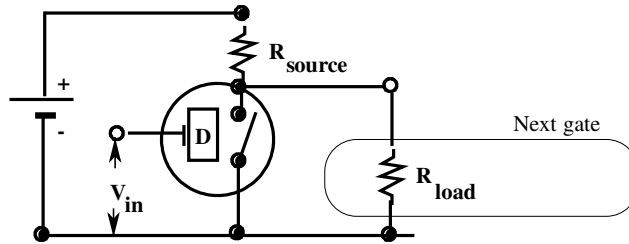
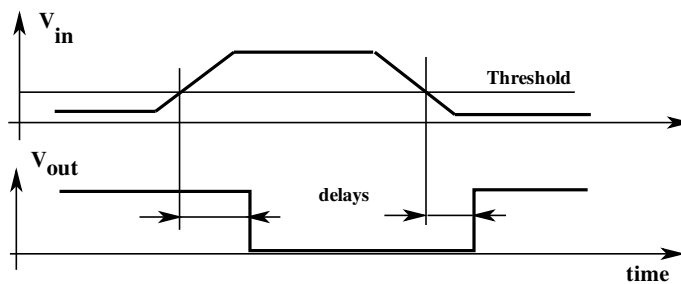


**Lecture 6: Physical Time-Dependent Behaviour of Digital Circuits and Feedback**

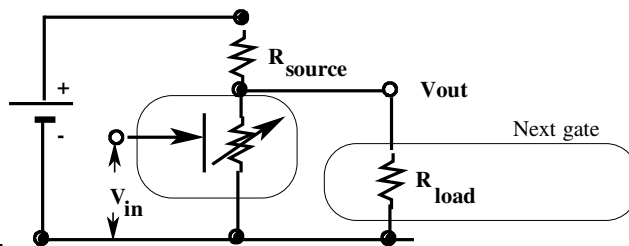
Two rather different quasi-physical models of an inverter gate have been shown in the previous lecture. The first one is the **Delay Model**. This model uses a perfect switch (on/off) which connects the output to ground (zero volts) when closed. It has no effect on the circuit when it is open. When the 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 delay between the input voltage attaining a **threshold** positive value and the switch closing.



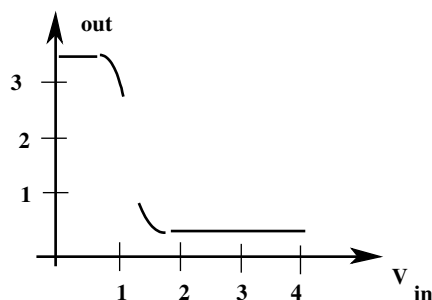
The time dependent behaviour of the input and output voltages are shown on the next diagram:



The second quasi-physical model of an inverter gate uses a **variable resistor**.

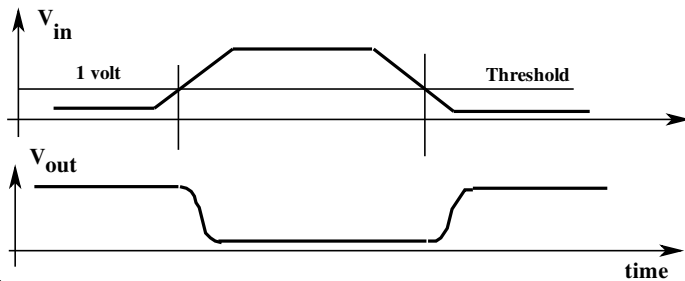


Like in the switch in this model the output does not change instantaneously; however, here  $V_{out}$  vs  $V_{in}$  (as shown in the next diagram) is not a linear function. The output vs. input voltage of a given configuration (with a given  $R_{load}$ ) will have the form:



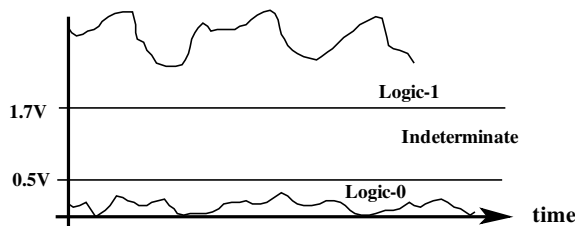
We can see that the output remains relatively high (around 3.5 volts) as long as the input voltage is below 1V. When the input voltage is above 1.5 V then the output is already very low, near to zero, in practice it is near to 0.3V. In fact, this model is closer to the actual physical truth because physical voltage signals cannot change instantaneously.

We can use this graph ( $V_{out}$  vs.  $V_{in}$ ) to produce the input and output voltage waveforms as functions of time and get:



Which is similar to the one generated by our first model. Thus, the two models seem to agree as far as their output behaviours are concerned.

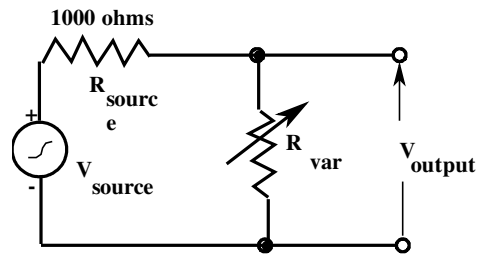
The output is an **analogue** signal, it is continuous, but we want to build a **digital** computer! How can we convert analogue signals to indicate digital values? Actually, the solution is quite ingenious and involves the concept of **noise margin**.



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-zero**. Otherwise, its "digital" value is **Indeterminate**. Now, with our second (non-linear resistor) model, 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 (our rule) then the output will be around 0.3 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, assuming that we have waited long enough for all the devices to switch to their final state. We have to wait in a real situation because of gate delays. During switching, all devices **must go through the Indeterminate** phase.

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.

In order to see why we have different noise margins at the **Logic-0** and the **Logic-1** states, we need a bit of analysis. Let's go back to our second model with a typical value of 1000 Ohms for  $R_{source}$ .



Electrical engineers learn **Ohm's law** early in their student carrier and can tell you that the output  $V_{output}$  can be calculated by:

$$V_{output} = V_{source} * R_{var} / ( R_{source} + R_{var} )$$

Here  $V_{source}$  represents noise. Since the variable resistor is non-linear, it will have different values for its **Logic-1** and **Logic-0** states. For example, if we measure it, we may find that for the **Logic-1** state its resistance is equal to 3000 Ohms. If we assume a noise signal with 3 volts peak-to-peak (this is rather pessimistic), we get:

$$V_{noise} = 3.0 * 3000 / ( 3000+1000 ) = 2.2V \text{ p-to-p}$$

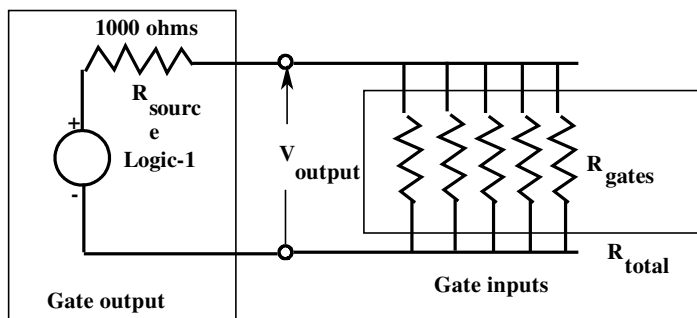
which is below our danger limit. Now if we switch to the **Logic-0** state, the resistance drops to a low 20 Ohms (these are reasonably realistic values), and we have:

$$V_{noise} = 3.0 * 20 / ( 20 + 1000 ) = 0.06V \text{ p-to-p}$$

still well below the danger limit of 0.4 volts peak-to-peak. Even with this large amount of noise our digital circuit will work reliably.

### Fan-Out

The term fan-out refers to the connection of the output of one gate to many inputs. The important question is, how many input devices could be connected to one gate? We can answer this question with our model if we ask again the advice of our Electrical Engineering friends about the total resistor value of resistors connected in parallel. We can approximate the circuit by ignoring  $R_{var}$  when we have:



Our electrical engineering friend tells us that the combined resistor value of "N" resistors connected in parallel is:

$$R_{\text{total}} = (1/N) R_{\text{gate}}$$

The total load resistor decreases as we connect more and more input gates to our device. Let us calculate the output voltage at the **Logic-1** level with a load resistor of 10,000 Ohms.

$$V_{\text{out}} = (3.5 \text{ V}) * (10000/N) / [ 1000 + (10000/N) ] > 1.7 \text{ V}$$

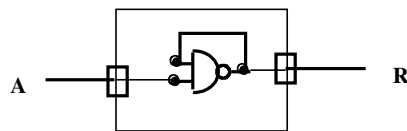
we get:

$$1,700 N < 18,000$$

and the maximum number of gates we can connect is ten or so. This is the fan-out of an ordinary 7400 series gate (normally). There are special "amplifier" gates whose fan-out is much larger.

### Feedback Connections

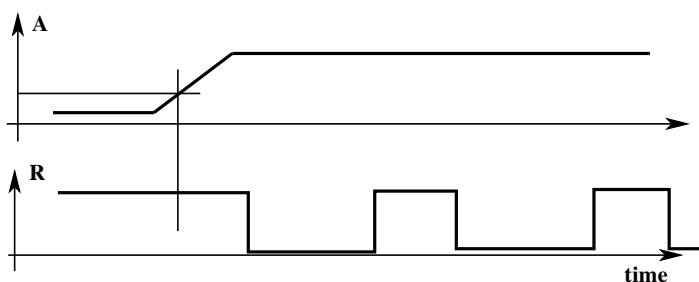
You may well ask: why do we need these physical models? We have shown that digital gates work reliably. We connect them and that's all to it. Unfortunately, things are never that simple. If we want to look at the general case, when any input can be connected to any output then we have to consider **feedback** 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** which wants to make the output to be **0** (the opposite 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 found an inconsistent circuit.**

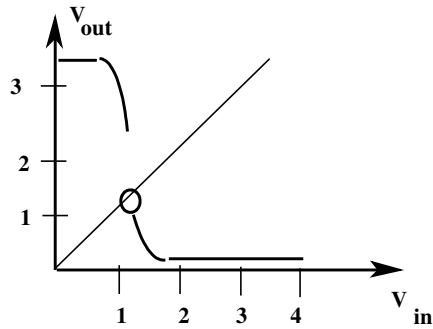
Now let us use our physical models to try to figure out what will happen. The switch with the delay indicates **oscillation**. Say, the output is high. This is sensed by the circuit but the output will not change for a small time interval (delay time). After changing, another delay time elapses before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



We have not finished yet. 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:

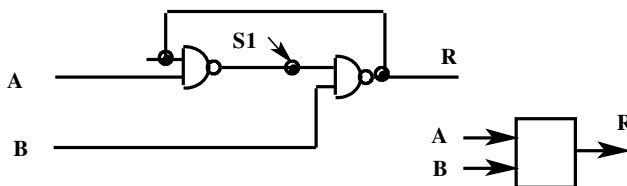
$$V_{in} = V_{out}$$

In the following diagram the above equation can be indicated by a line at 45°. The line crosses the output/input curve at 1.13 volts and there is **no oscillation**. The output settles to the **Indeterminate** value (so, it is not good at all).



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.

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:



For this method we use the **Delay Model**. The first step is to give a signal name to all inputs and outputs. The signals **A**, **B**, and **R** have been specified with the circuit, we use the symbol **S1** for the output of the first **NAND** gate.

The second step is to find the Boolean equation for all outputs in terms of the signal inputs.

$$S1 = (A \cdot R)'$$

$$R = (S1 \cdot B)'$$

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 will call this small interval of time **now**. According to the circuit equations, the outputs may change or not change depending on their input values at **now**. After some small delay time, signals will change to new values and will remain again constant for a short interval, which we call **later**. We have to look at all possible combinations of signal values at time **now** and calculate their new values for time **later** in order to analyse a circuit with feedback. In our example, let us apply the proper voltage values for **A** and **B** as logic **0** and logic **1**. **U** stands for “unknown”, i.e. either **1** or **0**. The rules of Boolean algebra can be stated for these **UNKNOWN** values as:

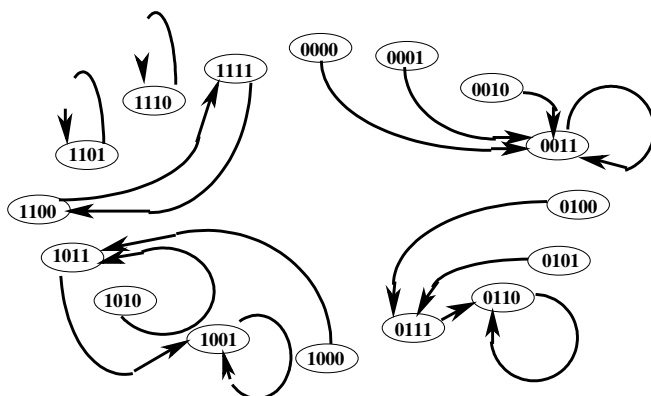
$$U \cdot U = U ; U \cdot 0 = 0 ; U \cdot 1 = U ; U + U = U ; U + 0 = U ; U + 1 = 1 ; U' = U$$

	<u>now</u>	<u>1 gate delay later</u>	<u>2 gate delays later</u>	<u>N gate delays later</u>
<b>A</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>B</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>S1</b>	<b>U</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>R</b>	<b>U</b>	<b>U</b>	<b>0</b>	<b>0</b>

Finally, for each possible **now** combination of all the signals (both inputs and outputs) we calculate the signal values at the **later** time. The independent inputs (**A**, **B**) do not change. There are four possible cases for these. We encircled the **stable** states, indicating that these states will not change (the **now** and **later** values are the same)

A	B	S1	R (now)	A	B	S1	R (later)	
0	0	0	0	0	0	1	1	
0	0	1	0	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	1	1	stable
1	0	1	0	1	0	1	1	
1	0	1	1	1	0	1	1	
1	1	0	0	1	1	1	1	stable
1	1	0	1	1	1	0	1	stable
1	1	1	0	1	1	1	0	
1	1	1	1	1	1	0	0	

We can also construct a **transition diagram** for this circuit. Every one of the sixteen possible states as a **now** state turns into one of the states as a **later** state. This can be indicated by an arrow:



Both the list and especially the diagram demonstrates the following:

- If the input values **AB** are **00** the outputs settle to values **11**
- If the input values **AB** are **01** the outputs settle to values **10**
- If the input values **AB** are **10** the outputs settle to values **01**
- If the input values **AB** are **11** the outputs settle **either** to values **10** **or** to values **01** or they could possibly oscillate between the values **11** and **00**.

A fascinating result! When we set the inputs to be **11** the **R** output is either **0** or it is **1** and it is stable. We discovered **memory!!**