V_{in} is not a linear function. Its shape is governed by the characteristics of the particular transistor used, and will be of the form shown below

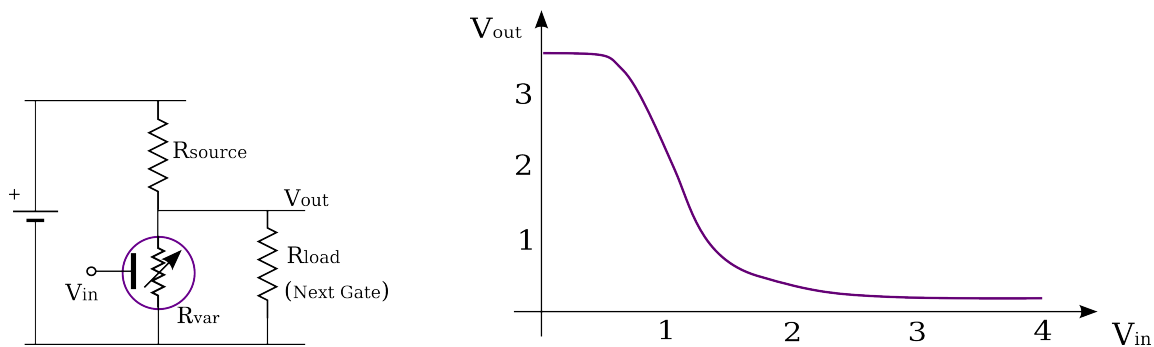
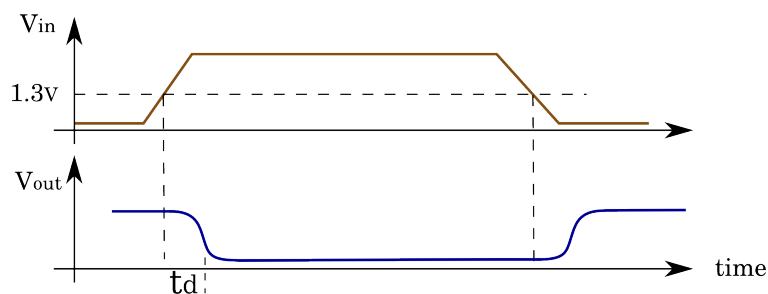


Figure 1: Variable resistance model

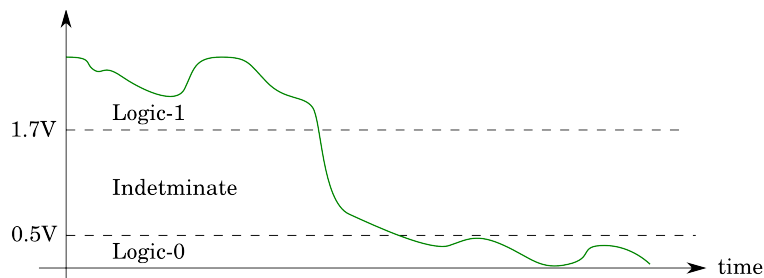
Figure 2: Inverter behaviour

We can see that the output remains relatively high (above 3 volts) as long as the input voltage is below 1V. When the input voltage rises to 1.5 V then the output is already very low, and it quickly falls to near zero, in practice to around 0.3V. This second model is closer to the actual physical truth because physical voltages cannot change instantaneously. We can use the graph of V_{out} against V_{in} to determine the output voltage waveform as a function of time given the *ramp-up / ramp-down* input voltage. Again we include the time delay and note that it is caused by the capacitance of the gate to which it is connected. Unfortunately, as a consequence of the manufacturing process, every gate has a capacitor between its input terminal and earth.



Both models produce a similar result, and so seem to agree as far as their output behaviours are concerned. Notice that the output of the second model is an analogue signal, it is continuous taking values between 0.3V

and 3.5V, but we want a digital computer! How can we reliably convert analogue signals to indicate digital values? The answer involves the concept of noise margin.



Real physical signals are never steady. They vary, often randomly in time and we say that they have *noise*. In order to associate such signals with digital values, we define three distinct signal values: Logic-0, Logic-1, and Indeterminate. If the signal is larger than 1.7 volts then it has the value of Logic-1. If the signal is lower than 0.5 volts then it is Logic-0, otherwise, its "digital" value is Indeterminate. Now, with the non-linear transistor behaviour of figure 1, if the input is below 0.5 volts, the output will be around 3.5 volts which is definitely a Logic-1. If the input is above 1.7 volts then the output will be below 0.5 volts, which is definitely a Logic-0. This means that after we have wired up a combinational circuit (of any complexity) and provided it with proper input values, no Indeterminate values will be ever present given that we have waited long enough for all the devices to reach their final state. During switching, all devices must go through the indeterminate phase, but the non-linearity helps to keep this time to a minimum.

What is even more exciting, noise (if moderate) will not destroy our "perfect" digital system. The nominal Logic-1 voltage is 3.5 volts and Indeterminate (trouble) state starts only at 1.7 volts; therefore, any noise wave with a peak-to-peak noise wave of less than $2 \times (3.5 - 1.7) = 3.6$ volts will not upset our steady Logic-1 output. Similarly, at Logic-0 the nominal voltage is 0.3 volts but should be less than 0.5 volts, so the noise margin is peak-to-peak $2 \times (0.5 - 0.3) = 0.4$ volts.

Fan out

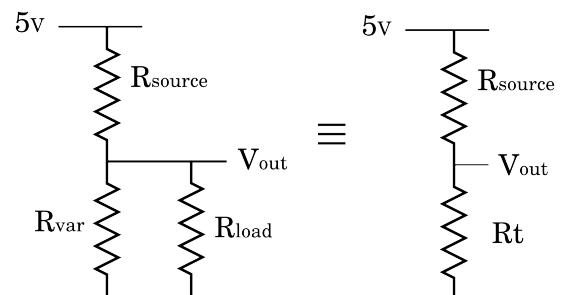
Fan out refers to the number of inputs to which the output of a gate is connected. It turns out that as fan out increases the noise tolerance is reduced and the speed of response is reduced.

If the fanout is 1, we can calculate the output voltage as follows. For the two lower resistors in parallel we can calculate as single equivalent resistance from the formula:

$$\frac{1}{R_t} = \frac{1}{R_{var}} + \frac{1}{R_{load}}$$

The output voltage is computed simply from the ratio of the upper and lower resistors:

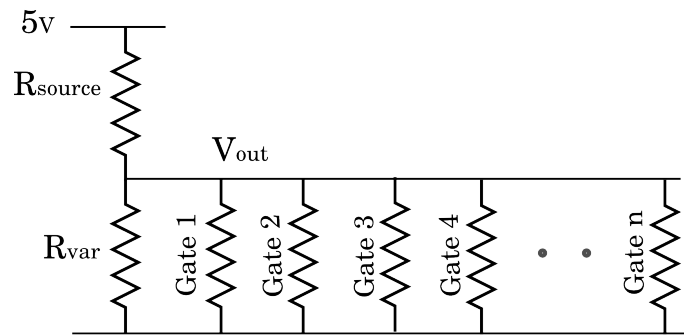
$$V_{out} = \frac{5R_t}{R_{source} + R_t}$$



To see how our circuit performs let's supply some typical values for the resistors, and calculate its output.

	$R_{source}(\Omega)$	$R_{load}(\Omega)$	$R_{var}(\Omega)$	$R_t(\Omega)$	V_{out} (Volts)
Logic-1	1000	10000	3000	2308	3.5
Logic-0	1000	10000	60	59	0.28

Under these conditions the circuit works well, producing a clearly defined logic output. However if we connect its output to more than one gate things don't look quite so good. The input load resistance of each gate appears in parallel as shown below. The resistors in parallel are combined using the inverse sum as before. Since they all have the same resistance, say R_l , if we have a fan out of n the combined load resistance will be R_l/n .

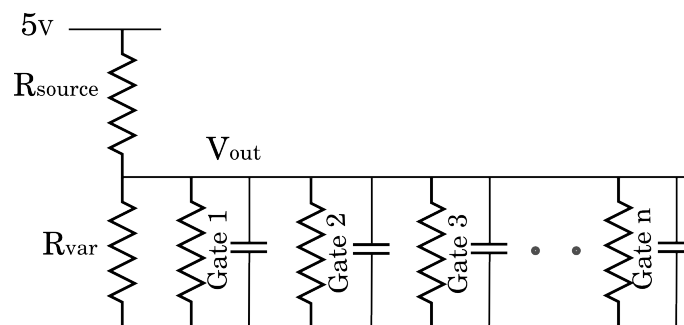


Using the figures we had before, for a fan out of 10 the value of the combined load resistance becomes 1000Ω , and the table becomes:

	$R_{source}(\Omega)$	$R_{load}(\Omega)$	$R_{var}(\Omega)$	$R_t(\Omega)$	V_{out} (Volts)
Logic-1	1000	1000	3000	750	2.1
Logic-0	1000	1000	60	57	0.27

Now we see that for the Logic-1 state, the output voltage has been drawn down by the extra load, and is in fact perilously near to the indeterminate value. The Logic 0 state is not affected. In practice it is unwise to use a fanout of more than ten times.

The other problem with fan out is that it slows down the circuit. This is because of the input capacitance of each gate, which we have so far ignored in this lecture. If we include it, the load attached to an inverter looks like this.



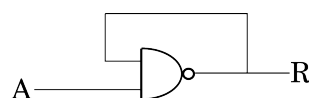
Capacitors in parallel simply add their values, so if the capacitance of an individual gate is C_l , the total load capacitance for a fan out of n is:

$$C_{load} = C_l + C_l + \dots C_l = n \times C_l$$

We saw last lecture that the time taken to charge a capacitor from zero volts to the threshold for logic 1 is proportional to the load capacitance. Hence, if the time delay t_d for a gate to switch when the fan out is 1 is t_d , then the time to switch when the fan out is n will be $n \times t_d$. This is the most compelling reason for limiting the fan out, since we want our circuits to operate as fast as possible.

Feedback

You may well ask: why do we need these physical models? We have shown that digital gates work reliably. We connect them together and that's all there is to it. Unfortunately, things are never that simple. If we want to look at the general case, when any input can be connected to any output then we have to consider feedback connections. One such feedback connection is shown in the next diagram which uses a two-input NAND gate.



The question is: what is the truth table of this one-input (A), one-output (R) device?

If A is at Logic-0 then the output of the NAND gate will be high. Since this is "fed back" to the other input, we have 10 as inputs which is still consistent with a Logic-1 output of the NAND gate. Thus, we have a

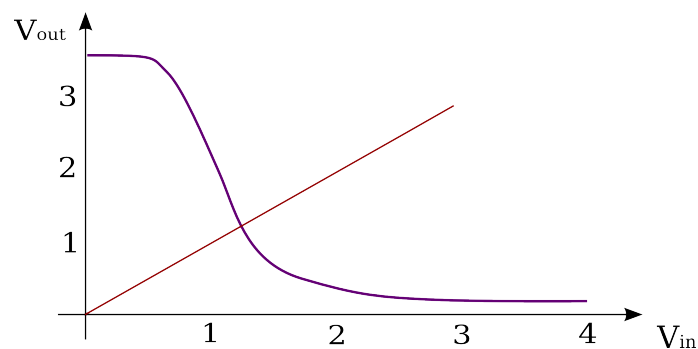
consistent circuit. However, if the input A is set to Logic-1 then it depends what the output R is before we can tell what the other input is. Since R was high when A was low, we start with this value. Now the NAND inputs are at 11 and the output should be 0 (the opposite of what it was). However, if the output is 0 then it is fed back to the input and we have 01 for inputs which should make the output to be 1. We have found an inconsistent circuit.

Now let us use our physical models to try to figure out what will happen. The switch with the delay indicates oscillation. Say, the output is high. This is sensed by the circuit but the output will not change for a small time interval (t_d). After changing, another delay time elapses before the output will change again.

What kind of behaviour will our second model give us? When the A input of the NAND gate is set to Logic-1 the device becomes a one-input, one-output inverter. We can use feedback to connect this output to the input and in fact we have the simple equation for the inverter:

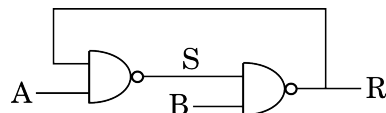
$$V_{in} = V_{out}$$

We can use this fact when looking at the input output behaviour predicted by the second model as shown in figure 2. The condition $V_{in} = V_{out}$ can be shown on this figure by a line drawn at 45° .



The line crosses the output/input curve at 1.2 volts, so this model suggests that the output of the circuit becomes indeterminate, and there is no oscillation. The fascinating thing is that in the laboratory either output behaviour of an inconsistent circuit can occur. It may oscillate or settle at an Indeterminate value, though it is more likely to do the latter.

As a last example of a general circuit with feedback, we demonstrate a method by which the behaviour of a feedback circuit may be determined. Consider the following circuit which uses two NAND gates:



We will analyse it using the Delay Model. The first step is to give a signal name to all inputs, outputs and internal wires in the circuit: A,B,R and S. The second step is to find the Boolean equation for all outputs in terms of the signal inputs.

$$\begin{aligned} S &= (A \cdot R)' \\ R &= (S \cdot B)' \end{aligned}$$

With the delay model the outputs change after some delay; thus we can assume that all signals have constant values for a small interval. We have to look at all possible combinations of signal values at time t and calculate their new values for time $t + t_d$ in order to analyse a circuit with feedback. Let U stand for unknown (either 1 or 0). The rules of Boolean algebra can be stated for these UNKNOWN values as:

$$U \cdot U = U \quad U \cdot 0 = 0 \quad U \cdot 1 = U \quad U + U = U \quad U + 0 = U \quad U + 1 = 1 \quad U = U$$

Now let us apply the proper voltage values to A and B for logic 0 and logic 1.

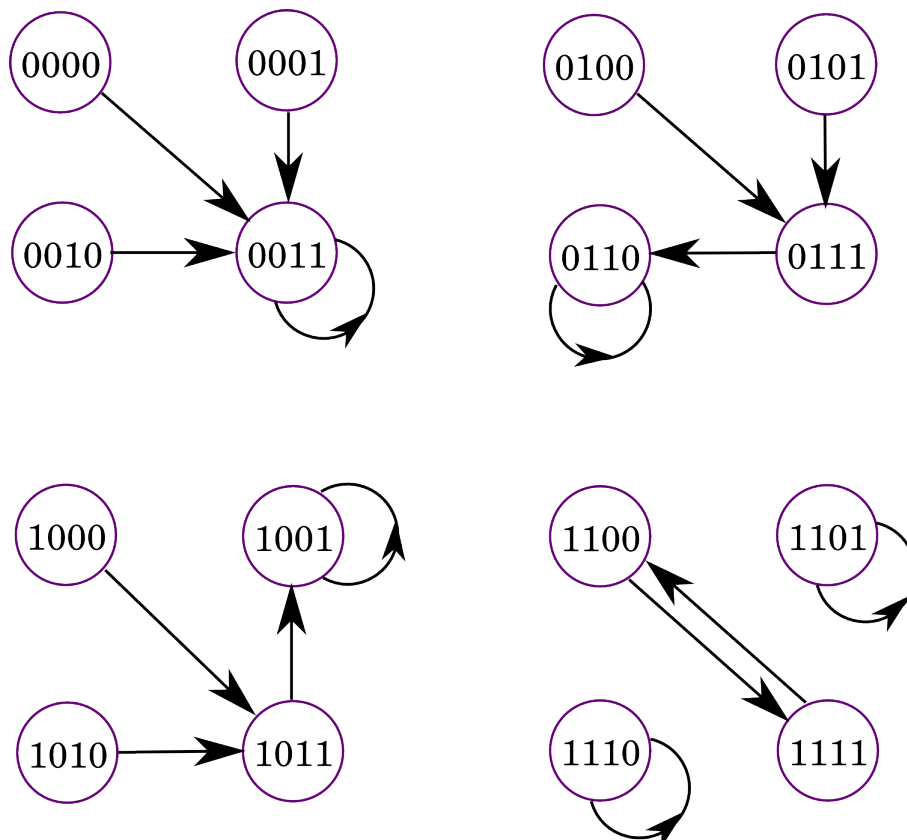
	t	$t + t_d$	$t + 2t_d$		$t + nt_d$
A	0	0	0	.	0
B	1	1	1	.	1
S	U	1	1	.	1
R	U	U	0	.	0

We see that after two gate delays the circuit ends in a well defined stable logic state. However, not every input will produce a stable output.

Let's see what happens to the circuit in one time step from each possible logic state that it can be in.

t				$t + t_d$				
A	B	S	R	A	B	S	R	
0	0	0	0	0	0	1	1	
0	0	0	1	0	0	1	1	
0	0	1	0	0	0	1	1	
0	0	1	1	0	0	1	1	Stable
0	1	0	0	0	1	1	1	
0	1	0	1	0	1	1	1	
0	1	1	0	0	1	1	0	Stable
0	1	1	1	0	1	1	0	
1	0	0	0	1	0	1	1	
1	0	0	1	1	0	0	1	Stable
1	0	1	0	1	0	1	1	
1	0	1	1	1	0	0	1	
1	1	0	0	1	1	1	1	
1	1	0	1	1	1	0	1	Stable
1	1	1	0	1	1	1	0	Stable
1	1	1	1	1	1	0	0	

We see that some states are stable, for other starting states the circuit may cycle. It's difficult to see this looking at the table so we will construct a state transition diagram to show how the states change. This representation is sometimes called a the finite state machine representation.



Looking at the diagram we see that the behaviour of this circuit is really interesting. If the inputs are 00, 01 or 10 then the circuit will end up in a stable logic state. For the input 11, if the circuit starts with RS=01 or RS=10, then it remains in those states. Otherwise it cycles between RS=00 and RS=11. We will see next lecture that this behaviour can be harnessed to make a memory.