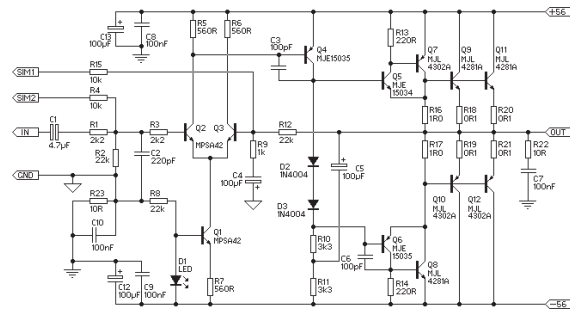
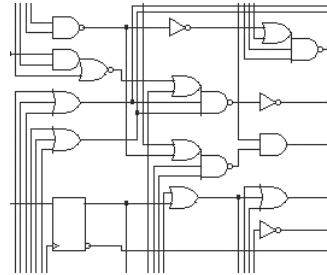




Analog Simulation

- **Digital simulation**
 - discrete values
 - bit, boolean, enumerated, integer
 - exception - floating point
 - discrete timing
 - cycle based - uniform time intervals
 - event based - nonuniform time intervals

- **Analog simulation**
 - continuous values
 - represented as floating point numbers
 - continuous timing
 - time or frequency domain



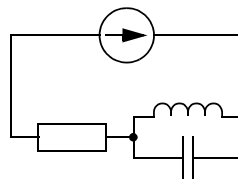
Analog Simulation

- **Low abstraction level descriptions**
 - detailed delay, power consumption, etc. estimation
 - transistors, polygons, etc.
- **Netlist of components**
 - Differential equations to describe components
 - Small signal modelling
 - frequency domain (Fourier transformation)
 - Large signal modelling
 - time domain (Laplace transformation)
- **SPICE**
 - Simulation Program with Integrated Circuit Emphasis
 - Donald O. Pederson (Prof. Emer. of UC, Berkeley)



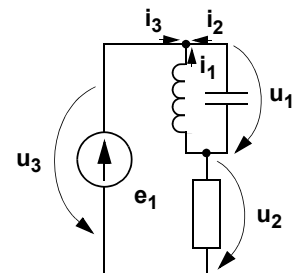
Electric Circuit Model

- **Circuit – graphical model**
 - node – connection point of three or more components
 - branch – path between two nodes
 - loop – closed path in a circuit
- **Series connected components – constant current**
- **Parallel connected components – constant voltage**



Kirchoff's Laws

- **Kirchoff's first law**
 - applied onto nodes
 - the sum of currents entering a node is zero
 - $i_1 + i_2 + i_3 = 0$
- **Kirchoff's second law**
 - applied onto loops
 - the sum of voltage drops of elements in a loop is equal to the sum of electromotoric powers in the loop
 - $u_1 + u_2 = e_1$
 - $u_3 = e_1$





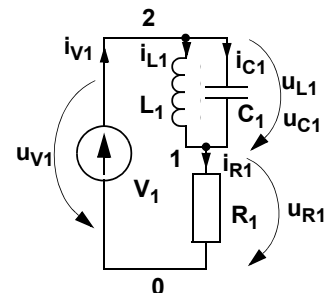
Circuit Calculation

- Set of equations
- How to select loops?
- p branches and q nodes --> p variables
 - p' – independent current sources
 - or the current does not depend on the voltage of the source
 - $q-1$ equations based on the Kirchoff's first law
 - $p-p'-(q-1)$ equations based on the Kirchoff's second law
- Inductors and capacitors --> differential equations
 - $C = q_C / u_C \quad i_C = dq_C/dt = C(du_C/dt)$
 - $L = \Psi_L / i_L \quad u_L = d\Psi_L/dt = L(di_C/dt)$



Circuit Calculation

- Kirchoff's first law
 - (trivial)node 0 - $i_{R1} = i_{V1}$
 - node 1 - $i_{L1} + i_{C1} = i_{R1}$
 - node 2 - $i_{V1} = i_{L1} + i_{C1}$
- Kirchoff's second law
 - loop 1 - $u_{L1} = u_{C1}$
 - loop 2 - $u_{V1} = u_{L1} + u_{R1}$
 - loop 3 - $u_{V1} = u_{C1} + u_{R1}$
- $p = 3, \quad q = 2, \quad p' = 0$

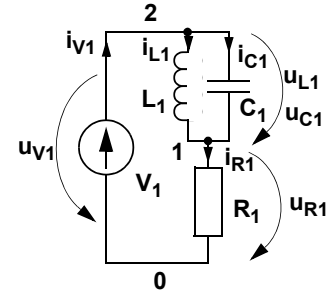




Circuit Calculation

- $p = 3, \quad q = 2, \quad p' = 0 \quad \rightarrow 1 \text{ K1 \& 2 K2}$

- $i_{C1} = C1 \cdot (du_{C1}/dt)$
- $u_{L1} = L1 \cdot (di_{L1}/dt)$
- $u_{R1} = R1 \cdot i_{R1}$
- $e_{V1} = u_{V1}$



- 1: $i_{L1} + C1 \cdot (du_{C1}/dt) = i_{R1}$
- 2: $e_{V1} = L1 \cdot (di_{L1}/dt) + R1 \cdot i_{R1}$
- 3: $L1 \cdot (di_{L1}/dt) = u_{C1}$



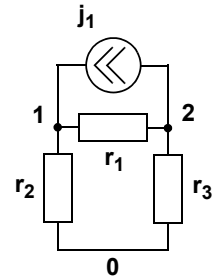
Complex Representation

- $u = U_m \cdot \sin(\omega \cdot t + \psi) = J_m(\bar{U}_m \cdot e^{j \cdot \omega \cdot t})$
- $du/dt = J_m((j \cdot \omega \cdot \bar{U}_m) \cdot e^{j \cdot \omega \cdot t})$
- Ohm's law – $\bar{U} = \bar{i} \cdot Z \quad - \quad Z = \bar{U} / \bar{i}$
- Resistor – $i = u / R \quad - \quad Z = r$
- Capacitor – $i = C (du/dt) \quad - \quad Z = 1/(j \cdot \omega \cdot C)$
- Inductor – $u = L (di/dt) \quad - \quad Z = j \cdot \omega \cdot L$



Node Voltage Method

- q nodes
 - $q-1$ equations – base node, Kirchoff's first law
- Matrices - $(u) = (g)^{-1} \cdot (j)$
 - (u) - node voltages, vector $[q-1]$
 - (g) - conductivity between nodes, matrix $[(q-1) \times (q-1)]$
 - (j) - current sources (entering the nodes), vector $[q-1]$
- $(u)^T = [(u_{10}) (u_{20})]$
- $(j)^T = [(j_1) (-j_1)]$
- (g) :
 - $g_{11} = (1/r_1) + (1/r_2)$
 - $g_{12} = -(1/r_1)$
 - $g_{21} = -(1/r_1)$
 - $g_{22} = (1/r_1) + (1/r_3)$



Node Voltage Method

- $(j) = (g) \cdot (u)$ $(u) = (g)^{-1} \cdot (j)$

$$\begin{bmatrix} u_{10} \\ u_{20} \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{r_1}\right) + \left(\frac{1}{r_2}\right) & -\left(\frac{1}{r_1}\right) \\ -\left(\frac{1}{r_1}\right) & \left(\frac{1}{r_1}\right) + \left(\frac{1}{r_3}\right) \end{bmatrix}^{-1} \times \begin{bmatrix} j_1 \\ -j_1 \end{bmatrix}$$

- $A^{-1} = (1/D) \cdot |A_{ij}|^T$

$$(g)^{-1} = \left(\frac{1}{D}\right) \times \begin{bmatrix} \left(\frac{1}{r_1}\right) + \left(\frac{1}{r_3}\right) & \left(\frac{1}{r_1}\right) \\ \left(\frac{1}{r_1}\right) & \left(\frac{1}{r_1}\right) + \left(\frac{1}{r_2}\right) \end{bmatrix}$$

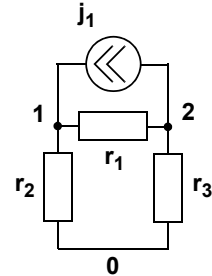


Node Voltage Method

- $u_{10} = (1/D) \cdot (j_1 / r_3)$
- $u_{20} = (1/D) \cdot (-j_1 / r_2)$
- $D = 1 / r_1 r_2 + 1 / r_1 r_3 + 1 / r_2 r_3$

- $r_1=2k\Omega, \quad r_2=0.5k\Omega, \quad r_3=1.5k\Omega$
- $j_1=10mA$

- $D=2.6667$
- $u_{10} = 10 / 1.5 \cdot 2.6 = 2.5 \text{ V}$
- $u_{20} = -10 / 0.5 \cdot 2.6 = -7.5 \text{ V}$



SPICE

- **Uses numerical techniques to solve nodal analysis of circuit**
 - Textual input to specify circuit & simulation commands
 - Text or graphical output format for simulation results
 - **Circuit elements:**
 - Resistors
 - Capacitors
 - Inductors
 - Independent sources (V, I)
 - Dependent sources (V, I)
 - Transmission lines
 - Active devices (diodes, BJTs, JFETS, MOSFETS)
 - **Analysis types:**
 - non-linear d.c., non-linear transient
 - linear a.c.
 - noise, temperature



SPICE

- **A wide variety of active device models**
- **Process parameter variation**
- **Effects of worst / best case & statistical spreads of process**
- **Design optimization**
- **Component Libraries**
- **Behavioral modelling**

- **With some versions of SPICE there can be algorithm related problems (e.g., failure to converge to a solution) which require a repertoire of fixes. With HSPICE many of these problems are avoidable.**
- **For effective application, the designer must have accurate MOSFET models for the process being used, and an understanding of the models and their parameters.**



Input Format

- **A SPICE file is made up of a series of statements**
- **Each statement is on one line, unless continued onto the next by starting it with '+' as the first character**
- **Each statement is made up of fields**
- **Fields are separated by ',', '=', '(' ')' or one or more spaces**
- **Fields consist of SPICE key words, SPICE symbols, names (alpha-numeric up to 16 characters) numbers (integer or floating point) or scale-factors**



Input Format

- The scale-factors recognized by SPICE are:
 - T = 1E12 G = 1E9 MEG = 1E6
 - K = 1E3 MIL = 25.4E-6 M = 1E-3
 - U = 1E-6 N = 1E-9 P = 1E-12 F = 1E-15
- Letters immediately following a number that are not scale factors are ignored, as are letters immediately following a scale factor, so 1K, 1000volts, 1KV, 1.0E3 are all the same
- Comment lines start with '*' or '\$' as the first character; comments on the same line as a SPICE statement must be at the end and be preceded by '\$' (surrounded by spaces)
- A SPICE file must start with a title statement and finish with an end statement
- The order of statements between start and end is arbitrary



SPICE Syntax

- Description of the circuit - every element has name and every node has label
- The general SPICE syntax for a circuit element is as follows:
 - NAME node1 node2 ... nodeN <model ref> value <parameters>
- The simplest syntax for each element in the above circuit is:
 - Resistor
 - Rxxx n1 n2 value
 - Capacitor
 - Cxxx n1 n2 value
 - MOSFET
 - Mxxx nd ng ns nb mname L=val W=val
 - + <AD=val> <PD=val> <AS=val> <PS=val>



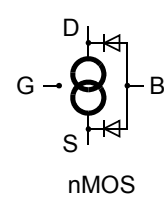
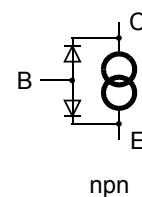
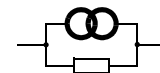
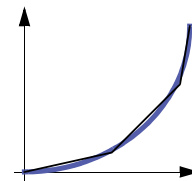
SPICE Syntax

- **Element identifiers xxx are chosen by the designer and may be numbers or letters**
- **Nodes may be identified by unique numbering; node number 0 is reserved for ground. Alternatively, nodes may be identified by unique names, with gnd reserved for ground.**
- **Note the sequence of MOSFET terminals: drain, gate, source, bulk. L is length, W is width. AD, AS, PD, PS are drain/source areas & perimeters used for estimating capacitances. mname must correspond to a defined model with this identifier.**



Non-linear components

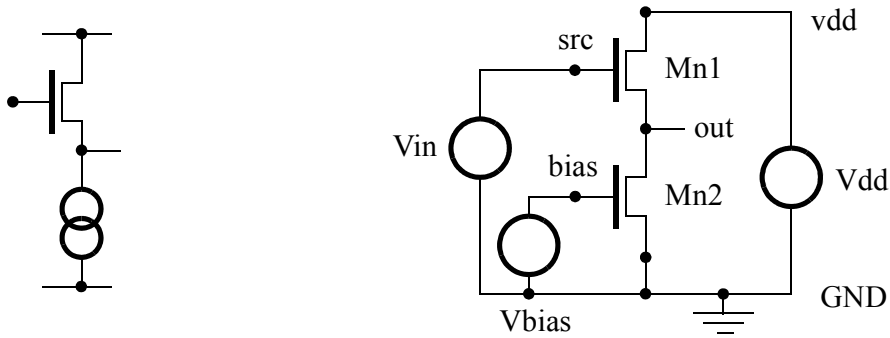
- **Non-linear models are replaced with a set of linear models**
 - **Newton Raphson algorithm**
 - **working point determines which of the linear models is used**
- **Diodes**
 - **resistor + voltage controlled current source**
- **Transistors**
 - **different models for small and large signals**
 - **bipolar – diodes + current controlled current source**
 - **diodes – pn-junctions**
 - **MOS – diodes + voltage controlled current source**
 - **diodes – substrate**





Example – SAQ Circuit

- The current sink has been replaced by a transistor and the circuit has been annotated as follows:



Example – SAQ Circuit

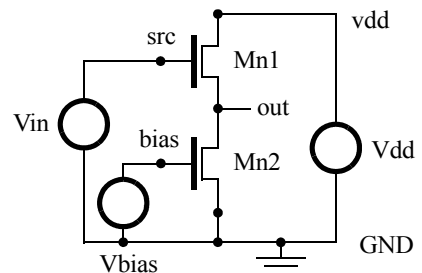
*single transistor source-follower - dc sweep of voltage I/O - no body effect
 *file:fol-2.hsp replace ideal current sink by single transistor current sink

```
*parameters
.PARAM ref=2.0
*circuit net list
Vdd vdd GND DC 5.0
Vin src GND DC 0.0
Vbias bias GND DC ref
```

```
*assume twin-tub or p-well process so Vbs=0 can be designed
Mn1 vdd src out out nmos11 W=8 L=8
Mn2 out bias GND GND nmos11 W=2 L=16
```

```
*options & analysis
.OPTIONS POST=2 SCALE=1U
.DC Vin 0 5 0.1
*output
.plot DC v(src) v(out) $cp. Lect1-4.vwg slide No 9
```

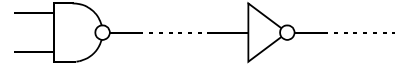
```
*mosfet models - level 1 included from separate file
.INC 'l1typ.inc' $Note this is lower case L 1 ...
.END
```



Example #2 – NAND gate with load Hands-on exercise

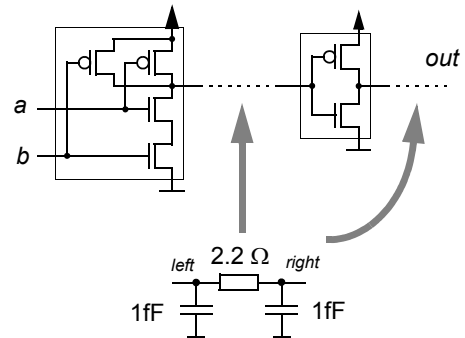
- NAND gate loaded with an inverter**

- 180 nm technology
- wire length - 10 μm



- Modeling the system**

- gate models
 - NAND gate - 2 nMOS & 2 pMOS transistors
 - inverter - 1 nMOS & 1 pMOS transistors
 - $L = 180 \text{ nm}$ & $W \geq 360 \text{ nm}$
- wire model
 - Π -model - 1 resistor & 2 capacitors
 - 10.0-0.36 μm \rightarrow 2.2 Ω & 2 fF



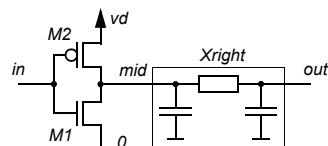
Example #2 – NAND gate with load

- Wire model**

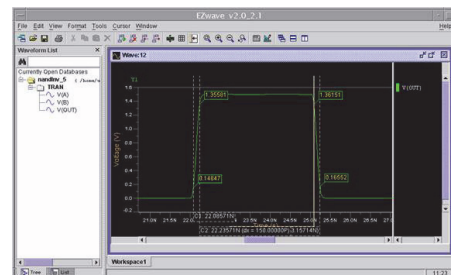
```
.SUBCKT wire left right
R1 left right 2.2
C1 left 0 1f
C2 right 0 1f
.ENDS
.end
```

- Inverter with wire**

```
.INCLUDE wire.cir
.MODEL nM NMOS LEVEL=2
.MODEL pM PMOS LEVEL=2
Vdd vd 0 1.5
M1 mid in 0 0 nM L=180n W=360n
M2 mid in vd vd pM L=180n W=936n
Xright mid out wire
Vin in 0 pulse(0 1.5 0 10p 10p 5n 10n)
.TRAN 1ps 50ns
.plot TRAN v(in) v(out)
.end
```



Add the NAND gate and get slopes right...





How to optimize? Complexity of the task...

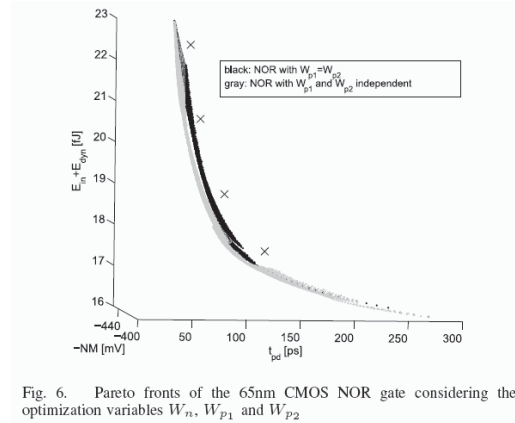
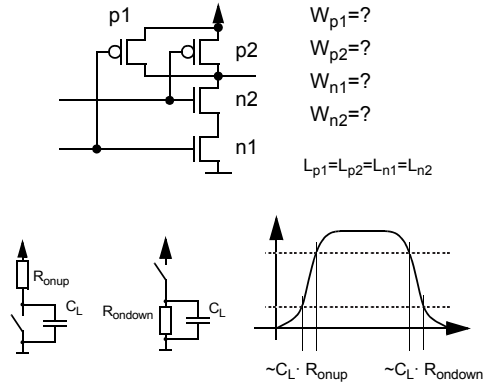


Fig. 6. Pareto fronts of the 65nm CMOS NOR gate considering the optimization variables W_n , W_{p1} and W_{p2}

- Matthias Blesken, Ulrich Rückert, Dominik Steenken, Katrin Witting, Michael Dellnitz, "Multiobjective Optimization for Transistor Sizing of CMOS Logic Standard Cells Using Set-Oriented Numerical Techniques." The 27th Norchip Conference, Trondheim, Norway, Nov. 2009.



How to optimize? Approximation... a.k.a. Why NAND?

- Mobility –
 $\mu_n = 1250 \text{ cm}^2 / \text{V sec}$ & $\mu_p = 480 \text{ cm}^2 / \text{V sec}$
 - depends on the actual technology
- $R \sim \mu^{-1}$ & $R \sim L w^{-1}$ (L-constant)

