DIC L6: SPICE

Sung-Min Hong (smhong@gist.ac.kr)

Semiconductor Device Simulation Lab.
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

GIST Lecture on October 1, 2019

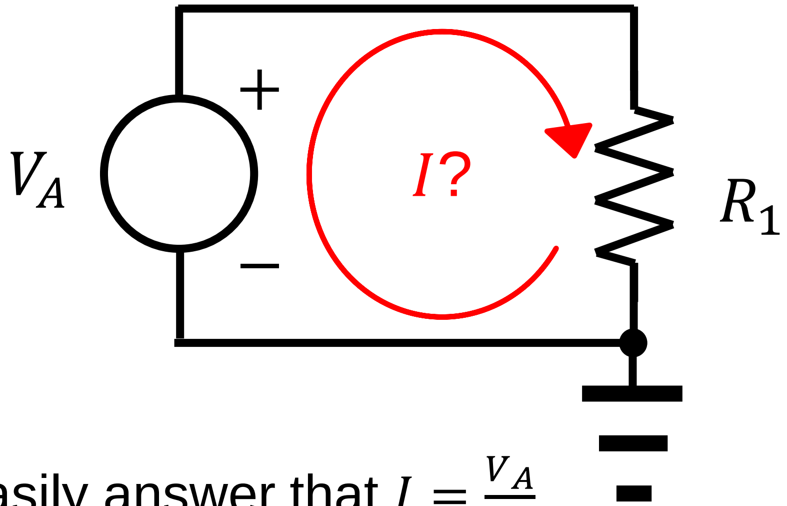
8.1. Introduction to SPICE

- **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis
  - Developed in 1970's at UC Berkeley
  - Many commercial versions are available.
  (For example, HSPICE)
- Initially, written in FORTRAN
for punch-card machines
  - Circuit elements are called cards.
  - Compled description is called a SPICE deck.

Larry Nagel, the main author of SPICE (Google Images)

8.1. Introduction to SPICE

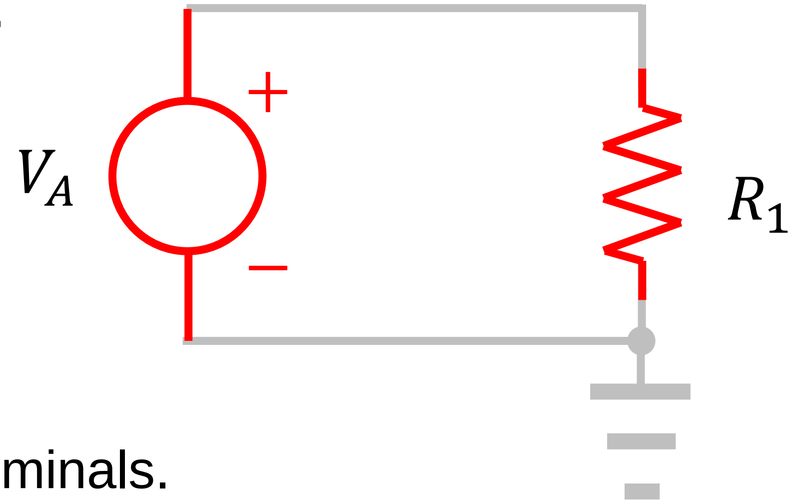
- Consider a simple problem.
  - What is the current?



  - Of course, you can easily answer that $I = \frac{V_A}{R_1}$.
  - But, how can we teach our computer to solve this problem?

8.1. Introduction to SPICE

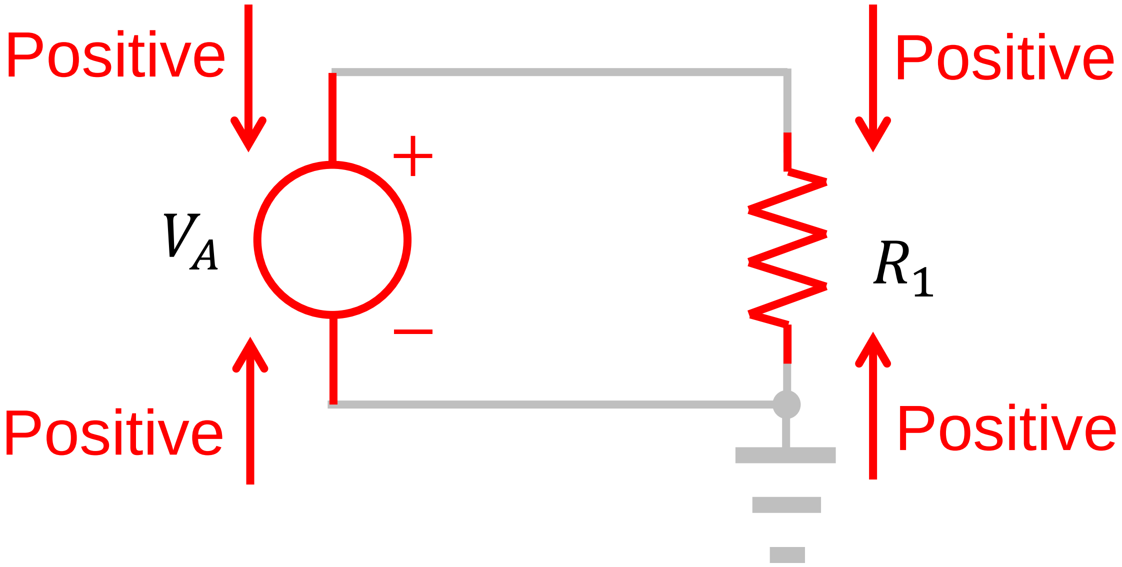
- Elements: Resistors, capacitors, etc
  - A circuit is made by connecting the elements.
  - They can have multiple terminals.
  - A resistor has two terminals.
  - A diode has two terminals.



  - A MOSFET has three (or four) terminals.

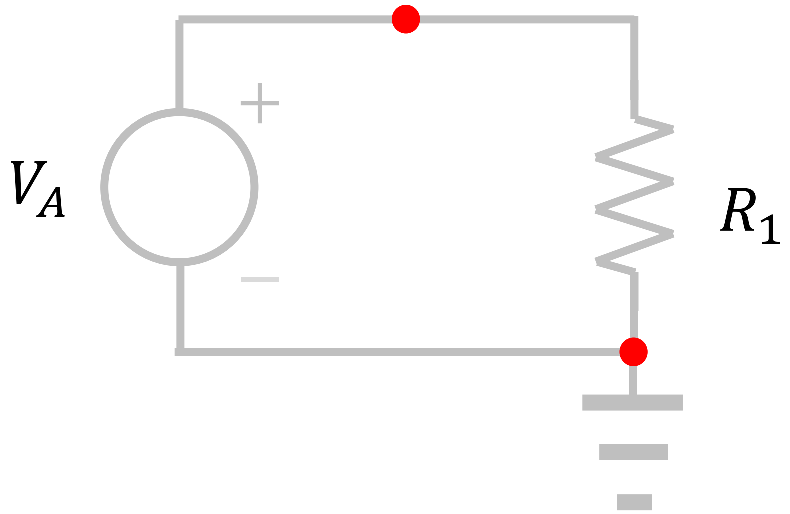
8.1. Introduction to SPICE

- Terminal current
  - Conventionally, an in-coming current is regarded as a positive one.



8.1. Introduction to SPICE

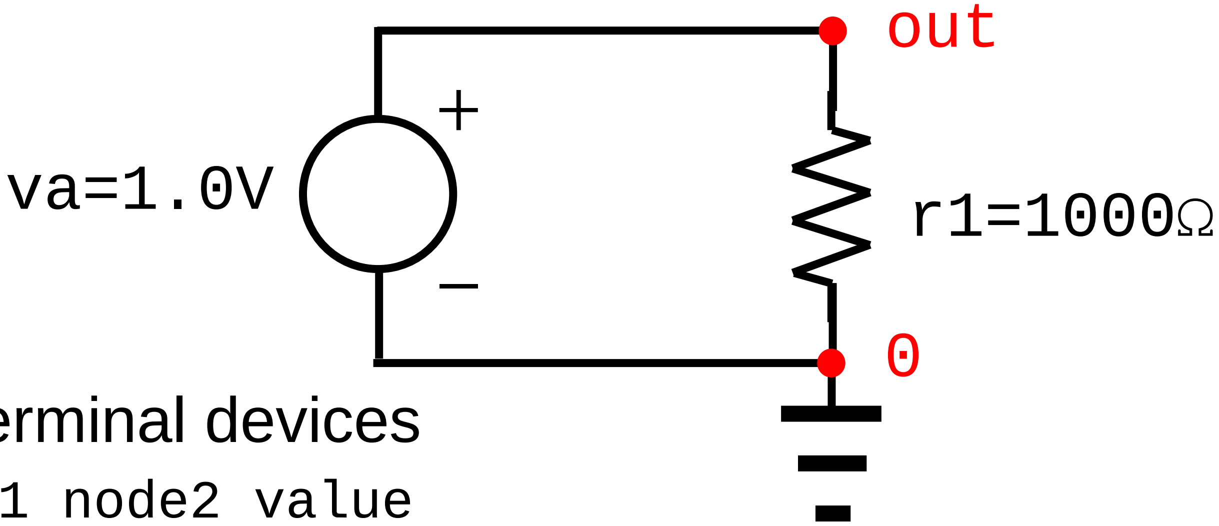
- Node: A point to which multiple terminals are tied.
  - Usually, a dot is used to represent a node.
  - There is a special node, GND.



8.1. Introduction to SPICE

- How can we describe a circuit?
- Of course, we can draw a circuit schematic. What else?
- A netlist for this circuit looks like:

```
va out 0 1.0
r1 out 0 1000
```



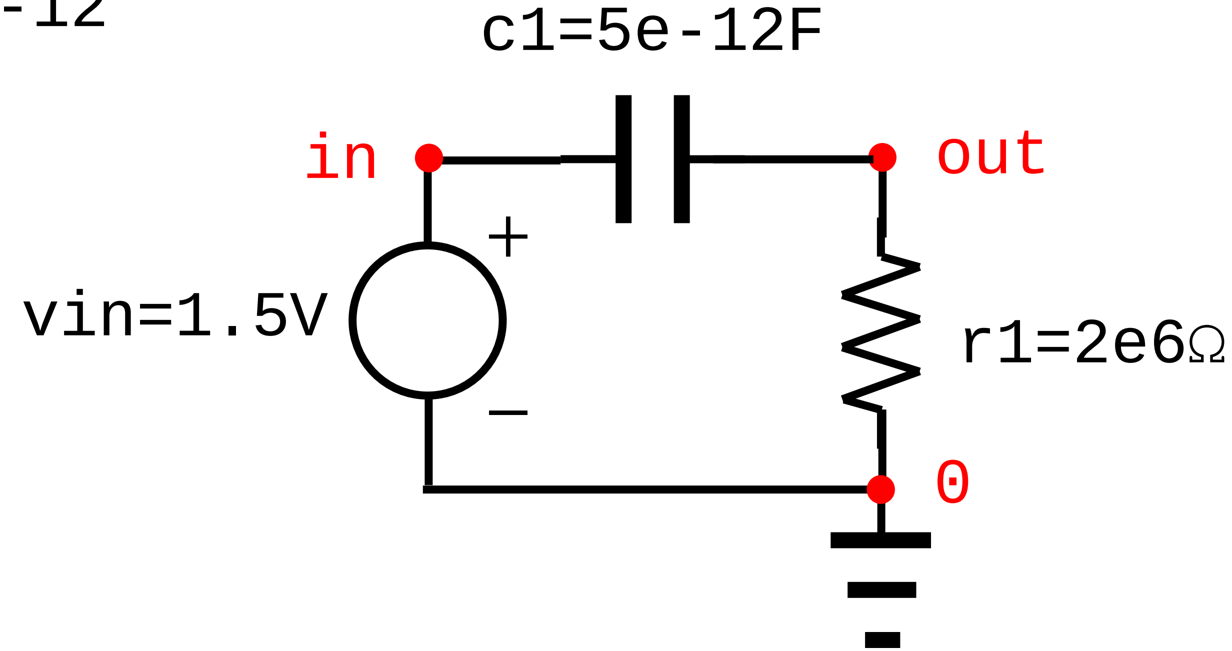
- Format for two-terminal devices

```
elementlabel node1 node2 value
```

8.1. Introduction to SPICE

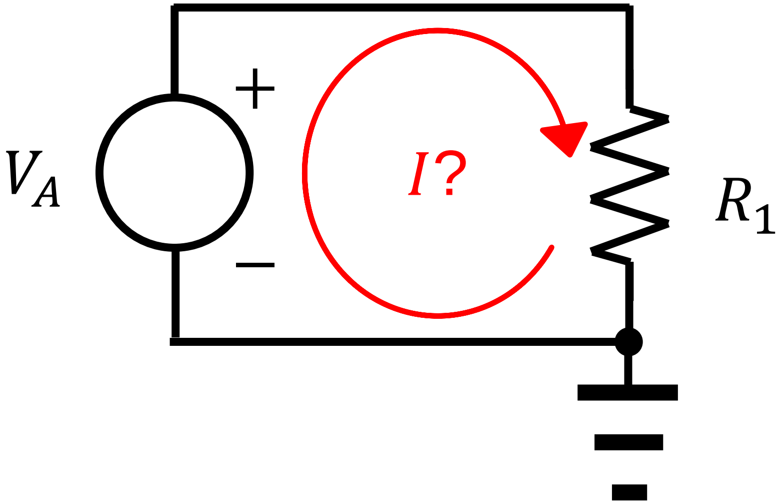
- A netlist for this circuit looks like:

```
c1 in out 5e-12
r1 out 0 2e6
vin in 0 1.5
```



8.1. Introduction to SPICE

- Solve a simple problem by a numerical means.
  - Identifying the governing equation



8.1. Introduction to SPICE

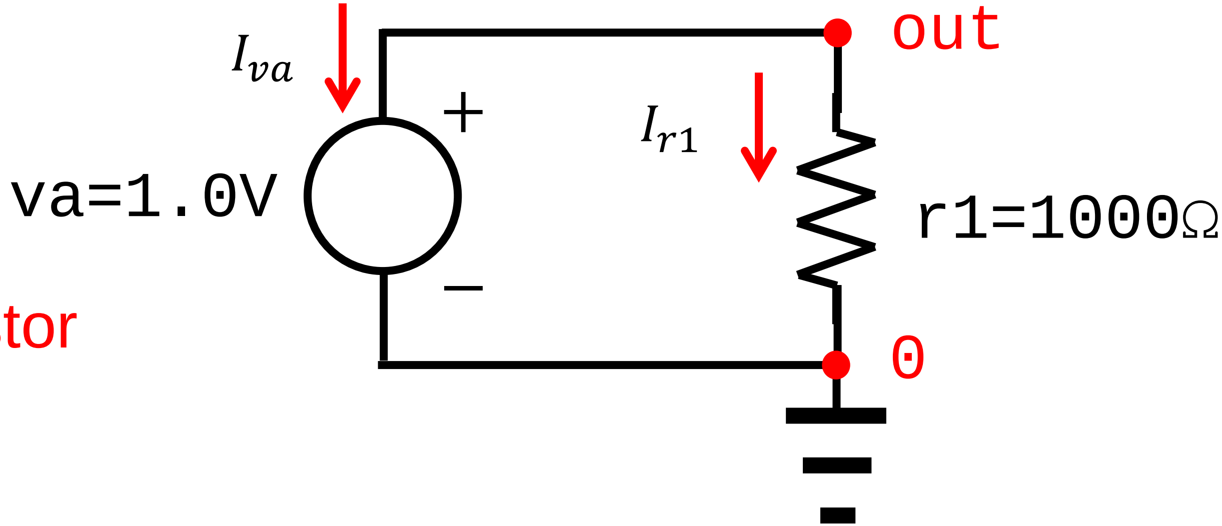
- Our simple problem
  - Three equations:

Voltage source

$$V(out) - 0.0 = 1.0$$

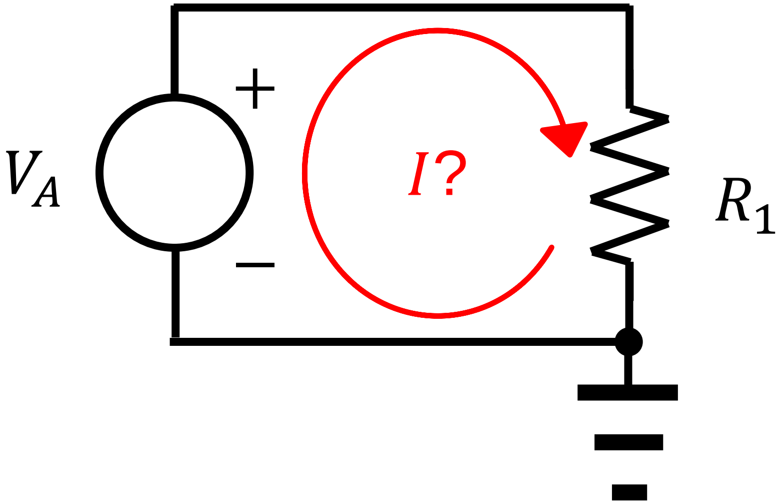
$$I_{r1} = \frac{V(out)}{1000}$$ Resistor

$$I_{va} + I_{r1} = 0$$ KCL



8.1. Introduction to SPICE

- Solve a simple problem by a numerical means.
  - Identifying the governing equation



8.2. SPICE tutorial (1)

- Meaning of characters

Letter	Unit	Magnitude
a	atto	10^{-18}
f	fempto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
u	micro	10^{-6}
m	milli	10^{-3}
k	kilo	10^{3}
x	mega	10^{6}
g	giga	10^{9}

Table 8.2

Table 8.1

Letter	Element
R	Resistor
C	Capacitor
L	Inductor
K	Mutual Inductor
V	Independent voltage source
I	Independent current source
M	MOSFET
D	Diode
Q	Bipolar transistor
W	Lossy transmission line
X	Subcircuit
E	Voltage-controlled voltage source
G	Voltage-controlled current source
H	Current-controlled voltage source
F	Current-controlled current source

8.2. SPICE tutorial (2)

- Example: RC circuit
  - We want to follow Figure 8.2 of our textbook. (Tested with SPICE3)

```
* rc.sp for SPICE3F5

Vin in 0 pwl 0ps 0 100ps 0 150ps 1.0 1ns 1.0
R1 in out 2k
C1 out 0 100f

.tran 20ps 1ns
.plot tran v(in) v(out)
.end
```

8.2. SPICE tutorial (3)

- M element for MOSFET

```
Mname drain gate source body type
+ W=<width> L=<length>
+ AS=<area source> AD = <area drain>
+ PS=<perimeter source> PD=<perimeter drain>
```

  - Valid for NMOSFETs and PMOSFETs! (I made a mistake in Lecture5.)

8.2. SPICE tutorial (4)

- Example: DC analysis
  - We want to follow Figure 8.6 of our textbook. (Tested with SPICE3)
  - SPICE3 does not support the BSIM3 model.

```
* mosiv.sp for SPICE3F5

.include models_1p2mu.sp

Vgs g 0 0
Vds d 0 0
M1 d g 0 0 NMOS W=2.4 L=1.2

.dc Vds 0 5.0 0.05 Vgs 0 5.0 1.0
.print dc V(g) I(Vds)
.end
```

8.2. SPICE tutorial (5)

- Example: Inverter
  - Calculate the voltage transfer curve. (Tested with SPICE3)

```
* inv.sp for SPICE3F5

.include models_1p2mu.sp

Vdd vdd 0 5.0
Vin a 0 0.0
M1 y a 0   0   NMOS W=2.4 L=1.2
M2 y a vdd vdd PMOS W=4.8 L=1.2

.dc Vin 0 5 0.01
.print dc V(a) V(y)
.end
```

Homework#2

- Report on the previous three examples
  - Run those examples.
  - Show the simulation results.

- Due: October 8, 2019 (Before the lecture starts)
  - Upload your Homework to our GitHub repository.