Nanoelectronic Circuit Design

Design, simulation and analysis in SPICE

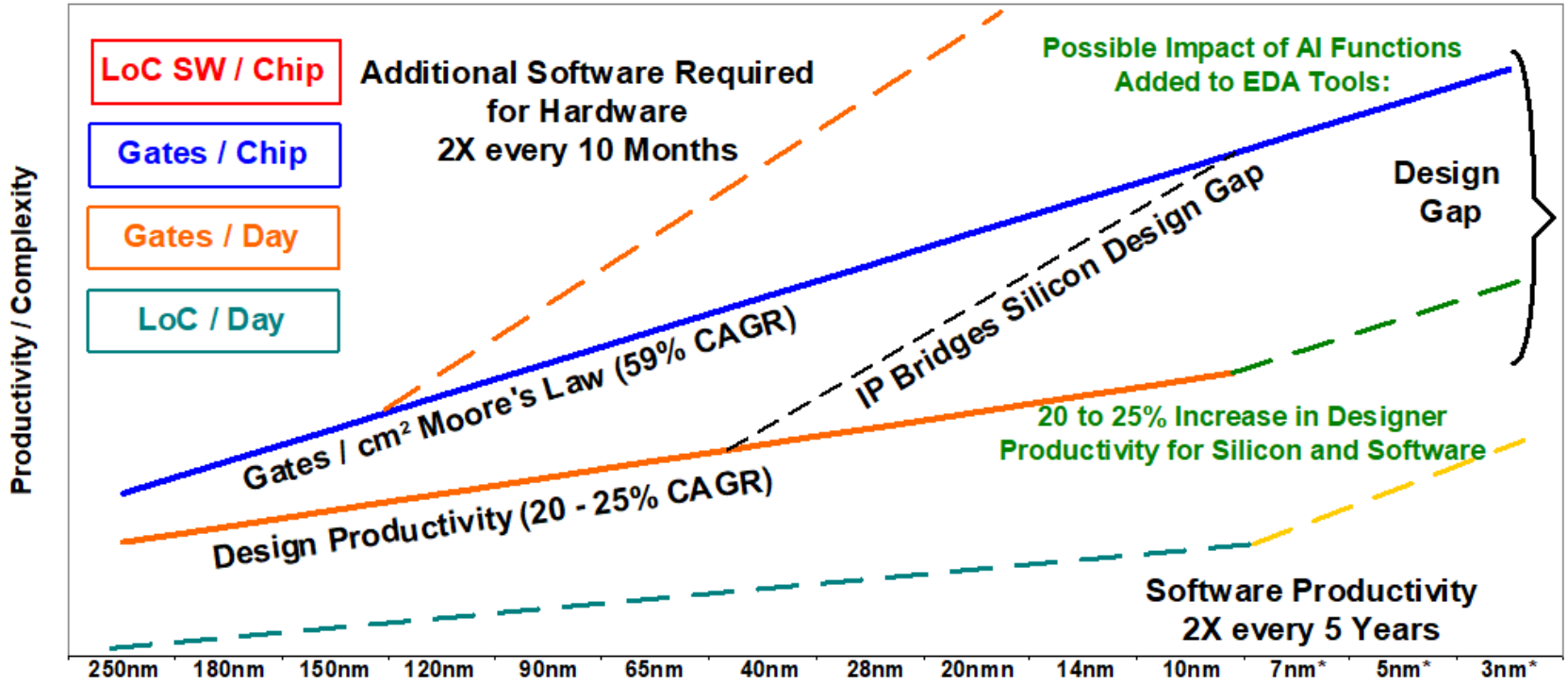
Ramon Canal

Dept. Arquitectura de Computadors (DAC)

UPC-Barcelona Tech

Based on the slides made by Prof. Enric Pastor (DAC, UPC)

The Design Problem

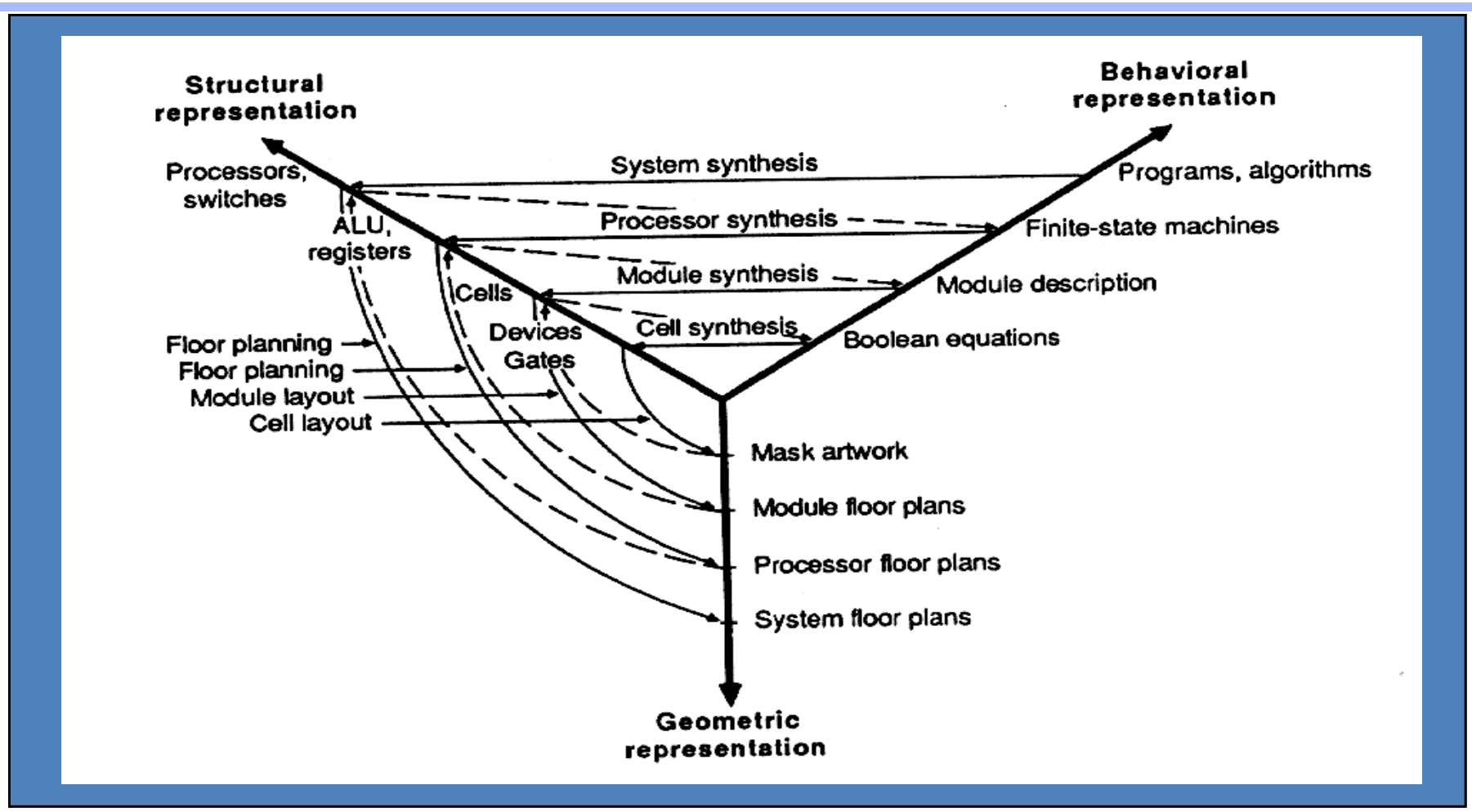


Source: Semico Research Corp. June, 2018

*Note: Further explanation of the graph above is provided in Semico's report ***Silicon and Software Design Costs*** (SC103-18).

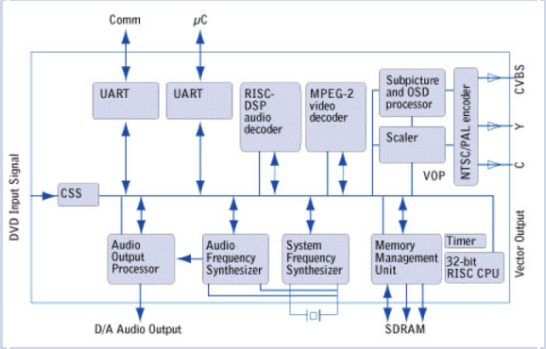
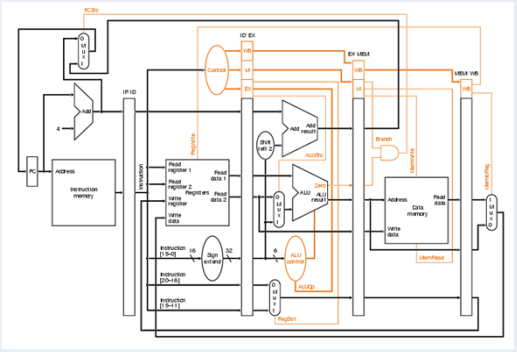
A growing gap between design complexity and design productivity

Design Methodology

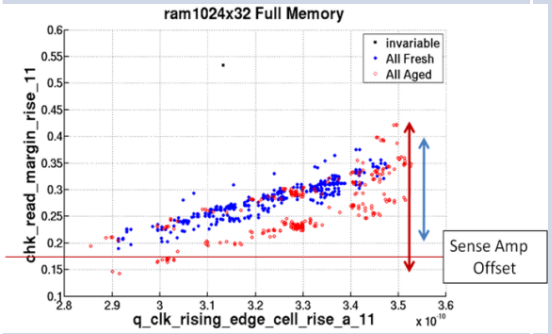
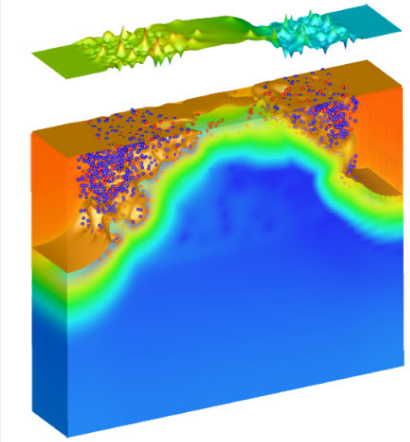


- Design process traverses iteratively between three abstractions: behavior, structure, and geometry
- More and more automation for each of these steps

Tools across Design Space

Design Level		Tools	Simultation Type
CPU/SoC		Custom built in HLL (Simplescalar, Gem5, Marssx86, Multi2Sim, GPGPU, ...) Input: Block timing, benchmarks [extra power/temperature models (i.e. Wattch, McPAT, CACTI, or self-made)	Functional simulation Output: performance characterization
Circuit Level (functional [+ timing]) *Bigger blocks, whole CPU*		VHDL, verilog Input: circuit [+ circuit timing]	Logic Output: Behavior, Results [timing].

Tools across Design Space (cont.)

Design Level		Tools	Simultation Type
Circuit Level (analog) *Usually small blocks*		SPICE Input: Compact Model	Electrical simulation Output: characterization (delay, power, etc.)
Device (i.e. Transistor)		TCAD, atomistic simulators or similar Input: physical description of materials	Particle Physics Output: Compact Model

Content

- Overview of SPICE
- SPICE Description:
  - Nodes
  - Basic components
  - Transistors: dimensions and capacity
  - Modular Design
  - Definition of inputs
  - Power measurement
  - Buses
- Technology: BSIM 28nm
- Example: an inverter, Brent-Kung 32 bit adder

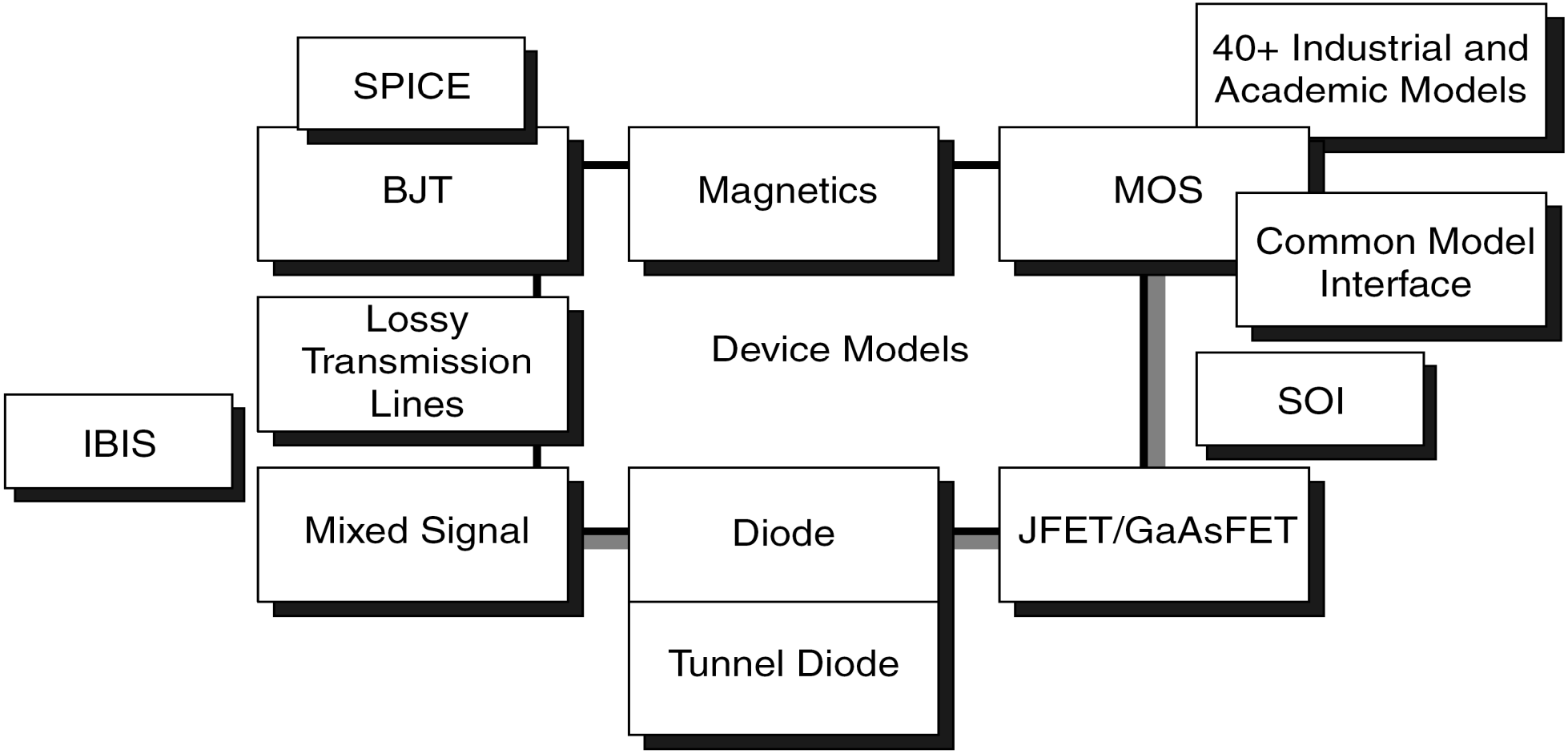
Content

- Simulation Program with Integrated Circuit Emphasis
- General purpose analog circuit simulator

- Used in IC and board-level design for check of integrity of circuit designs and prediction of circuit behavior
- Developed at Electronics Research Laboratory of the University of California, Berkeley (1973)

- SPICE simulation is industry-standard for verification of circuit operation at transistor level before manufacturing

Modeling Technologies

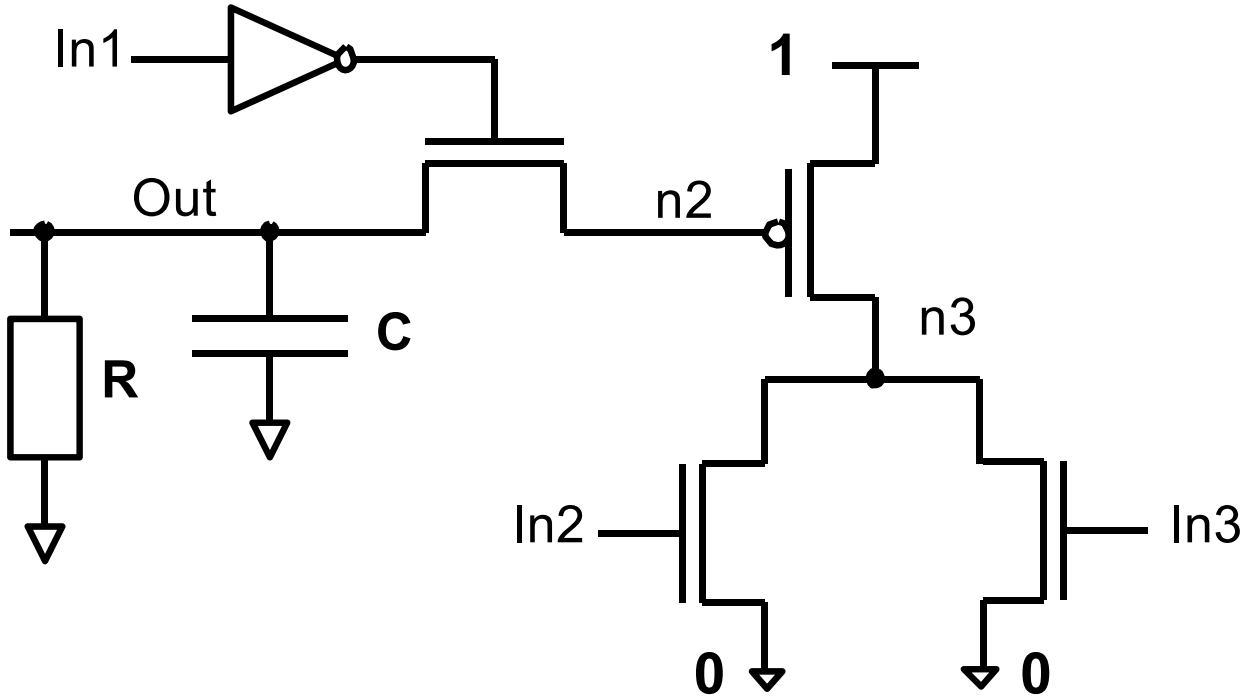


SPICE: electrical simulator

- SPICE supports several types of components:
  - Resistance, capacitors, voltage sources...
  - Transistors NMOS, PMOS
  - Buses
  - Modular description
- Behavior is simulated as nonlinear differential equations:
  - Simple models for resistances and capacitances
  - Several different models for transistors
- SPICE simulates by discretizing time
  - Solving by implicit integration methods, Newton's method and sparse matrix techniques
  - May not converge or reduce simulation intervals
  - No convergence may mean a wrong design

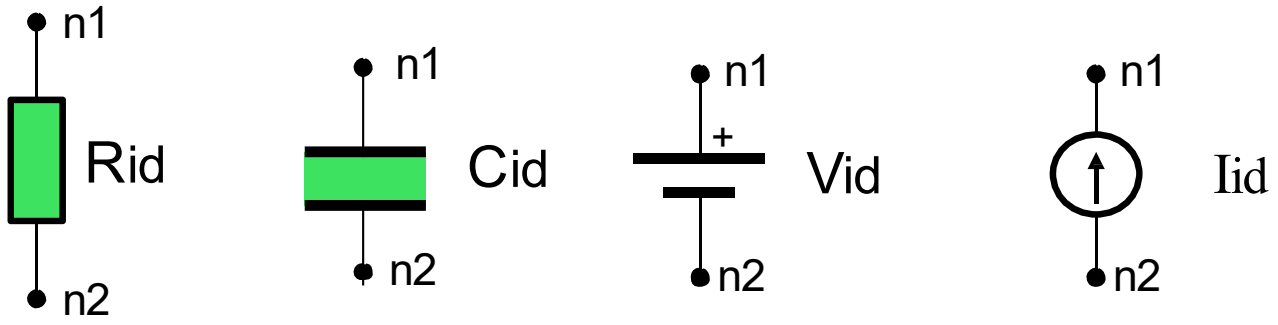
SPICE description: Nodes

- SPICE analyzes networks of nodes:
  - Each node can be connected to other ones through components
  - Each node has a unique name or number
  - Nodes do not have a special direction (current)
  - Special nodes are **1** (Vdd) and **0** (Gnd)



SPICE description: Components

- The first character of the description defines the component type
- Most common components are:
  - Resistance **R**id node1 node2 value*(ohms)*
  - Capacitance **C**id node1 node2 value *(farads)*
  - Voltage source **V**id node1 node2 value *(volts)*
  - Current source **I**id node1 node2 value *(ampers)*



SPICE description: Components

- Other components:

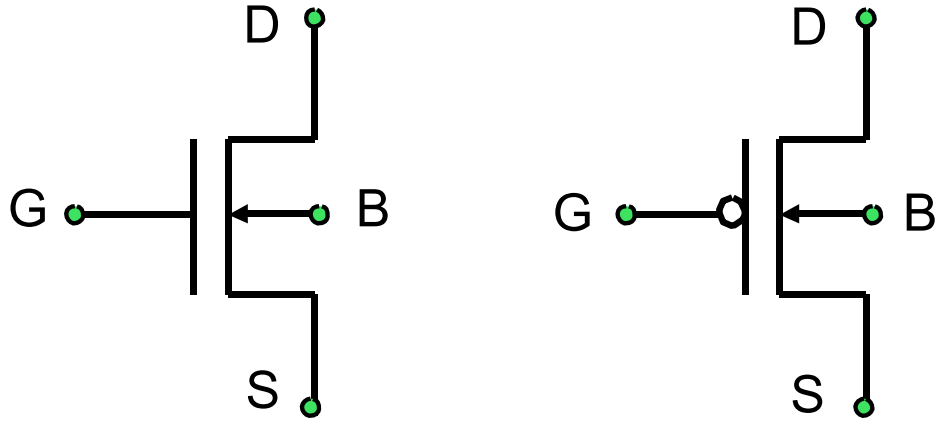
– Diode	**D**id	– Bipolar Transistor	**Q**id
– Linear Inductor	**L**id	– Transmission Line	**T**id, **U**id, **W**id
– JFET or MESFET	**J**id		

– Subcircuit **X**id <subcircuit parameters>

SPICE description: Transistors

- We can model a plethora of devices
- We will concentrate on MOS type transistors:

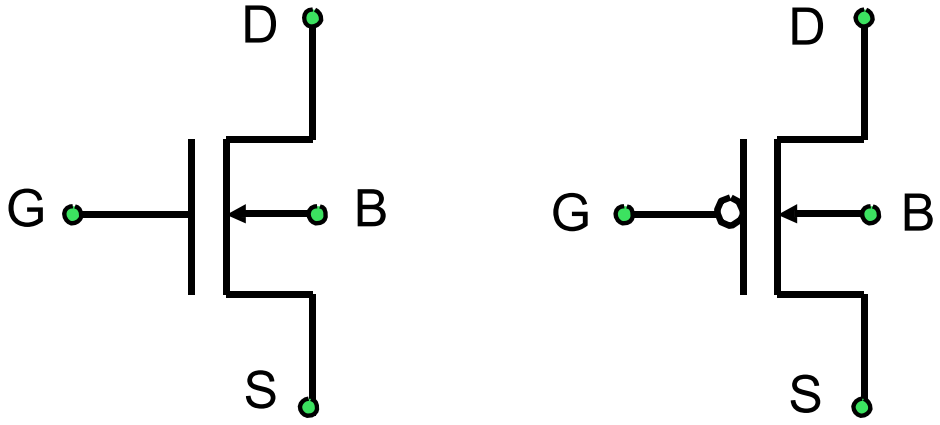
  M*id* *nodeD nodeG nodeS nodeB type*
- Description:
  - ***nodeD***: drain ***nodeG***: gate
  - ***nodeS***: source ***nodeB***: bulk
  - **type**:NMOS / PMOS



SPICE description: Transistors

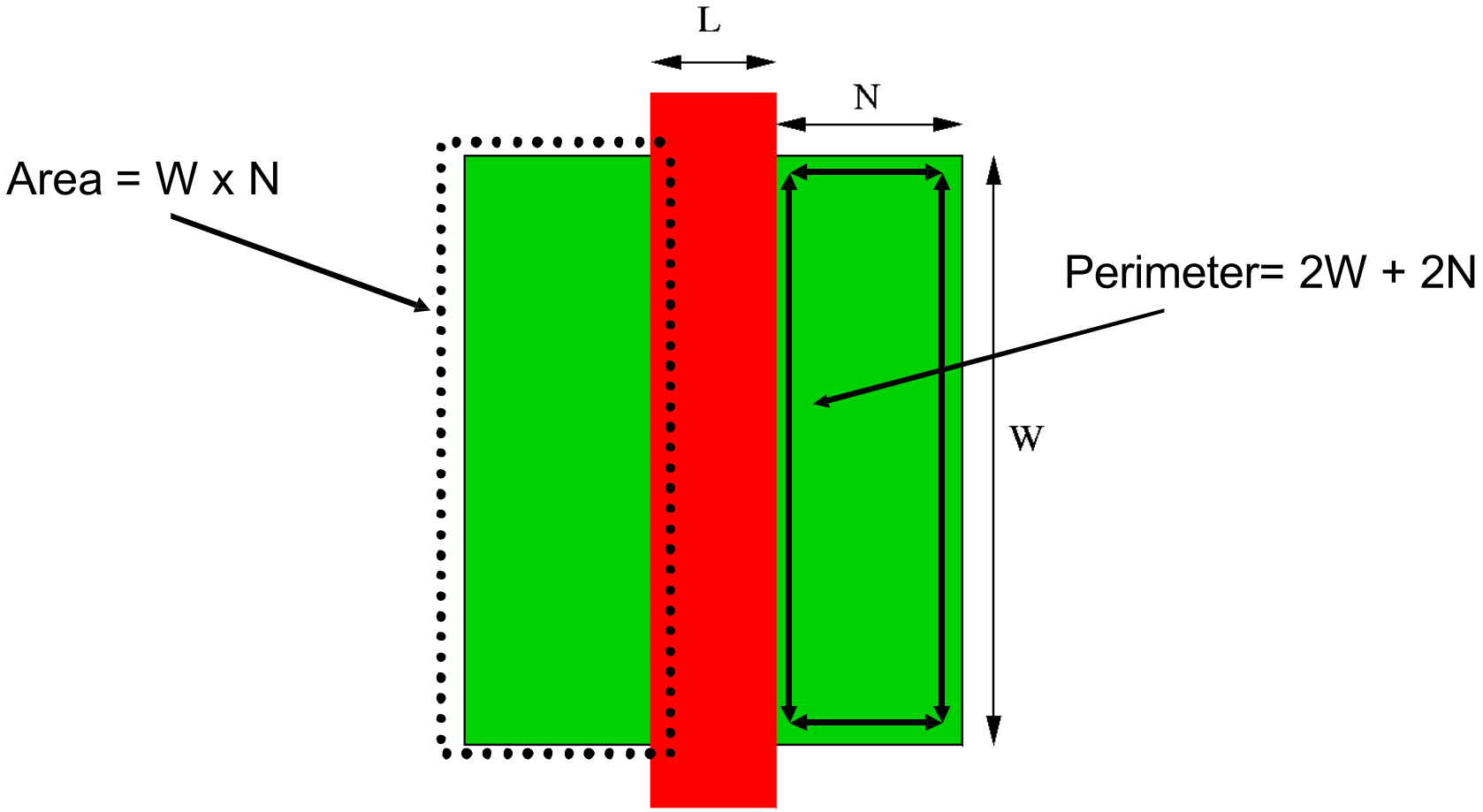
- Additional parameters:
  - ***W***: channel width ***L***: channel length
  - ***PD / PS***: perimeter of the Drain/Source
  - ***AD / AS***: Area of the Drain/Source

M*id nD nG nS nB type W= L= PD= AD= PS= AS=*

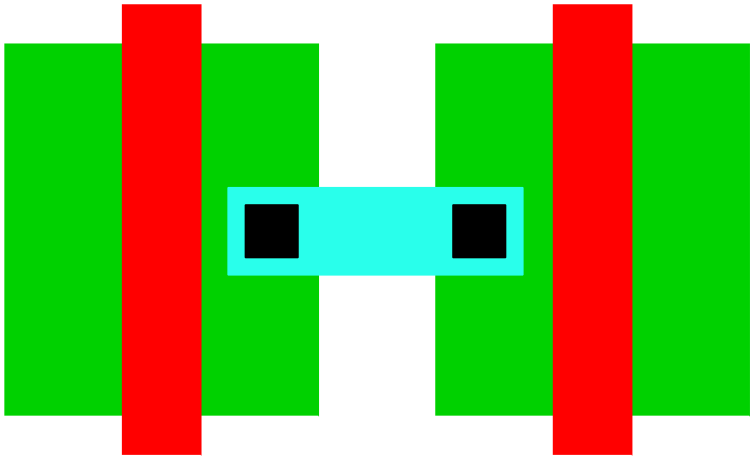


SPICE description: Transistors

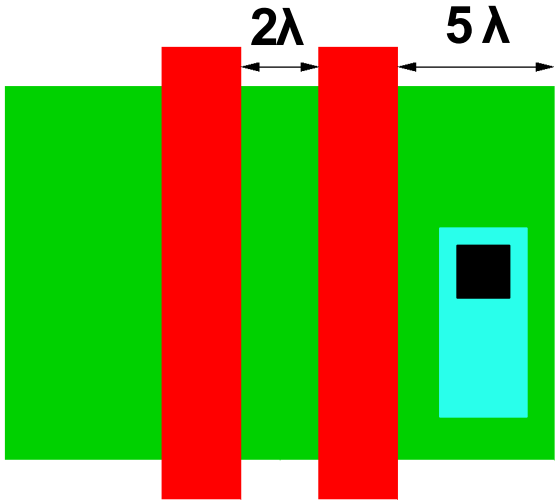
- How do we compute the perimeters and areas in a transistor?

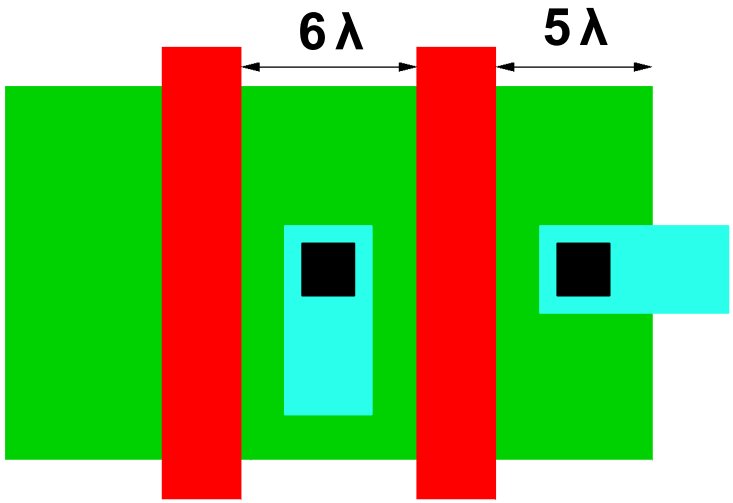


SPICE description: Transistors



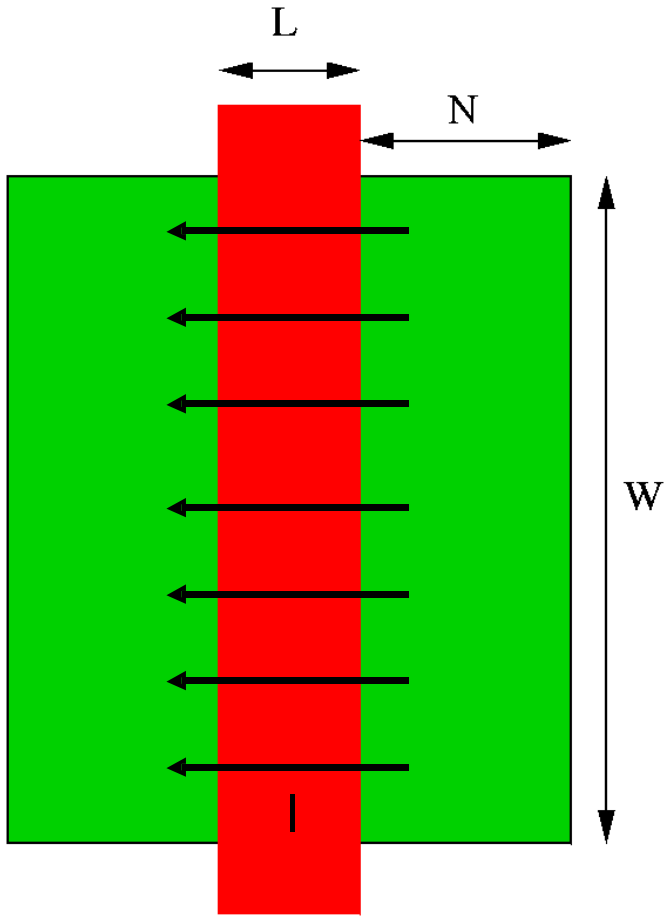
- 2λ difussion w/o contact
- 6λ difussion w contact
- 5λ difussion w contact at the end of the structure





SPICE description: Transistors

- L and W determine the performance of the transistor:
  - Resistance ~ L/W
  - L: higher resistance
  - W: smaller resistance
- PMOS transistor:
  - Smaller current than NMOS
  - 2-3 times slower



SPICE description: Modules

- Subcircuit definition:

```
.SUBCKT Name NodeList

*Definition of content

.ENDS Name
```

- Highly recommended:

```
NodeList: InputList OutputList 1 (Vdd) 0 (Gnd)
```

- Subcircuit instantiation:

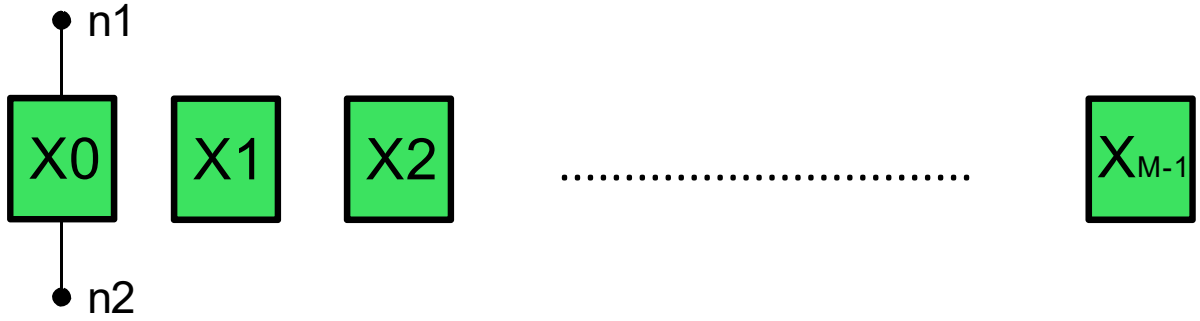
```
Xid NodeList Name
```

SPICE description: Multiple Modules

- The M (multiply) parameter:

```
* Definition Val copies of SubcircuitName connected
in parallel
  Xid NodeList SubcircuitName M=Val
```

- Useful for large parallel structures, such as:
  - Memories (register file, caches, etc.)



SPICE description: Inputs

- Definition of inputs:
  - Signal wave (chronogram)
  - List of points in the wave (time-value pairs)
  - Lineal interpolation between consecutive points

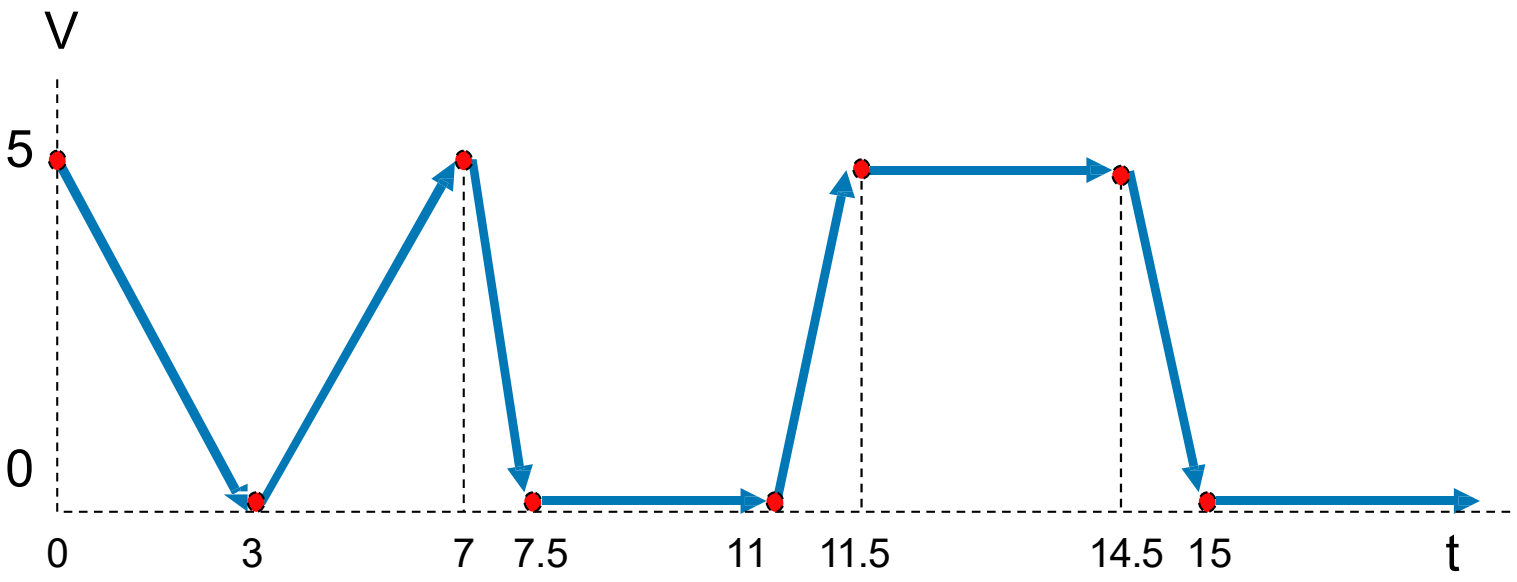
```
Vnode Node 0  pwl (Time1 Value1 Time2 Value2 ...)
```

- Intricacies:
  - Define the value for initial (0) time
  - $Time_x < Time_{x+1}$ (no 0-delay transitions allowed!)
  - Each edge needs 2 points to define the transition slope

SPICE description: Inputs

- Example:

```
Vnode Node 0  pwl (   0ns    5      3ns    0
+                     7ns    5      7.5ns  0
+                     11ns   0      11.5ns 5
+                     14.5ns 5      15ns   0)
```



SPICE description: Pulses

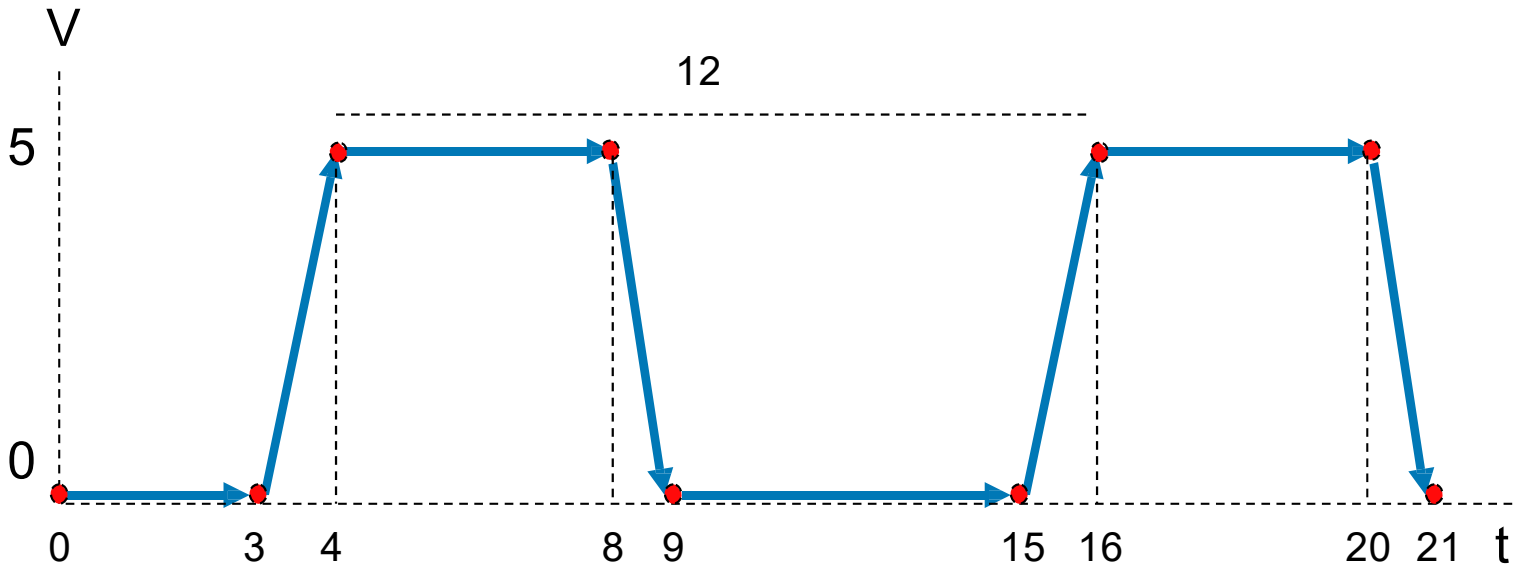
- Description of periodic signals:
  - V1 Initial value
  - V2 Value during the pulse
  - TD Initial delay of the pulse
  - TR Raise time
  - TF Fall time
  - PW Pulse width
  - PER Pulse period

```
Vnode Node 0  PULSE (V1  V2  TD  TR  TF  PW  PER)
```

SPICE description: Pulses

- Example:

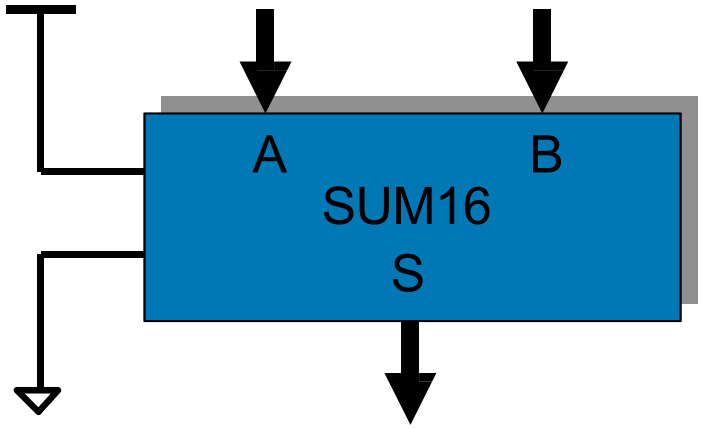
```
Vnode Node 0   PULSE ( 0      5
+                      3ns    1ns    1ns    4ns
+                      12ns)
```



SPICE description Power Delivery

- Voltage source:
  - Defines the electrical values for *Vdd* and *Gnd*
  - Same common values for all the system.

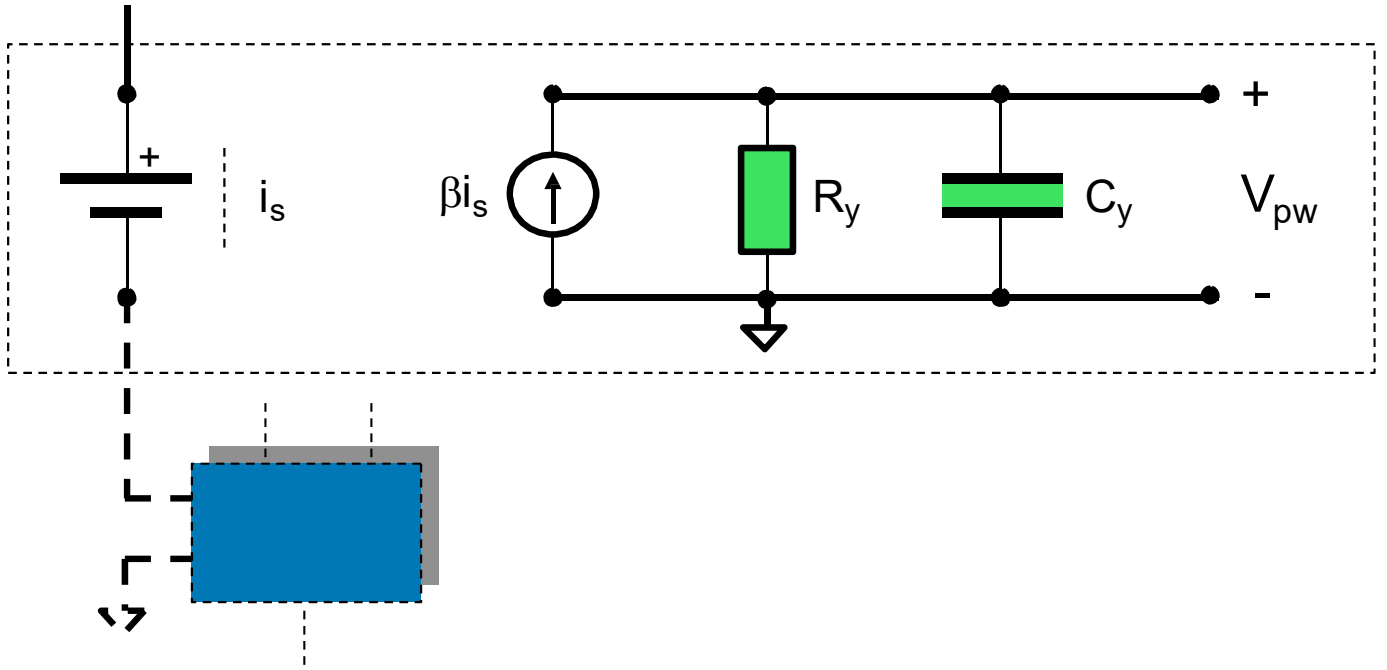
```
* 5v between vdd (node 1) and gnd (node 0)
VCC 1 0 DC 5V
XSUM   A15 A14 ... A2 A1 A0 B15 B14 ... B2 B1 B0
+      S15 S14 ... S2 S1 S0 1 0 sum16
```



SPICE description: Power Delivery

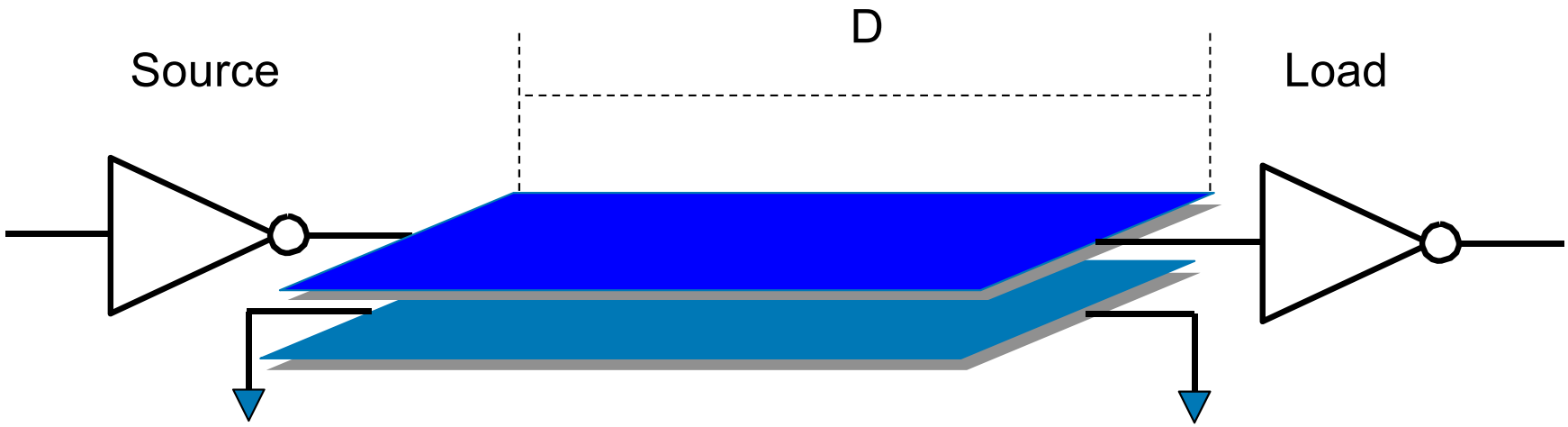
- Voltage source:
  - Current sensor: current + total accumulated power

```
XPS   Vsupp Vpw 1 0 Pmeter
XSUM    A15 A14 ... A2 A1 A0   B15 B14 ... B2 B1 B0
+       S15 S14 ... S2 S1 S0 Vsupp 0 sum16
```



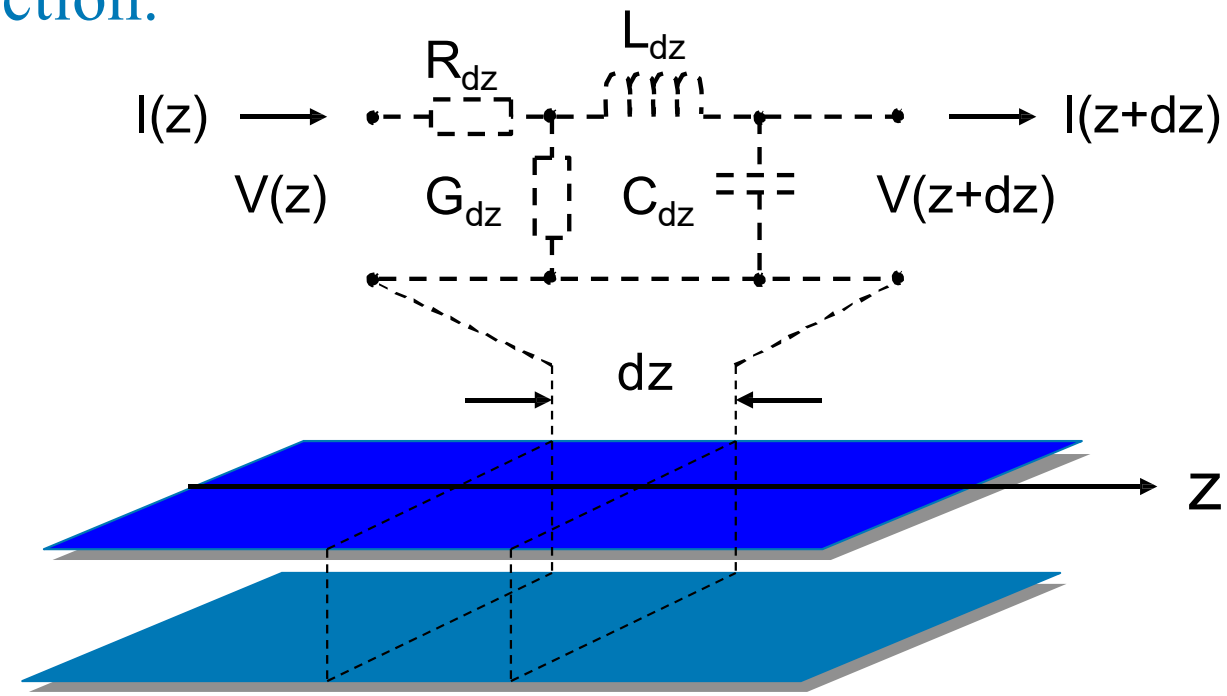
SPICE description: Interconnects

- Data transmission:
  - A conductor over a substrate
- Communication models:
  - Transmission line analysis
  - Lumped-element analysis



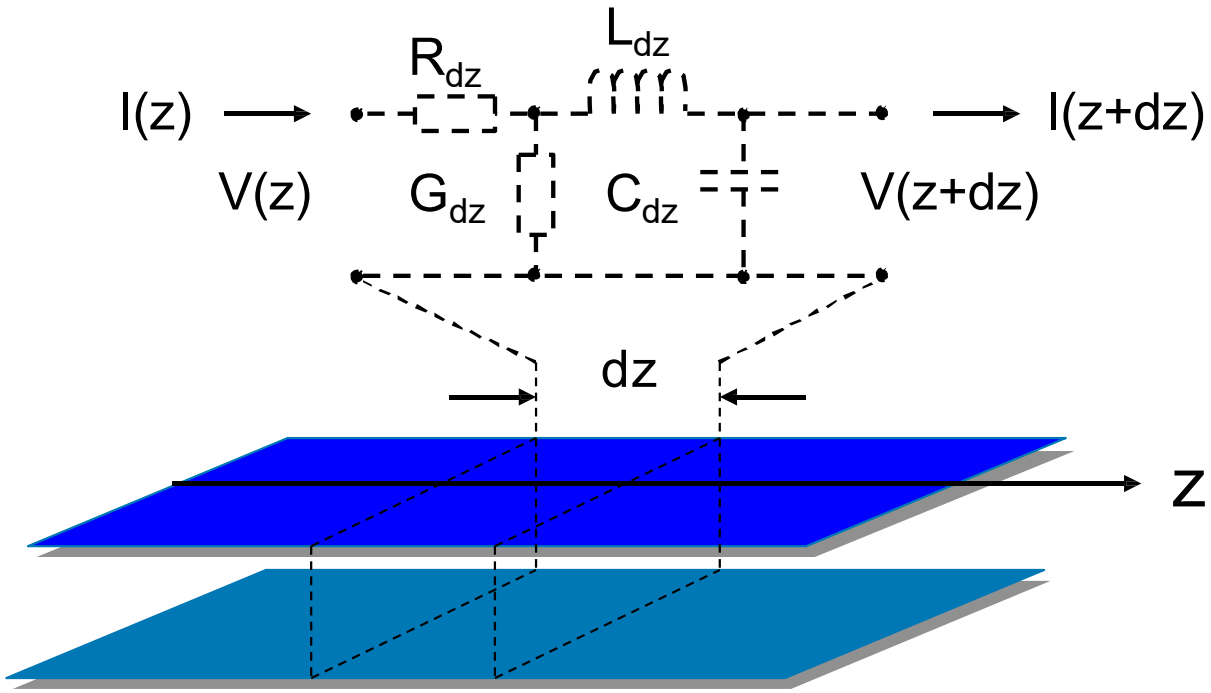
SPICE description: Transmission line

- Segment analysis:
  - Resistance (R_{dz}), capacity (C_{dz}), inductance (L_{dz}) and leakage (G_{dz}) per unit of length (i.e. nanometer) in the connection.



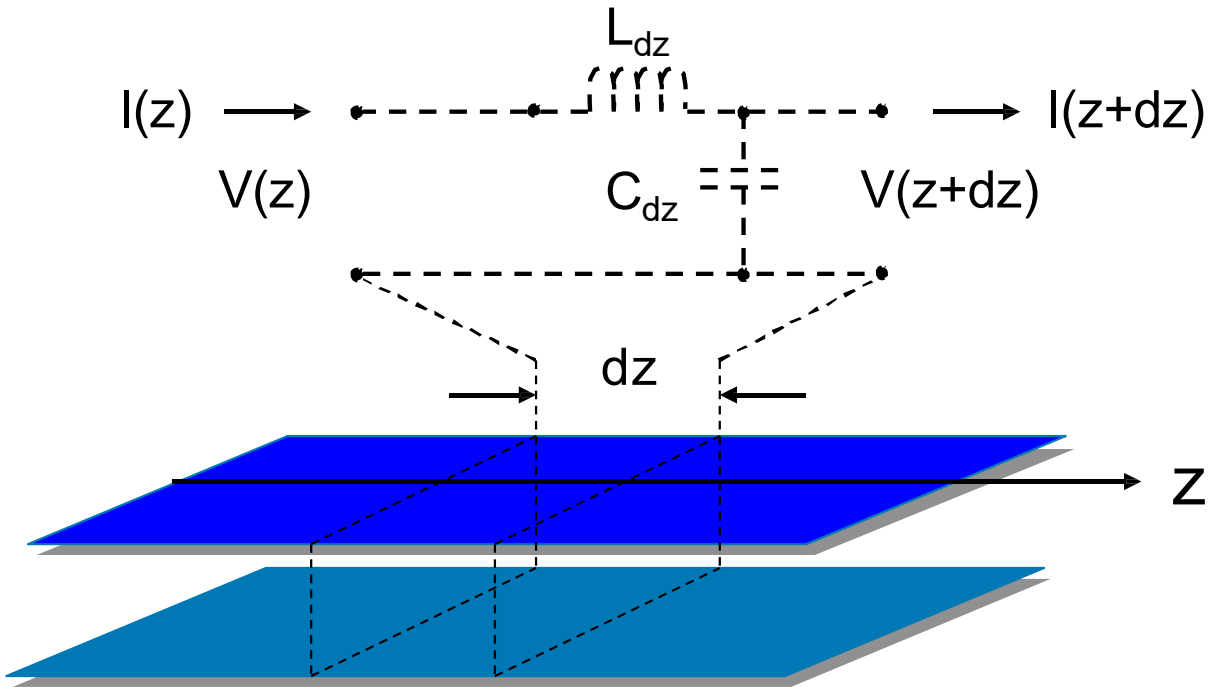
SPICE description: Transmission line

- Channel with losses:
  - Resistance (R_{dz}) and leakage (G_{dz}) cause a reduction of current and a drop of voltage.



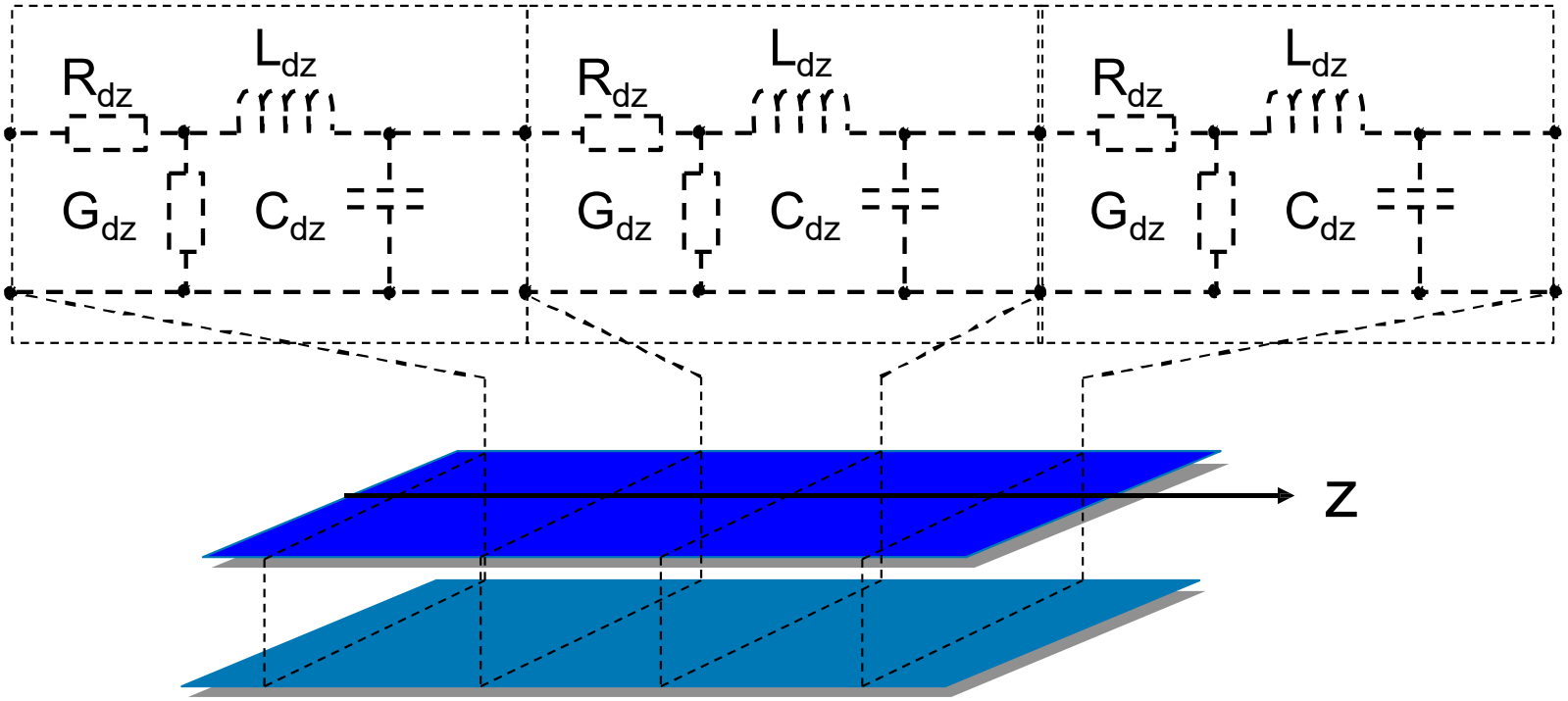
SPICE description: Transmission line

- Channel without losses:
  - We need to model just capacity (C_{dz}) and inductance (L_{dz}). This affects ***only*** delay.



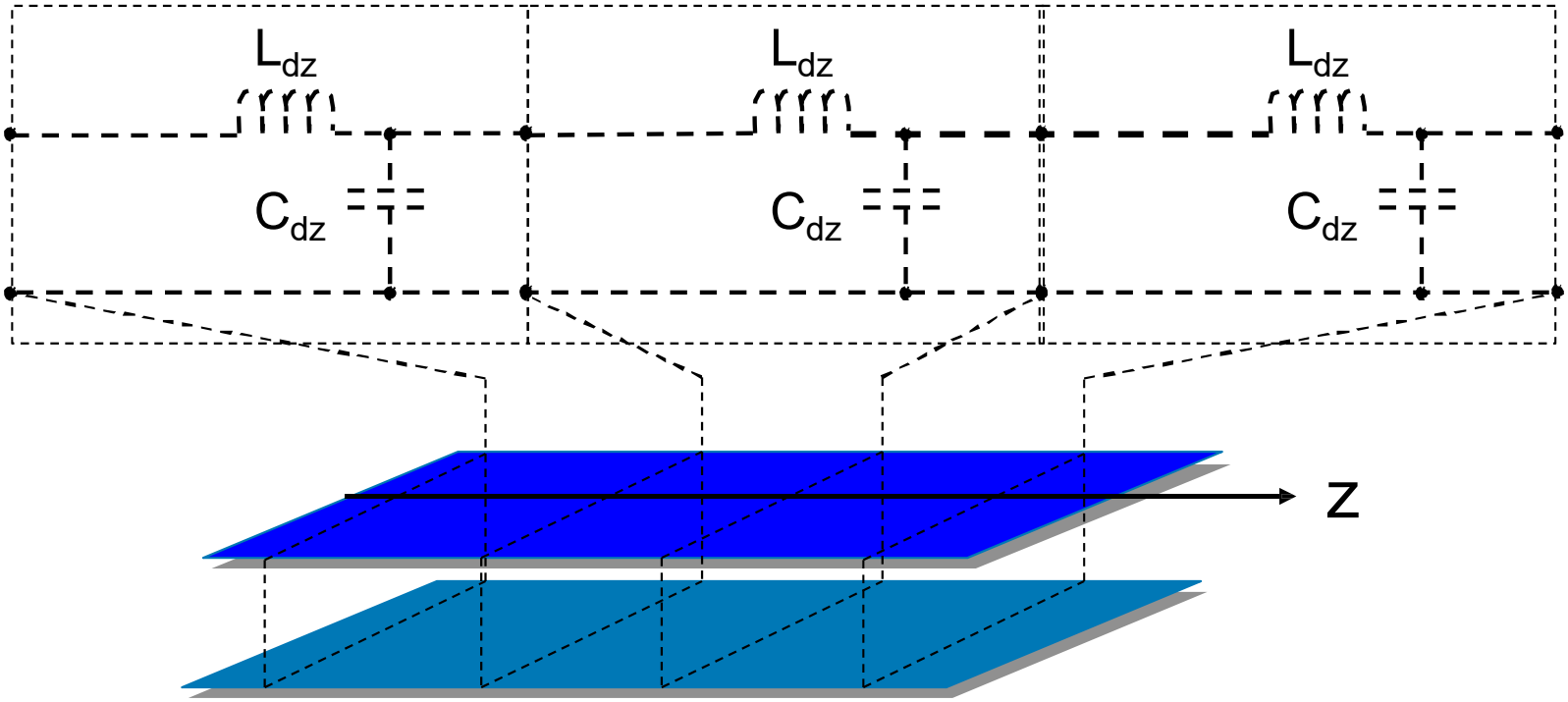
SPICE description: Lumped-Element

- Simulation of a channel with losses:
  - Use multiple transmission elements

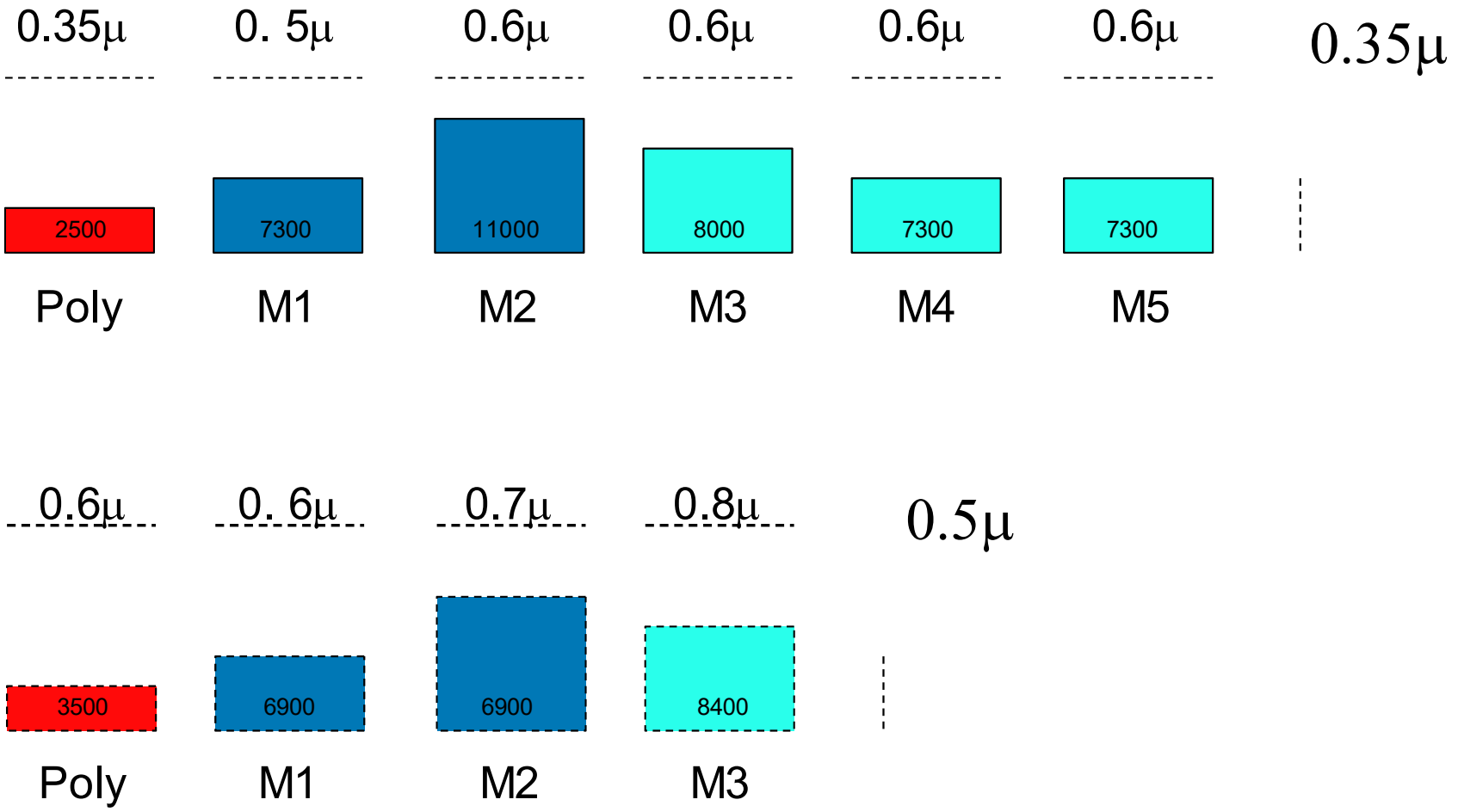


SPICE description: Lumped-Element

- Simulation of a channel without losses :
  - Use multiple transmission elements



SPICE description: 0.35µ/0.5µ Technology



SPICE description: 0.35µ Technology

- Resistance parameters:

Poly Sheet R	10 - 30 Ω/•
N+ Sheet R	10 - 30 Ω/•
P+ Sheet R	10 - 30 Ω/•
M1-M5 Sheet R	35 - **55** - 75 mΩ/•
High poly Sheet R	800 - **1000** - 1200 mΩ/•
Contact R	2 - 15 Ω/cnt
Via R	1 - 3 Ω/cnt

SPICE description: 0.35µ Technology

- Capacity parameters:

M1 to DIFF	**0.036** fF/ µm^2
M1 to POLY	**0.047** fF/ µm^2
M1 to SUB	**0.033** fF/ µm^2
M2 to SUB	**0.012** fF/ µm^2
M3 to SUB	**0.008** fF/ µm^2
M4 to SUB	**0.005** fF/ µm^2
M5 to SUB	**0.004** fF/ µm^2
POLY to SUB	**0.126** fF/ µm^2
POLY	**4.93** fF/ µm^2

SPICE description: 0.35µ/0.5µ Technology

- Resistance parameters:

Poly Sheet R	10 Ω/•	30 Ω/•
N+ Sheet R	10 Ω/•	90 Ω/•
P+ Sheet R	10 Ω/•	115 Ω/•
M1-M5 Sheet R	55 mΩ/•	85 - 55 mΩ/•
High poly Sheet R	1000 mΩ/•	- - - -
Contact R	2 - 15 Ω/cnt	40 - 80 Ω/cnt
Via R	1 - 3 Ω/cnt	1 - 3 Ω/cnt

SPICE description: 0.35µ/0.5µ Technology

- Capacity parameters:

M1 to DIFF	**0.036** fF/ µm²	**0.031** fF/ µm²
M1 to POLY	**0.047** fF/ µm²	**0.049** fF/ µm²
M1 to SUB	**0.033** fF/ µm²	**0.031** fF/ µm²
M2 to SUB	**0.012** fF/ µm²	**0.011** fF/ µm²
M3 to SUB	**0.008** fF/ µm²	**0.007** fF/ µm²
M4 to SUB	**0.005** fF/ µm²	- - - -
M5 to SUB	**0.004** fF/ µm²	- - - -
POLY to SUB	**0.126** fF/ µm²	**0.12** fF/ µm²
POLY	**4.93** fF/ µm²	**2.56** fF/ µm²

Numbers

- Numbers can be
  - Integer
  - Floating point
  - Floating point with integer exponent
  - Integer or floating point with one scale factor
- Numbers can use
  - Exponential format
  - Engineering key letter format
  - Not both (1e-12 or 1p, but not 1e-6u)

Prefix	Scale Factor	Multiplying Factor
Tera	T	1e+12
Giga	G	1e+9
Mega	MEG or X	1e+6
Kilo	K	1e+3
Milli	M	1e-3
Micro	u	1e-6
Nano	n	1e-9
Pico	p	1e-12
Femto	f	1e-15
Atto	a	1e-18

Comments

* **** Parameters *****

Comments:

- First letter of line is asterisk (*) → whole line is comment
- Dollar sign ($) anywhere on the line → text after is comment

For example:

```
* <comment_on_a_line_by_itself>
```

-or-

```
<HSPICE_statement> $ <comment_following_HSPICE_input>
```

Comment statements can be placed anywhere in circuit description

Parameters and Expressions

```
.param Wn=2u L=0.6u
.param Wp='2*Wn'
```

- Definition of netlist parameters
- Parameter can be defined with expressions
- Definition can occur after use in elements
- Parameter names must begin with alphabetic character
- At redefinition last parameter's definition is used
- Expressions cannot exceed 1024 characters

Transient analysis with “Sweeps”

- Transient analysis simulates circuit in a specific time
- Simple syntax:

  .TRAN <Tstep> <Tstop>

  <Tstep>: time step

  <Tstop>: End time (duration) of simulation
- Also more complex commands possible

```
.TRAN 200P 20N SWEEP TEMP -55 75 10
```

  - *Time step: 200 ps, Duration: 20 ns*
  - *Multipoint simulation: temperature is swept from -55 to 70°C by 10°C steps*

PLOT statement

```
.plot ov1 [ov2 ... ovN]
```

- Generate a plot for including all variables ov1…ovN
- oVx can be:
  - V(n): voltage at node n.
  - V(n1<,n2>): voltage between the n1 and n2 nodes.
  - Vn(d1): voltage at nth terminal of the d1 device.
  - In(d1): current into nth terminal of the d1 device.
  - ‘expression’: expression, involving the plot variables above

Measure statement

.MEASURE <ana_type> <param_name> <meas_mode>

<param_name>: Parameter name

<Meas_mode> Measurement mode, e.g.:

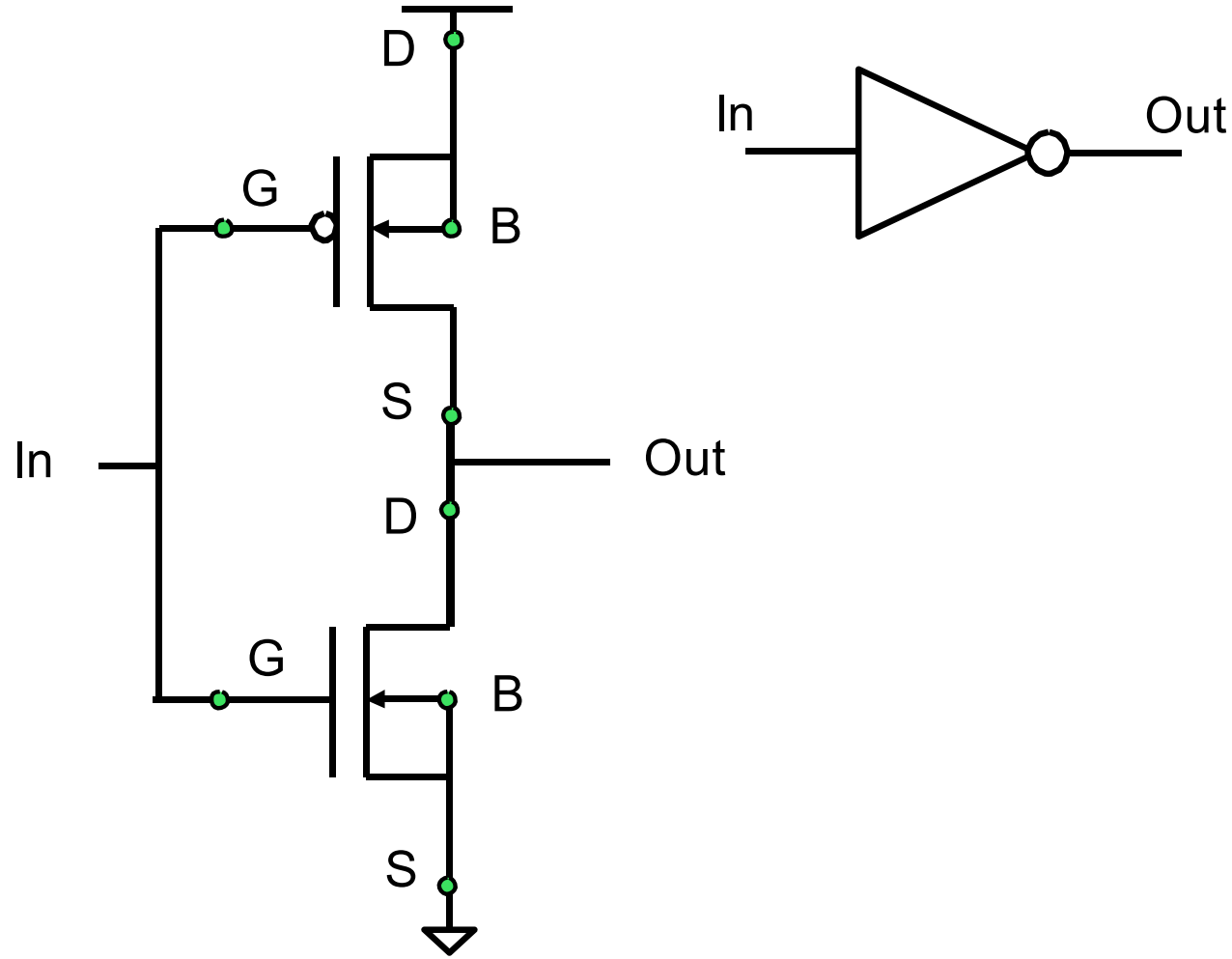
- Rise, fall, and delay
- Find-when
- Average, RMS, min, max, and peak-to-peak
- Integral evaluation
- Derivative evaluation

.MEASURE tran vin AVG V(nt1) from=0 to=1n

- Parameter name: vin
- Measurement type: Average
- Value: Voltage of net n1

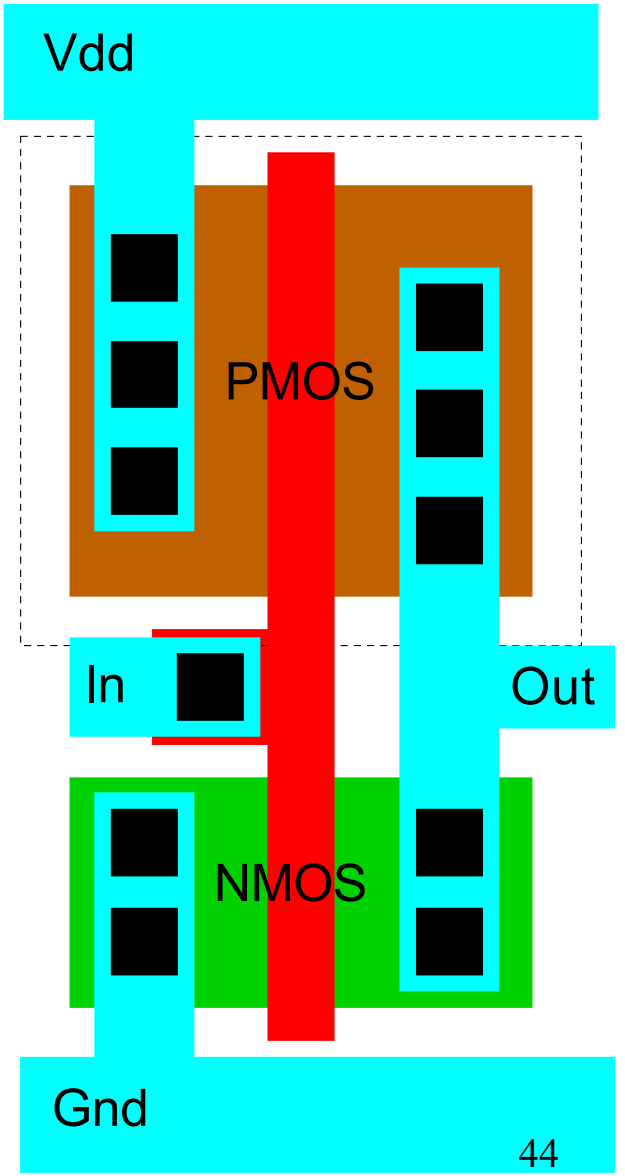
Example: inverter

- Schematic and gate:



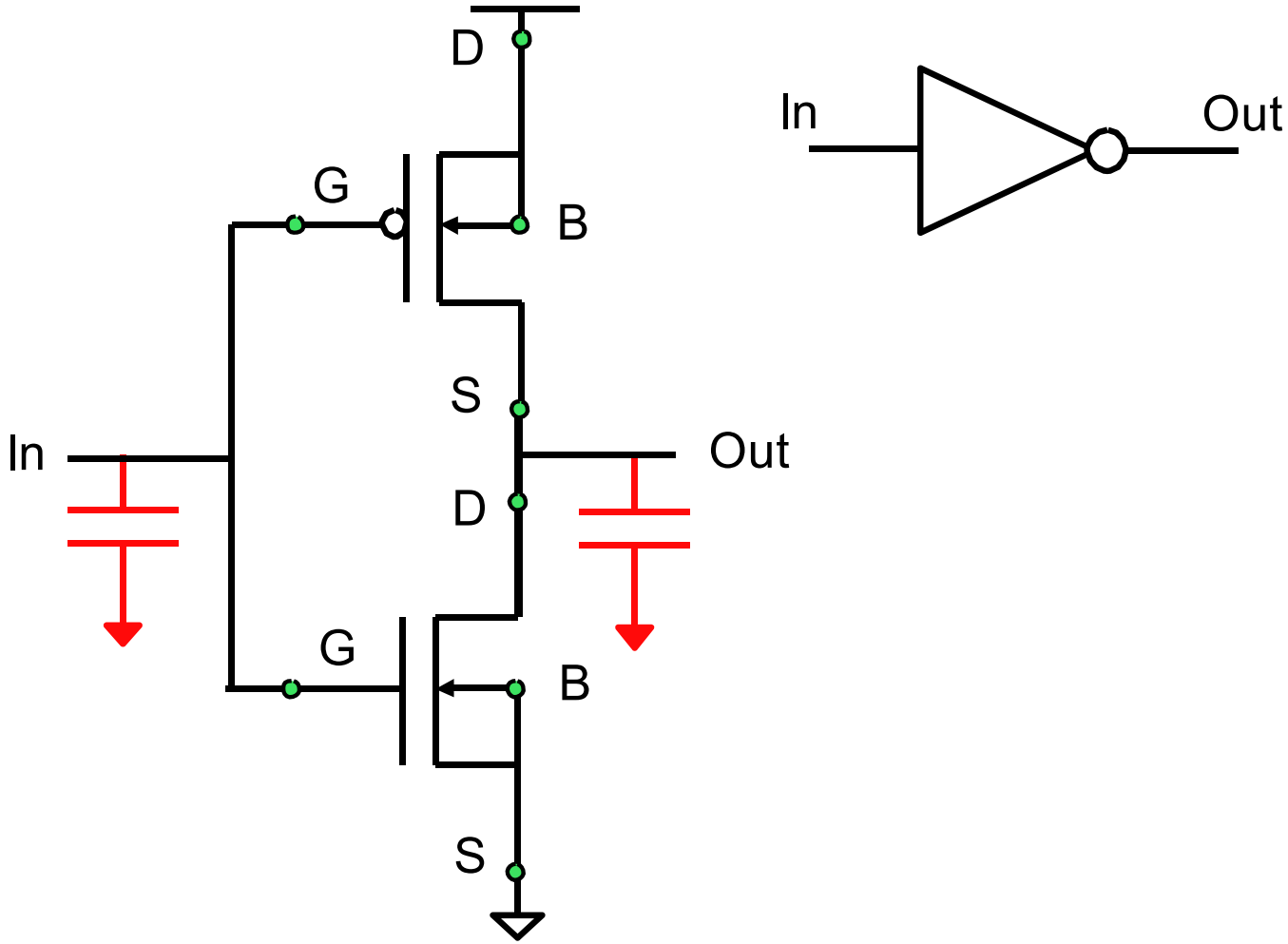
Example: inverter (@28nm) Layout

- L/W values:
  - L_p = 56n (2λ) W_p = 448n (16λ)
  - L_n = 56n (2λ) W_n = 224n (8λ)
- Areas: (N=6λ)
  - $AD_p = AS_p$ = 448n * 6(28n) = 75.3f
  - $AD_n = AS_n$ = 224n * 6(28n) = 37.6f
- Perimeters:
  - $PD_p = PS_p$ = 2(448n + 168n) = 1.23u
  - $PD_n = PS_n$ = 2(224n + 168n) = 784n



Example: inverter

- Schematic and gate:

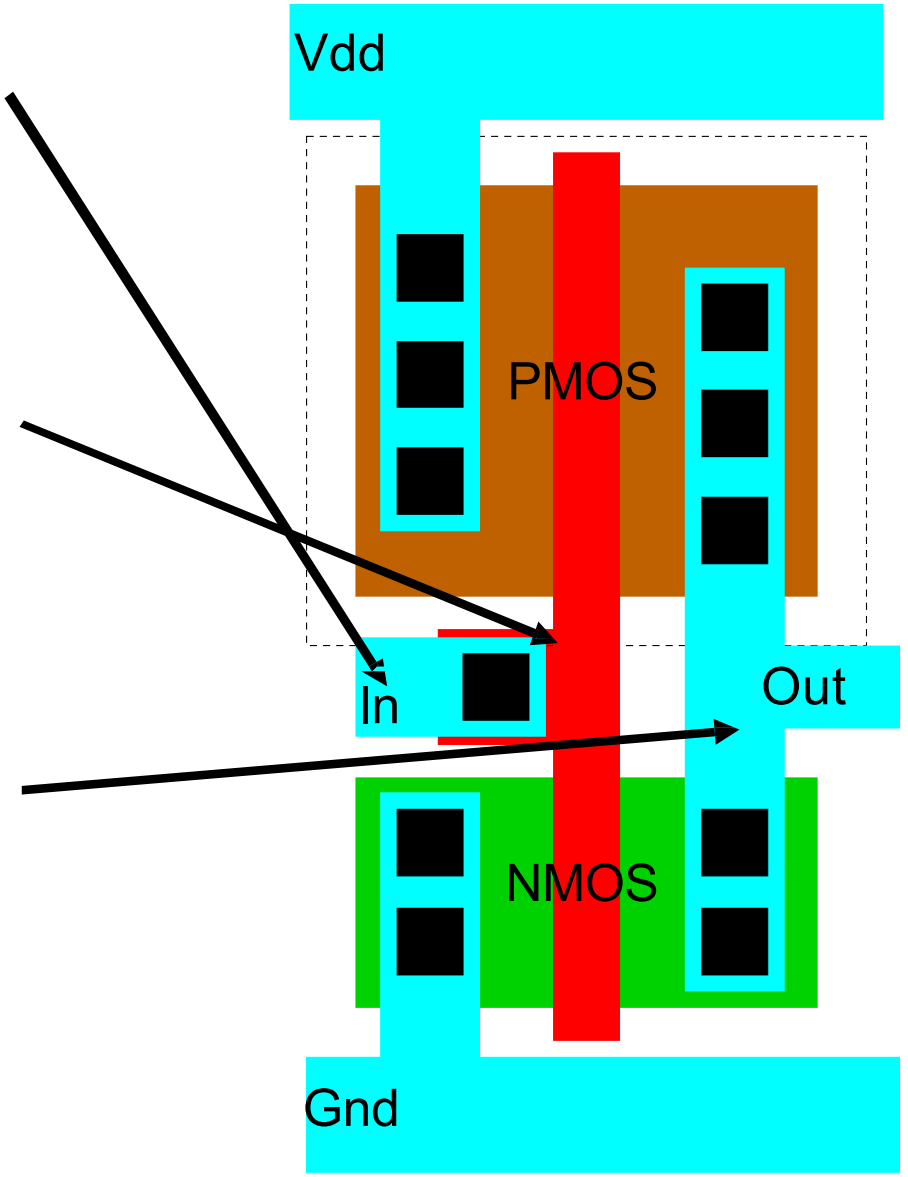


Example: inverter SPICE Model

```
*Definition of the inverter subcircuit In Out
.SUBCKT inv In Out 1 0

*Pull-up
M1  1 In Out 1 tp  L=56n W=448n AS=75.3f AD=75.3f PS=1.23u PD=1.23u

*Pull-down
M2  Out In 0 0 tn  L=56n W=224n AS=37.6f AD=37.6f PS= 784n PD=784n

*Optional: metal input/output capacitances
C1   In   0  24P
C2   Out  0  10P
.ENDS inv
```

Example: inverter Layout

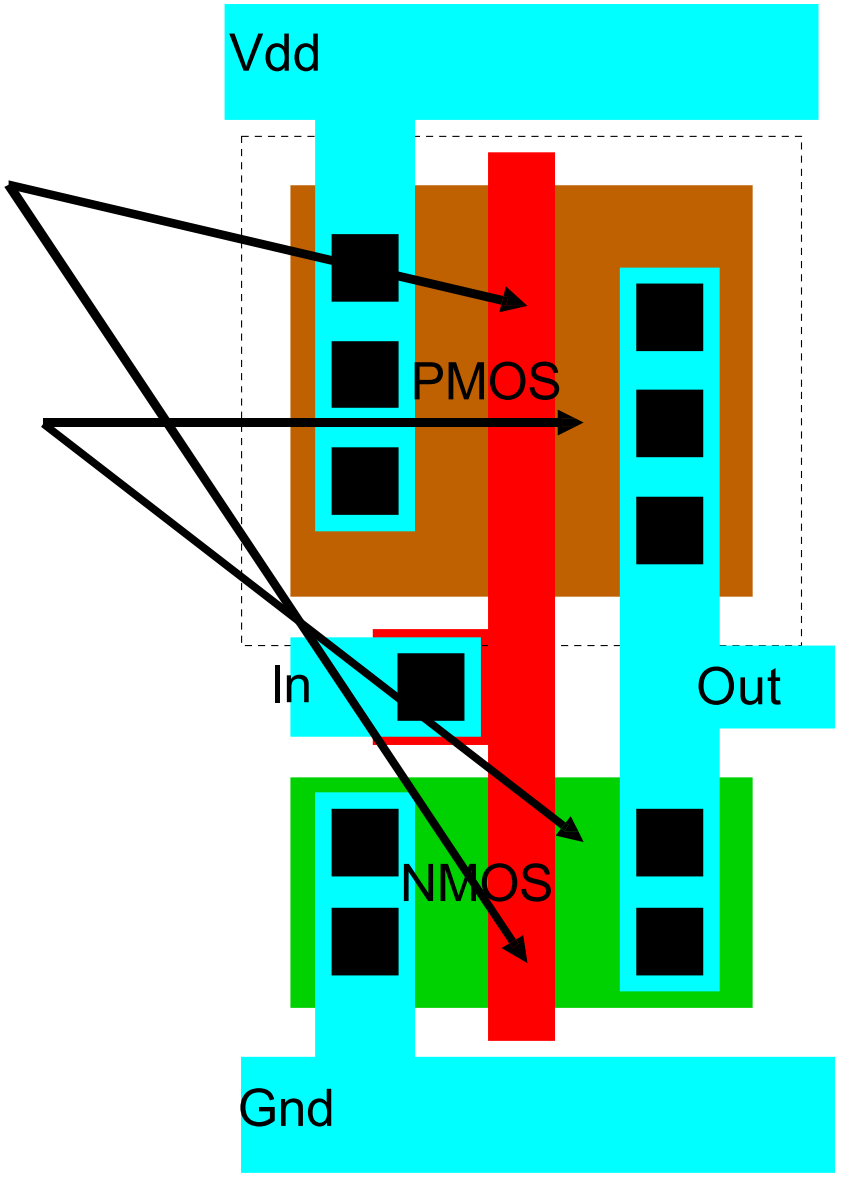
- Input metal capacitance:
  - Area: ? Need layout
  -
- Input poly capacitance:
  - Area: ? Need layout
  -
- Output metal capacitance:
  - Area: ? Need layout
  -



Example: inverter Layout

- Gate capacitance:
  - Area: automatic

  Difussion capacitance:
  - Area: automatic
- SPICE takes these capacitances already into account.



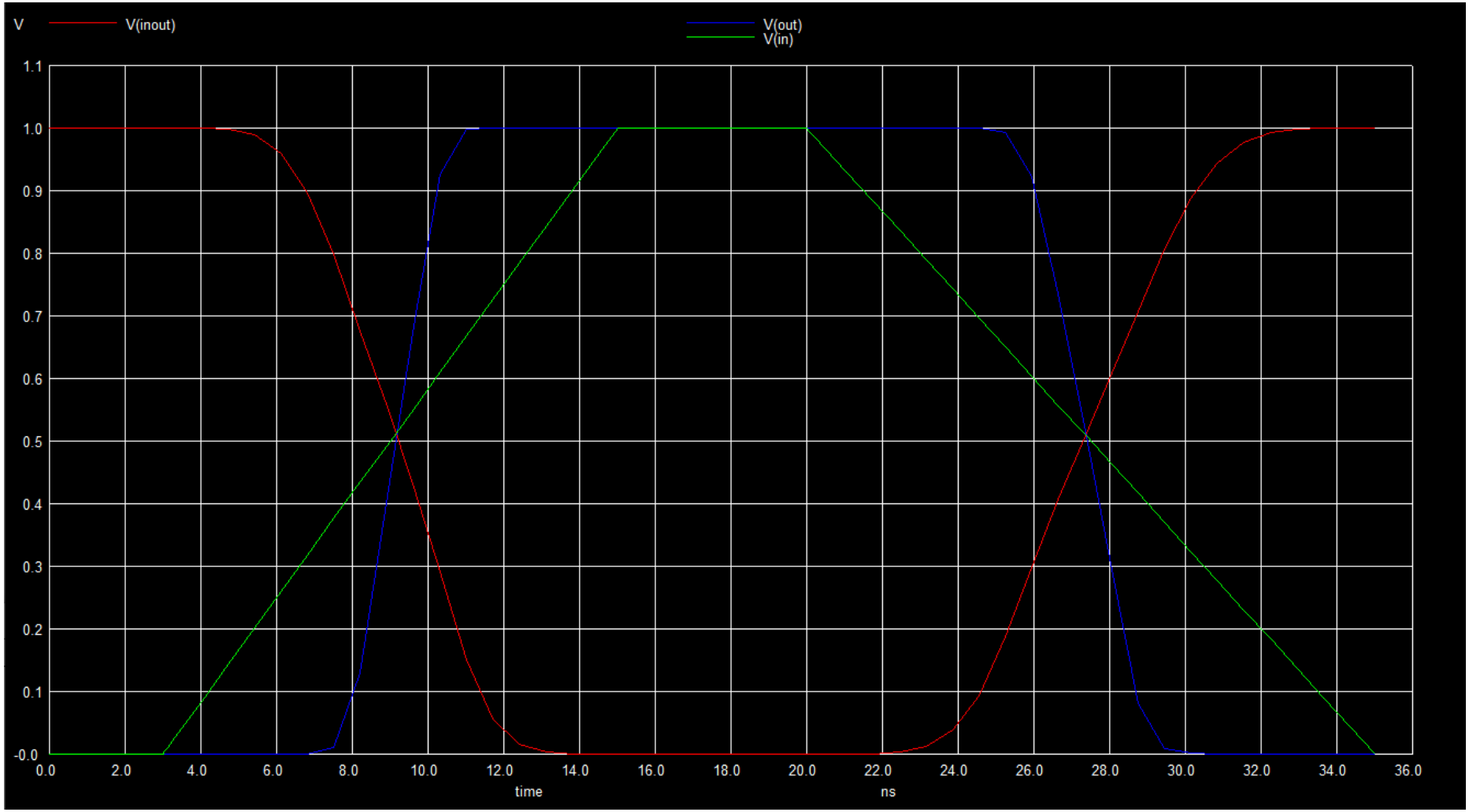
Example: inverter chain Simulation

```
*Load the modules
.INCLUDE model28nm.spi
.INCLUDE inv.spi
*Instantiation of the gate simulated
X1 In Out 1 0 inv
*1.0v between Vdd and Vss
VCC 1 0 DC 1.0V
*Simulation input
Vin In 0 pwl(0ns 0  3ns 0 15ns 1.0  20ns 1.0  35ns 0)

*Duration of simulation (step time and total time)
.TRAN 1ns 35ns
.END
```

Example: inverter Simulation

```
*Load the modules
.INCLUDE model28nm.spi
.INCLUDE inv.spi

*Instantiation of the gate simulated
X1 In InOut 1 0 inv
X2 InOut Out 1 0 inv

*1.0v between Vdd and Vss
VCC 1 0 DC 1.0V

*Simulation input
Vin In 0 pwl(0ns 0  3ns 0 15ns 1.0  20ns 1.0  35ns 0)


*Duration of simulation (step time and total time)
.TRAN 1ns 35ns
.END
```

Example: inverter Simulation



Example: complex gate Layout

- Function:

  F

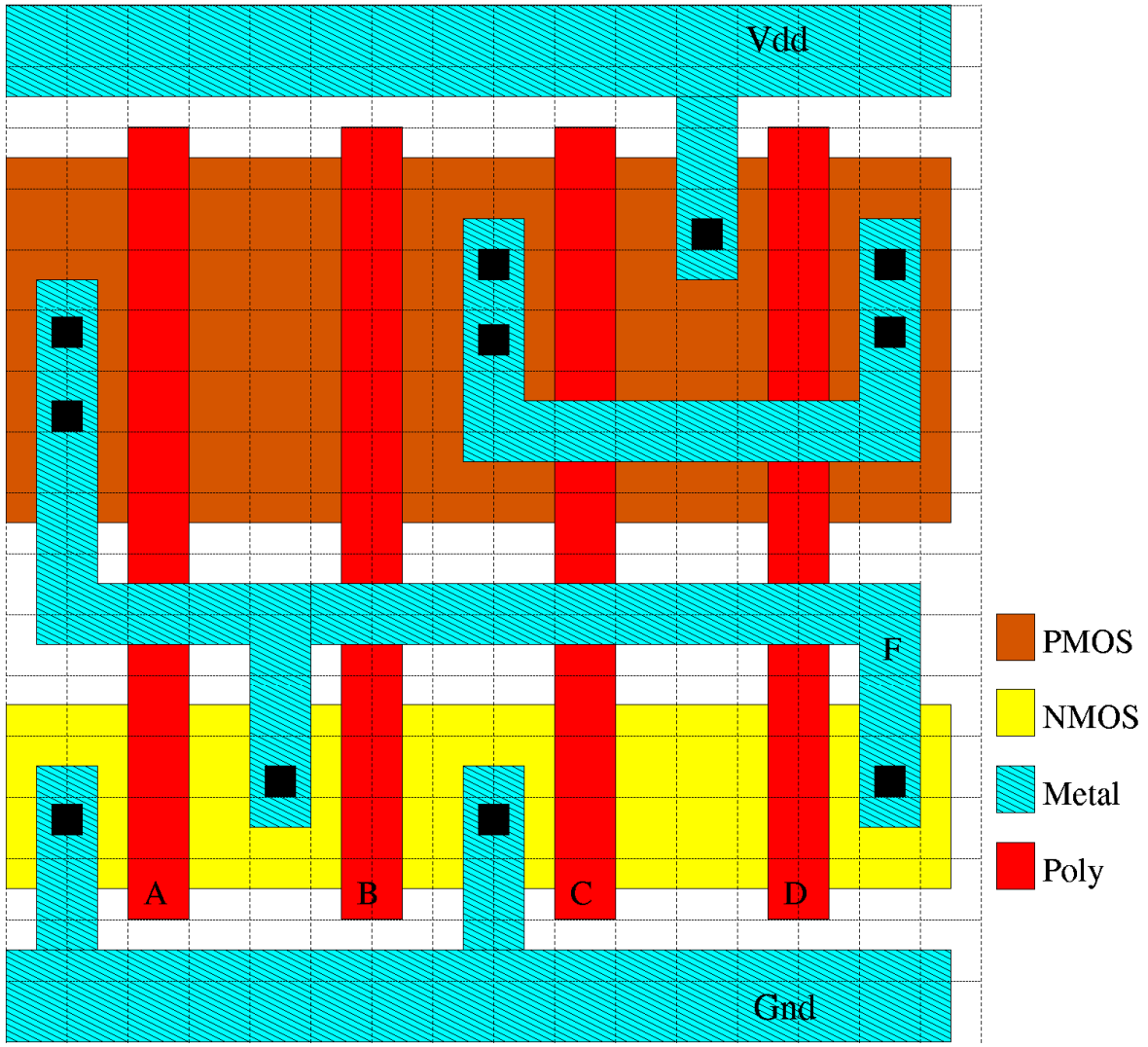
- Inputs:

  A, B, C, D

- Objective:

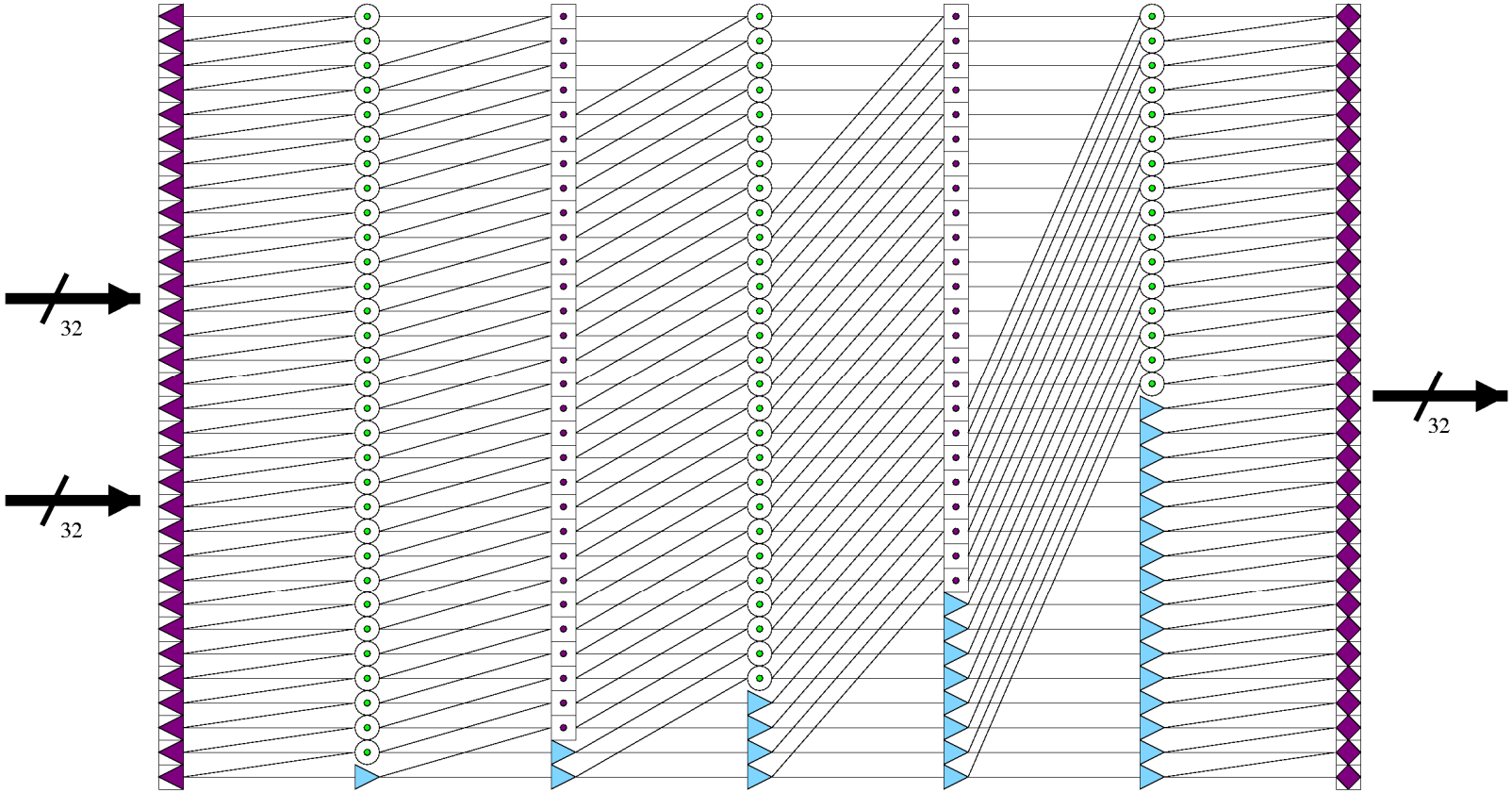
  Extract function

  Parasitic capacitances



Example: Brent-Kung Adder

- 32-bit Brent-Kung Adder:
  - Static CMOS
  - SPICE model (without connections)



Conclusions

- Electric simulation of digital circuits.
- Much more detailed than logic simulators:
  - Capacitances, resistances, transistors.
  - Can model interconnections (i.e. buses).
- Large simulation times.
- No (built-in) modelling of effects such as:
  - Cross-talk
  - Power consumption / energy