

SPICE

Elektronika pro informační technologie - IEL

Michal Bidlo, L330

Ústav počítačových systémů

FIT VUT v Brně

bidlom@fit.vutbr.cz

Obsah

- Úvod: co je SPICE, historie, některé zajímavosti
- Vybrané implementace SPICE (Linux, Win, Web)
- Základní princip použití (netlist, zadání vstupu, řízení simulace, zobrazení výstupu)
- SPICE v příkladech (typy simulace: ustálený stav, přechodové děje, frekvenčně závislé obvody)
- Pokročilé aspekty SPICE (modely nelineárních součástek, editory schémat, hierarchický popis, příklady simulace složitějších obvodů)

Úvod, co je SPICE?

Jedná se o SW nástroj určený pro simulaci elektrických a elektronických obvodů. Název SPICE vyjadřuje zkratku Simulation Program with Integrated Circuit Emphasis. Obvod je popsán ve formátu tzv. **netlistu**, což je textová reprezentace zapojení jednotlivých součástek (případně i složitějších komponent a podobvodů – hierarchický popis) a specifikace jejich parametrů – hodnoty odporů, kapacity, indukčnosti, zdroje a popis modelů složitějších součástek (diody, tranzistory, operační zesilovače apod.). Alternativně lze k zadání obvodu použít též různé editory schémat, z nichž je pak netlist vygenerován automaticky.

Historie SPICE

První verze (SPICE1 napsaná v jazyku FORTRAN) byla představena na University of California, Berkeley, v roce 1973 (L. W. Nagel a D. O. Pederson).

Značná vylepšení a rozšíření přinesla verze SPICE2 v roce 1975 (např. pokročilé simulační algoritmy, podpora základních polovodičových součástek).

Aktuální řada SPICE3 (napsána v C, autor Thomas Quarles, v různých modifikacích používána dodnes) vznikla v roce 1989. Zahrnuje mimo jiné možnost řízení simulace z příkazového řádku a grafické rozhraní X Window System pro zobrazování výsledků simulace.

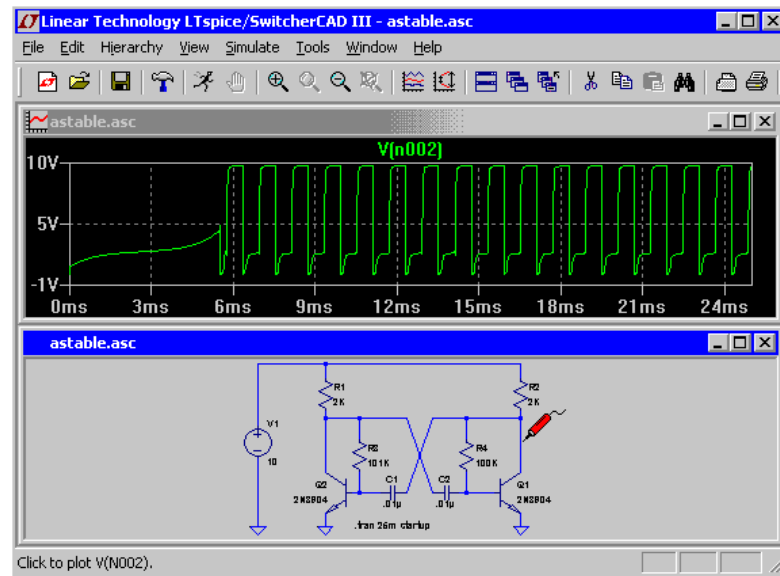
V roce 2011 organizace IEEE deklarovala SPICE jedním z významných milníků v oboru elektroniky a výpočetní techniky:

<http://theinstitute.ieee.org/tech-history/technology-history/spice-circuit-simulator-named-ieee-milestone765>

SPICE je v současné době průmyslovým standardem pro simulaci elektronických obvodů.

Distribuce SPICE

- **NGSPICE** – open-source implementace řízená z příkazového řádku, k dispozici pro všechny běžné OS, dostupná na <http://ngspice.sourceforge.net/>
- Windows, Mac: volná verze **Ltspice** (f. Linear Technology) – základní simulační funkce, včetně editoru schémat
- Komerční verze: na FIT k dispozici **H-spice** (f. Synopsis)
- **OrCAD Pspice** (f. Cadence), komplexní profi nástroj, možno zdarma získat Lite verzi pro studijní účely: <http://www.orcad.com/buy/try-orcad-for-free>
- Existují též verze SPICE online, např. www.partsim.com



Základní použití SPICE

- Pro účely simulace je třeba zadat vstupní obvod pro SPICE v podobě tzv. **netlistu**, který popisuje jednotlivé komponenty obvodu (odpory, diody atd.), jejich parametry a vzájemné propojení. Jedná se o **textovou reprezentaci**.
- Základní součástky mají ve SPICE symbolický popis vyjádřený jednopísmenným prefixem v rámci identifikátoru (uživatelského názvu) konkrétní součástky v obvodu.
- | Součástka | Označení | Příklady pojmenování |
|--------------|----------|----------------------|
| odpor | R | R, R1, Rvstup, R2x |
| kondenzátor | C | C, Cvaz,... |
| cívka | L | |
| dioda | D | |
| transistor | Q | |
| zdroj napětí | V | |
| a další... | | |

Základní použití SPICE

- Popis každé součástky sestává z jejího **pojmenování** a mezerami odděleného seznamu popisujícího **body připojení** a **parametry součástky** (např. hodnota odporu, napětí, kapacity..., ale u složitějších komponent např. název modelu, popis chování jako funkce času apod.).
- **Netlist** je posloupnost řádků, každý popisující jednu komponentu. První znak na řádku musí být prefix specifikující typ součástky.
- Propojení komponent je specifikováno pomocí pojmenovaných **bodů připojení**.
- **Každý obvod musí obsahovat bod připojení 0 (nula) – referenční potenciál, zem.**

Příklad obvodu, netlist

- Dolní propust – napěťový signál o nízkých frekvencích je na výstupu **out** utlumen oproti vstupu **in** jen velmi málo. Naopak signál o vyšších frekvencích je utlumen značně.

Netlist:

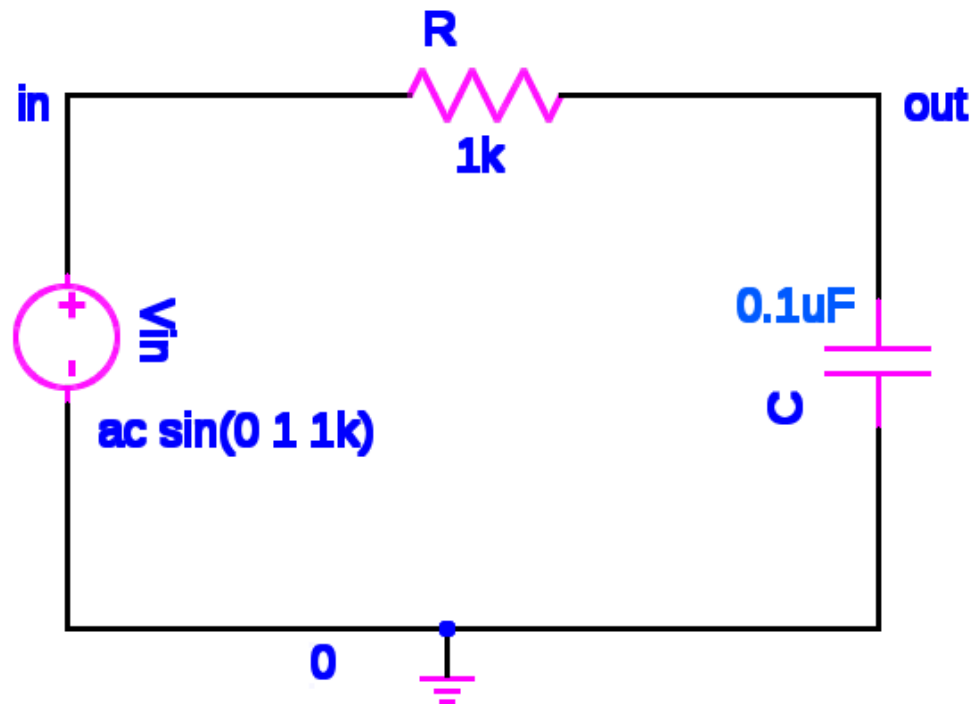
```
Vin in 0 ac sin(0 1 1k)
```

```
R in out 1k
```

```
C 0 out 0.1uF
```

Poznámky:

Vše je case insensitive, tj. $R \equiv r$ atd. Nezáleží na pořadí zápisu součástek v netlistu. Též nezáleží na pořadí bodů připojení součástek, u nichž se **nerozlišuje polarita**.



Poznámky k zadávání jednotek

- Zadání samostatné konstanty specifikuje hodnotu v základních jednotkách dané součástky, např.
 - V in 0 10: zdroj napětí 10V
 - R in out 500 (\equiv 0.5kohm): odpor 500 Ω , ale
 - C out 0 100 pro kapacitor 100F zřejmě ne => nutno specifikovat menší násobek základní jednotky (100M nebo 100m = 100 mili-Farad, 10u...mikro atd).
 - 1meg = mega-...

F	femto	10^{-15}
P	pico	10^{-12}
N	nano	10^{-9}
U	micro	10^{-6}
M	milli	10^{-3}
K	kilo	10^{+3}
MEG	mega	10^{+6}
G	giga	10^{+9}
T	tera	10^{+12}

* RC low-pass filter

Vin in 0 ac sin(0 1 1k)

R in out 1k

C 0 out 0.1uF

* nasledujici blok obsahuje prikazy spusteni simulace a zobrazeni vystupu

.control

* AC analysis (0.1 Hz - 6 kHz)

ac lin 1000 0.1 6khz

* Transient (time) analysis (step=10 us, t_max=2 ms)

tran 10us 2ms

* plot the results

plot ac.v(out)

plot tran.v(in) tran.v(out)

.endc

.end

Vstup pro NGSPICE, řízení simulace

**kompletní příklad kódu pro simulaci
výše uvedeného obvodu**

**Dalších příklady zde uvedené
budou řešeny pomocí ngspice.**

Anatomie kódu pro SPICE

* RC low-pass filter

Vin in 0 ac sin(0 1 1k)

R in out 1k

C 0 out 0.1uF

První řádek je povinný komentář (uvozený *)

* **nasledující blok obsahuje příkazy spuštění simulace a zobrazení
výstupu**

.control

* AC analysis (0.1 Hz - 6 kHz)

ac lin 1000 0.1 6khz

Existují různé typy simulace: zde např. tzv. AC (zkoumá odezvu obvodu na měnící se frekvenci) a časová (sledování činnosti obvodu v čase – přechodové děje, apod.).

* Transient (time) analysis (step=10 us, t_max=2 ms)

tran 10us 2ms

* plot the results

plot ac.v(out)

plot tran.v(in) tran.v(out)

Zde specifikujeme, co vykreslit, např.: hodnotu napětí v bodě out v závislosti na frekvenci vstupního signálu, časový průběh napětí v bodech in a out.

.endc

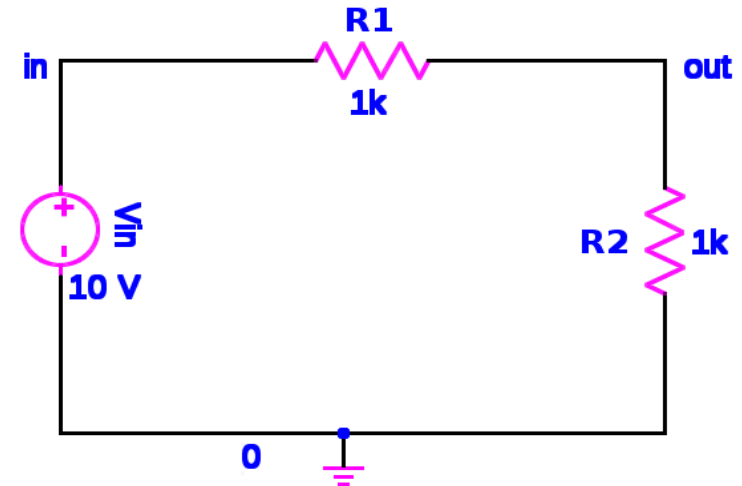
.end

Poslední řádek musí být .end

Simulace typu operační bod ve SPICE

- Simulace **operačního bodu** obvodu je nejjednodušším typem simulace – vypočte hodnoty napětí a proudů ve všech částech obvodu pro dané **konstantní vstupní parametry**.

- Příklad: dělič napětí (vdivider.net)
- Úkol: určit napětí U_{R2} za předpokladu daného napětí zdroje $V_{in} = 10\text{ V}$ a hodnot $R1 = R2 = 1000\ \Omega$.



- Netlist se specifikací simulace „operační bod“:

```
* Voltage divider
```

```
Vin in 0 10
```

```
R1 in out 1k
```

```
R2 out 0 1k
```

```
.control
```

```
* operating point analysis
```

```
op
```

```
.endc
```

```
.end
```

Simulace příkladu vdivider.net v ngspice

V adresáři s příkladem spustíme: `ngspice vdivider.net`

Jelikož má netlist uveden i příkaz pro spuštění simulace (`op`), ngspice načte netlist a simulaci rovnou vykoná.

Pro výpis výsledků
v ngspice zadáme:

`print all`

Případně pro výpis
jednotlivých veličin:

`print v(in)`

`print v(out)`

`print -i(Vin) ...proud obvodem`

`print v(in, out) ...napětí na R1`

Ctrl-d ukončí ngspice.

```
ngspice 2 -> print all
in = 1.000000e+01
out = 5.000000e+00
vin#branch = -5.00000e-03
ngspice 2 -> print v(in)
v(in) = 1.000000e+01
ngspice 2 -> print v(out)
v(out) = 5.000000e+00
ngspice 2 -> print v(in, out)
v(in, out) = 5.000000e+00
ngspice 2 -> print -i(Vin)
-i(vin) = 5.000000e-03
ngspice 2 -> █
```

Popis komplexních součástek

Pokročilé součástky (zejména polovodiče) a komponenty s více než dvěma elektrodami nelze obvykle charakterizovat jednou hodnotou, tudíž je nutné použít pro jejich simulaci ve SPICE tzv. **model**.

Model představuje komplexní matematicko-fyzikální popis součástky (jako např. diody, tranzistoru aj.), případně se může jednat o složitější obvod sestavený z více součástek (např. operační zesilovač, stabilizátor apod.), který je použit jako subsystém (stavební blok) většího celku – jedná se v zásadě o hierarchický popis.

Modely obvykle nevytváří sám uživatel, nýbrž se použijí již hotové – k dispozici jsou z webů výrobců elektroniky, případně za poplatek jako komerční produkt. Některé též z webů různých zájmových skupin z oblasti elektroniky jako „open-source“ materiál.

DC simulace, příklad modelu diody

- **DC simulace** simuluje chování obvodu pro daný rozsah **stejnosměrného napětí** zdroje.
- Příklad: nelineární obvod s diodou (viz soubor diode_va.net)
- Úkol: určit závislost proudu protékajícího obvodem na hodnotách napětí zdroje V1 pro dané parametry obvodových součástek.
- Model dioda 1N4007 k dispozici na <http://www.onsemi.com/PowerSolutions/supportDoc.do?type=models&part=1N4007>

* DC simulation example

.MODEL 1n4007 d

+IS=7.02767e-09 RS=0.0341512 N=1.80803 EG

+XTI=5 BV=1000 IBV=5e-08 CJO=1e-11

+VJ=0.7 M=0.5 FC=0.5 TT=1e-07

+KF=0 AF=1

D1 out 0 **1n4007**

V1 in 0 10

R1 in out 1k

.control

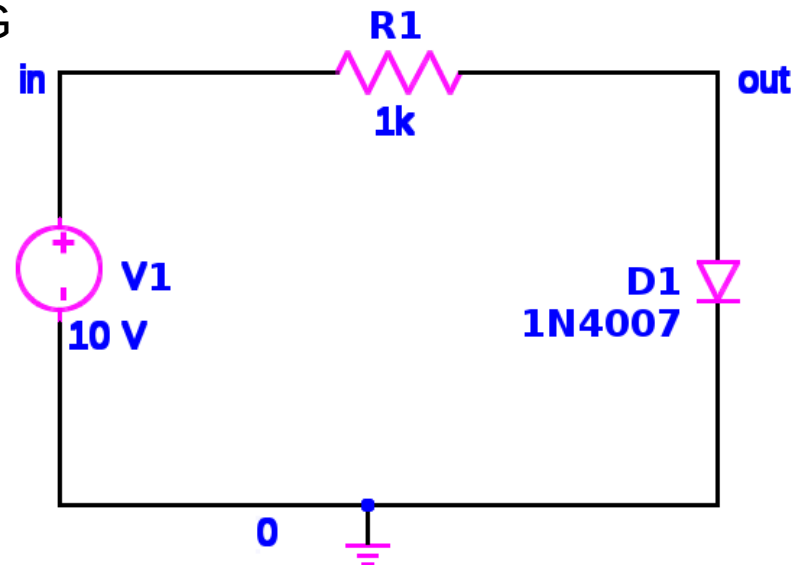
*** DC analysis from -5V to 5V with 10mV steps**

dc V1 -5 5 10m

plot -i(V1)

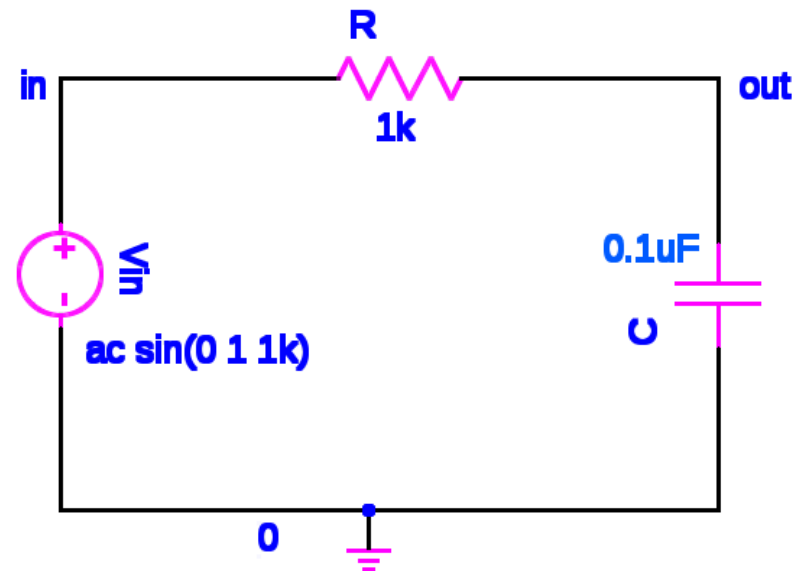
.endc

.end



Další typy simulací

- **AC simulace** (frekvenční analýza) zkoumá chování obvodu v závislosti na měnící se frekvenci daného vstupního (střídavého) signálu.
- **Časová simulace** – zkoumá chování obvodu v čase. Vhodné např. pro sledování přechodných jevů a obecně pro simulaci obvodů, u nichž nás zajímá časový průběh (tvar) signálu.
- Příklad: analogový filtr dolní propust (viz soubor low_pass.net)
- Úkol 1 (AC simulace): zjistit závislost amplitudy výstupního napětí na frekvenci vstupního střídavého napětí zdroje V1 pro dané parametry obvodu.
- Úkol 2 (časová simulace.): zjistit průběh vstupního a výstupního napětí v daném časovém úseku.



Řešení AC a časové simulace RC článku

Netlist

* RC low-pass filter

* Circuit netlist

Vin in 0 ac sin(0 1 1k)

R in out 1k

C 0 out 0.1uF

.control

* AC analysis: here for 1000 samples from 0.1 Hz - 6 kHz (linear)

ac lin 1000 0.1 6khz

* Transient (time) analysis from t = 0 to 2 ms with a step of 10 us

tran 10us 2ms 0ms

.endc

.end

Editory schémat pro SPICE

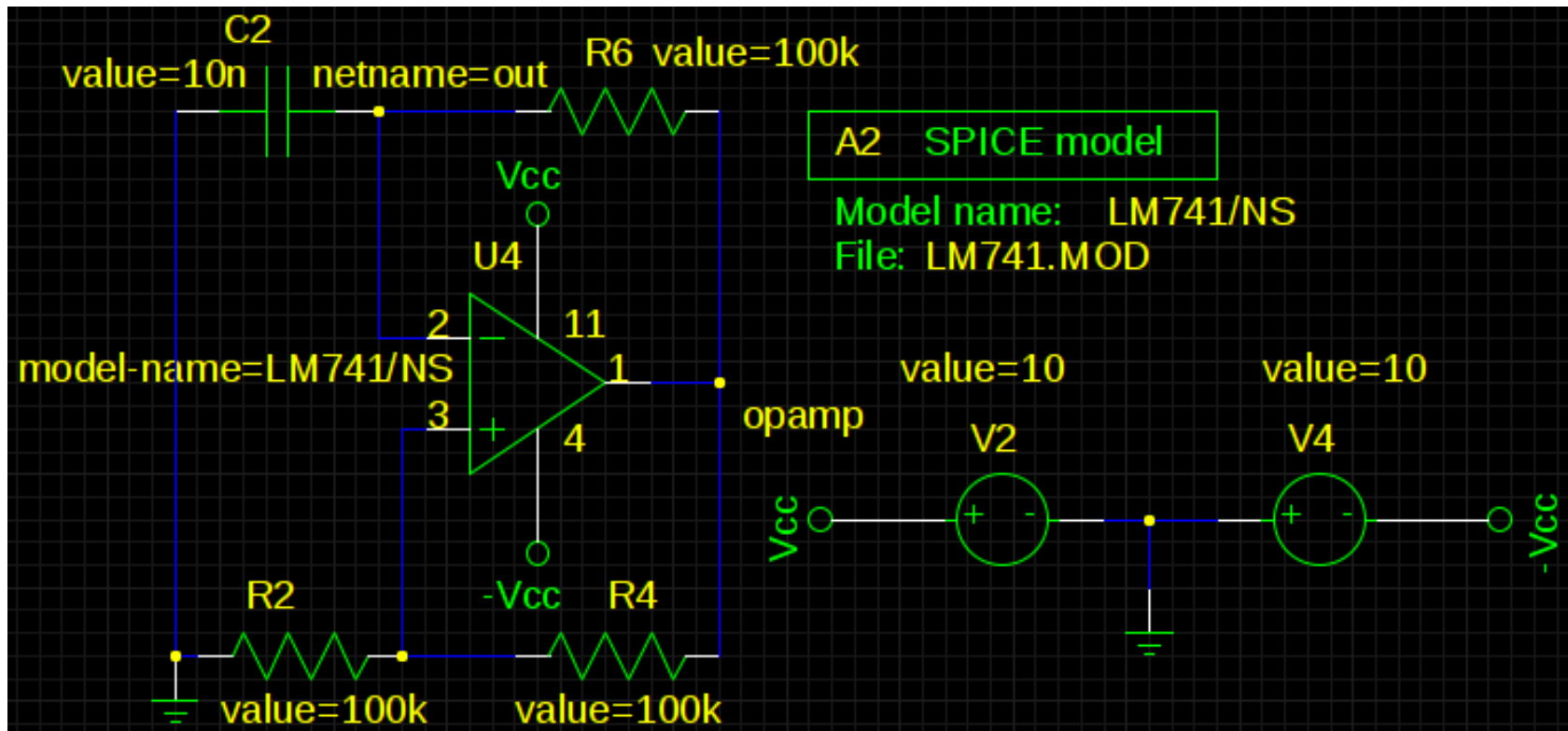
- Existuje řada aplikací pro grafický návrh schémat obvodů, případně simulaci či generování netlistu pro SPICE:
 - LTspice (Win) – demo-ukázka v rámci přednášky, další podrobnosti viz literatura [1]
 - gEDA – sada nástrojů pro Linux, zahrnuje editor schémat gschem; <http://www.geda-project.org/>, použití viz další slidy
 - PartSim – online editor + SPICE simulator; <http://www.partsim.com/simulator#>
 - CircuitLab – <https://www.circuitlab.com/>
 - Falstad Circuit Simulator Applet – online, včetně řady příkladů obvodů; <http://www.falstad.com/circuit/>
 - a další...

gEDA (Linux): gschem + gnetlist

- Jedná se o prostředky pro tvorbu schémat s možností automatického generování netlistu pro SPICE.
- V .deb-like Linux např. dostupné balíčky **geda-gschem**, **geda-gnetlist** a **ngspice** (součásti většího celku **geda**).
- Příklad (viz dále): Schmitt trigger – střídavý oscilátor s operačním zesilovačem LM741.
- Základní kroky:
 - 1. návrh schématu pomocí gschem,
 - 2. generování netlistu pomocí gnetlist,
 - 3. doplnění příkazů pro řízení simulace,
 - 4. simulace pomocí nativní linux-verze **ngspice**, zobrazení výsledků simulace.

Příklad obvodu v aplikaci gschem

- Veškeré symboly jsou v gschem dostupné z knihoven prostředí geda, je však třeba dodat model pro LM741: <http://www.ti.com/product/lm741>

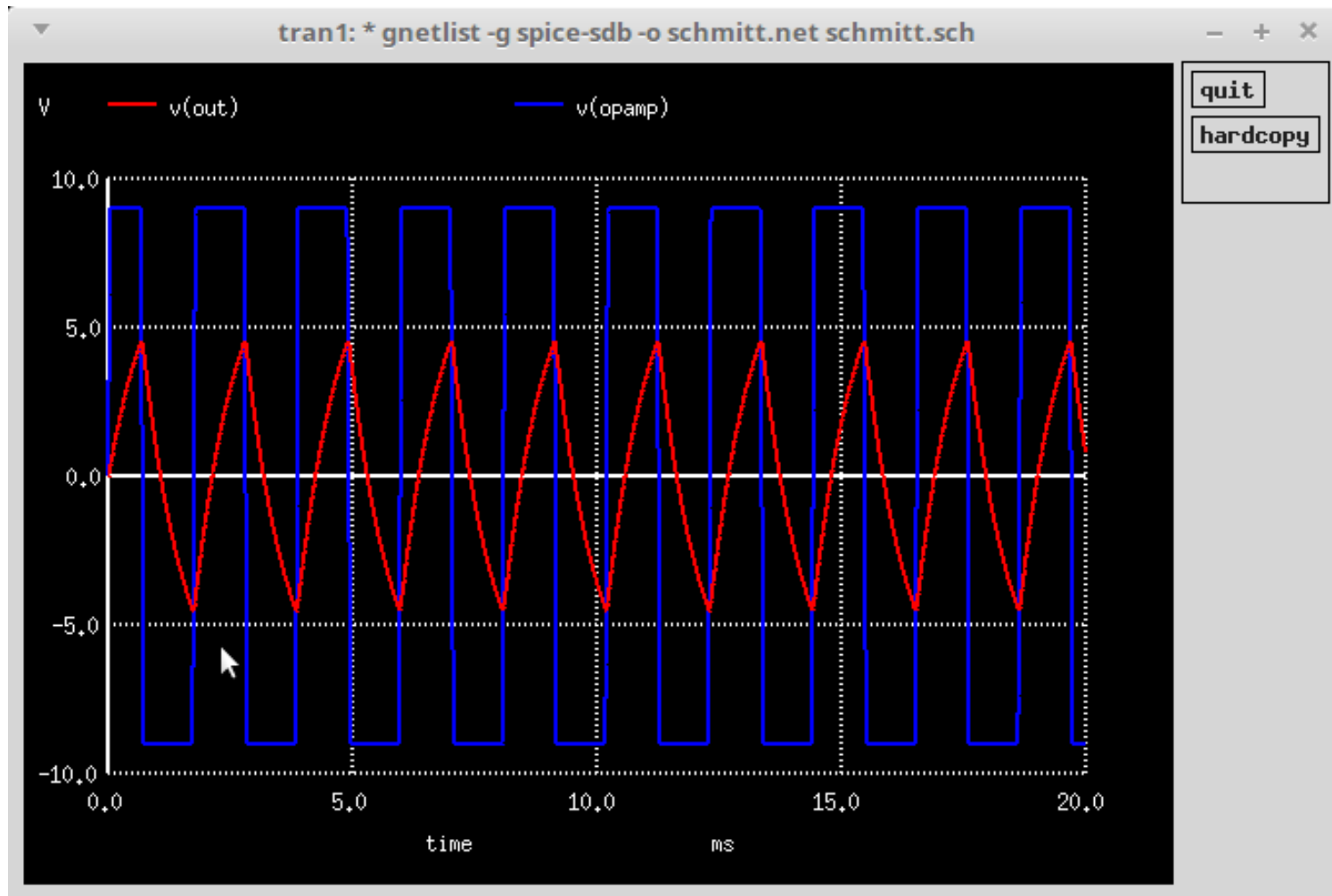


Generování netlistu nástrojem gnetlist

- Vstupní soubory
 - Schéma z aplikace gschem: schmitt.sch
 - Model operačního zesilovače: LM741.MOD
- Výstup: soubor s netlistem pro SPICE: schmitt.net
- Generování netlistu ze schématu:
`gnetlist -g spice-sdb -o schmitt.net schmitt.sch`
- Do netlistu je nutno dopsat příkazy řízení simulace:
`.control`
`tran 10us 20ms`
`plot v(out) v(opamp)`
`.endc`

Simulace pomocí ngspice (Linux)

- Vstup: soubor schmitt.net s doplněnými příkazy pro spuštění simulace a zobrazení výsledků, spuštění ngspice pomocí: `ngspice schmitt.net`
- Výstup: časový diagram simulace dle zadaných parametrů:



Textový výstup simulace z ngspice do souboru

- Je možné zadat přesměrování k příkazu print, např.:

```
print v(opamp) > schmitt.dat
```

- SPICE standardně vypisuje data se stránkovými oddělovači (prázdné řádky + záhlaví), pro potřeby dalšího zpracování je toto možné vypnout (mimo sekci .control) zadáním:

```
.opt nopage
```

```
* gnetlist -g spice-sdb -o schmitt.net schmitt.sch
Transient Analysis Tue Oct 6 23:37:05 2015
-----
Index    time                v(opamp)
-----
0    0.000000e+00    1.658650e-02
1    1.000000e-07    1.658650e-02
2    2.000000e-07    1.658651e-02
3    4.000000e-07    1.658653e-02
4    8.000000e-07    1.658662e-02
5    1.600000e-06    1.659088e-02
6    1.893517e-06    1.659547e-02
7    2.480552e-06    1.663468e-02
8    2.767563e-06    1.668328e-02
9    2.974561e-06    1.674499e-02
10   3.145443e-06    1.682400e-02
11   3.319093e-06    1.694481e-02
12   3.513636e-06    1.715544e-02
13   3.738649e-06    1.756092e-02
14   3.954236e-06    1.821646e-02
15   4.183477e-06    1.940776e-02
16   4.404033e-06    2.136409e-02
```

- Více k použití SPICE na Linuxu lze nalézt např. v původním článku [2].
- Zajímavé pojednání p řízení ngspice z jazyka Python - vizte článek [3].

Literatura

- [1] C. A. Schuler: Electronics Principles and Applications (8th Edition), McGraw-Hill Edu., 2012 + doplňkový materiál:
http://highered.mheducation.com/sites/dl/free/0073373796/938955/LTSpice_Final.ppt
- [2] K. Cosgrove: *Analyzing Circuits with SPICE on Linux*. Linux Journal, issue 39, July 1, 1997; dostupné online:
<http://www.linuxjournal.com/article/2169>
- [3] P. Fabo, M. Kuba: Python a simulácia elektronických obvodov. Root.cz [05/2018], dostupné online:
<https://www.root.cz/clanky/python-a-simulacia-elektronickych-obvodov/>
- [4] A. Vladimirescu: *The SPICE Book*. Wiley, 1994
- [5] P. Tuinenga: *SPICE: A Guide to Circuit Simulation and Analysis Using Pspice* (3rd Edition), Prentice Hall, 1995
- [6] R. Jacob Baker: *CMOS Circuit Design, Layout and Simulation*. Wiley + IEEE Press, 2010