Modeling Logic Gates with Delay- Part#2

by George Lungu

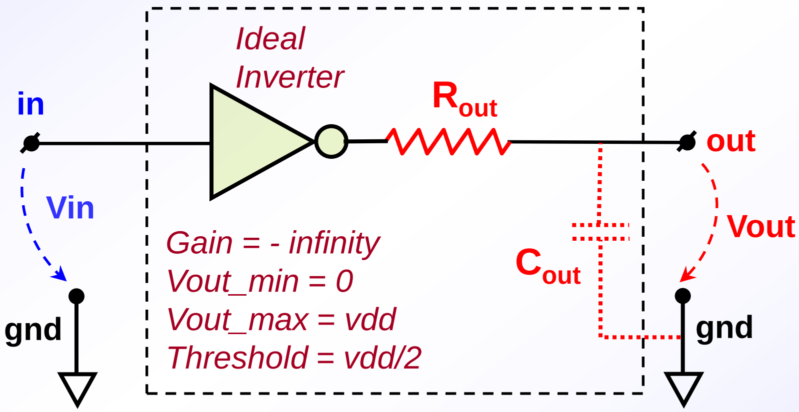
- In the previous section of this tutorial we concluded that in order to be able to simulate with decent precision delay in single-stage logic gates, we only need to introduce a propagation delay caused solely by rise or fall time. This makes the model of a single-stage gate fairly simple and easy to program in a single spreadsheet cell, using built-in formulas (hence making the model much faster than employing user defined functions).
- This second half of the tutorial derives the final formulas and a delay based model is implemented in Excel 2003.

The simplified equivalent model:

- We use the model to the right, which (as we will see) can be written using spreadsheet built-in functions in a single cell without the need to use (very slow) user defined functions =>>>>>

- In the previous series of tutorials about ideal logic gates we have already seen how to describe an inverter and a few other ideal logic gates in equations.

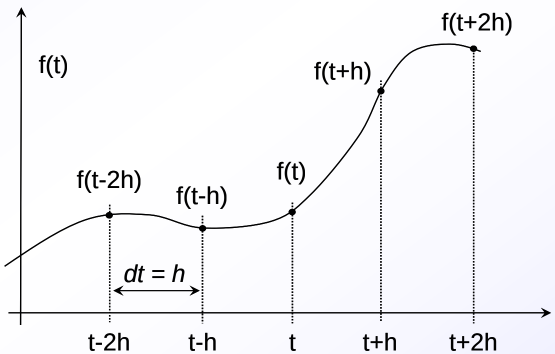
Let's now see how we can combine that the ideal logic gate description with a numerical low pass filter solution.

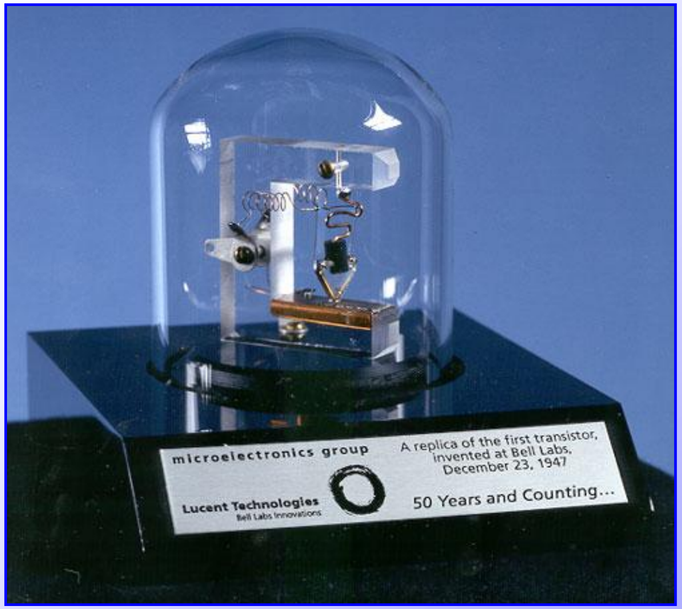


A 1-pole Low Pass Filter numerical solution:

- This topic was treated on this blog before, therefore it will be treated lightly here.

- In order to numerically model a time dependant process we first need to sample its time function at discrete intervals **dt = h**





The first transistor - invented at Bell Labs in 1947

- Most of the physical processes can be described by differential equations. For easy solutions we need to find a way of expressing derivatives in an approximate but easy way. The easiest way would be to approximate the tangent to the curve in point T with the secant to the curve (there are three options available).

Three ways of approximating the tangent (first derivative) to a curve:

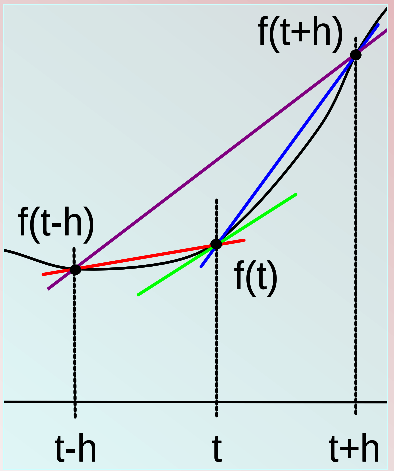
Definition: $\frac{df(t)}{dt} = \lim_{dt \to 0}\left[\frac{f(t+dt)-f(t)}{dt}\right]$

- To the left there are three different ways of simple approximations to the tangent using the actual value of the function in various points in time.

Forward estimate: $\frac{df(t)}{dt} \approx \frac{f(t+h)-f(t)}{h}$

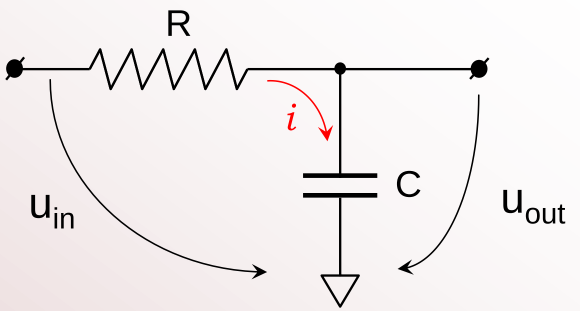
Backward estimate: $\frac{df(t)}{dt} \approx \frac{f(t)-f(t-h)}{h}$

Central estimate: $\frac{df(t)}{dt} \approx \frac{f(t+h)-f(t-h)}{2 \cdot h}$



Pay attention to colors

- Let's use the forward estimate to numerically solve the system below:



From Ohm's law: $u_{in}(t) - u_{out}(t) = R \cdot i$

From of current intensity definition: $i = \frac{dq}{dt}$

From the capacitance definition: $C = \frac{dq}{du_{out}}$

=> $u_{in}(t) - u_{out}(t) = R \cdot C \cdot \frac{du_{out}}{dt}$

using the forward estimate

$$u_{in}(t) - u_{out}(t) = R \cdot C \cdot \frac{u_{out}(t+h) - u_{out}(t)}{h}$$

<www.excelunusual.com>

After some minor algebraic manipulation we reach the final formula for $u_{out}(t)$:

$$u_{out}(t+h)=\frac{h}{R\cdot C}\cdot[u_{in}(t)-u_{out}(t)]+u_{out}(t)$$

The equation is valid for any moment in time. Moving it back one time step "h" (which means t = t-h) we get:

$$u_{out}(t)=\frac{h}{R\cdot C}\cdot[u_{in}(t-h)-u_{out}(t-h)]+u_{out}(t-h)$$

The final formulas for seven different logic gate models with delay:

$$u_{out\_INV}(t)=\frac{h}{R\cdot C}\cdot\left[if\,(u_A(t-h)>\frac{vdd}{2},0,vdd)-u_{out}(t-h)\right]+u_{out}(t-h)$$

$$u_{out\_AND}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(AND\left(u_A(t-h)>\frac{vdd}{2},u_B(t-h)>\frac{vdd}{2}\right),vdd,0\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

$$u_{out\_NAND}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(AND\left(u_A(t-h)>\frac{vdd}{2},u_B(t-h)>\frac{vdd}{2}\right),0,vdd\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

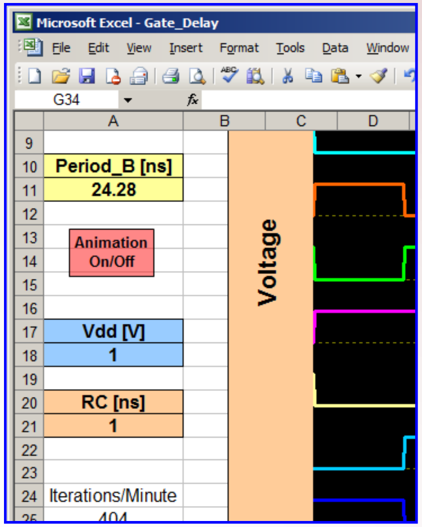
$$u_{out\_OR}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(AND\left(u_A(t-h)<\frac{vdd}{2},u_B(t-h)<\frac{vdd}{2}\right),0,vdd\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

$$u_{out\_NOR}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(AND\left(u_A(t-h)<\frac{vdd}{2},u_B(t-h)<\frac{vdd}{2}\right),vdd,0\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

$$u_{out\_XOR}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(OR\left(AND\left(u_A(t-h)>\frac{vdd}{2},u_B(t-h)<\frac{vdd}{2}\right),AND\left(u_A(t-h)<\frac{vdd}{2},u_B(t-h)>\frac{vdd}{2}\right)\right),vdd,0\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

$$u_{out\_XNOR}(t)=\frac{h}{R\cdot C}\cdot\left[if\left(OR\left(AND\left(u_A(t-h)>\frac{vdd}{2},u_B(t-h)<\frac{vdd}{2}\right),AND\left(u_A(t-h)<\frac{vdd}{2},u_B(t-h)>\frac{vdd}{2}\right)\right),0,vdd\right)-u_{out}(t-h)\right]+u_{out}(t-h)$$

Excel 2003 implementation:

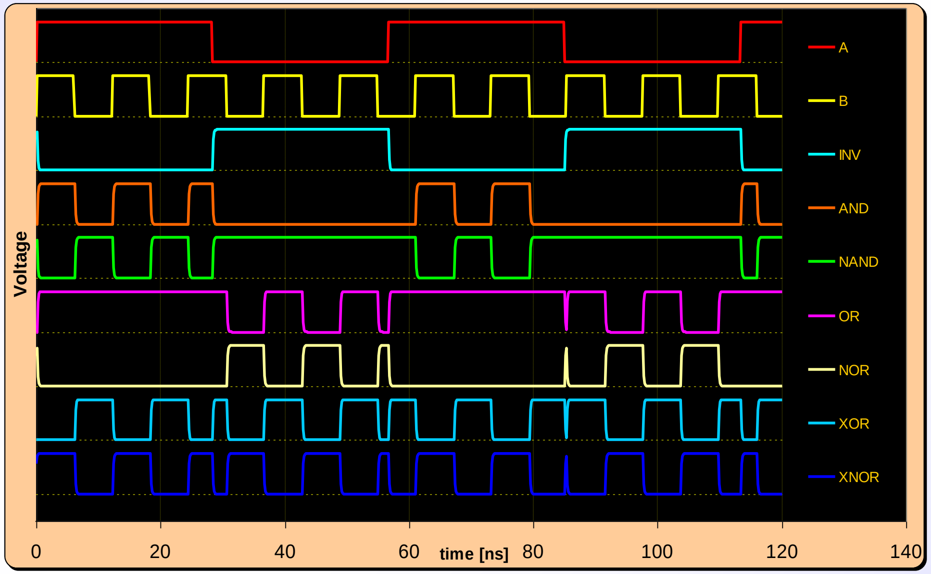


- Copy the last file "Logic_Gates.xls" and rename the new file "Gate_Delay.xls"
- Delete the first four worksheets and leave only the last one named "Basic_4"
- Rename the remaining worksheet: "Seven_Gates_with_Delay"
- Insert a new name for cell A21: "r_c" and type a "0.5" in cell A21.
- Change the formulas in the following cells:
- Cell D37: "=time_step*(IF(B38>vdd/2,0,vdd)-D38)/r_c+D38"
- Cell E37: "=time_step*(IF(AND(B38>vdd/2,C38>vdd/2),vdd,0)-E38)/r_c+E38"
- Cell F37: "=time_step*(IF(AND(B38>vdd/2,C38>vdd/2),0,vdd)-F38)/r_c+F38"
- Cell G37: "=time_step*(IF(AND(B38<vdd/2,C38<vdd/2),0,vdd)-G38)/r_c+G38"
- Cell H37: "=time_step*(IF(AND(B38<vdd/2,C38<vdd/2),vdd,0)-H38)/r_c+H38"
- Cell I37: "=time_step*(IF(OR(AND(B38>vdd/2,C38<vdd/2),AND(B38<vdd/2,C38>vdd/2)),vdd,0)-I38)/r_c+I38"
- Cell J37: "=time_step*(IF(OR(AND(B38>vdd/2,C38<vdd/2),AND(B38<vdd/2,C38>vdd/2)),0,vdd)-J38)/r_c+J38"
- After all these formula alterations copy range D37:J37 down to row 837 and you are ready to test the model.

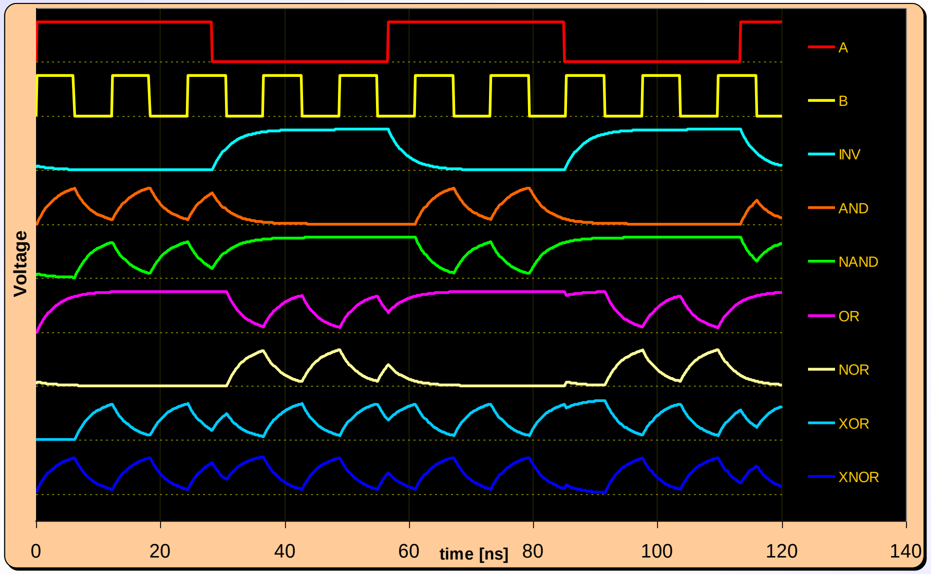
Running and testing the animated model:

- The snapshots are taken at the following values of RC (r_c) constants : 0.2ns, 0.6ns and 2ns.

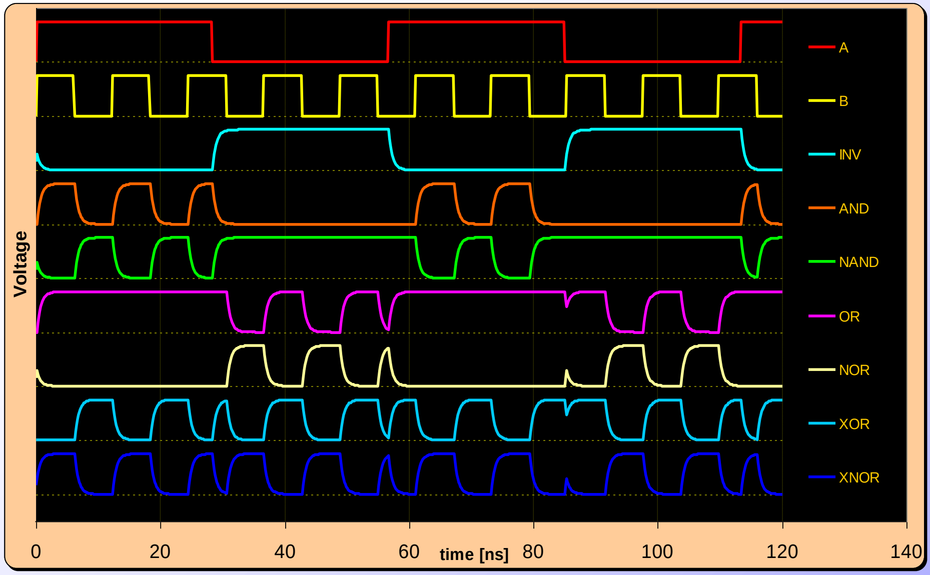
- The speed of the model is confirmed very good, namely 663 frames/minutes with the chart visible and 5168 frames/minute with the chart out of sight. This is because no user defined functions were used throughout the model.



RC = 0.2 ns



RC = 3 ns



RC = 0.7 ns