

Modelování a simulace v mikroelektronice

Garant předmětu:

Prof. Ing. Dalibor Biolek, CSc.

Autor textu:

Prof. Ing. Dalibor Biolek, CSc.

Obsah

1	ÚVOD.....	5
2	ZAŘAZENÍ PŘEDMĚTU VE STUDIJNÍM PROGRAMU	5
2.1	ÚVOD DO PŘEDMĚTU	6
2.2	VSTUPNÍ TEST	7
3	ÚVOD DO SIMULACE OBVODŮ PROGRAMY TYPU SPICE.....	10
3.1	PROGRAMY TYPU „SPICE-LIKE“ A „SPICE-COMPATIBLE“ [2]	10
3.2	TYPICKÁ STRUKTURA SIMULAČNÍHO PROGRAMU ZALOŽENÉHO NA PSPICE	11
3.3	POSLOUPNOSTI ČINNOSTÍ V SIMULAČNÍM PROGRAMU	12
3.4	CO VŠE LZE ANALYZOVAT POMOCÍ PROGRAMŮ „SPICE – COMPATIBLE“	14
3.5	PRAVIDLA PRO TVORBU TEXTOVÝCH SOUBORŮ.....	14
3.6	SHRnutí KAPITOLY 3	15
3.7	KONTROLNÍ OTÁZKY A PŘÍKLADY KE KAPITOLE 3	16
4	ZÁKLADY MODELOVÁNÍ V PSPICE, ČÁST 1.....	17
4.1	KLÍČOVÉ ZNAKY ANALOGOVÝCH SOUČÁSTEK.....	17
4.2	DEFINICE SOUČÁSTKY V NETLISTU, KTERÁ NEVYUŽÍVÁ PŘÍKAZ .MODEL.....	18
4.3	DEFINICE SOUČÁSTKY V NETLISTU, KTERÁ VYUŽÍVÁ PŘÍKAZ .MODEL	18
4.4	PŘÍKAZY .LIB A .INC	19
4.5	NĚKTERÉ DALŠÍ ZÁKLADNÍ PŘÍKAZY, VYUŽÍVANÉ PRO MODELOVÁNÍ V PSPICE	20
4.6	SHRnutí KAPITOLY 4	22
4.7	ŘEŠENÉ PŘÍKLADY	22
4.8	KONTROLNÍ OTÁZKY KE KAPITOLE 4	28
4.9	NEŘEŠENÉ PŘÍKLADY	30
5	ZÁKLADY MODELOVÁNÍ V PSPICE, ČÁST 2.....	32
5.1	PRÁCE S PODOBVODY SPICE	32
5.2	FUNKCE ZABUDOVANÉ V PSPICE PRO MODELOVÁNÍ PRVKŮ	33
5.3	VYTVÁŘENÍ VLASTNÍCH FUNKCÍ	34
5.4	ŘÍZENÉ (ZÁVISLÉ) ZDROJE PRO BEHAVIORÁLNÍ MODELOVÁNÍ (ABM –ANALOG BEHAVIORAL MODELING)	35
5.5	ŘÍZENÉ (ZÁVISLÉ) SPÍNAČE PRO BEHAVIORÁLNÍ MODELOVÁNÍ	37
5.6	SHRnutí KAPITOLY 5	38
5.7	ŘEŠENÉ PŘÍKLADY	39
5.8	KONTROLNÍ OTÁZKY KE KAPITOLE 5	45
5.9	NEŘEŠENÉ PŘÍKLADY	46
6	HLEDÁNÍ STEJNOSMĚRNÝCH USTÁLENÝCH STAVŮ – ZÁKLAD VĚTŠINY ANALÝZ V PSPICE	47
6.1	HLEDÁNÍ STEJNOSMĚRNÉHO ŘEŠENÍ V NELINEÁRNÍM OBVODU	47
6.1.1	„Ruční“ řešení.....	48
6.1.2	Algoritmický postup při řešení simulačním programem.....	49
6.1.3	Problémy s konvergencí - úvod.....	51
6.1.4	Globální podmínky simulace - úvod.....	52
6.2	OZNAČOVÁNÍ STEJNOSMĚRNÝCH NAPĚTÍ A PROUDŮ V PSPICE A V PROBE	54
6.3	DC ANALÝZA = STEJNOSMĚRNÁ ANALÝZA.....	55
6.4	SHRnutí KAPITOLY 6	60
6.5	ŘEŠENÉ PŘÍKLADY.....	60
6.6	KONTROLNÍ OTÁZKY KE KAPITOLE 6	72
6.7	NEŘEŠENÉ PŘÍKLADY	73
7	ANALÝZA TRANSIENT NEBOLI ČASOVÁ ANALÝZA.....	75
7.1	CÍLE ČASOVÉ ANALÝZY = ANALÝZY SIGNÁLŮ V OBVODU	75
7.2	PŘÍKAZ .TRAN	77
7.3	ZADÁVÁNÍ POČÁTEČNÍCH PODMÍNEK PRO ČASOVOU ANALÝZU	77

7.4	MECHANISMUS PŮSOBNÍ PŘÍKAZŮ .IC A .NODESET V ANALÝZE TRANSIENT	79
7.5	TYPICKÉ KOMBINACE (NE)POVOLENÍ VÝPOČTU PRACOVNÍHO BODU A IMPLICITNÍCH/VLASTNÍCH POČÁTEČNÍCH PODMÍNEK	80
7.6	GENEROVÁNÍ SIGNÁLŮ V PSPICE.....	80
7.6.1	<i>Nezávislé zdroje napětí a proudu jako zdroje signálů</i>	81
7.6.2	<i>Závislé (řízené) zdroje jako zdroje signálů</i>	83
7.7	SHRNUTÍ KAPITOLY 7	83
7.8	ŘEŠENÉ PŘÍKLADY	84
7.9	KONTROLNÍ OTÁZKY KE KAPITOLE 7.....	92
7.10	NEŘEŠENÉ PŘÍKLADY	93
8	ANALÝZA AC NEBOLI KMITOČTOVÁ ANALÝZA	95
8.1	CÍLE KMITOČTOVÉ ANALÝZY	95
8.2	JAK POSTUPUJE SIMULÁTOR PŘI KMITOČTOVÉ ANALÝZE	95
8.3	SYNTAXE ANALÝZY AC	96
8.4	JAKÉ JSOU ZDROJE STŘÍDAVÉHO SIGNÁLU V OBVODU	96
8.5	REPREZENTACE PROMĚNNÝCH PŘI ANALÝZE AC V PSPICE	97
8.6	NĚKTERÉ FUNKCE V PSPICE A V PROBE PRO ANALÝZU AC.....	97
8.7	ZÁVISLÉ (ŘÍZENÉ) ZDROJE PRO TRANSFORMACI SIGNÁLŮ V ANALÝZE AC	98
8.8	POUŽITÍ PŘÍKAZU .PRINT V ANALÝZE AC.....	100
8.9	SHRNUTÍ KAPITOLY 8	101
8.10	ŘEŠENÉ PŘÍKLADY	101
8.11	KONTROLNÍ OTÁZKY KE KAPITOLE 8.....	109
8.12	NEŘEŠENÉ PŘÍKLADY	111
9	PŘÍLOHA: SIMULACE V ORCADPSPICE 10 NA ÚROVNI VSTUPNÍCH SOUBORŮ.....	113
10	SPRÁVNÉ VÝSLEDKY ZE VSTUPNÍHO TESTU.....	115
11	ODPOVĚDI NA KONTROLNÍ OTÁZKY	116
12	VÝSLEDKY NEŘEŠENÝCH PŘÍKLADŮ	117
12.1	KAPITOLA 4	117
12.2	KAPITOLA 5	120
12.3	KAPITOLA 6	121
12.4	KAPITOLA 7	126
12.5	KAPITOLA 8	130
13	SEZNAM POUŽITÉ LITERATURY	136

Seznam obrázků

OBR. 3.1:	ZJEDNODUŠENÁ BLOKOVÁ STRUKTURA PROGRAMU TYPU PSpICE.....	11
OBR. 3.2:	SCHÉMA, VSTUPNÍ SOUBOR, NETLIST, SIMULACE [1]	13
OBR. 3.3:	TYPY ANALÝZ, ANALYZAČNÍ REŽIMY A MÓDY PRO „SPICE-COMPATIBLE“ PROGRAM MICROCAP. ŠKRTY JSOU VYZNAČENA OMEZENÍ PSpICE.....	14

Ostatní obrázky nejsou očíslovány, což vyplývá ze způsobu jejich zasazení do učebního textu.

1 Úvod

Tyto elektronické učební texty jsou určeny zejména pro studenty 1. ročníku magisterského studijního programu Elektrotechnika, elektronika, komunikační a řídicí technika, studijního oboru Mikroelektronika, předmětu „Modelování a simulace v mikroelektronice (MMSI)“.

2 Zařazení předmětu ve studijním programu

Předmět „Modelování a simulace v mikroelektronice“ je vyučován v letním semestru 1. ročníku magisterského studia oboru Mikroelektronika v rozsahu 26 hodin přednášek, 26 hodin počítačových cvičení a 13 hodin práce na individuálních projektech, čemuž odpovídá jeho ohodnocení šesti kredity. Předmět je zakončen zápočtem a zkouškou.

Nejdůležitější předměty bakalářského studia, na které tento předmět obsahově navazuje, jsou „Analogové elektronické obvody“ (BAEY), „Elektronické součástky“ (BESO), „Modelování a počítačová simulace“ (BMPS), z volitelných oborových předmětů pak „Mikroelektronické praktikum“ (BMPE). Předpokládá se aktivní znalost základních zákonů a principů teoretické elektrotechniky, metod analýzy lineárních a nelineárních obvodů, jakož i znalost vlastností a funkce základních elektrotechnických součástek.

Z předmětů zimního semestru 1. ročníku je navazováno zejména na „Analogové integrované obvody“ (MAIO) a „Mikroelektronické prvky a struktury“ (MMPR). Z teorie signálů je vyžadována znalost spektrální analýzy a praktických zásad používání algoritmu FFT.

Pokud jde o navazování na matematiku, v předmětu „Modelování a simulace v mikroelektronice“ je běžně používán matematický aparát pro popis a analýzu lineárních a nelineárních elektrických obvodů. To představuje práci se soustavami lineárních algebraických rovnic a manipulace s těmito soustavami prostřednictvím maticového počtu. Lineární diferenciální rovnice budou formálně převáděny na algebraické prostřednictvím operátorového počtu. Nelineární rovnice budou řešeny numerickými iteračními metodami. O těchto metodách je třeba mít alespoň uživatelský přehled ve smyslu globálního porozumění mechanismů jejich fungování. Dále je doporučováno absolvování předmětu „Diferenciální rovnice a jejich použití v elektrotechnice“ (MDRE) z bloku teoretické nástavby v 1. semestru. K zvládnutí simulačních metod Monte Carlo a Worst Case jsou zapotřebí základní znalosti z teorie pravděpodobnosti a statistiky.

2.1 Úvod do předmětu

Texty jsou koncipovány pro **samostatné** studium. Jednotlivé kapitoly obsahují vždy teoretickou a praktickou část. V teoretické části jsou objasňovány základní postupy počítačového modelování, analýzy a simulace elektronických systémů. V praktické části jsou tyto postupy ozřejmovány formou řešených příkladů v PSPICE. Úroveň nabytých vědomostí si student může průběžně ověřovat řešením dalších příkladů z učebního textu.

Doporučené zásady práce s příklady v textu na cvičeních nebo během samostatného studia:

1. Východiskem práce je „papírové“ zadání schématu obvodu a požadavky na cíle analýzy.
2. Prvním cílem práce je napsání vstupního souboru. Student může mít k dispozici všechny dostupné materiály a literaturu bez jakéhokoli omezení, během cvičení i vydatnou pomoc učitele.
3. Při použití nového příkazu, funkce apod. by si student měl tyto techniky fixovat jako vzory možného řešení jiných konkrétních problémů počítačové simulace. Dané syntaxe nemusí zpočátku umět z paměti, měl by však vědět, kde hledat, až je bude později potřebovat. Měl by si později v manuálech nalézt další varianty a možnosti daných příkazů a orientačně se s nimi seznámit. Doporučeným zdrojem informací je soubor **pspsref.pdf**, který je součástí instalace programu OrCADPSpice 10 a nachází se v adresáři **doc/pspcref**. Další dokumenty jsou v elektronické formě k dispozici na webových stránkách předmětu.
3. Preferovanou metodou při práci se SPICE je metoda pokusu a omylu ve smyčce vstupní soubor-simulace, která by měla postupně konvergovat k bezchybným výsledkům. Důležité je, aby student přicházel na řešení alespoň některých problémů sám. Neméně důležité je, aby se zpětně naučil lépe rozumět obvodům, které simuluje. Pro vyučujícího to znamená dvě věci: Dát studentům přiměřený návod a pak i čas na samostatné řešení, pak předvést správné postupy, a v závěru příkladu komentovat výsledky simulací **jako elektrikář**. Pro studenty to znamená mít po ruce vhodné přehledy syntaxí jazyka SPICE, jakož i zápisky z přednášek, a schopnost provádět efektivní záznamy o probíhajících počítačových experimentech do pracovních sešitů. Důležité je i samostatné procházení příkladů mimo organizovanou výuku na vlastním počítači.

Výklad je orientován na simulační program OrCADPSpice 10. Probírané obecné principy počítačového modelování, analýzy a simulace jsou však využitelné pro práci s jakýmkoli komerčním simulačním programem z „rodiny SPICE“. Výklad se opírá především o simulace na úrovních vstupních souborů. Proto není nutné, aby studenti museli používat přímo program OrCADPSpice v. 10. Je to pouze doporučováno. Omezenou volně šířitelnou verzi tohoto programu lze stáhnout z webových stránek předmětu na

<http://user.unob.cz/biolek>

Po absolvování předmětu by měl být student schopen samostatně řešit úlohy, vyžadující tvůrčí využívání možností modelování a simulací soudobých verzí PSpice.

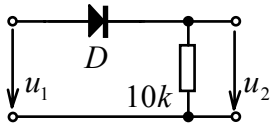
2.2 Vstupní test

Průchod následujícím „autotestem“ vám ukáže, nakolik vaše současné znalosti odpovídají vstupním požadavkům na úspěšné další studium předmětu. Výsledky jsou uvedeny v kapitole 10 na str. 115.

Vyznačte správnou odpověď (ke každé otázce existuje právě jedna):

č.	obvod	otázka	varianty odpovědí
1		Napětí U_X [V] je	a) 2, b) 3, c) 4, d) -6
		Napětí U_Y [V] je	a) -2, b) -3, c) -4, d) 6
		Proud I_X je	a) $>I_Y$, b) $<I_Y$, c) I_Y , d) $-I_Y$
		Proud I_X [A] je	a) 1, b) -1, c) 2, d) -2
2		Napětí [mV] na R_1 je	a) 200, b) 300, c) 400, d) 0
		Napětí baterie U_B [V] je	a) 1, b) 2, c) 3, d) 5
		Napětí U_X [V] je	a) 1, b) 2, c) -1, d) 0
		Výkon [mW] dodávaný baterií je	a) 200, b) 300, c) 400, d) 0
3		Proud I_X [mA] je	a) -8, b) 1, c) 2, d) 8
		Proud I_Y [mA] je	a) -8, b) -1, c) 2, d) 8
		Napětí U_X [V] je	a) 2, b) 8, c) 16, d) -2
		Poměr výkonů na R_1 a na R_2 je	a) 0,5, b) 2, c) 1, d) 4
4		Po připojení baterie se obvod dostane do ustáleného stavu řádově za několik	a) sekund, b) milisekund, c) mikrosekund, d) nanosekund
		V ustáleném stavu bude C nabit na napětí [V]	a) 0, b) 5, c) 10, d) -10
5		Amplituda proudu C [mA] v ustáleném stavu bude asi	a) 0, b) 5, c) 10, d) 100
		Obvod se chová jako filtr typu	a) dolní propust, b) horní propust, c) pásmová propust, d) pásmová zádrž
		Mezní kmitočet [kHz] filtru je zhruba	a) 0,1, b) 0,3, c) 1, d) 100
6		Napětí na R [V] je zhruba	a) 0, b) 0,7, c) 4,35, d) 5
		Proud diodou [mA] je zhruba	a) 0, b) 10, c) 22, d) 25
		Při změně R na 150Ω se napětí na R	a) nezmění, b) klesne o několik procent, c) vzroste o několik procent, d) klesne o desítky procent
7		Napětí na R [V] je zhruba	a) 0, b) 0,7, c) 4,35, d) 5
		Proud diodou [mA] je asi	a) 0, b) 10, c) 22, d) 25
		Při změně R na 150Ω se napětí na R	a) nezmění, b) klesne o několik procent, c) vzroste

			o několik procent, d) klesne o desítky procent
8		Tranzistor je v režimu	a) aktivním, b) saturace, c) nevodivém, d) inverzním
		Při zvětšení odporu R_b se napětí U_{CE}	a) zvětší, b) zmenší, c) nezmění se
		Při zvětšení teploty se napětí U_{CE}	a) zvětší, b) zmenší, c) nezmění se
9		Střídavé zesílení obvodu je zhruba	a) 1, b) -20, c) -250, d) 500
		Střídavý vstupní odpor zesilovače je zhruba	a) 50Ω, b) 2kΩ, c) 750kΩ, d) 1MΩ
		Obvod je schopen zesilovat signály o kmitočtech od	a) 0Hz, b) 8Hz, c) 8kHz, d) 80kHz
10		Napětí U_d [V] je	a) 0, b) 0,73, c) 10, d) -1
		Proud I_i [mA] je	a) 0, b) 1m, c) 10, d) 11
		Napětí U_2 [V] je	a) 0, b) 1, c) 11, d) 14
11		Napětí U_d [V] je	a) 0, b) 0,73, c) 10, d) -1
		Proud I_i [mA] je	a) 0, b) 1m, c) 10, d) 11
		Napětí U_2 [V] je	a) 0, b) 1,27, c) 10, d) 14
12		Napětí U_d [V] je	a) 0, b) 2,27, c) -1,27, d) -1
		Proud I_i [mA] je	a) 0, b) 1, c) 10, d) 11
		Napětí U_2 [V] je	a) 0, b) -1,27, c) 10, d) -14
13		Signál má stejnosměrnou složku [V]	a) 0, b) 2, c) 5, d) 10
		První harmonická signálu má kmitočet [kHz]	a) 0, b) 0,5, c) 1, d) 2
		Po dvojnásobném „zpomalení“ signálu se amplituda 1. harmonické	a) 2x sníží, b) 2x zvýší, c) nezmění se, d) vynuluje
14		Po průchodu výše uvedeného signálu obvodem bude signál u_2 změněn oproti u_1 takto:	a) bude konstantní o velikosti 5V, b) bude nulový, c) nezmění se tvar, ale rozkmit bude od -5V do +5V, d) podstatně se změní tvar

		Po průchodu harmonického signálu o kmitočtu 1kHz obvodem zaznamená spektrální analyzátor na výstupu čáru na kmitočtu [kHz]	a) 1, b) 1,5, c) 2, d) 2,5
15		Po průchodu harmonického signálu o kmitočtu 1kHz obvodem zaznamená spektrální analyzátor na výstupu (mj.) čáru na kmitočtu [kHz]	a) 0,5, b) 1,5, c) 2, d) 2,5

3 Úvod do simulace obvodů programy typu SPICE

Cíle kapitoly:

- ✚ Objasnit pojmy „SPICE-like“ a „SPICE-compatible“ programy.
- ✚ Objasnit typickou strukturu programů typu PSPICE a jaké možnosti a styl práce uživatele z toho vyplývají.
- ✚ Seznámit se s typy metod analýzy elektronických obvodů v „SPICE-like“ simulačních programech.

3.1 Programy typu „SPICE-like“ a „SPICE-compatible“ [2]

V roce 1971 vytvořil student „University of California“, Berkeley, USA Larry Nagel program SPICE1 (SPICE = Simulation Program with Integrated Circuit Emphasis) jako vývojově vyšší verzi svého předchozího programu CANCER (Computer Analysis of Nonlinear Circuits Excluding Radiation). Program umožňoval analýzu dějů v obvodech, obsahujících zejména bipolární a unipolární tranzistory. O věrohodnost výsledků bylo usilováno propracovaností modelů (Gummel-Poonův model bipolárního tranzistoru a Shichman-Hodgesův model tranzistorů MOSFET a JFET) i matematických algoritmů řešení rovnic. Uživatel měl navíc možnost prakticky neomezeného rozšiřování sortimentu analyzovaných součástek technikou makromodelů zakládáním tzv. podobvodů (subcircuits) SPICE.

Protože program byl v podstatě volně šířitelný, stal se brzo standardním simulačním nástrojem pro elektrotechnické úlohy. Usilovně se pracovalo na jeho zdokonalování.

V roce 1975 byla představena verze SPICE2 s podstatně zdokonalenými modely i numerickými algoritmy. Tato verze byla v průběhu téměř 20 let postupně zdokonalována na Berkeleyjské univerzitě až do dnes všeobecně známého standardu SPICE2G.6, který byl v r. 1983 zpřístupněn k volnému používání.

Zdrojové texty SPICE1 a SPICE 2 byly napsány ve Fortranu. Vzhledem k zvýšenému využívání Unixových pracovních stanic padlo v Berkeley rozhodnutí přepsat SPICE 2 do jazyka C. Tak začala vznikat verze SPICE3. Dnes je rozšířena verze SPICE 3F.2. Oproti SPICE2G.6 se vyznačuje řadou vylepšení, ovšem z různých důvodů došlo k ztrátě zpětné kompatibility se SPICE2G.6.

S růstem výkonnosti počítačů PC došlo k přepisování programů, dosud běžících na výkonných pracovních stanicích, na programy spustitelné na „PCčkách“. Tak vznikl standard PSpice.

Dnes existuje více simulačních programů, které využívají v podstatě tři ne zcela kompatibilní standardy: SPICE2, SPICE3, PSPICE. Všechny lze rozdělit na tzv. „Spice-like“ a „Spice-compatible“ simulátory.

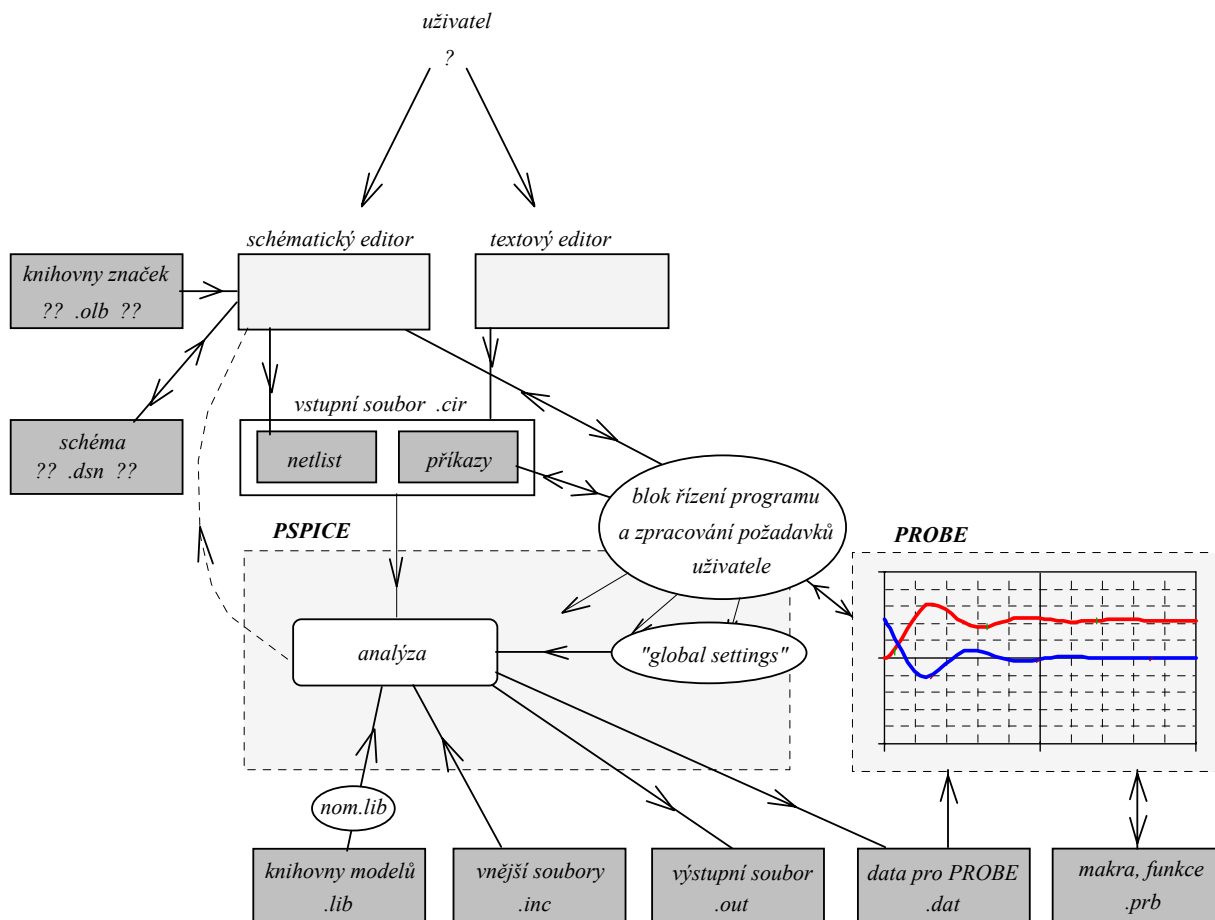
Označení „**Spice-like**“ znamená, že simulátor je schopen generovat podobné výsledky analýzy jako SPICE, avšak nemusí být schopen číst standardní vstupní soubory SPICE. Typickými příklady jsou staré verze programů MicroCap nebo TINA, program SABER apod.

Termínem „**Spice-compatible**“ se označují simulační programy, které dokáží číst standardní vstupní soubory SPICE, provádět klasické SPICE analýzy, a generovat výsledky v standardním SPICE2G.6 tvaru. Ze současných programů jsou to například PSpice, HSpice (standard SPICE3), WINSpice (standard SPICE3), MicroCap od verze IV, Multisim a další.

V tomto textu je pozornost věnována standardu PSpice.

3.2 Typická struktura simulačního programu založeného na PSPICE

Typická struktura simulačního programu typu OrCadPSPice je na **obr. 3.1**.



Obr. 3.1: Zjednodušená bloková struktura programu typu PSpice.

Uživatel má dvojí možnost zadávání modelu obvodu: buď prostřednictvím textového editoru (většinou zabudovaného přímo v simulačním programu), nebo prostřednictvím schématického editoru. Každý z těchto způsobů má své výhody i nevýhody. Hlavní výhodou používání schématického editoru je větší přehlednost modelu, zejména pokud jde o složitější obvody. Nevýhodou je mj. to, že schématických editorů je celá řada a jejich ovládání se může v novějších verzích simulátoru i podstatně lišit. Zatímco struktura textové informace, vytvářené textovým editorem, je v podstatě standardizovaná, nelze toto říci o struktuře pomocných souborů, které k práci využívá schématický editor (viz **obr. 3.1** - knihovny schématických značek, soubor s informací o schématu obvodu, apod.). Ovládání současné verze programu OrCadPSPice10 je poměrně velmi komplikované hlavně díky schématickému editoru, zatímco výpočetní jádro je již léta v podstatě neměnné a stejně snadno použitelné pro starší i novější verze PSPICE.

Z uvedených důvodů se bude převážná část výkladu týkat zadávání obvodu ve formě tzv. vstupního souboru (circuit file – přípona .cir) v textovém módu. Pak je možné pracovat v podstatě s libovolným volně šířitelným „Spice-compatible“ programem bez nutnosti vázat se na složitý systém OrCadPSPice a jeho „update“.

Jak uvidíme v další části, vstupní soubor, jakoukoliv metodou vytvořený, se skládá z tzv. netlistu – popisu modelu obvodu – a z příkazové části, která mj. obsahuje naše požadavky na analýzu obvodu. Tento soubor tvoří vstupní data pro výpočetní jádro – vlastní simulátor (na

obr. 3.1 označeno slovem PSPICE). Úkolem výpočetního jádra je řešit soustavu rovnic, které charakterizují analyzovaný obvod. K sestavení těchto rovnic jsou zapotřebí matematické modely součástek, z nichž je obvod sestaven. Některé „komplikovanější“ součástky mají své složité modely z praktických důvodů uloženy v souborech, kterým se říká knihovny modelů (standardní přípona .lib). Pro simulátor jsou však dostupné jen ty, jejichž názvy jsou vypsány v speciálním souboru nom.lib. Existují jednoduché postupy, jak umožnit simulačnímu programu číst různé modely - ať už modely dodávané s programem, umístěné v nejrůznějších souborech, nebo naše vlastní modely, buď speciálně napsané uživatelem nebo třeba stažené z Internetu.

Kromě knihoven může simulátor čerpat další data, ovlivňující jeho činnost, z dalších vnějších souborů, jejichž obsah je přičleněn (inc = include) k textové informaci ze vstupního souboru. Podrobněji o této možnosti bude pojednáno dále.

Činnost výpočetního jádra lze optimalizovat pomocí „ladicích parametrů“, které ovlivňují běh numerických algoritmů. Standardní přednastavení je vyhovující pro řešení běžných simulačních úloh. V některých případech však není simulátor schopen při standardním nastavení nalézt řešení nebo je řešení zatíženo chybami. Pak je nutné provést kvalifikovanou změnu v „global settings“.

Výsledky simulací, které jsou standardně v textovém tvaru, se automaticky zapisují do tzv. výstupního souboru (output file – přípona .out). V PSpice se navíc objevuje možnost využití výstupu dat pro jejich zobrazení v grafickém postprocesoru PROBE. Na rozdíl od výše uvedených souborů však jde o soubor v binárním tvaru. Jeho standardní přípona je .dat. Binární forma je volena s ohledem na větší kompresi dat, která mohou být značného objemu, zejména při náročných simulacích složitých obvodů. Nicméně existuje možnost, aby PSPICE generoval tato data v „srozumitelném“ ascii formátu a PROBE jim bude přesto rozumět.

Pro úplnost dodejme, že PSPICE může kromě těchto dvou výstupních souborů generovat další soubory s výsledky analýz. O těchto detailech bude pojednáno na příslušných místech.

Postprocesor PROBE využívá k své funkci pomocné soubory typu .prb, v nichž jsou mj. uloženy informace o předchozím nastavení postprocesoru a tzv. makra a měřicí funkce, usnadňující analýzu získaných křivek.

Pro zajímavost ještě uvedme, že v starších verzích PSpice bylo možné nalézt na disku nezávislé soubory pspice.exe a probe.exe. Ve verzi 10 se v souboru pspice.exe skrývá v podstatě grafický postprocesor probe a výpočetní jádro je schováno do jednoho z ovladačů.

3.3 Posloupnosti činností v simulačním programu

Obr. 3.2 ilustruje postupy modelování a simulace chování obvodů na příkladu jednoduchého článku RLC. Grafickým modelem reálného obvodu je schéma, v němž vyznačíme uzly. Protože PSpice interně popisuje obvody metodou uzlových napětí, je třeba v obvodu zvolit – v principu libovolně - tzv. referenční uzel a ten označit číslicí nula. Ostatní uzly se očíslovají přirozenými čísly nebo označí povolenými jmény (viz dále).

Na základě schématu a zadání požadavků na simulaci se vytvoří vstupní soubor, a to buď uživatelem, nebo automatizovaně programem. Tento soubor začíná vždy tzv. hlavičkou (popiskou), následuje tzv. netlist, pak příkazy pro simulaci, a vše je zakončeno příkazem .end.

Poté se spustí simulátor, na základě netlistu sestaví matematický model obvodu ve formě rovnic, a tyto rovnice vyřeší v závislosti na tom, co je požadováno analyzovat. Grafický postprocesor, pokud je v příkazové části požadavek na jeho aktivaci, pak zobrazí výsledky.

V příkladu na **obr. 3.2** je na 5 řádcích netlistu popsán celý obvod v jazyce PSPICE. Následuje požadavek na analýzu TRANSIENT (časové průběhy) s „přeskočením“ výpočtu stejnosměrného pracovního bodu (Skip Bias Point). Pak se vygenerují data pro zobrazení časových průběhů na kapacitou C_p a induktoru L_s .

Obr. 3.2: Schéma, vstupní soubor, netlist, simulace [1]

VSTUPNÍ SOUBOR (*.cir)

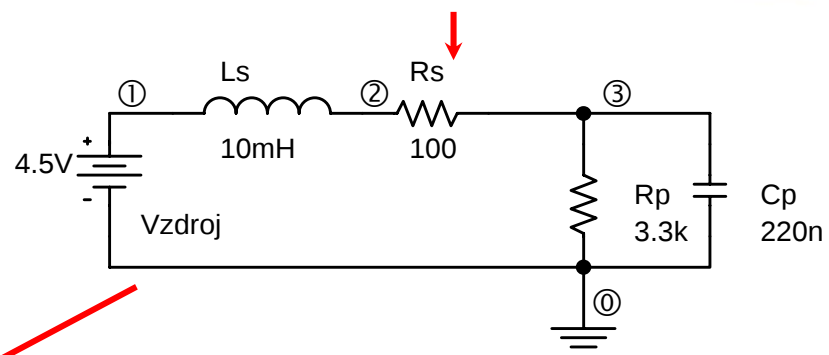
Hlavička
Popis obvodu = netlist
Příkazy pro řízení simulace
.END

požadavky, jaké vlastnosti obvodu se mají simulovat

SIMULÁTOR

Simulace chování obvodu podle dat ve vstupním souboru

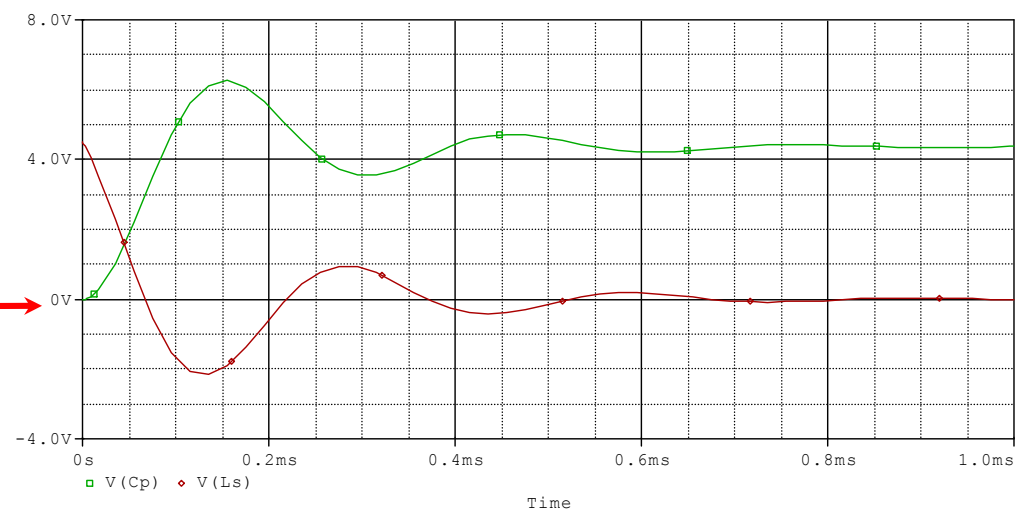
Pořadí zápisu není nutno dodržovat, je však doporučeno v zájmu přehlednosti



```

RLC obvod
*
Vzdroj 1 0 4.5V
Ls 1 2 10mH
Rs 2 3 100
Rp 3 0 3.3k
Cp 3 0 220nF
*
.TRAN 1u 1m SKIPBP
.PROBE V(Cp) V(Ls)
*
.end

```



3.4 Co vše lze analyzovat pomocí programů „SPICE – compatible“

Tyto programy nabízejí tzv. základní analýzy Transient, AC a DC (viz dále) a navíc tzv. rozšiřující typy analýz. Kromě toho lze spouštět analýzy v různých analyzačních režimech a módech. Příklad členění pro program MicroCap [8] je na obr. 3.3. Škrty jsou označeny funkce, které jsou v současných verzích PSpice nedosažitelné.

<i>analyzační módy</i>	<i>základní typy analýz</i>	<i>analyzační režimy</i>
- Klasický - Probe (sonda)	- Transient (časová)... + ss. prac. bod + Fourier. analýza a DSP - AC (kmitočtová, střídavá) + šumová analýza a DSP - DC (stejnoseměrná)	- Klasický - Stepping (krokování jednoduché a vícenásobné) - Teplotní analýza - Performance Analysis (vyhodnocovací analýza) - Monte Carlo a Worst Case (statistická analýza) - Optimization (optimalizace)
	<i>rozšiřující typy analýz</i> - Dynamic DC (dynamická stejnosměrná) - Dynamic AC (dynamická střídavá) - Transfer Function (přenosová funkce) - Sensitivity (citlivostní analýza) 	

Obr. 3.3: Typy analýz, analyzační režimy a módy pro „Spice-compatible“ program MicroCap. Škrty jsou vyznačena omezení PSpice.

O analyzačních režimech bude podrobně pojednáno na příslušných místech. Analyzační mód Probe (sonda) v MicroCapu umožňuje uživateli interaktivním způsobem zjišťovat časové i jiné průběhy v obvodu pouhým klikáním „sondy“ do jednotlivých uzlů ve schématu, což připomíná přikládání měřicích sond k rychlému testování obvodu.

Určitou zvláštností programu PSPICE je to, že lze spustit analýzu, i když nejsou zadány požadavky na to, co se má analyzovat. Pak PSPICE vypočítá to, co standardně počítá **vždy** – stejnosměrný pracovní bod obvodu.

Zmíňme hned zpočátku některá omezení PSPICE pro každou analýzu:

Odpor, kapacita a indukčnost nemohou nabývat nulových hodnot.

Ke každému uzlu musí být připojeny aspoň 2 prvky (výjimky z tohoto pravidla budou zmíněna později).

V obvodu musí být uzel „0“ – „uzemnění“.

Každý uzel musí mít stejnosměrnou cestu na zem.

3.5 Pravidla pro tvorbu textových souborů

- Malá a velká písmena se nerozlišují.
- Názvy všech obvodových prvků musí být jedinečné. Nelze např. nazvat dvě baterie stejně, i když mají stejná napětí.
- První řádek ve vstupním souboru je „hlavička“, PSPICE jej ignoruje.
- Prázdné řádky se ignorují.

- Vícenásobné mezery „navíc“ v textu se ignorují (místo jedné mezery jich může být více po sobě).
- Každý řádek (kromě prázdného řádku a prvního řádku) musí začínat jedním z těchto symbolů:
 - Klíčovým znakem určeným pro identifikaci obvodového prvku, např. R pro rezistor
 - * Obsah řádku je pak chápán jako poznámka.
 - + pokračování textu z předchozího řádku (pro lepší čitelnost v případě dlouhých řádek).
 - . Tečkou začínají dohodnuté příkazy.
- Umístíme-li kamkoliv na řádek středník ; , pak vše vpravo od něj je chápáno jako poznámka.
- Vstupní soubor musí končit řetězcem .END
- Uzly můžeme označovat přirozenými čísly (původní standard SPICE), nebo textovými řetězci, které nesmí obsahovat mezery (umožňuje většina současných verzí SPICE). Jeden z uzlů musí být označen číslem 0.
- Čísla vyjadřujeme v přirozeném tvaru (0.052) s desetinnou tečkou, nikoliv čárkou, nebo v exponenciálním tvaru (5.2E-2), nebo pomocí inženýrské notace (52m). V rámci řetězce se nesmí vyskytovat mezery. V inženýrské notaci jsou tyto zvláštnosti: M nebo m je mili, MEG je mega, u je mikro (velikost písmen je nevýznamná). **Textový řetězec za číslem se ignoruje (1u = 1uF).**

3.6 Shrnutí kapitoly 3

Řada současných programů pro počítačovou simulaci elektronických obvodů patří do tzv. rodiny SPICE programů. Některé z nich se pouze „tváří“ jako programy SPICE, protože mají schopnost produkovat potřebné výstupy analýzy, avšak nejsou schopny číst vstupní soubory standardu SPICE a nejsou tak tedy v plné míře se standardem SPICE kompatibilní. Tyto programy se označují přívlastkem „SPICE – like“. Patří k nim většina starších verzí současných programů. Programy, které navíc dokáží plnohodnotně zpracovat vstupní data formátu SPICE, vykazují v tomto směru úplnou kompatibilitu a označují se termínem „SPICE – compatible“. Pro uživatele těchto programů to představuje mj. výhodu v tom, že k modelování může využívat SPICE knihovny součástek, dostupných např. na Internetu.

Jedním z nejznámějších „SPICE – compatible“ simulátorů je program PSPICE, který je v současnosti distribuován firmou OrCad. Program OrCadPSpice je dodáván s profesionálním schématickým editorem a dalšími programovými nastaveními tak, aby jako celek představoval výkonný nástroj pro profesionální návrháře. Vlastní simulátor PSpice je jen malou součástí tohoto balíku. Zajímá-li nás pouze základní simulace obvodů a již ne dejme tomu optimalizace nebo počítačový návrh plošných spojů, vystačíme s jednoduššími „SPICE – compatible“ programy.

K simulaci programem PSpice nutně potřebujeme pouze „výpočetní jádro“, knihovny součástek a prostředek k tvorbě vstupního souboru, což může být buď jednoduchý textový editor nebo schématický editor (pokud již nejsou zabudovány přímo v simulačním programu).

K vizualizaci výsledků simulace v grafické formě potřebujeme vhodný postprocesor (pokud není součástí simulačního programu). Grafický postprocesor k programu PSPICE se nazývá PROBE.

„SPICE – like“ programy poskytují základní typy analýz Transient, DC a AC. Kromě toho většinou poskytují další speciální možnosti analýz, v nichž jsou výjimečné a kde překonávají standardní výstupy SPICE programů.

Pro počáteční studium počítačové simulace s programem OrCadPSpice doporučujeme práci na úrovni textového vstupního souboru, nikoliv na úrovni schématického editoru. Pak je třeba získat základní přehled o pravidlech, která je nutno dodržovat při tvorbě textových souborů SPICE. Až po detailnějším porozumění toho, jak simulátor funguje na úrovni příkazů, je vhodné přejít k práci se schématickým editorem, který řadu činností automatizuje, ovšem za cenu toho, že uživatel vlastně příliš nerozumí tomu, co dělá, a musí mnohdy vynakládat hodně úsilí k řešení nestandardních problémů, které by jinak snadno a rychle vyřešil, kdyby uměl pracovat na úrovni textových souborů. Typickým příkladem je začlenění modelu nové součástky do simulačního programu. Bez schématického editoru je to velmi snadné: nemusíme řešit problém tvorby nové schématické značky, přiřazení matematického modelu k této značce, a další podobné problémy, které vlastně s počítačovou simulací přímo nesouvisí...

3.7 Kontrolní otázky a příklady ke kapitole 3

Pokud dospějete k závěru, že na některou z níže uvedených otázek nedovedete vyčíst úplnou odpověď z tohoto elektronického textu, pokuste se dočerpat dodatečné informace z Internetu.

Správné odpovědi naleznete na konci učebního textu.

1. Na Internetu si vyhledejte informace o posledních verzích těchto simulačních programů: Multisim, TINA, SNAP, MicroCap, SABER. U každého rozhodněte, zda jde o program „SPICE-like“, „SPICE-compatible“ nebo zda není z rodiny SPICE.
2. Co představují zkratky SPICE2G.6, SPICE3F.2, PSPICE?
3. Přiřaďte správně standardy SPICE2, SPICE 3, PSPICE k programům HSPICE, WinSPICE, AimSPICE, OrCadPSpice.
4. Jaké jsou základní typy analýz poskytované programy „SPICE-like“ a „SPICE-compatible“?
5. Jaký je rozdíl mezi netlistem a vstupním souborem?
6. Jaké jsou možnosti tvorby vstupního souboru v OrCadPSPICE?
7. Může začínat vstupní soubor prázdným řádkem?
8. Co se stane, jestliže je ze vstupního souboru smazán první řádek s hlavičkou?
9. Co je to PROBE?
10. 10M znamená 10^6 , 10^7 , nebo 10^{-2} ?

4 Základy modelování v PSPICE, část 1

Cíle kapitoly:

-  Ukázat základy definice součástek v netlistu, bez a s využitím příkazu `.MODEL`.
-  Vysvětlit způsob práce s příkazy `.LIB`, `.INC`, `.TEMP`, `.PARAM`, `.STEP`, `.OP`.

4.1 Klíčové znaky analogových součástek

Z příkladu na obr. 3.2 je zřejmé, že na každém řádku netlistu je uvedena informace o jedné součástce v obvodu včetně toho, k jakým uzlům je připojena.

Prvním textovým řetězcem v definici součástky je její jméno. První znak ve jménu nazvěme klíčovým znakem. Podle klíčového znaku PSPICE pozná, o jaký typ součástky jde.

Přehled klíčových znaků analogových součástek v SPICE:

R rezistor **C** kapacitor **L** induktor

K .. přiřazení magnetické vazby mezi induktory nebo mag. jádra induktoru (induktorům)

V .. nezávislý zdroj napětí **.I** .. nezávislý zdroj proudu

E .. zdroj napětí řízený napětím

F.. zdroj proudu řízený proudem

G.. zdroj proudu řízený napětím

H.. zdroj napětí řízený proudem

D .. dioda

B .. tranzistor GaAsFET

J .. tranzistor JFET

M .. tranzistor MOSFET

Q .. bipolární tranzistor

Z .. IGBT tranzistor

(pozor – v SPICE3 jde o univerzální nelineární zdroj!!!)

← součástky vyžadující příkaz `.MODEL`, viz dále

S .. spínač řízený napětím

W .. spínač řízený proudem

T .. vedení

X .. podobvod (subobvod, funkční blok, makro)

Výše vyznačené součástky mají poměrně složitý model, který je třeba kvantifikovat několika číselnými parametry. Například model bipolárního tranzistoru obsahuje více než 40 takových parametrů. Součástí definice těchto součástek je příkaz `.MODEL`, obsahující definici těchto parametrů. Příkaz může být umístěn přímo ve vstupním souboru obvodu. Z praktických důvodů se ale doporučuje jeho umístění do knihovního souboru (`.lib`), na nějž je pak ze vstupního souboru odkázáno.

Obdobně definice podobvodu ve vstupním souboru musí být doplněna odkazem na zdrojový text podobvodu. Podrobnosti viz část 4.5.

4.2 Definice součástky v netlistu, která nevyužívá příkaz .MODEL

(např. pasivní součástky RLC a zdroje)

<jméno součástky> <seznam uzlů, na které je připojena> <její vlastnosti> [upřesnění]

Ostré závorky < > značí, že obsah, který ohraničují, je povinný.

Hranaté závorky [] značí, že obsah, který ohraničují, je nepovinný.

Jméno součástky musí začínat jejím klíčovým znakem.

Nejjednodušší možné syntaxe pasivních prvků R, L, C:

	Příklad
Rxx..xx <N+> <N-> <odpor>	Rochranny 3 5 150; odpor 150 ohmu mezi uzly 3 a 5
Lxx..xx <N+> N-> <indukčnost>	Ltlumivka 1 4 9H; inductor 9H mezi uzly 1 a 4
Cxx..xx <N+> N-> <kapacita>	Capacitor 2 0 15u; kapacitor 15uF mezi uzly 2 a 0

Napětí i proud se uvažují v orientaci od uzlu N+ do N-.

Nejjednodušší možné syntaxe nezávislých stejnosměrných zdrojů napětí a proudů:

	Příklad
Vxx..xx <N+> <N-> <napětí>	Vbaterie 1 0 4.5V; baterie o napeti 4.5V mezi uzly 1 a 0
Ixx..xx <N+> N-> <proud>	Ix 3 12 2m; zdroj o proudu 2mA mezi uzly 3 a 12

Napětí i proud se uvažují v orientaci od uzlu N+ do N- (pozor, není dodržována konvence o zdrojové orientaci čítacích šipek napětí a proudu!!!).

4.3 Definice součástky v netlistu, která využívá příkaz .MODEL

(zejména diody a tranzistory)

<jméno součástky> <seznam uzlů, na které je připojena> <jméno modelu> [upřesnění]

Seznam uzlů musí nyní být zapsán v dohodnutém pořadí, neboť např. záměna anody a katody by měla pro funkci obvodu fatální důsledky.

Příklady seznamu uzlů pro diodu a bipolární tranzistor:

<seznam uzlů> = <anoda katoda> ... dioda

<seznam uzlů> = <kolektor báze emitor> ... bipolární tranzistor

Jméno modelu si volíme libovolně. Model pak přesně specifikujeme příkazem .MODEL, který může být umístěn buď přímo v netlistu nebo v jiném souboru (většinou knihovně), na který „vidí“ SPICE.

Obecná syntaxe modelu:

.MODEL <jméno modelu> <kód modelu> [(] parametry modelu oddělené mezerami [)]

Kulaté závorky jsou nepovinné.

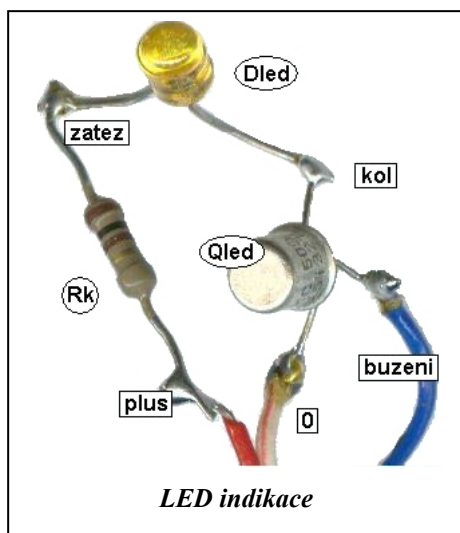
Vybrané kódy modelu:

D ..dioda, **NPN** ..bipolární tranzistor NPN, **PNP** ..bipolární tranzistor PNP.

Nepovinně lze model přiřadit i pasivním prvkům, např.:

CAP .. kapacitor, **RES** ..rezistor, **IND** ..induktor

Příklad, převzatý z [1]:



```
LED indikace
*
*zapojeni
*
Rk plus zatez 100
Dled zatez kol ledka
Qled kol buzeni 0 KC509
*
*modely soucastek
*
.model ledka D (N=1.7 RS=.7 CJO=23.9P
+ IS=85.3p BV=6 IBV=10U VJ=0.75 TT=4.32U)
.model KC509 NPN (BF=1.50K BR=2 IS=12.0P
+ CJC=2.80P CJE=1.40P RC=1.01U VAF=100
+ TF=718.4P TR=10N MJC=375.7M VJC=690.0M
+ MJE=375.77M VJE=690.00M NF=1.259
+ ISE=33.7P ISC=0.001566F IKF=9.44M
+ IKR=4.62K NE=2.18 RE=32.4M VTF=113.5
+ ITF=533.9K XTF=19.25M)
.end
```

Parametry modelů lze nalézt v manuálech. Každý typ modelu má číselnou sadu svých tzv. implicitních hodnot parametrů. Pokud daný parametr nedefinujeme příkazem **.MODEL**, platí implicitní hodnota. Pokud v příkazu neuvedeme žádný parametr, modelujeme součástku s implicitními parametry.

4.4 Příkazy **.LIB** a **.INC**

Modely prvků můžeme zapisovat přímo ve vstupním souboru, nebo – což je častější – využíváme modely, které jsou umístěny v různých souborech na disku, v tzv. knihovnách. Standardní přípona těchto souborů je **.lib**.

.LIB .. napojení knihovny modelů na vstupní soubor

.LIB [název souboru]

Příklady:

.LIB pokus.lib ; napojení na soubor s názvem pokus.lib

.LIB "c:\data\ana.lib" ; napojení na soubor v adresáři, na který normálně SPICE „nevidí“

.LIB NOM.LIB ;napojení na všechny knihovny, které jsou vypsány v souboru **NOM.LIB**

.LIB ;stejný význam jako **.LIB NOM.LIB** (**NOM.LIB** je tzv. Master Library)

Co všechno může být obsaženo v souboru *.lib:

Poznámky

Definice modelů (.MODEL)

Definice podobvodů (.SUBCKT)

Definice globálních parametrů (.PARAM)

Definice vlastních funkcí (.FUNC)

Příkazy .LIB

Podobný, ale ne stejný význam má příkaz .INC:

.INC .. včlenění obsahu vnějšího souboru

.INC <název souboru>

Příklady:

.INC zesilovac.cir začlenění celého netlistu do vstupního souboru

.INC "c:\data\pok.txt"

Význam .INC:

Efekt je stejný, jako kdyby se v místě příkazu .INC objevil text z včleněného souboru. Například ve vstupním souboru místo celého netlistu umístíme pouze odkaz na něj.

Co nesmí být obsaženo ve včleňovaném souboru:

Úvodní hlavička, pokud není označena jako poznámka (* nebo ;)

Je povoleno „vnořování“ těchto souborů do 4 úrovní.

4.5 Některé další základní příkazy, využívané pro modelování v PSPICE

Teplota součástek je standardně nastavena na 27 stupňů Celsia. Tuto tzv. globální teplotu lze změnit pomocí příkazu .TEMP.

.TEMP .. definice teploty (temperature)

.TEMP <výčet teplot oddělených mezerami>

Příklady:

.TEMP 58 ; analýza proběhne jednou, pro teplotu 58 stupňů Celsia

.TEMP -28 11 54.5 ; analýza proběhne opakovaně pro teploty -28, 11 a 54,5 stupňů Celsia

.PARAM .. definice globálních parametrů (proměnných)

.PARAM <název parametru>=<hodnota parametru>

Při použití parametru ve výrazu se parametr musí umístit do složených závorek { }.

Příklady:

.PARAM odpor1=1k odpor2=5.2k kapacita=5nF

Rparalelni vstup zem {odpor1*odpor2/(odpor1+odpor2)}

.STEP .. krokování, vícenásobná analýza pro různé hodnoty krokované proměnné

.STEP <název proměnné> LIST <výčet hodnot oddělených mezerami>

.STEP [LIN] < název proměnné > <počáteční hodnota> <konečná hodnota> <krok>

.STEP OCT < název proměnné > <počáteční hodnota> <konečná hodnota> <počet bodů>

.STEP DEC < název proměnné > <počáteční hodnota> <konečná hodnota> <počet bodů>

LIN (nepovinné): lineární krokování, t.j. další hodnota=současná hodnota+krok

OCT, DEC: logaritmické krokování, t.j. další hodnota=současná hodnota*číslo větší než 1.

Počet bodů u logaritmického krokování: číslo větší než 1 se zvolí tak, aby se každá oktáva (OCT) nebo dekáda (DEC) rozdělila na daný počet bodů.

Dekáda je interval ohraničený hodnotami, jejichž poměr je 10, např. 8 a 80.

Oktáva je interval ohraničený hodnotami, jejichž poměr je 2, např. 8 a 16.

Jaké jsou možné názvy krokované proměnné:

- Název nezávislého zdroje napětí nebo proudu. Krokují se napětí nebo proud tohoto zdroje.
- Typ modelu a jméno modelu, následované krokovaným parametrem v závorkách ().
- Teplota TEMP; Toto krokování nelze současně používat s příkazem .TEMP.
- Globální parametr, formát **PARAM=<název parametru>**. Rovnítko není povinné.

Příklady:

.STEP Vin 0V 3V 1V; analýza proběhne pro napětí zdroje Vin = 0V, 1V, 2V a 3V

.STEP PARAM=odpor LIST 1k 0.1K 2meg; analýza proběhne opakovaně pro 1kΩ, 100Ω, 2MΩ

.STEP DEC NPN Q1(BF) 10 1000 1; krokují se parametr BF tranzistoru Q1, BF=10, 100, 1000

.STEP TEMP 10 100 20; krokují se teplota v hodnotách 10, 30, 50, 70 a 90 stupňů Celsia

Současně nelze krokovat více proměnných – podstatné omezení SPICE.

.OP .. způsobí detailní výpis stejnosměrného pracovního bodu do výstupního souboru

Příklad:

.OP

Pozor, nejde o příkaz spouštějící analýzu, pouze jde o výpis výsledků. Pracovní bod se počítá vždy, i při chybějícím příkazu .OP.

Co se vypisuje bez příkazu .OP:

Uzlová napětí, proudy přes zdroje napětí, celkový ztrátový výkon.

Co se navíc vypisuje s příkazem .OP:

Linearizované malosignálové parametry polovodičových prvků a nelineárních řízených zdrojů.

4.6 Shrnutí kapitoly 4

Model každé součástky je v netlistu definován na samostatném řádku.

Definice začíná klíčovým znakem, podle kterého PSPICE identifikuje typ součástky. Nejčastěji používané klíčové znaky jsou R, L, C, V, I, E, Q, X (nalezněte si jejich význam a zapamatujte si).

Součástky s jednoduchými modely, např. rezistory, definujeme pouze řádkovým zápisem v netlistu. Složitější modely součástek obsahují řadu parametrů, které jsou deklarovány pomocí příkazu `.MODEL`. Tyto příkazy je možné vkládat buď přímo do vstupního souboru, nebo do speciálních knihovních souborů, na které se ze vstupního souboru odkážeme příkazem `.LIB`, případně `.INC`.

Je vhodné si pamatovat nejjednodušší syntaxe zápisu pasivních součástek typu R, L, C a zdrojů napětí a proudu.

Je vhodné si pamatovat způsob definice součástky, využívající příkaz `.MODEL`.

PSPICE analyzuje modely součástek při přednastavené teplotě 27 stupňů Celsia. Tuto teplotu lze změnit příkazem `.TEMP`.

Důležitými prostředky k modelování jsou globální parametry, kterými můžeme definovat konstanty nebo proměnné, používané ve vzorcích. Globální parametr se zavádí příkazem `.PARAM`.

Velmi praktickou vícenásobnou analýzu lze aktivovat příkazem `.STEP`, kdy dojde k tzv. krokování sledovaného parametru součástky. Analýza proběhne opakovaně tolikrát, kolikrát je parametr krokován. Pak můžeme sledovat závislost chování obvodu na parametru.

PSPICE vždy počítá stejnosměrný pracovní bod obvodu, i když uživatel nezadá žádný příkaz k analýze. Příkazem `.OP` můžeme vygenerovat do výstupního souboru detailnější výpis informací o nalezeném pracovním bodu.

4.7 Řešené příklady

Příklad 4.1 *Odporový dělič napětí, tvorba vstupního souboru*

Sestavte vstupní soubor pro výpočet napěťových a proudových poměrů v děliči napětí podle obrázku.

Řešení:

Odporový dělič

*

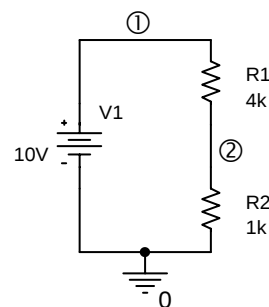
R1 1 2 4k; rezistor mezi uzly 1 a 2, odpor 4kohm

R2 2 0 1k; rezistor mezi uzly 2 a 0, odpor 1kohm

V1 1 0 DC 10V; zdroj stejnosměrného napětí 10V mezi uzly 1 a 0

*

.END



Není specifikována žádná analýza. Přesto proběhne výpočet ss pracovního bodu (implicitně proběhne vždy) s výsledkem

```
NODE VOLTAGE  NODE VOLTAGE  NODE VOLTAGE  NODE VOLTAGE
( 1) 10.0000 ( 2) 2.0000
```

Vyzkoumejte, co značí údaj TOTAL POWER DISSIPATION, ověřte vypočítanou hodnotu.
Vyzkoumejte, co značí údaj VOLTAGE SOURCE CURRENTS, ověřte vypočítanou hodnotu.
Jak je to se znaménkem?

Příklad 4.2 *Odporový dělič napětí, použití globálního parametru*

V předchozím příkladu zaveďte globální parametr Rpromenny, nastavte jej na hodnotu 4kohmy a hodnotu odporu R1 pak definujte tímto parametrem.

Řešení:

Odporovy delic

*

R1 1 2 {Rpromenny}

R2 2 0 1k

V1 1 0 10V

*

.PARAM Rpromenny=4k

*

.END

Přesvědčte se o tom, že výsledky analýzy budou stejné jako v příkladu 4.1.

Příklad 4.3 *Odporový dělič napětí, krokování globálního parametru*

Pomocí příkazu .STEP proveďte opakovanou analýzu napěťových a proudových poměrů v děliči napětí pro odpor R1 v hodnotách (1, 2, 4)kohmy.

Řešení:

Odporovy delic

*

R1 1 2 {Rpromenny}

R2 2 0 1k

V1 1 0 10V

*

.PARAM Rpromenny=4k

.STEP param=Rpromenny LIST 1k 2k 4k

*

.END

Snažte se pochopit, co vyvolal příkaz .STEP (opakovaný výpočet stejnosměrného pracovního bodu). Ověřte výsledky.

Nastudujte další metody krokování parametrů, metoda LIST není jediná.

Zkuste do seznamu krokovaných hodnot zařadit odpor 0. Není povoleno!

Příklad 4.4 *Obvod s diodou, výpočet napětí a proudů*

Napište vstupní soubor pro analýzu poměrů v obvodu s diodou podle obrázku. Uvažujte „typickou“ polovodičovou diodu s implicitními parametry.

Řešení:

Obvod s diodou

*

R1 1 2 4k; rezistor mezi uzly 1 a 2, odpor 4kohmy

D1 2 0 obycejna_dioda; dioda mezi uzly 2 a 0, jmenuje se

+ obycejna_dioda, anodu ma na uzlu 2,

+ katodu na uzlu 0

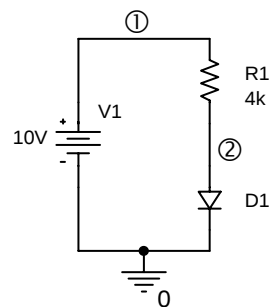
V1 1 0 10V; zdroj stejnosmerneho napeti 10V mezi uzly 1 a 0

*

.model obycejna_dioda D; diode obycejna_dioda je prirazen implicitni model diody D

*

.END



V řádku s příkazem .model nejsou za písmenem D uvedeny parametry modelu diody. PSPICE proto při simulaci přebírá všechny přednastavené (implicitní) parametry, viz manuál.

Spusťte simulaci a zjistěte napětí na diodě. Odpovídá to vaší představě o velikosti úbytku napětí na otevřené diodě?

Přidejte nad příkaz .END příkaz .OP. Co se stane?

Pro jakou teplotu součástek vlastně analýza probíhá?

Příklad 4.5 *Obvod s diodou, analýza pro teplotu 50 °C*

Modifikujte vstupní soubor z příkladu 4.4 tak, aby PSPICE provedl analýzu obvodu při teplotě 50°C.

Řešení:

Obvod s diodou

*

R1 1 2 4k

D1 2 0 obycejna_dioda

V1 1 0 10V

*

.model obycejna_dioda D

*

.TEMP 50

.OP

*

.END

Změnilo se napětí na diodě? Vzrostlo nebo pokleslo? Proč?

Příklad 4.6 *Obvod s diodou, krokování teploty*

Modifikujte vstupní soubor z příkladu 4.5 tak, aby PSPICE provedl vícenásobnou analýzu obvodu při teplotách (27, 50, 100)°C.

Řešení:

Obvod s diodou

*

R1 1 2 4k

D1 2 0 obycejna_dioda

V1 1 0 10V

.model obycejna_dioda D

.TEMP 27 50 100

*

.END

Příklad 4.7 *Obvod s diodou s pozměněnými parametry diody*

Modifikujte vstupní soubor z příkladu 4.4 tak, aby PSPICE provedl analýzu obvodu s diodou, která bude mít konkrétní parametry $I_s=80\text{pA}$ (saturační proud) a $R_s=2\text{ohmy}$ (sériový odpor).

Řešení:

Obvod s diodou

*

R1 1 2 4k

D1 2 0 obycejna_dioda

V1 1 0 10V

.model obycejna_dioda **D(Is=80p Rs=2)**

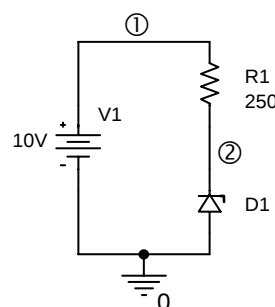
*

.END

Nalezněte si v dokumentaci PSPICE implicitní hodnoty parametrů I_s a R_s . Prozkoumejte, nakolik se změní výsledky analýzy oproti modelu s implicitními parametry. Pokuste se o zdůvodnění.

Příklad 4.8 *Stabilizátor se Zenerovou diodou*

Napište vstupní soubor pro stabilizátor se Zenerovou diodou na obr. Jde o diodu 1N750. Model diody lze získat buď vyhledáním na Internetu, nebo může být přímo v některé z knihoven PSPICE. Například v programu OrCadPSpice10 se model nazývá D1N750 a nachází se v knihovně EVAL.LIB, pokud pracujete s evaluační verzí, nebo v knihovně DIODE.LIB, která je součástí profí verze. Model okopírujte přímo do vstupního souboru.



Řešení:

Stabilizátor se Zenerovou diodou

*

R1 1 2 **250**; rezistor mezi uzly 1 a 2, odpor 250ohmu

D1 **0 2 D1N750**; Zenerova dioda polovana v zaverne smeru,

+

Zenerovo napeti je asi 4.7V pri proudu 20mA

V1 1 0 10V; zdroj stejnosmerneho napeti 10V mezi uzly 1 a 0

*

```
.model D1N750 D(Is=880.5E-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p M=.5516
+      Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=4.7 Ibv=20.245m Nbv=1.6989
+      Ibvl=1.9556m Nbv1=14.976 Tbv1=-21.277u)
*
.END
```

Pomocí výsledků simulace zjistíte napětí na diodě a proud diodou.

Příklad 4.9 Stabilizátor se Zenerovou diodou, napojení na model z knihovny

Modifikujte vstupní soubor z příkladu 4.8 tak, abyste přímo využívali model diody z „externí“ knihovny bez nutnosti kopírování modelu do vstupního souboru.

Řešení:

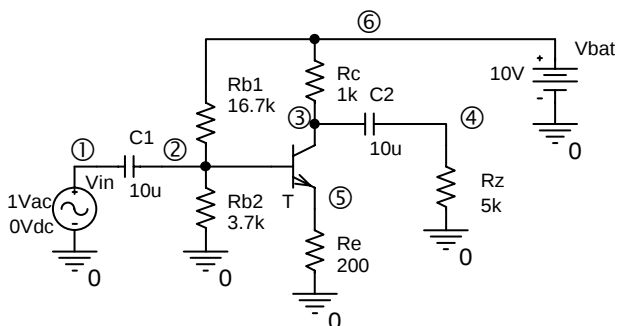
Stabilizátor se Zenerovou diodou

```
*
R1 1 2 250;
D1 0 2 D1N750
V1 1 0 10V
*
.lib eval.lib
*
.END
```

Místo **eval** případně napíšeme správný název knihovny s diodou D1N750.

Příklad 4.10 Jednostupňový tranzistorový zesilovač

Napište vstupní soubor pro analýzu stejnosměrných poměrů v tranzistorovém zesilovači na obrázku. Uvažujte tranzistor s implicitními parametry.



Řešení:

zesilovac

```
*
Vin 1 0 AC 1; význam parametru AC bude objasněn později,
+      na výsledek stejnosměrné analýzy nemá vliv
Cv1 1 2 10uF
Rb1 6 2 16.7k
Rb2 2 0 3.7k
QT 3 2 5 obycejny_tranzistor; pořadí vývodů tranzistoru: kolektor, báze, emitor
Rc 6 3 1k
Re 5 0 200
```

```
Cv2 3 4 10u
Rz 4 0 5k
Vbat 6 0 10V
*
```

```
.model obycejny_tranzistor NPN; NPN je jméno modelu bipolárního tranzistoru
.OP
*
.end
```

Opište si souřadnice vypočteného pracovního bodu, parametry BF, betaDC, betaAC tranzistoru. Poznatek: BF (Beta Forward) není přesně rovno „betě“ tranzistoru, od tohoto čísla se však proudové zesilovací činitele odvíjejí.

Pomocí odečteného parametru betaDC ověřte výpočtem daný pracovní bod.

Všimněte si nezvykle velkého napětí mezi bází a editorem. Ne všechny implicitní hodnoty parametrů tranzistoru jsou „optimální“ pro konkrétní aplikaci. Pro konkrétní aplikaci se vyplatí použít model konkrétní součástky – viz dále.

Příklad 4.11 *Jednostupňový tranzistorový zesilovač, krokování parametru BF tranzistoru*

Zjistěte změny pracovního bodu zesilovače z příkladu 4.10 po výměně tranzistoru za jiný s parametrem BF 100, 200 a 500.

Řešení:

```
zesilovac
*
Vin 1 0 AC 1 sin 0 10mV 1kHz
Cv1 1 2 10uF
Rb1 6 2 16.7k
Rb2 2 0 3.7k
QT 3 2 5 obycejny_tranzistor
Rc 6 3 1k
Re 5 0 200
Cv2 3 4 10u
Rz 4 0 5k
Vbat 6 0 10V
*
.param BF=100
.model obycejny_tranzistor NPN BF={BF}
.step param=BF list 100 200 500
.OP
*
.end
```

Všimněte si, že parametry modelu nemusí být v závorkách, neboli že jsou přípustné oba zápisy:

```
.model obycejny_tranzistor NPN BF={BF}
.model obycejny_tranzistor NPN (BF={BF})
```

Ověřte si, že proud kolektoru a napětí mezi kolektorem a zemí se mění relativně málo, i když parametr BF vykazuje velký rozptyl. Příčinou je působení stabilizující záporné zpětné vazby v obvodu (odpor R_e).

Příklad 4.12 *Jednostupňový tranzistorový zesilovač s tranzistorem 2N2222*

Najděte v knihovnách model tranzistoru 2N2222 a nahraďte jím původní implicitní model tranzistoru v zesilovači z příkladu 4.10. Buď si model překopírujte do vašeho vstupního souboru nebo se napojte na knihovnu příkazem `.lib`.

Řešení:

```

zesilovac
*
Vin 1 0 AC 1 sin 0 10mV 1kHz
Cv1 1 2 10uF
Rb1 6 2 16.7k
Rb2 2 0 3.7k
QT 3 2 5 Q2N2222
Rc 6 3 1k
Re 5 0 200
Cv2 3 4 10u
Rz 4 0 5k
Vbat 6 0 10V
*
.model Q2N2222 NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=255.9 Ne=1.307
+      Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1
+      Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75
+      Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
*
.OP
*
.end

```

Ověřte, že klidové napětí báze-emitor je nyní 0,688V.

Doporučujeme zopakovat s tímto tranzistorem experimenty z příkladu 4.10.

4.8 Kontrolní otázky ke kapitole 4

Pokud dospějete k závěru, že na některou z níže uvedených otázek nedovedete vyčíst úplnou odpověď z tohoto elektronického textu, pokuste se dočerpát dodatečné informace z dalších zdrojů.

Správné odpovědi naleznete na konci učebního textu.

1. Rozhodněte, který z uvedených vstupních souborů SPICE je správně napsán (jde o kompletní výpis):

a) zdroj

```
Vin 1 2 3
.end
```

b) ;zdroj

```
V a 0 -5V
.end
```

c) *zdroj

```
Vx in out value=5
.end
```

2. Bezchybný zápis modelu bipolárního tranzistoru je:

a) .MODEL T1 NPN, b) .model Q1 1 2 3 NPN (), c) .Model tran c b e NPN (BF=100)

3. Knihovna modelů SPICE

a) může, b) musí, c) nesmí
obsahovat hlavičku.

4. Příkaz .OP způsobí výpis těchto dat do výstupního souboru:

a) jen napětí všech uzlů v pracovním bodu, b) kromě standardního výpisu se vypíše detailní parametry polovodičových prvků a nelineárních řízených zdrojů v pracovním bodu, c) stejný výpis jako při vynechání tohoto příkazu.

5. Krokování teploty v hodnotách 10, 20, 30, 40 a 50 stupňů Celsia se zabezpečí příkazem

a) .TEMP 10 50 10, b) .TEMP LIST 10 20 30 40 50, c) .TEMP 10 20 30 40 50

6. Rozhodněte, kolik chyb je třeba odstranit ze vstupního souboru, aby SPICE provedl simulaci (jde o kompletní výpis):

```
delic
Vin 1 0 3
R1 1 2 {r}
R2 2 0 {1k-r}
.param r 100
.step r list 0 1k
.end
```

a) 3, b) 2, c) 1 Pozn.: Chyby vyznačte s návrhem na jejich odstranění.

7. Uvažujte obvod, popsany tímto vstupním souborem:

```
obvod
Ix 1 0 1mA
Rx 1 0 1k
.end
```

Napětí V(1) vyjde:

a) 1V, b) 0V, c) -1V.

8. Řádek v netlistu

```
.STEP DEC NPN Q1(BF) 10 1000 1
```

znamená krokování parametru BF tranzistoru Q1. Při krokování bude parametr nabývat hodnot:

a) 10, 11, 12, ..., 1000, b) 10, 100, 1000, c) 10, 1000

9. Odpor 1 megaohm se v netlistu definuje takto:

a) Rx a b 1M, b) Rx a b 1megaohm, c) Rx a b value=1MEG

10. Bipolární tranzistor má uzemněný emitor. Báze s kolektorem jsou spojeny do uzlu 8. Příslušná část netlistu bude:

a) T8 8 0 0 tr, b) Q8 8 8 0 Q, c) Q 8 8 8 tran

11. Označte správnou syntaxi krokování globálního parametru:

a) .STEP param Rx 1k 2k 1k, b) .STEP LIN Rx 1k 2k 1k, c) .STEP DEC param Rx 0 1k 10

12. Příkaz .LIB znamená totéž co příkaz:

a) .LIB MASTER.LIB, b) .LIB NOM.LIB, c) bez příkazu

13. Tranzistor s implicitními parametry namodelujeme takto:

a) .MODEL Qx 1 2 3 PNP, b) .model tran npn, c) .MODEL Qtran Xnpn

4.9 Neřešené příklady

Výsledky řešení naleznete v příloze.

Příklad 4.13 *Dělič napětí z příkladu 4.1*

Proveďte následující experimenty se vstupním souborem z příkladu 4.1:

- 1) Vynechejte u zdroje V1 výraz DC. Výsledek bude stejný. Proč? (syntaxe!!!)
- 2) U zdroje V1 zaměňte pořadí uzlů, tedy 0 1. Výsledná napětí budou záporná. Proč?
- 3) Přidejte před poslední příkaz .END příkaz .OP. Ve výsledku není žádná změna. Proč?
- 4) Vyzkoušejte různé jiné názvy než R1, R2, V1. Vyzkoušejte si, že důležité je neměnit první identifikační znak ve jméně a že to funguje i bez dalších znaků, tedy že jsou povolena i jména R a V.

Příklad 4.14 *Zatížený dělič napětí*

Na výstupní svorky děliče napětí z příkladu 4.1 připojte rezistor o odporu 1kohm. Napište vstupní soubor pro výpočet výstupního napětí. Zjistěte velikost tohoto napětí.

Příklad 4.15 *Modelování potenciometru*

Na odpory R1 a R2 v děliči napětí z příkladu 4.1 můžeme pohlížet jako na dvě části potenciometru o odporu R1+R2 a s jezdcem, který je vyveden na spojnici mezi R1 a R2. Na základě této představy modelujte obvod z příkladu 4.1, obsahující potenciometr o odporu 5kohmů. „Otáčení“ hřídele potenciometru mezi krajními polohami modelujte globálním parametrem alfa, který nabývá hodnot v intervalu (0,1): 0 znamená, že R1=0, R2=5kohmů, 1 značí, že R1=5kohmů, R2=0. Proveďte výpočet výstupního napětí děliče pro alfa v krocích 0.1, 0.5, 0.9. Výsledky ověřte ručními výpočty.

Příklad 4.16 *Obvod s diodou z příkladu 4.4*

Analyzujte obvod s diodou z příkladu 4.4, jestliže v původním obvodu „otočíme“ diodu, tj. zaměníme anodu a katodu. Vypočítejte napětí na diodě a ověřte váš předchozí odhad tohoto napětí.

Příklad 4.17 *Obvod s diodou z příkladu 4.6*

Analyzujte obvod s diodou z příkladu 4.6 za účelem zjištění teplotního součinitele napětí (poměru změny napětí na diodě a změny teploty, která zapříčiňuje změnu napětí). Teplotní součinitel zjistěte analýzou při teplotách 27°C a 37°C.

Příklad 4.18 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

Paralelně k stabilizační diodě z příkladu 4.8 připojte zátěž o odporu 1 kohm. Zjistěte, zda jsou při této zátěži ještě zachovány stabilizační schopnosti obvodu. Krokováním tohoto odporu odhalte kritickou hodnotu zátěže, při níž již stabilizace selhává.

Příklad 4.19 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

Zjistěte teplotní součinitel výstupního napětí stabilizátoru z příkladu 4.8: vypočtěte stabilizované napětí při teplotě 27 stupňů (V27) a pak 37 stupňů (V37), součinitel=(V37-V27)/V27/10.

Příklad 4.20 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

Zjistěte činitel stabilizace $\Delta V(1)/\Delta V(2)$ u obvodu z příkladu 4.8. Návod - Vypočtěte výstupní napětí při vstupním napětí 10V a pak při 20V. Můžete provést opakovaný výpočet nebo využít příkaz .STEP.

Příklad 4.21 *Jednostupňový tranzistorový zesilovač z příkladu 4.10*

Jak donutit PSPICE, aby vypsál do výstupního souboru proud rezistorem Rb2? PSPICE vypisuje proudy procházející zdroji napětí. Proto umístíme do série s Rb2 zdroj napětí, ale jeho napětí nastavíme na 0V, aby nebyla ovlivněna funkce zesilovače. Jaký je proud procházející tímto zdrojem?

Příklad 4.22 *Jednostupňový tranzistorový zesilovač z příkladu 4.10*

Zjistěte stabilitu pracovního bodu zesilovače při změnách teploty. Konkrétně pro BF=100 „změřte“ klidový kolektorový proud pro teplotu 20 stupňů (Ic20) a 30 stupňů (Ic30) a pak vypočtěte činitel teplotní stabilizace $100 \cdot (Ic30 - Ic20) / (10 \cdot Ic20)$, tj. procentuální změnu Ic při změně teploty o 1 stupeň.

5 Základy modelování v PSPICE, část 2

Cíle kapitoly:

-  Objasnit význam podobvodu SPICE, jeho definování a využívání k modelování relativně samostatných obvodových bloků.
-  Seznámit se základními matematickými funkcemi, zabudovanými v PSPICE, které mohou usnadnit tvorbu vlastních podobvodů SPICE.
-  Vysvětlit filozofii tzv. behaviorálního modelování a naučit se využívat řízené zdroje a další nástroje tohoto modelování.

5.1 Práce s podobvody SPICE

Podobvod (subcircuit) nebo také makroobvod je relativně samostatný obvod, který komunikuje s okolím prostřednictvím konkrétních vývodů. Nejprve jej musíme definovat (*deklarovat*), tj. vytvořit jeho model, uložit jej do dané knihovny nebo přímo do vstupního souboru, a pak jej můžeme využívat (*volat*) jako každou jinou součástku.

Pokud podobvod deklarujeme přímo ve vstupním souboru analyzovaného obvodu, tento model podobvodu bude k dispozici jen pro analýzu tohoto obvodu.

Pokud podobvod umístíme do externí knihovny knihovna.lib, pak model podobvodu bude k dispozici každému, kdo může využívat tuto knihovnu. Na knihovnu se napojíme tak, že do vstupního souboru umístíme příkaz

```
.lib knihovna.lib
```

Podobvod může být sestavený ze součástek, jejichž parametry můžeme měnit při volání podobvodu. Pak se jedná o volání s parametrem (parametry). Této možnosti se běžně příliš nevyužívá. Asi nejvíce se podobvodů využívá k modelování konkrétních typů operačních zesilovačů a dalších integrovaných obvodů nejrozumnějších výrobců.

Deklarace podobvodu - syntaxe:

```
.SUBCKT <jméno podobvodu> <seznam vývodů> [params: <parametr=číselná hodnota>]
```

```
<popis vnitřního zapojení>
```

```
.ENDS
```

Použití (volání) podobvodu - syntaxe:

```
Xxx.xx <seznam uzlů podo..u> <jméno podo..u> [params: <parametr>=<číselná hodnota>]
```

Příklad – podobvod bez parametrů:

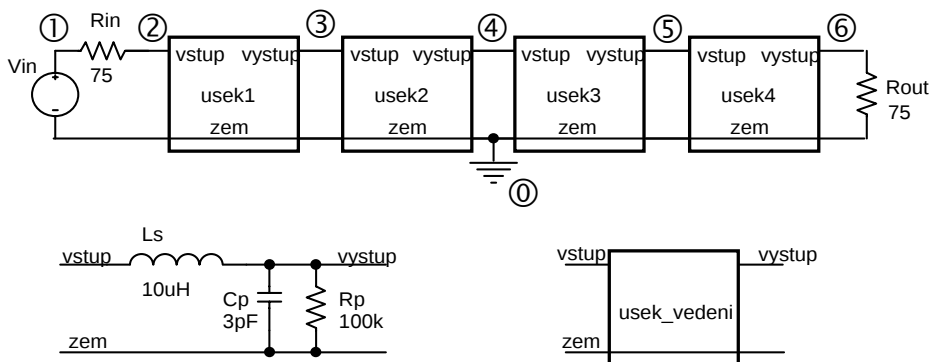
model vedeni

Vin 1 0 1V

Rin 1 2 75ohmu

Xusek1 2 3 0 usek_vedeni**Xusek2 3 4 0 usek_vedeni****Xusek3 4 5 0 usek_vedeni****Xusek4 5 6 0 usek_vedeni**

Rout 6 0 75ohmu



*definice podobvodu

.subckt usek_vedeni vstup vystup zem

Ls vstup vystup 10uH

Cp vystup zem 3pF

Rp vystup zem 100kohmu

.ends usek_vedeni**.end**

Nepovinné, ale je dobrým zvykem zařazovat za **.ends** název „končícího“ podobvodu. Přispívá to k větší přehlednosti.

Příklad – podobvod s parametry:

.....

Xusek1 2 3 0 usek_vedeni params: Ls=10u Cp=3p Rp=100k

.....

.subckt usek_vedeni vstup vystup zem params: indukcnost=10u kapacita=3p

+

odpor=100k

Ls vstup vystup {indukcnost}

Cp vystup zem (kapacita)

Rp vystup zem {odpor}

.ends usek_vedeni**.end****5.2 Funkce zabudované v PSPICE pro modelování prvků**

Výběr:

funkce	význam	poznámka
ABS(x)	$ x $	
COS(x)	$\cos(x)$	x je v radiánech
DDT(x)	dx/dt	jen u analýzy TRANSIENT
EXP(x)	e^x	
IF(a,x,y)	x když a=TRUE, jinak y	a je logický výraz, například IF(I(L1)>0,V(1),-V(1))
IMG(x)	imaginární část x	
LIMIT(x,MIN, MAX)	MIN když x<MIN, MAX když x>MAX, jinak x	oboustranný okrajovač
LOG(x)	$\ln(x)$	základ e
LOG10(x)	$\log(x)$	základ 10
M(x)	$ x $	modul komplexního čísla x

MAX(x,y)	větší z čísel x a y	
MIN(x,y)	menší z čísel x a y	
P(x)	argument komplexního čísla	
PWR(x,y)={x**y}	$ x ^y$	mocninná funkce
PWRS(x,y)	$ x ^y$ když $x>0$, $- x ^y$ když $x<0$	mocninná funkce se znaménkem
R(x)	reálná část x	
SDT(x)	$\int_0^t x(t)dt$	jen u analýzy TRANSIENT
SGN(x)	1 pro $x>0$, -1 pro $x<0$, 0 pro $x=0$	znaménková funkce
SIN(x)	$\sin(x)$	x je v radiánech
STP(x)	1 když $x>0$, jinak 0	jednotkový skok
SQRT(x)	$\sqrt{x}$	
TABLE(x,x1,y1,x2,y2,...,xn,yn)	Graf funkce vznikne pospojováním bodů o souřadnicích [xk,yk] lomenou čarou. Pro $x \leq x1$ je výsledek y1, pro $x > xn$ je výsledek yn.	

Kromě toho lze definovat vlastní funkce s využitím zabudovaných funkcí. K tomu slouží příkaz .FUNC.

5.3 Vytváření vlastních funkcí

.FUNC <jméno> ([argumenty]) {<vzorec>}

Před vzorcem je přípustné i rovnítko, složené závorky nejsou nutné.

Příklady:

.FUNC LIMITER(x,gain) {LIMIT(gain*x,-15,15)} ; omezovač se zesílením GAIN, signál x větší než +15V a menší než -15V ořeže
 .FUNC MIN3(A,B,C) {MIN(A,MIN(B,C))} ; minimální z čísel A, B, C

Funkce lze psát do zvláštního souboru. Ze vstupního souboru na ně bude „vidět“ po vložení příkazu

.INC soubor

Jednou z užitečných aplikací funkcí – ať už zabudovaných nebo uživatelských – je modelování vlastností obvodových prvků nebo celých bloků rovnicemi. Hovoříme o behaviorálním modelování: ABM = Analog Behavioral Modeling. Například celý integrovaný zesilovač se zesílením $A=100$ můžeme zjednodušeně modelovat blokem o dvojici vstupních a výstupních svorek, na nichž jsou definována napětí V_{in} a V_{out} , a mezi nimiž platí vzorec $V_{out}=100V_{in}$.

Důležitým nástrojem pro behaviorální modelování jsou tzv. řízené (závislé) zdroje.

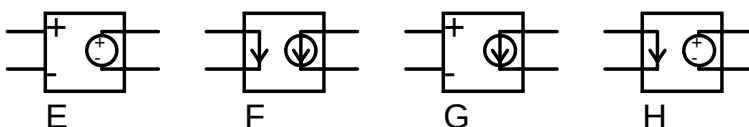
5.4 Řízené (závislé) zdroje pro behaviorální modelování (ABM –Analog Behavioral Modeling)

E..zdroj napětí řízený napětím

F..zdroj proudu řízený proudem

G..zdroj proudu řízený napětím

H..zdroj napětí řízený proudem



Napětí na svorkách i proudy uvnitř zdrojů jsou orientovány od svorky + do svorky -.

Zdroje E a G mají stejnou syntaxi. Níže vysvětleno zejména pomocí zdrojů E.

1) Zdroj s řídicími svorkami, lineární zařízení:

E<název> <výstup+> <výstup-> <vstup+> <vstup-> <přenos>

Vzorec: $\text{výstup} = \text{přenos} * \text{vstup}$

Příklady:

Ezesil 2 0 1 0 100 ; lineární zesilovač, vstup 1-0, výstup 2-0, zesílení 100

GOTA out 0 plus minus {gm} ; OTA zesilovač s výstupem out, vstupy plus-minus
+ a transkonduktancí gm, která je definována jako globální
+ parametr

Přenos může být i vzorec sestavený z konstant a globálních parametrů.

2) Zdroj s řídicími svorkami, nelineární zařízení:

E<název> <výstup+> <výstup-> POLY(<dimenze>) {<vstup+> <vstup->} {<koefficient>}

seznam dvojic řídicích svorek

seznam koeficientů polynomu

Vzorec: Nelineární polynomiální závislosti výstupu na řídicích vstupech, viz tabulka

exp. u	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	...
x1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
x1 x2		1 0	0 1	2 0	1 1	0 2	3 0	2 1	1 2	0 3	4 0	3 1	2 2	1 3	0 4	...
x1 x2 x3		1 0	0 1	0 0	2 0	1 1	1 0	0 2	0 1	0 0	3 0	2 1	2 0	1 2	1 1	...
...		0	0	1	0	0	1	0	1	2	0	0	1	0	1	...

Dimenze je rovna počtu řídicích vstupů.

a) Dimenze = 1, je jeden vstup x1, výstup y:

$$y = p_0 + p_1 x_1 + p_2 x_1^2 + \dots$$

Příklady:

Ekvadrat 2 0 POLY(1) 1 0 0 0 0.1 ; vstupní napětí mezi 1 a 0, výstupní napětí mezi 2 a 0. Jde
+ o kvadrátor o vzorci $V(2)=V(1)*V(1)/10$.

Ex out 0 POLY(1) in1 in2 0.5 -0.5 ; vstupní napětí mezi in1 a in2, výstupní napětí mezi out a
+ 0. Funkční měnič $V(\text{out})=(1-V(\text{in1},\text{in2}))/2$.

b) Dimenze = 2, jsou dva vstupy x1 a x2, výstup y:

$$y = p_0 + p_1 x_1 + p_2 x_2 + p_3 x_1^2 + p_4 x_1 x_2 + p_5 x_2^2 + p_6 x_1^3 + p_7 x_1^2 x_2 + p_8 x_1 x_2^2 + p_9 x_2^3 + \dots$$

Příklady:

Emulti 3 0 POLY(2) 1 0 2 0 0 0 0 0 .1 ; dvojice vstupních napětí mezi 1-0 a 2-0, výstupní
+ napětí mezi 3-0. Jde o násobičku $V(3)=V(1)*V(2)/10$.

Esum out 0 POLY(2) in1 0 in2 0 0 1 1 ; dvojice vstupních $V(in1)$ a $V(in2)$, výstupní napětí
+ $V(out)$. Sumátor $V(out)=V(in1)+V(in2)$.

3) Zdroj bez řídicích svorek, výstupní napětí je modelováno vzorcem:

E<název> <výstup+> <výstup-> VALUE [=] {<vzorec>}

Rovnička je nepovinné.

Příklady:

Ekvadrat out 0 VALUE={ $V(in)^2/10$ } ; kvadrátor o vzorci $V(out)=V(1)*V(1)/10$.

Ex 2 0 VALUE { $10*(-0.5+STP(V(1)))$ } ; komparátor: $V(2)$ je buď +5V nebo -5V podle toho,
+ zda $V(1)$ je kladné nebo záporné.

4) Zdroj bez řídicích svorek, výstupní napětí je modelováno po úsecích lineární závislosti:

E<název> <výstup+> <výstup-> TABLE {<vzorec>} [=] (seznam dvojic <xk yk>)

Příklady:

Ekvadrat out 0 VALUE={ $V(in)^2/10$ } ; kvadrátor o vzorci $V(out)=V(1)*V(1)/10$.

Ex 2 0 VALUE { $10*(-0.5+STP(V(1)))$ } ; komparátor: $V(2)$ je buď +5V nebo -5V podle toho,
+ zda $V(1)$ je kladné nebo záporné.

Ex 2 0 TABLE { $V(1)$ } (-4 -5 4 5) ; pro $V(1)<-4V$ je $V(2)=-5V$,

+ pro $V(1)>4V$ $V(2)=5V$

+ pro $-4V<V(1)<4V$ je graf dán spojnicí obou bodů

Ex 2 0 TABLE { $10*V(1)$ } (-4 -5 4 5) ; pro $10*V(1)<-4V$ je $V(2)=-5V$,

+ pro $10*V(1)>4V$ $V(2)=5V$

+ pro $-4V<10*V(1)<4V$ je graf dán spojnicí bodů

+ [-0.4, -5], [0.4, 5]

Poznámka:

Tyto zápisy jsou ekvivalentní: (-4 -5 4 5) a (-4 -5) (4 5) a (-4, -5, 4, 5) a (-4, -5) (4, 5) atd.

5) Zdroj bez řídicích svorek, výstupní signál je modelováno klíčovými slovy LAPLACE nebo
FREQ:

Bude popsáno později v souvislosti s analýzou AC.

Zdroje F a H mají stejnou syntaxi. Níže vysvětleno zejména pomocí zdrojů F.

Jsou to zdroje napětí (H) a proudu (F), řízené **proudem**. Proud je snímán vždy na nezávislém zdroji napětí, který musí být specifikován v definici řízeného zdroje namísto řídicích svorek. Chceme-li, aby se tento zdroj choval jako ideální „ampérmetr“, musíme jeho napětí položit rovné nule.

1) Lineární řízený zdroj:

F<název> <výstup+> <výstup-> V<název řídicího zdroje> <přenos>

Vzorec: $\text{výstup} = \text{přenos} * \text{proud řídicím zdrojem}$

Příklad:

Fzesil 2 0 Vin 100 ; lineární proudový zesilovač, vstup 1-0, výstup 2-0, zesílení 100

Vin 1 0 0V ; pomocný zdroj napětí mezi svorkami 1-0 o nulovém napětí

2) Nelineární řízený zdroj:

F<název> <výstup+> <výstup-> POLY(<dimenze>) {V<název říd. zdroje>} {<koef.>}

seznam řídících zdrojů

seznam koeficientů polynomu

Vzorec: Nelineární polynomiální závislosti výstupu na řídících vstupech, viz zdroje E a G

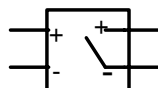
Příklad:

Fdif out 0 POLY(2) V1 V2 0 1 -1 ; výstupní proud je rozdíl proudů tekoucích přes zdroje V1
+ a V2

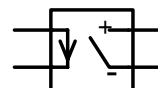
5.5 Řízené (závislé) spínače pro behaviorální modelování

S .. spínač řízený napětím

W .. spínač řízený proudem



S



W

Spínač „S“:

S<název> <spínač+> <spínač-> <řízení+> <řízení-> <jméno modelu>

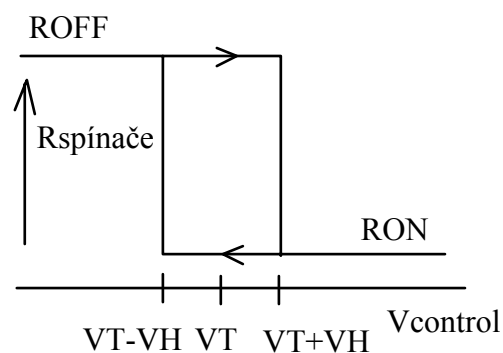
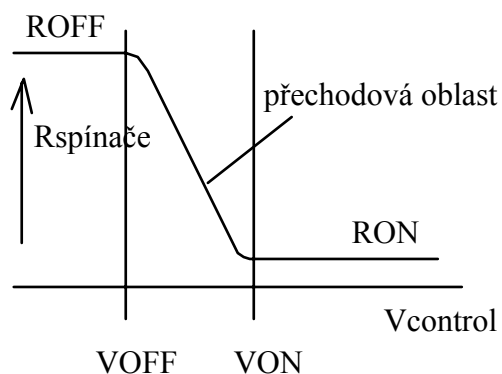
Definice modelu spínače:

.MODEL <jméno modelu> VSWITCH [parametry modelu]

V PSpice existují dva různé modely spínače: „Variable-Resistance Switch“ (VRS) a „Short-Transition Switch“ (STS). PSpice rozezná typ modelu podle jeho zadaných parametrů. Implicitním modelem je model VRS.

VRS			
parametr	popis	jednotka	default
ROFF	odpor v rozpojeném stavu	Ohm	1MEG
RON	odpor v sepnutém stavu	Ohm	1
VOFF	vypínací napětí	V	0
VON	spínací napětí	V	1

STS			
parametr	popis	jednotka	default
ROFF	odpor v rozpojeném stavu	Ohm	1T
RON	odpor v sepnutém stavu	Ohm	1
VT	prahové napětí	V	0
VH	hysterezi napětí	V	0
TD	časové zpoždění	s	0



Parametr TD znamená časové zpoždění od přivedení překlápěcí úrovně napětí k faktickému přepnutí spínače.

Spínač „W“:

S<název> <spínač+> <spínač-> V<název řídicího zdroje> <jméno modelu>

Definice modelu spínače:

.MODEL <jméno modelu> ISWITCH [parametry modelu]

Spínač se nyní řídí proudem, který teče řídicím zdrojem napětí. Parametry modelu jsou obdobné jako u modelu VSWITCH, pouze napětí VOFF, VON, VT, VH jsou nahrazena proudy IOFF, ION, IT, IH.

5.6 Shrnutí kapitoly 5

Podobvody SPICE slouží k modelování samostatných obvodů na principu „černé skříňky“. Uživatel se podobvod jeví jako zapouzdřený celek, který lze využívat prostřednictvím vývodů. Dovnitř není třeba vidět.

Takto jsou modelovány v podstatě všechny integrované obvody. Příslušné podobvody je možno nalézt například na Internetu.

Podobvody lze využívat buď napojováním vstupního souboru na soubory, v nichž se podobvody nacházejí (většinou se jedná o knihovny), nebo lze definice podobvodů umísťovat přímo do vstupních souborů (pak mají pouze lokální platnost).

Existují podobvody „fixní“ a podobvody, které lze volat s parametry.

Uživatel libovolného podobvodu potřebuje znát konvenci – standardizovaný způsob volání – podobvodu. Tvůrce podobvodu potřebuje znát standardizovanou strukturu podobvodu. Student tohoto předmětu potřebuje znát obojí.

Behaviorální modelování je zjednodušené modelování elektronických prvků a bloků. Jde o modelování chování obvodů z hlediska jejich vstupů a výstupů, ne o modelování toho, z čeho se obvod skládá. Jde o modelování chování celku bez ohledu na vlastnosti jednotlivých částí tohoto celku. Tento pragmatický přístup k modelování reality bývá velmi efektivní co do rychlosti výpočtů.

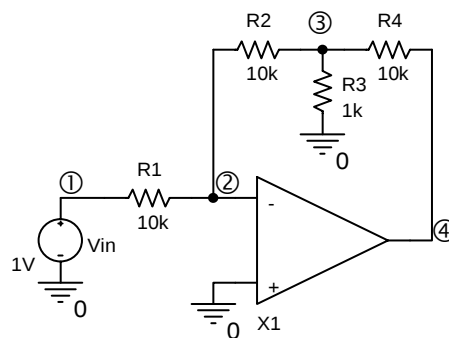
Důležitou metodou behaviorálního modelování je tedy náhrada složitého systému několika standardizovanými bloky. Důležitým nástrojem modelování jsou vzorce pro popis vstupně-výstupního chování těchto bloků.

V PSPICE jsou základní standardizované stavební bloky řízené zdroje, pomocí nichž lze vybudovat behaviorální model libovolného elektrického systému. Matematický popis chování těchto bloků je zabezpečován funkcemi, zabudovanými v PSPICE. Kombinací těchto funkcí si může uživatel definovat své vlastní funkce.

5.7 Řešené příklady

Příklad 5.1 Zesilovač s T-článkem s lineárním modelem operačního zesilovače

Napište vstupní soubor pro zesilovač s T-článkem podle obrázku. Operační zesilovač modelujte podobvodem, který je zobrazen níže. Analýzou určete výstupní napětí. Ověřte výpočtem.



Řešení:

obvod s operacním zesilovacem

*

Vin 1 0 1V

R1 1 2 10k

R2 2 3 10k

R3 3 0 1k

R4 3 4 10k

X1 0 2 4 operak_jednoduchy

*

.subckt operak_jednoduchy 1 2 4

Rin 1 2 1meg

Rout 3 4 50

E1 3 0 1 2 200k

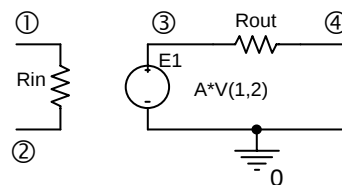
.ends

*

.OP

*

.end



Na základě schématu podobvodu a schématu celého obvodu se snažte pochopit mechanismus tohoto hierarchického modelování.

Zopakujte si z teorie způsob modelování zdroje napětí řízeného napětím E.

Proveďte analýzu pracovního bodu a ručním výpočtem ověřte její správnost. Zdůvodněte nepatrné numerické odchylky.

Všimněte si výrazu

(X1.3) -12.0540

(interní uzel č. 3 podobvodu X1 a jeho napětí; tento uzel není připojen k vnějším uzlům obvodu, proto jeho napětí PSPICE rovněž zviditelnil).

Příklad 5.2 Zesilovač s T-článkem s lineárním modelem operačního zesilovače, volání podobvodu s parametry

Modifikujte vstupní soubor z příkladu 5.1 tak, aby při volání podobvodu bylo možno zadat tyto konkrétní parametry operačního zesilovače: zesílení, vstupní odpor, výstupní odpor.

K simulaci prakticky ideálního OZ použijte hodnoty výše uvedených parametrů 1T, 1Tohm, 100pikoohmů.

Řešení:

obvod s operacním zesilovcem

*

Vin 1 0 1V

R1 1 2 10k

R2 2 3 10k

R3 3 0 1k

R4 3 4 10k

X1 0 2 4 operak_jednoduchy **params: A=1T Rin=1e12 Rout=1e-10**

*

.subckt operak_jednoduchy 1 2 4 **params: A=200k Rin=1meg Rout=50**

Rin 1 2 {**Rin**}

Rout 3 4 {**Rout**}

E1 3 0 1 2 {**A**}

.ends

*

.OP

*

.end

Nyní je podobvod volán s parametry, a to tak, abychom simulovali „téměř ideální“ operační zesilovač. Poznamenejme, že Rout nelze nastavit na nulu a žádný číselný údaj na nekonečno.

Volání podobvodu s parametry je zvlášť výhodné, pokud je podobvod umístěn v knihovně. Pak jej může využívat více uživatelů ve svých konkrétních vstupních souborech, které jsou na tuto knihovnu napojeny příkazem .lib.

Výsledky analýzy srovnajte s teorií.

Příklad 5.3 *Tvorba vlastní knihovny s podobvodem OZ*

Vytvořte si svou vlastní knihovnu se jménem pokus.lib, do které vložíte definiční text podobvodu z příkladu 5.2. Pak zjednodušte vstupní soubor z příkladu 5.2, v němž budeme využívat modelu z knihovny pokus.lib.

Řešení:

Knihovna pokus.lib:

Moje pokusna knihovna

*

.subckt operak_jednoduchy 1 2 4 **params: A=200k Rin=1meg Rout=50**

Rin 1 2 {**Rin**}

Rout 3 4 {**Rout**}

E1 3 0 1 2 {**A**}

.ends

Vstupní soubor:

obvod s operacním zesilovcem

*

```

Vin 1 0 1V
R1 1 2 10k
R2 2 3 10k
R3 3 0 1k
R4 3 4 10k
X1 0 2 4 operak_jednoduchy params: A=1T Rin=1e12 Rout=1e-10
*
.lib pokus.lib
.OP
*
.end

```

Ověřte, zda simulace funguje stejně jako v příkladu 5.2.

Příklad 5.4 *Tvorba podobvodu s nelineárním modelem OZ (modelování saturace OZ)*

Rozšiřte knihovnu pokus.lib o podobvod s modelem OZ, který bude navíc oproti modelu z příkladu 5.2 modelovat saturaci operačního zesilovače. Výstupní napětí operačního zesilovače pak nemůže být větší než 13V a menší než -13V. Využijte k tomu sřízeného zdroje E spolu s funkcí LIMIT.

Řešení:

Vstupní soubor:

```

obvod s operacnim zesilovacem
*
Vin 1 0 1V
R1 1 2 10k
R2 2 3 10k
R3 3 0 1k
R4 3 4 10k
X1 0 2 4 operak_se_saturaci params: A=200k Rin=1meg Rout=50
*
.lib pokus.lib
.OP
*
.end

```

podobvody v knihovně:

moje pokusna knihovna

```

*
.subckt operak_jednoduchy 1 2 4 params: A=200k Rin=1meg Rout=50
Rin 1 2 {Rin}
Rout 3 4 {Rout}
E1 3 0 1 2 {A}
.ends

.subckt operak_se_saturaci 1 2 4 params: A=200k Rin=1meg Rout=50
Rin 1 2 {Rin}

```

```

Rout 3 4 {Rout}
E1 3 0 value={limit(A*V(1,2), -13,13)}
.ends

```

Analýzou se přesvědčete o tom, že naprogramování saturace funguje (například zvětšete vstupní napětí na 20V nebo zmenšete R3 na 100 ohmů).

Příklad 5.5 Použití uživatelské funkce k modelování operačního zesilovače

Upravte model podobvodu operačního zesilovače z příkladu 5.4 s využitím uživatelem definované funkce tak, aby vykazoval stejné vlastnosti. Pomocí příkazu .FUNC definujte novou funkci $y = \text{OUTPUT}(x)$, která bude modelovat závislost výstupního napětí zdroje E1, tj. y , na vstupním diferenčním napětí, tj. x . Tuto funkci pak použijte v podobvodu operak_se_saturaci.

Řešení - podobvod:

```

.subckt operak_se_saturaci 1 2 4 params: A=200k Rin=1meg Rout=50
Rin 1 2 {Rin}
Rout 3 4 {Rout}
E1 3 0 value={output(V(1,2))}
.FUNC output(x)=limit({A}*x, -13,13); funguje i bez rovnítka a taky {limit(A*x, -13,13)}
.ends

```

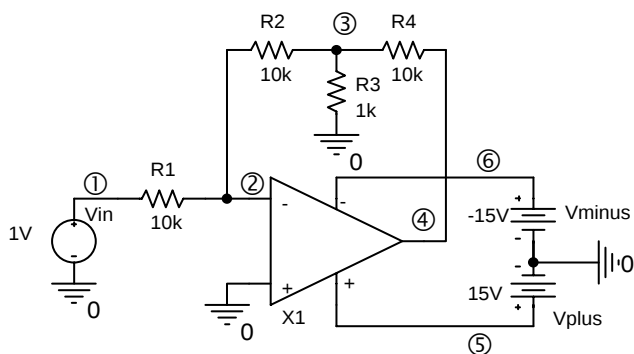
Funkce **output** má pouze lokální platnost v rámci podobvodu.

Zkuste změnit parametry operačního zesilovače tak, aby představoval ideální OZ, viz příklad 5.2. SPICE bude mít problémy s nalezením řešení. O této problematice bude pojednáno později.

Příklad 5.6 Tvorba složitějšího modelu OZ

Vytvořte podobvod, který bude vycházet z modelu OZ z příkladu 5.4, tj. modelu respektujícího saturaci. Podobvod nyní bude mít 5 vývodů: neinvertující vstup, invertující vstup, kladné ss napájení, záporné ss napájení, výstup. Saturační napětí však nyní bude záviset na napájecím napětí, konkrétně rozdíl mezi napájecím a saturačním napětím bude vždy 1.5V (například pro napájecí napětí $\pm 15\text{V}$ bude saturační napětí $\pm 13.5\text{V}$).

Napište rovněž vstupní soubor pro obvod na obrázku a vyzkoušejte fungování podobvodu.



Řešení:

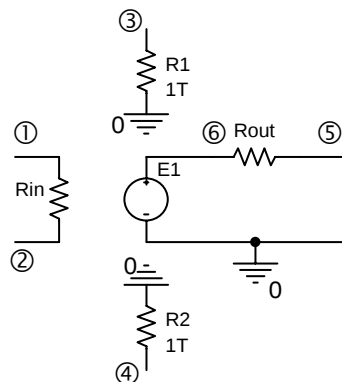
obvod s operacnim zesilovacem

*

```

Vin 1 0 1V
Vplus 5 0 15V
Vminus 6 0 -15V
R1 1 2 10k
R2 2 3 10k
R3 3 0 1k
R4 3 4 10k

```



```

X1 0 2 5 6 4 operak_se_saturacix params: A=200k Rin=1meg Rout=50
*
.lib pokus.lib
*
.OP
*
.end

* uzly:                neinvertující vstup
*                      | invertující vstup
*                      | | kladné napájecí napětí
*                      | | | záporné napájecí napětí
*                      | | | | výstup
*                      | | | | |
.subckt operak_se_saturacix 1 2 3 4 5 params: A=200k Rin=1meg Rout=50
Rin 1 2 {Rin}
Rout 5 6 {Rout}
R1 3 0 1T
R2 4 0 1T
E1 6 0 value={limit(A*V(1,2),V(4)+1.5,V(3)-1.5)}
.ends

```

Pomocné odpory 1Tohm (Terra = $1 \cdot 10^{12}$) jsou připojeny k napájecím vývodům z toho důvodu, že každý z uzlů obvodu musí mít stejnosměrné spojení se zemí (s uzlem 0).

Odzkoušejte nyní fungování celého obvodu v saturačním režimu při různých napájecích napětích.

Příklad 5.7 *Využívání profesionálního modelu OZ z knihovny*

V příkladu 5.6 nahraďte model OZ profesionálním SPICE modelem zesilovače typu 741 z knihovny PSPICE. Vyzkoušejte funkci zapojení.

Řešení:

viz předchozí příklad 5.6, pouze volání podobvodu bude jiné:

X1 0 2 5 6 4 uA741

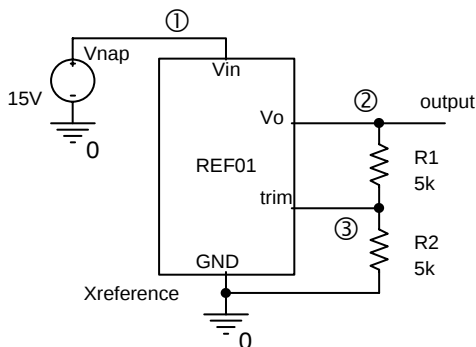
V demo verzi OrCadPSpice je tento model v knihovně EVAL.LIB, v profi verzi je v knihovně OPAMP.LIB. Prohlédněte si jej. Pořadí vývodů operačního zesilovače při volání podobvodu je standardně (viz též hlavička podobvodu v knihovně): neinvertující vstup, invertující vstup, kladné napájení, záporné napájení, výstup.

Pro úspěšnou simulaci potřebujeme tuto informaci, nepotřebujeme rozumět „vnitřku“ daného podobvodu.

Porovnáme-li chování obvodu s operačním zesilovačem 741 a s naším zesilovačem operak_se_saturacix, nezaznamenejme velké rozdíly. Je tomu tak proto, že náš model je dostatečně přesný pro analýzu stejnosměrných poměrů (při napájecím napětí nad 1,5V). Naprosto nevyhovující však bude pro analýzu v dynamickém režimu (řešení přechodných dějů a kmitočtových charakteristik).

Příklad 5.8 Modelování obvodu s napětovou referencí REF01

Z Internetu nebo odjinud si sežněte SPICE model integrované 10V napětové reference REF01 firmy Analog Devices. Příslušný podobvod si uložte do pomocné knihovny POKUS.LIB. Zjistěte výstupní napětí podle schématu na obrázku při teplotě 27 stupňů Celsia.



Řešení:

```
zdroj 10V
*
Vnap 1 0 15V
R1 2 3 5k
R2 3 0 5k
Xreference 1 0 3 2 REF01_AD
*
.lib pokus.lib
*
.end
```

```
*
*          VIN
*          | GND
*          | TRIM
*          | VOUT
*          |
.SUBCKT REF01_AD 2 4 5 6
.....
.....
.ENDS REF01_AD
```

Výstupní napětí vyjde 9,0977V.

Důležité je podívat se na hlavičku získaného podobvodu (viz obrázek), abychom mohli při volání podobvodu správně stanovit pořadí vývodů.

Příklad 5.9 Modelování obvodu s napětovou referencí REF01

Pomocí příkazu .PARAM realizujte odpory R1 a R2 z příkladu 5.8 jako potenciometr o odporu 10kohmů. Zjistěte, jak se mění výstupní napětí při „točení“ potenciometru. Pokuste se tak nastavit výstupní napětí přesně na 10V.

Řešení:

```
zdroj 10V
*
Vnap 1 0 15V
R1 2 3 {Rprom}
R2 3 0 {10k-Rprom}

Xreference 1 0 3 2 REF01_AD
*
.param Rprom=5k
.lib pokus.lib
.step param=Rprom list 8k 8.5k 9k
*
.end
```

Při růstu R1 roste výstupní napětí. Napětí 10V je nastaveno při R1 asi 8,77k.

5.8 Kontrolní otázky ke kapitole 5

Pokud dospějete k závěru, že na některou z níže uvedených otázek nedovedete vyčíst úplnou odpověď z tohoto elektronického textu, pokuste se dočerpat dodatečné informace z dalších zdrojů.

Správné odpovědi naleznete na konci učebního textu.

1. Operační zesilovače se v PSPICE modelují:

a) jako podobvody, ne přes příkaz `.MODEL`, b) jen přes příkaz `.MODEL`, c) buď jako podobvody nebo přes příkaz `.MODEL`

2. V PSpice se podobvod (subckt)

a) musí, b) může, c) nesmí
volat s parametry.

3. Zdroje typu F a H jsou řízeny proudem, který protéká:

a) jejich řídicími svorkami, b) pomocným zdrojem napětí, c) pomocným zdrojem proudu.

4. Podobvod má následující hlavičku:

`.subckt muj_obvod vstup vystup params: odpor=100k`

Ve vstupním souboru je volán takto:

`X1 muj_obvod 1 2`

Parametr odpor bude mít velikost:

a) 100k, b) 0, c) nebude definován, PSPICE ohlásí chybu.

5. Dimenze polynomiálního řízeného zdroje je rovna počtu:

a) všech vývodů zdroje, b) výstupů zdroje, c) řídicích bran zdroje.

6. Písmenem S začíná v PSPICE jméno

a) spínače řízeného napětím, b) zdroje proudu řízeného napětím, c) tranzistoru popsaného S-parametry.

7. Definice podobvodu SPICE končí řetězcem:

a) `.end`, b) `*ends`, c) `.ends`.

8. Jméno zdroje proudu řízeného napětím začíná v PSPICE písmenem

a) I, b) G, c) F

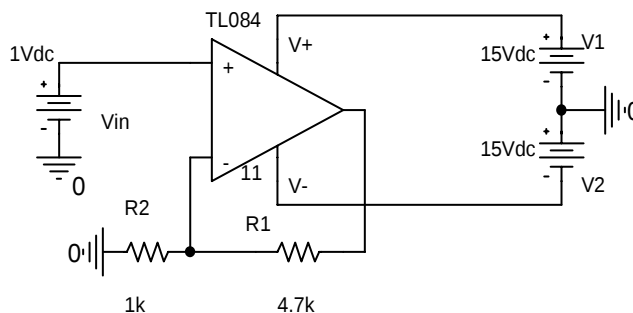
5.9 Neřešené příklady

Příklad 5.10 Použití funkce TABLE

V příkladu 5.4 je pomocí funkce LIMIT modelována nelineární převodní charakteristika operačního zesilovače se saturací. Napište obdobný podobvod SPICE s tím, že k modelování převodní charakteristiky použijete funkci TABLE.

Příklad 5.11 Neinvertující zesilovač s TL084

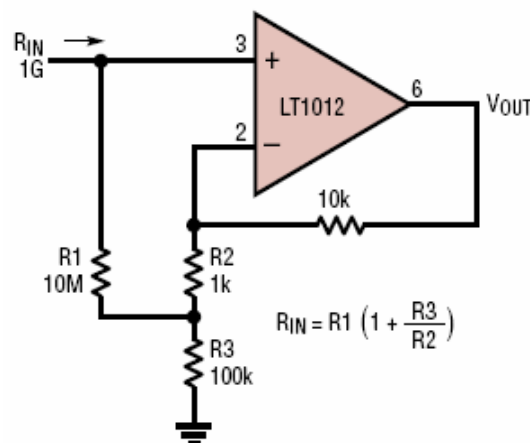
V obvodu na obrázku si označte uzly. Napište vstupní soubor pro výpočet výstupního napětí operačního zesilovače. Získejte podobvod zesilovače TL084 a využijte jej k analýze. Výsledek analýzy srovnajte s teoretickou hodnotou pro ideální operační zesilovač.



Příklad 5.12 Násobič odporu s LT1012

Z Internetu nebo odjinud získejte model operačního zesilovače LT1012. Sestavte vstupní soubor pro výpočet stejnosměrného odporu mezi neinvertujícím vstupem OZ a zemí podle obrázku. Operační zesilovač napájejte symetrickými zdroji +15V a -15V. Ověřte platnost vzorce uvedeného na obrázku.

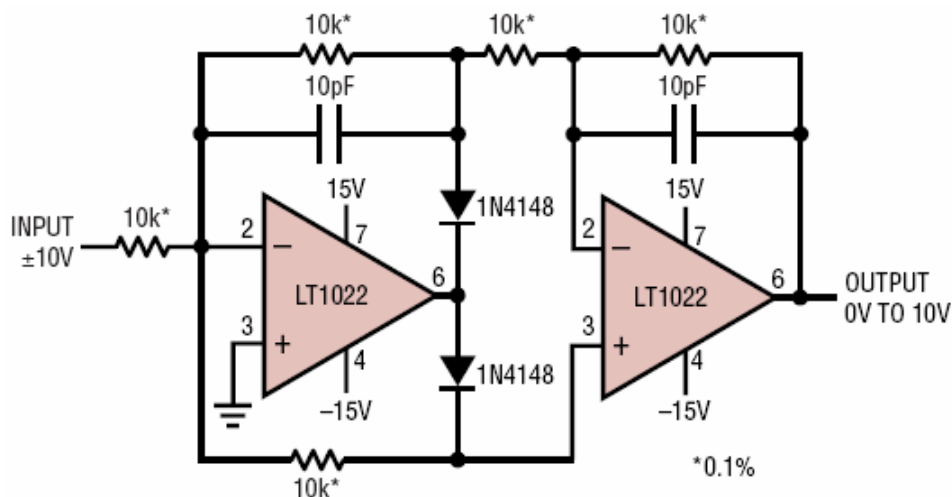
Návod na zjištění vstupního odporu: Mezi vstupní svorku a zem připojte stejnosměrný zdroj o napětí např. 1V. Zjistěte proud tekoucí tímto zdrojem a z něj vstupní odpor.



Příklad 5.13 Přesný usměrňovač s LT1022

Z Internetu nebo odjinud získejte model operačního zesilovače LT1022. Vypočtete výstupní napětí dvoucestného usměrňovače na obrázku, jestliže na vstupu je stejnosměrné napětí a) 0.1V, b)

-0.1V (při tak malých napětích se ještě neotevírají diody v klasických usměrňovačích).
Návod: Mezi vstupní svorku a zem připojte stejnosměrný zdroj o napětí a) +0.1V, b) -0.1V.



6 Hledání stejnosměrných ustálených stavů – základ většiny analýz v PSPICE

Cíle kapitoly:

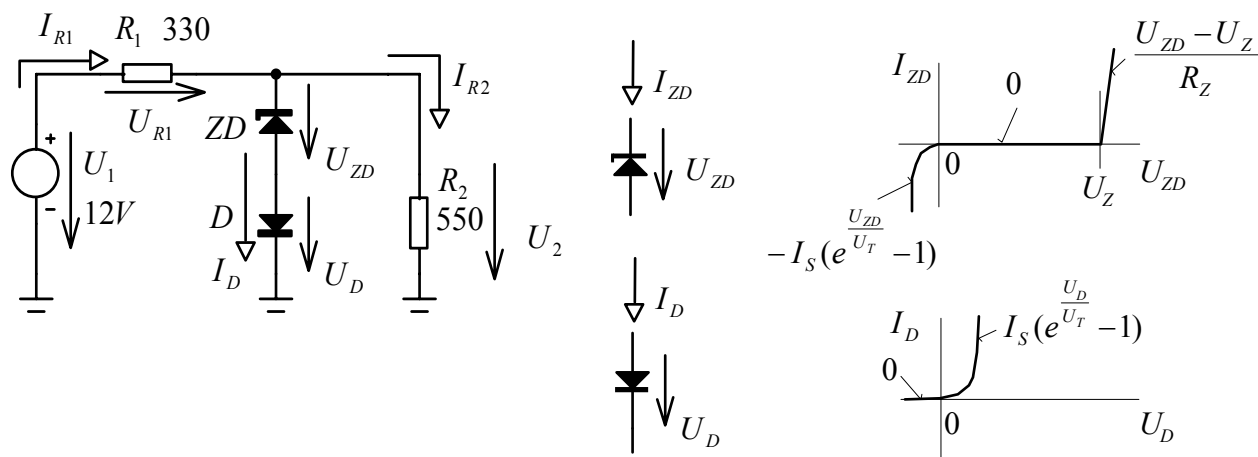
- ✚ Ukázat podstatu problémů s hledáním stejnosměrných řešení v nelineárních obvodech, včetně tzv. problémů s konvergencí, a způsobů jejich překonávání.
- ✚ Shrnout pravidla pro označování stejnosměrných napětí a proudů v PSPICE a v PROBE.
- ✚ Vysvětlit fungování a praktické používání DC analýzy.

6.1 Hledání stejnosměrného řešení v nelineárním obvodu

PSPICE počítá stejnosměrný pracovní bod v obvodu vždy, i když uživatel nespecifikuje žádný typ analýzy. T této skutečnosti jsme těžili při řešení všech dosavadních příkladů.

Později uvidíme, že algoritmus výpočtu stejnosměrného pracovního bodu se využívá ve všech základních analýzách, tj. v analýze DC, Transient i AC. Tento algoritmus řeší obecně velmi náročnou matematickou úlohu, hledání řešení soustavy nelineárních algebraických rovnic. Na „kvalitě“ tohoto algoritmu závisí chod simulačního programu a nepřímo i spokojenost jeho uživatele.

Problém vysvětlíme na konkrétním příkladu stabilizátoru napětí. Úkolem je určit velikost výstupního napětí, jestliže jsou zadány vzorce ampérvoltových charakteristik diod. ZD je Zenerova dioda 1N3826 o napětí $U_Z = 5,1$ V, D je „klasická“ křemíková dioda 1N459. Nejprve ukážeme, jak by úlohu řešil člověk „ručně“. Pak naznačíme, jak k řešení „přistupuje“ simulační program.



$U_Z = 5,1$ V ... Zenerovo napětí.

$R_Z = 7\Omega$... diferenciální odpor Zenerovy diody v oblasti Zenerova průrazu.

$I_S = 3,5 \cdot 10^{-13}$ A ... saturační proud; u obou diod je uvažována stejná hodnota.

$U_T = kT/q \doteq 25,8$ mV ... teplotní napětí při absolutní teplotě $T = 300$ K (27°C), k je Boltzmannova konstanta $1,38 \cdot 10^{-23}$ J/K, q je elementární elektrický náboj $1,602 \cdot 10^{-19}$ C.

6.1.1 „Ruční“ řešení

$$U_{zD} = R_z I_D + U_z, \quad U_D = U_T \ln\left(1 + \frac{I_D}{I_s}\right).$$

$$U_2 = R_z I_D + U_z + U_T \ln\left(1 + \frac{I_D}{I_s}\right), \quad I_D \geq 0 \quad (1)$$

$$U_2 = R_2 I_{R2} = R_2 (I_{R1} - I_D) = R_2 \left(\frac{U_1 - U_2}{R_1} - I_D \right).$$

$$U_2 = \frac{R_2}{R_1 + R_2} U_1 - \frac{R_1 R_2}{R_1 + R_2} I_D.$$

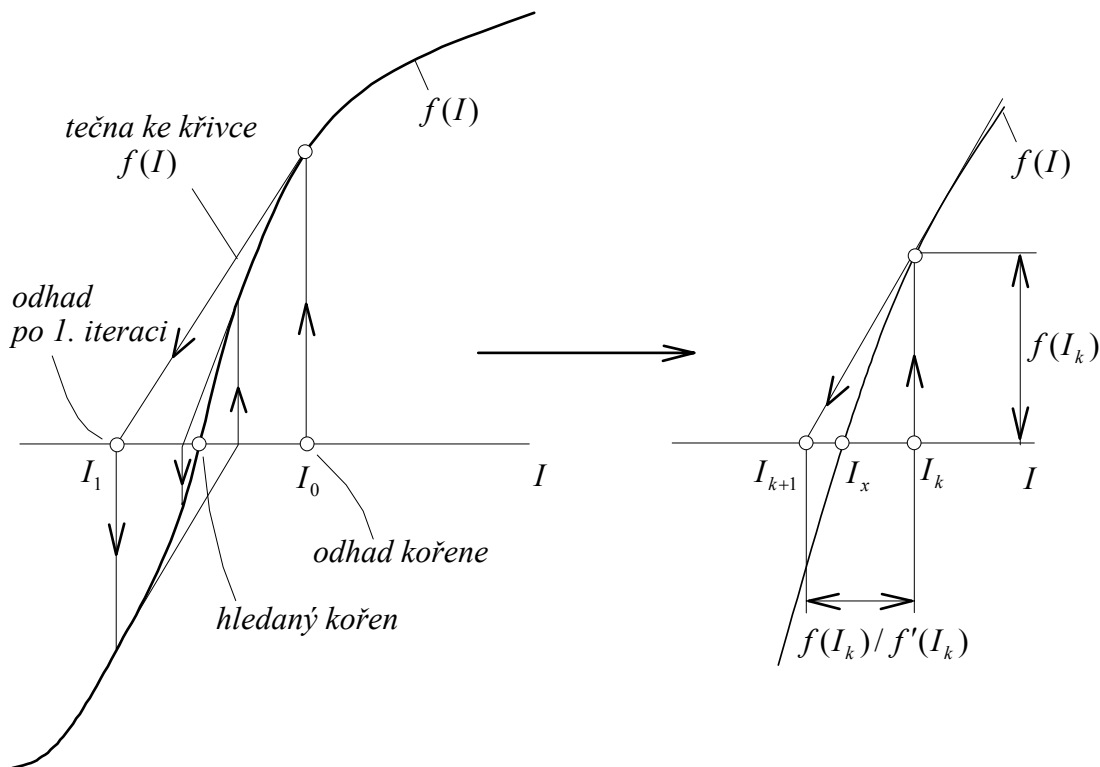
Dosadíme do levé strany rovnice (1). Tím dostáváme rovnici o jediné neznámé I_D , která získá po úpravě tvar

$$\left(R_z + \frac{R_1 R_2}{R_1 + R_2} \right) I_D + U_T \ln\left(1 + \frac{I_D}{I_s}\right) + U_z - \frac{R_2}{R_1 + R_2} U_1 = 0. \quad (2)$$

Na levé straně rovnice je nelineární funkce proudu I_D , kterou označíme $f(I_D)$. Hledáme takový proud, pro nějž je funkce nulová. Po dosazení číselných hodnot lze (2) zapsat takto:

$$f(I_D) \doteq I_D + 1,2118 \cdot 10^{-4} \ln(1 + 2,8571 \cdot 10^{12} I_D) - 1,1125 \cdot 10^{-2} = 0. \quad (3)$$

Rovnici vyřešíme Newtonovou iterační metodou, jejíž princip je ilustrován na obr. Vyjdeme z počátečního odhadu řešení I_0 a na základě naznačené konstrukce dospějeme k „zpřesněnému“ odhadu I_1 . Opakujeme tak dlouho, až se posloupnost $I_0, I_1, I_2, \dots$ ustálí v rámci akceptovatelné chyby na hodnotě hledaného řešení I_x .



Z geometrického významu derivace funkce f v bodě I_k vyplývá známý vztah (Newtonova formule) mezi řešeními v sousedních iteracích:

$$I_{k+1} = I_k - \frac{f(I_k)}{f'(I_k)}. \quad (4)$$

Aplikujeme-li tento vzorec na naši konkrétní funkci (3), dostaneme po úpravě výpočetní algoritmus

$$I_{k+1} = I_k - \frac{I_k + 1,2118 \cdot 10^{-4} \ln(1 + 2,8571 \cdot 10^{12} I_k) - 1,1125 \cdot 10^{-2}}{1 + \frac{3,4624 \cdot 10^8}{1 + 2,8571 \cdot 10^{12} I_k}}. \quad (5)$$

Zvolíme-li počáteční odhad $I_0 = 0A$, postupným dosazováním do (4), dostaneme posloupnost

$$(0, 3.2504e-8, 2.9345e-6, 2.4897e-4, 1.9975e-2, 1.3034, 7.9644, 8.3584, 8.3585, 8.3585, \dots) \text{ mA}.$$

Hledaný proud diodami je tedy (v rámci přesnosti použitých modelů) s přesností na 4 platné cifry

$$I_D = 8,3585 \text{ mA}.$$

Můžete se přesvědčit o tom, že při počátečním odhadu řešení 10 mA se ustálí iterační algoritmus na tomto řešení hned v 2. kroku.

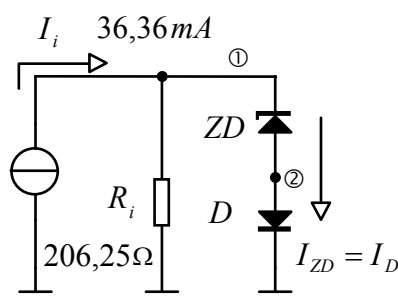
Pro danou hodnotu proudu I_D pak z modelů diod vycházejí napětí

$$U_{ZD} = 5,159V, \quad U_D = 0,617V.$$

6.1.2 Algoritmický postup při řešení simulačním programem

Simulační program nejprve algoritmicky sestaví obvodové rovnice metodou uzlových napětí a pak počítá všechny neznámé, tj. uzlová napětí. Program sestaví tolik nelineárních rovnic, kolik je uzlových napětí, a hledá iterací stejný počet neznámých napětí. Iterační metoda tedy musí být zobecněná pro více proměnných. Nazývá se *Newtonova-Raphsonova iterační metoda*. V různých modifikacích je zabudována do všech stávajících simulačních programů do procedur pro hledání stejnosměrných pracovních bodů.

Pro jednoduchost převedeme zdroj napětí v stabilizátoru na ekvivalentní zdroj proudu:



Pro obvod platí dvě rovnice 1. Kirchhoffova zákona s neznámými uzlovými napětími U_1 a U_2 :

$$\begin{array}{l} \textcircled{1} \\ \textcircled{2} \end{array} \begin{array}{|c|} \hline f_1(U_1, U_2) \\ \hline f_2(U_1, U_2) \\ \hline \end{array} = \begin{array}{|c|} \hline G_i U_1 + I_{ZD} - I_i \\ \hline I_D - I_{ZD} \\ \hline \end{array} = \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array} \quad (6)$$

vektor f

kde kladné proudy diodami můžeme vyjádřit rovnicemi

$$I_{ZD} = G_Z(U_1 - U_2 - U_Z), I_D = I_S \left(e^{\frac{U_2}{U_T}} - 1 \right). \quad (7)$$

Vektor $\mathbf{f}$ tedy po dosazení (7) do (6) vypadá takto:

$$\mathbf{f} = \begin{bmatrix} G_i U_1 + G_Z(U_1 - U_2 - U_Z) - I_i \\ I_S \left(e^{\frac{U_2}{U_T}} - 1 \right) - G_Z(U_1 - U_2 - U_Z) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8)$$

Úkolem iterační metody je nalézt takové hodnoty U_1 a U_2 , abychom vynulovali vektor $\mathbf{f}$, tedy abychom současně vynulovali obě dílčí funkce f_1 a f_2 .

Newtonova-Raphsonova metoda je zobecněním „jednorozměrné“ Newtonovy metody a je tedy rovněž založena na vzorci (4), který je však nyní zobecněný pro více proměnných. Pro náš případ dvou proměnných vychází:

$$\begin{bmatrix} U_{1,k+1} \\ U_{2,k+1} \end{bmatrix} = \begin{bmatrix} U_{1,k} \\ U_{2,k} \end{bmatrix} - \begin{bmatrix} \frac{\partial f_{1,k}}{\partial U_1} & \frac{\partial f_{1,k}}{\partial U_2} \\ \frac{\partial f_{2,k}}{\partial U_1} & \frac{\partial f_{2,k}}{\partial U_2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} f_{1,k} \\ f_{2,k} \end{bmatrix} \quad (9)$$

Obecně

$$\mathbf{U}_{k+1} = \mathbf{U}_k - \mathbf{J}^{-1} \cdot \mathbf{f}_k$$

Čtvercová matice $\mathbf{J}$ se nazývá *Jakobiho matice*, její determinant *Jakobián*. Derivujeme-li funkci (8) podle U_1 a U_2 , dostaneme pro analyzovaný obvod *Jakobiho matici* ve tvaru

$$\mathbf{J} = \begin{bmatrix} G_i + G_Z & -G_Z \\ -G_Z & \frac{I_S}{U_T} e^{\frac{U_2}{U_T}} + G_Z \end{bmatrix} = \begin{bmatrix} 0,1477 & -0,1429 \\ -0,1429 & 1,3566 \cdot 10^{-11} e^{38,76 U_2} + 0,1429 \end{bmatrix} \quad (10)$$

Při algoritmickém řešení stejnosměrných poměrů v nelineárním obvodu tedy simulační program postupuje zhruba takto:

Algoritmus stejnosměrné analýzy nelineárních obvodů:

- 1) Nastaví se počáteční hodnota vektoru neznámých veličin, tj. uzlových napětí.
- 2) Na základě Kirchhoffových rovnic pro nezávislé uzly, modelů součástek a aktuálních hodnot uzlových napětí se vyplní prvky vektoru $\mathbf{f}$ (8) a prvky Jakobiho matice (10).
- 3) Pomocí rekurentní Newtonovy-Raphsonovy formule (9) se vypočte nový odhad uzlových napětí.
- 4) Body 2 a 3 se opakují v cyklu tak dlouho, dokud není splněno zadané chybové kritérium.

Kdybychom si daný postup vyzkoušeli naprogramovat například v MATLABu, zjistili bychom, že při počátečním nastavení $[U_1 \ U_2] = [0 \ 0]$ bude algoritmu trvat 92 iterací, než se v rámci 4 platných cifer ustálí na řešení

$$[U_1 \ U_2] = [5.7751 \text{V} \ 0.6165 \text{V}].$$

6.1.3 Problémy s konvergencí - úvod

Nelineární obvody mohou mít několik stejnosměrných pracovních bodů. Např. bistabilní klopný obvod s 2 tranzistory T1 a T2 má 3 pracovní body – T1 otevřen a T2 zavřen, T1 zavřen a T2 otevřen, T1 i T2 částečně otevřeny. První dva mohou v praxi běžně nastat, třetí je nestabilní a v reálném obvodu neudržitelný, tedy nepozorovatelný. Bohužel program „s oblibou“ najde většinou tento pracovní bod.

Obecně mohou nastat tyto případy:

1. Program najde pracovní bod, který nás zajímá.
2. Program najde pracovní bod, o který nestojíme.
3. Program nenajde v daném počtu iterací žádné řešení.

Výsledek můžeme ovlivnit volbou prvního odhadu řešení, tj. volbou souřadnic počáteční, nulté iterace. Pokud bude odhad blízký pracovnímu bodu, který nás zajímá, je naděje, že k němu program dospěje. První odhad řešení, neboli tzv. iterační násada Newtonovy – Raphsonovy metody, se definuje příkazem `.NODESET`.

`.NODESET` ... stanovení počátečního odhadu řešení některých uzlových napětí a proudů induktory

Formát:

`.NODESET <V(<node> [,<node>])=<value> >*`

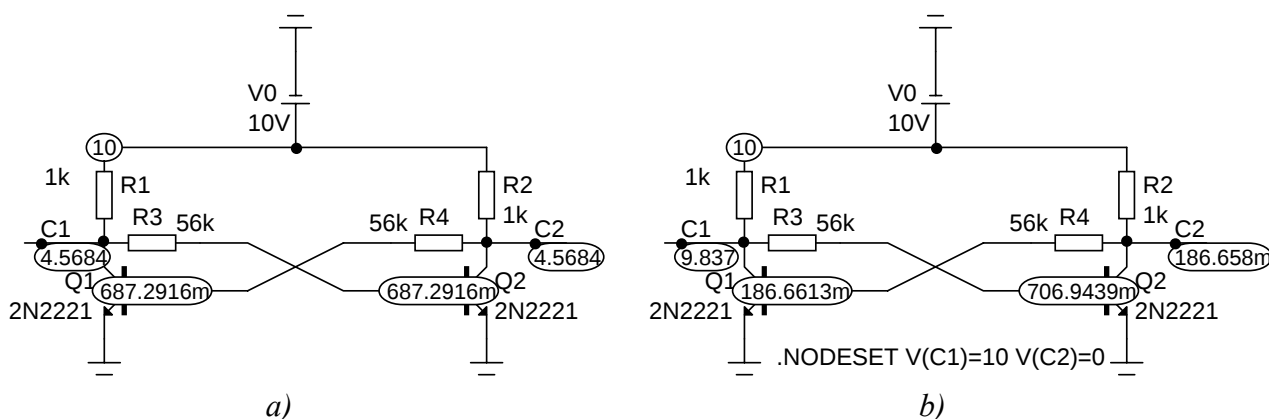
`.NODESET <I(<inductor>)=<value>>`

Příklad:

`.NODESET V(12)=5V V(in,out)=-5 I(Lx)=12mA`

Příklad analýzy bistabilního klopného obvodu [8]:

- a) Bez příkazu `.NODESET`, výpočet probíhá z nulových počátečních hodnot napětí. Program nalezne nestabilní pracovní bod.
- b) S příkazem `.NODESET`, který představuje otevřený tranzistor Q1 a uzavřený Q2. Program nalezne stabilní pracovní bod odpovídající tomuto stavu.



Další pravidla pro používání příkazu `.NODESET`:

Je-li současně přítomen příkaz `.IC`, pak je `.NODESET` ignorován.

`.NODESET` se uplatňuje při výpočtu klidového pracovního bodu:

-pokud není uveden žádný příkaz pro analýzu, tedy pokud se počítá pouze ss pracovní bod,

- při analýze .TRANSIENT (pak se uplatní při výpočtu počátečního bodu časového průběhu),
- při analýze .AC
- při analýze .DC (pak se uplatní při výpočtu prvního bodu křivky).

6.1.4 Globální podmínky simulace - úvod

	Use Original Value	Change To	
Relative accuracy of V's and I's:	☒ 0.001	0.001	(RELTOL)
Best accuracy of currents [amps]:	☒ 1e-012	1e-012	(ABSTOL)
Best accuracy of voltages [volts]:	☒ 1e-006	1e-006	(VNTOL)
Minimum conductance for any branch [1/ohm]:	☒ 1e-012	1e-012	(GMIN)
Run to time [seconds]:	☒ 1e-009	1e-009	(TSTOP)
Maximum step size [seconds]:	☒		(TMAX)
DC and bias "blind" iteration limit:	☒ 150	150	(ITL1)
DC and bias "buest guess" iteration limit:	☒ 20	20	(ITL2)
Transient time point iteration limit:	☒ 10	10	(ITL4)

Buttons: OK, OK & Resume Simulation, Cancel, Help

Přesnost výpočtů je řízena třemi kritérii:

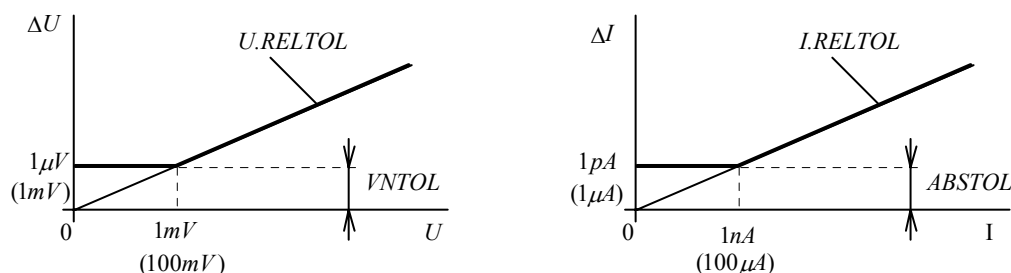
- *ABSTOL* – Absolutní chyba ve výpočtu proudů.
- *VNTOL* – Absolutní chyba ve výpočtu napětí.
- *RELTOL* – Relativní chyba ve výpočtu napětí a proudů.

Při iteračním řešení rovnic se uvedená čísla využívají k rozhodování, zda řešení ukončit, následujícím způsobem:

1. V dané iteraci se modifikovanou metodou uzlových napětí vypočtou všechna uzlová napětí plus případné další pomocné veličiny, například proudy singulárními prvky. Z těchto hodnot se aplikací Ohmova zákona dopočítají proudy všemi větvemi.
2. Pro každé vypočtené napětí U a pro každý vypočtený proud I se určí čísla ΔU a ΔI podle těchto vzorců:

$$\Delta U = \text{MAX}(\text{VNTOL}, U \cdot \text{RELTOL}), \quad \Delta I = \text{MAX}(\text{ABSTOL}, I \cdot \text{RELTOL}).$$

Grafické znázornění závislosti veličin ΔU a ΔI na napětí U a proudu I je na obr. Číselné hodnoty jsou uvedeny pro nastavení „Standard Default“, hodnoty v závorkách pro „Power Default“.



3. Porovnává se změna každého napětí U a každého proudu I oproti jejich velikostem v předchozím iteračním kroku. Výpočet se zastaví (tj. sada hodnot U a I je prohlášena za řešení), pokud jsou změny všech napětí menší než ΔU a současně změny všech proudů menší než ΔI .

Jestliže se programu nepodaří nalézt řešení, objeví se příslušné chybové hlášení, například „*Convergence problem in bias point calculation*“.

Pokud je při standardním nastavení testované napětí větší než 1 mV a testovaný proud větší než 1 nA , pak se iterace ukončí, když rozdíl po sobě jdoucích hodnot nepřekročí jednu tisícinu („*RELTOL*“) aktuálního napětí, resp. proudu. Při relativně nízkých hodnotách napětí nebo proudů by však tato chyba vycházela velmi malá (při nulových hodnotách dokonce nulová). Pokud by byla porovnatelná s numerickými chybami výpočtů, pak bychom se konvergence nemuseli dočkat vůbec. Z těchto důvodů jsou zavedeny absolutní chyby „*VNTOL*“ a „*ABSTOL*“, které při malých hodnotách U a I reprezentují povolené hranice odchylek ΔU a ΔI namísto parametru „*RELTOL*“.

Podmínka ukončení iteračního hledání řešení je i tak poměrně přísná. Proto se při řešení komplikovaných obvodů může stát, že v povoleném počtu iterací nedojde k jejímu naplnění. Maximální počet iterací je určen parametry *ITL1*, *ITL2*:

ITL1 (Standard Default = 100, Power Default = 150):

Určuje maximální počet iterací při počítání stejnosměrného pracovního bodu (inicializace časové analýzy, předstupeň kmitočtové analýzy).

ITL2 (Default = 20):

Určuje maximální počet iterací při počítání stejnosměrného pracovního bodu v každém kroku analýzy „*DC*“.

Z uvedeného plyne, že pokud nedojde k nalezení řešení v daném počtu iterací, je možné vyzkoušet buď zvětšení parametrů *RELTOL*, *ABSTOL* a *VNTOL* (zlepšení konvergence na úkor přesnosti výsledků), nebo zvětšit povolený počet iterací (pokusit se o konvergenci zvětšením počtu iterací, tj. na úkor rychlosti výpočtů).

Parametr *GMIN*: Jsou-li v modelech polovodičových diod použity nulové hodnoty paralelních vodivostí, pak parametr *GMIN* tyto nulové hodnoty automaticky nahradí. Zvětšování *GMIN* tedy může rovněž napomoci při problémech s konvergencí obvodů s polovodičovými strukturami.

6.2 Označování stejnosměrných napětí a proudů v PSPICE a v PROBE

Existují určité rozdíly mezi syntaxemi používanými v PSPICE a v PROBE. Další odlišnosti se objevují v případě, kdy pracujeme na úrovni schématického editoru. Podrobnosti naleznete v dokumentaci OrCadPSpice.

Označování stejnosměrných napětí a proudů v PSPICE

V(číslo uzlu) Příklad: V(18)	napětí uzlu oproti referenčnímu uzlu napětí mezi uzlem 18 a uzlem 0
V([název uzlu]) Příklad: V([vstup])	napětí uzlu, nazvaného jiným řetězcem než číslem, oproti referenčnímu uzlu napětí mezi uzlem „vstup“ a referenčním uzlem
V(číslo uzlu1, číslo uzlu2) Příklad: V(3,11)	Napětí mezi uzlem1 a uzlem2, stejné jako V(číslo uzlu1)-V(číslo uzlu2) napětí mezi uzlem č. 3 a uzlem č. 11, tj. V(3)-V(11)
V(jméno dvojpólu) Příklad: V(Rx)	napětí mezi svorkama 1 a 2 dvojpólu; svorka 1 je první svorka, uvedená v deklaraci dvojpólu napětí na rezistoru Rx
V<jméno svorky>(jméno vícepólu) Příklad: Vb(Q1)	napětí mezi svorkou vícepólu a referenčním uzlem napětí mezi bází tranzistoru Q1 a referenčním uzlem
V<jméno svorky1><jméno svorky2> (jméno vícepólu) Příklad: Veb(Q1)	napětí mezi svorkami 1 a 2 vícepólu napětí mezi emitorem a bází tranzistoru Q1
I(jméno dvojpólu) Příklad: I(D2)	proud tekoucí vnitřkem dvojpólu od svorky 1 do svorky 2; svorka 1 je první svorka, uvedená v deklaraci dvojpólu proud diodou D2
I<jméno svorky>(jméno vícepólu) Příklad: Ib(Qx)	proud vtékající do svorky vícepólu proud tekoucí do báze tranzistoru Qx

Označování stejnosměrných napětí a proudů v PROBE

Pozn.: při volání .PROBE můžeme uvést napětí a proudy v konvencích PSpice. Automaticky bude provedena konverze do formátu PROBE. Při volání „prázdného“ příkazu .PROBE je třeba zadávat napětí a proudy podle konvencí PROBE.

Zadávání v PROBE = zadávání v PSpice s těmito výjimkami:

V(název uzlu) Příklad: V(vstup)	napětí uzlu, nazvaného jiným řetězcem než číslem, oproti referenčnímu uzlu; oproti PSpice se nepoužívají závorky [] napětí mezi uzlem „vstup“ a referenčním uzlem
Nejsou podporovány zápisy	
V(jméno dvojpólu)	
V<jméno svorky>(jméno vícepólu)	

V<jméno svorky1><jméno svorky2> (jméno vícepólu)

(s výjimkou, že jsou volány s příkazem .PROBE, tj. že veličiny jsou vypočteny PSpicem a uloženy v souboru .dat)

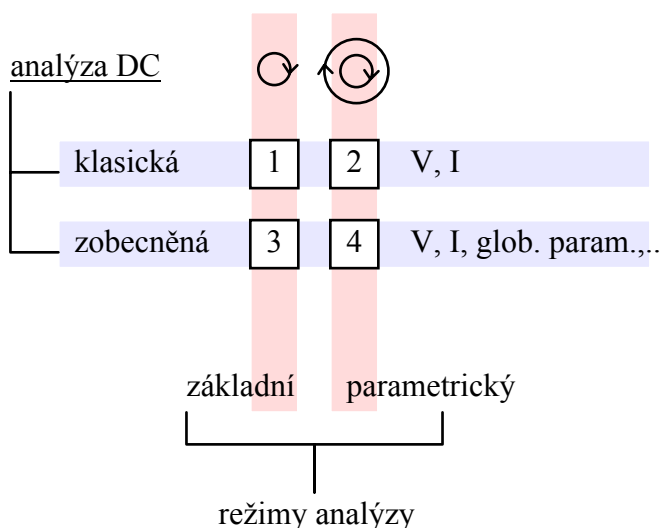
6.3 DC analýza = stejnosměrná analýza

- jednorázový výpočet stejnosměrného pracovního bodu (žádný příkaz pro analýzu, příp. .OP)
- analýza DC (příkaz .DC)
- analýza „Dynamic DC“ – dynamická, interaktivní DC analýza (umožňuje MicroCap, nikoliv SPICE)

Analýza DC

= napodobování funkce přístroje pro snímání stejnosměrných charakteristik nelineárních obvodů (AV charakteristiky diod...).

V PSPICE může být DC analýza provozována buď v tzv. základním nebo v tzv. parametrickém režimu. Měřenými veličinami mohou být buď pouze napětí a proud (klasická DC analýza), nebo i další veličiny, například globální parametry (zobecněná DC analýza).

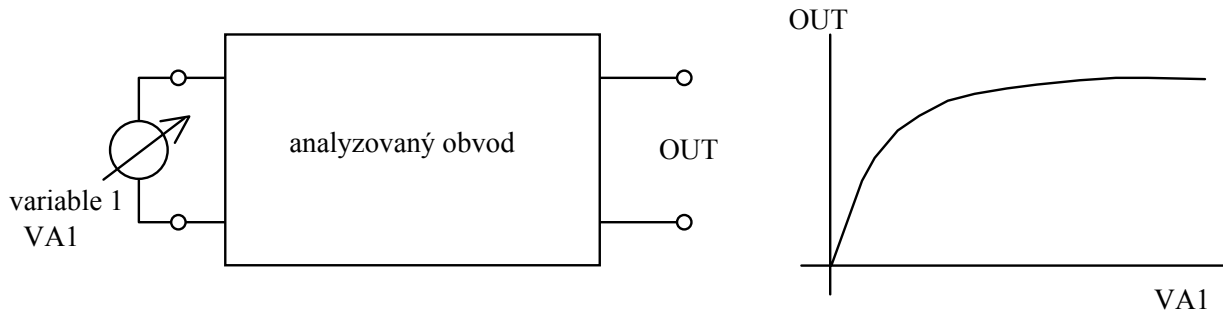


1: AV charakteristika diody, **2:** síť výstupních charakteristik tranzistoru, **3:** $I_c=f(R_c)$, **4:** $I_c=f(BF, TEMP)$.

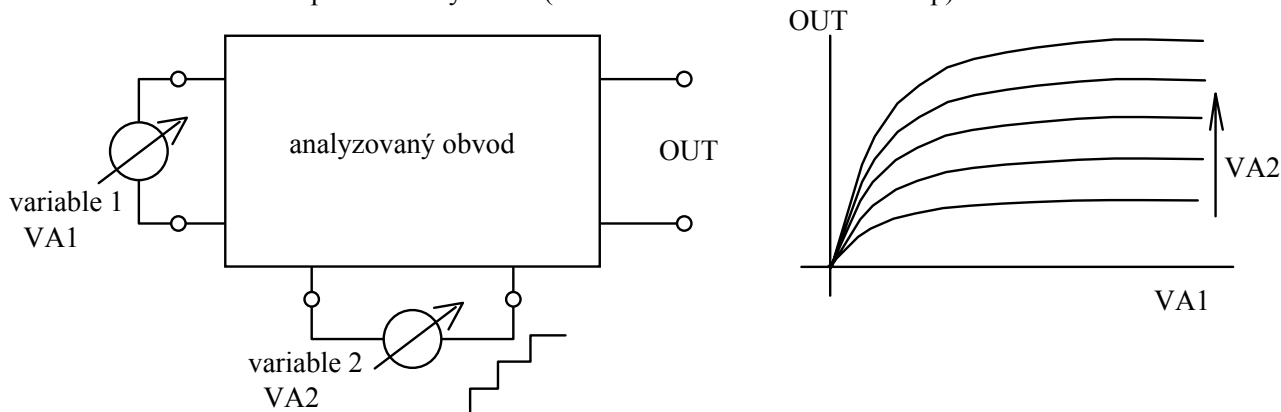
V základním režimu DC analýzy krokuje tzv. proměnná č. 1 (variable 1, VA1). V každém kroku se vypočte stejnosměrný ustálený stav. Výsledkem analýzy je závislost stejnosměrné výstupní veličiny (out) na VA1. Typickým příkladem je ampérvoltová charakteristika diody (VA1 je napětí na diodě, OUT je proud diodou).

V parametrickém režimu dochází rovněž ke krokování proměnné VA1, avšak ve vnější smyčce je krokována další proměnná, variable 2 (VA2). Výsledkem je síť stejnosměrných charakteristik. Typickým příkladem je síť výstupních charakteristik tranzistoru (VA1 je napětí kolektor-emitor, VA2 je proud báze, OUT je proud kolektoru).

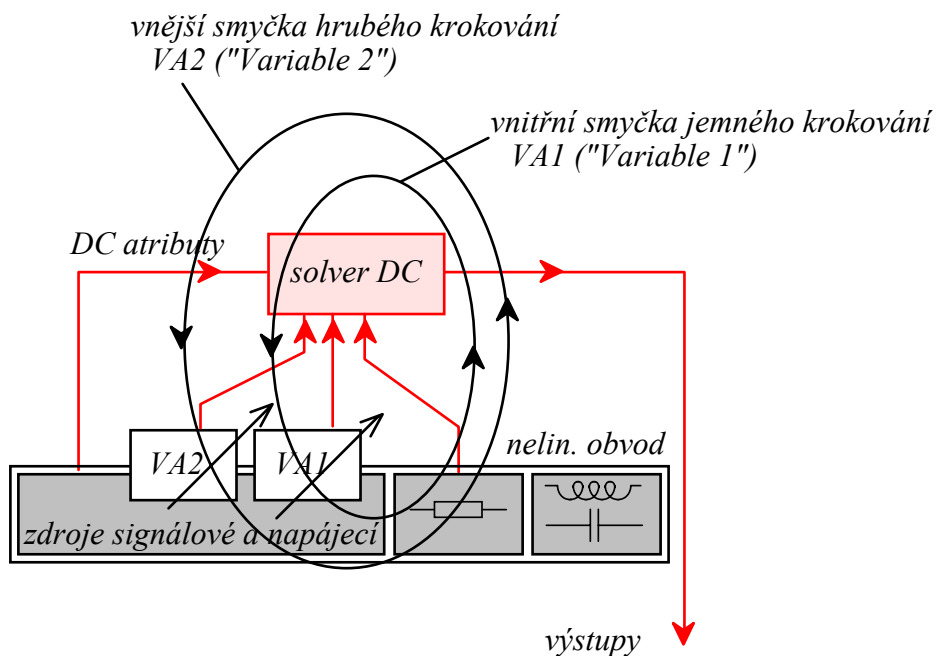
základní režim (jednoduché krokování / Single sweep):



parametrický režim (vnořené krokování / Nested sweep):



Jak postupuje simulátor při analýze DC:



Co všechno se dá krokovat (povolené typy proměnných VA1 a VA2):

Při **klasické** stejnosměrné analýze:

-napětí a proudy nezávislých zdrojů v obvodu.

Při **zobecněné** stejnosměrné analýze:

- napětí a proudy nezávislých zdrojů v obvodu,
- teplota,
- libovolný parametr z modelů součástek v obvodu,
- libovolná globální proměnná.

Jak se chovají ostatní zdroje a nekrokové parametry:

- nezávislé zdroje jsou nahrazeny zdroji o parametrech DC
- řízené zdroje pracují se stejnosměrnými vzorci
- v modelech prvků se uplatňují pouze stejnosměrné parametry.

Syntaxe příkazu .DC:

Základní režim – jednoduché krokování:

Lineární krokování

.DC [LIN] <jméno VA1> <počáteční hodnota> <konečná hodnota> <krok>

Krok musí být větší než 0. Počáteční hodnota může být i větší než konečná hodnota.

Příklady:

.DC LIN Vbat 0V 15V 1V	; rozmitání napětí zdroje Vbat od 0V do 15V po 1V
.DC Vbat 0V 15V 1V	; stejné jako v předchozím případě
.DC Idc 100mA -100mA 1mA	; zmenšování proudu zdroje Idc od 100mA do
+	-100mA s krokem 1mA
.DC NPN Qtranzistor(BF) 100 500 10	; rozmitání parametru BF všech tranzistorů, které mají
+	model Qtranzistor, od 100 do 500 s krokem 10
.DC PARAM Rx 1k 10k 1k	; rozmitání globálního parametru Rx od 1k do 10k po
+	1k

Logaritmické krokování

.DC DEC <jméno VA1> <počáteční hodnota> <konečná hodnota> <počet kroků na dekádu>

nebo

.DC OCT <jméno VA1> <počáteční hodnota> <konečná hodnota> <počet kroků na oktávu>

Konečná hodnota musí být větší než počáteční. Počet kroků musí být celé kladné číslo.

Příklady:

.DC DEC Vx 1mV 1V 1	; rozmitání napětí zdroje Vx v hodnotách 1mV,
+	10mV, 100mV, 1V
.DC OCT PARAM Cx 1n 16n 1	; rozmitání globálního parametru Cx v hodnotách 1nF,
+	2nF, 4nF, 8nF a 16nF

Krokování „LIST“

.DC <jméno VA1> LIST <výčet hodnot>

Příklady:

```
.DC Vbat LIST 5V 15V           ; provede se výpočet pro dvě napětí 5V a 15V
.DC PARAM beta LIST 100       ; provede se výpočet pro jedinou hodnotu globálního
+                               parametru beta = 100
.DC PARAM beta 100 100 1      ; stejný efekt jako v předchozím případě (jde o
+                               lineární krokování, kde konečná hodnota je rovna
+                               počáteční, na velikosti kroku nezáleží)
```

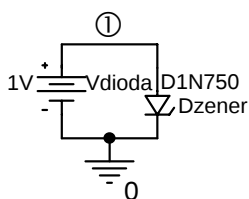
Parametrický režim – vnořené krokování:

Syntaxe navazuje na jednoduché krokování, následuje jméno druhé krokované proměnné VA2 a specifikace krokování, která je stejná jako u jednoduchého krokování.

Příklad:

```
.DC Vce 0 15 0.1 Ib 0 100u 10u ; ve vnější smyčce se nastavuje proud Ib lineárně od
+                               nuly do 100uA po 10uA; pro každou hodnotu Ib
+                               proběhnou výpočty pro napětí Vbe od nuly do 15V po
+                               0.1V.
```

Ukázky simulací:



Zenerka

*

Dzener 1 0 D1N750

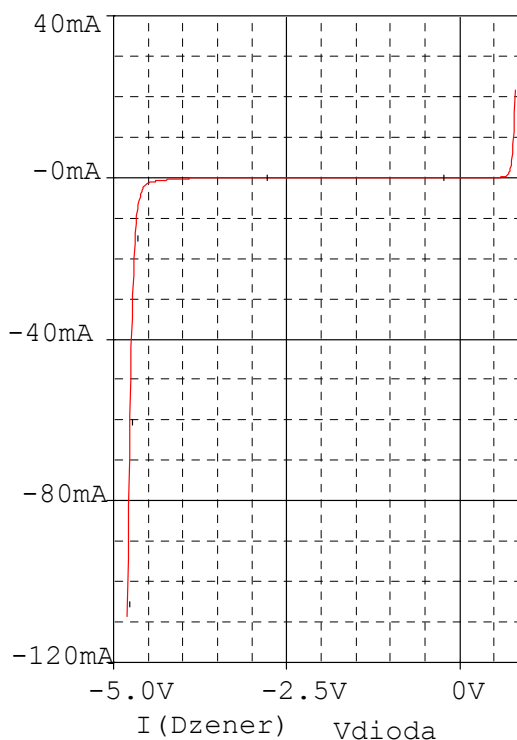
Vdioda 1 0 1V

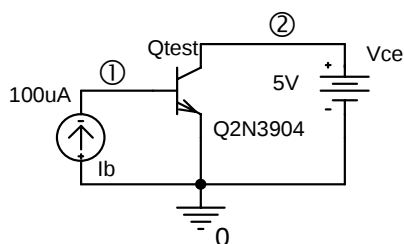
.DC Vdioda -4.8 0.8 0.01

.probe

.lib

.END





AV char

*

Qtest 2 1 0 Q2N3904

Vce 2 0 5V

Ib 0 1 100uA

.DC Vce 0V 10V 0.01V Ib 0A 100uA

+10uA

.probe

.lib

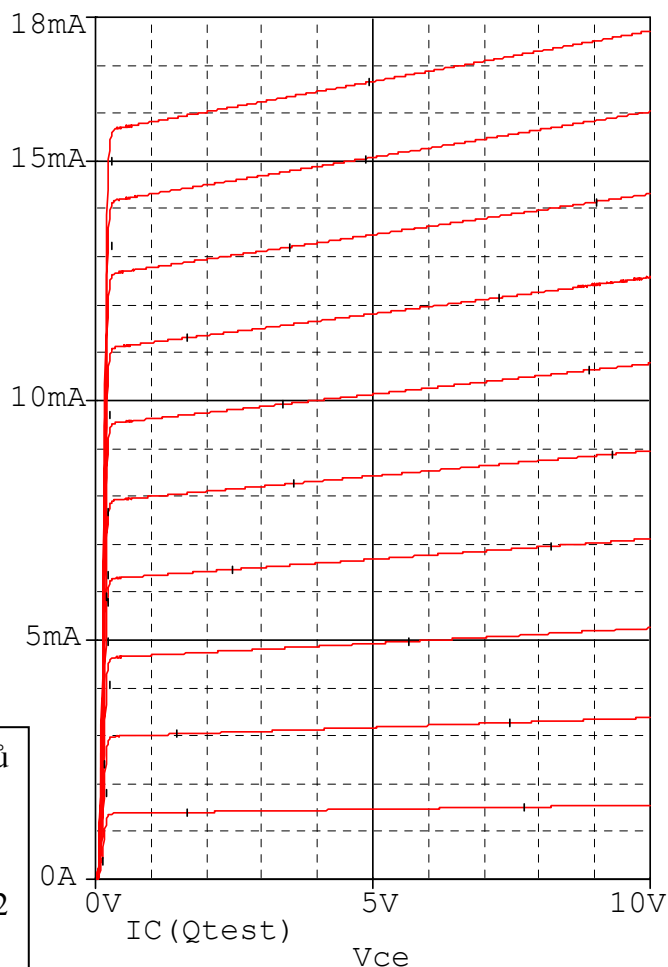
*

.END

Poznámka – které parametry modelů nelze krokovat:

L, W MOSFETu

Teplotní parametry, např. TC1 a TC2 rezistorů.



Příkaz .PRINT – vypisování výsledků analýzy do výstupního souboru

Tento příkaz funguje pouze ve spolupráci s příkazy pro základní analýzy, např. .DC.

.PRINT <označení analýzy> <výčet napětí a proudů>

Příklady:

.DC Vbat 15 15 1	; DC analýza pro jedinou hodnotu rozmítaného zdroje Vbat
.PRINT V(R1) I(Rx) V([uzel])	; do výstupního souboru se запиše 1 řádek s hodnotami napětí na R1, proudu tekoucího přes Rx, a napětí mezi uzlem o názvu uzlu a referenčním uzlem
+	
+	
.DC Vbat 5 15 1	; DC analýza pro 11 hodnot Vbat od 5V do 15V po 1V
.PRINT V(R1) I(Rx) V([uzel])	; do výstupního souboru se запиše 11 řádků s hodnotami napětí na R1, proudu tekoucího přes Rx, a napětí mezi uzlem o názvu uzlu a referenčním uzlem; každému řádku budou odpovídat hodnoty, vypočtené pro jednu z krokovaných hodnot Vbat
+	
+	
+	
+	

Počet míst vypisovaných číselných údajů je určován položkou NUMDGT v globálních podmínkách simulace. NUMDGT = počet míst mantisy za desetinnou tečkou plus 1.

Standardně je NUMDGT=4. Chceme-li např. výpis na 5 desetinných míst, umístíme do vstupního souboru příkaz

```
.OPTIONS NUMDGT=6
```

6.4 Shrnutí kapitoly 6

Hledání stejnosměrného řešení patří k nejdůležitějším operacím v simulačním programu, protože probíhá v rámci všech základních analýz. Z matematického hlediska se jedná o iterační řešení soustavy nelineárních rovnic. U některých obvodů nemusí vyjít výsledek podle našich představ: buď řešení není vůbec nalezeno v důsledku tzv. problémů s konvergencí, nebo je nalezeno řešení, o které nemáme zájem. Řešení těchto problémů nebývá jednoduché. Jedním ze základních nástrojů je příkaz .NODESET. Většinou je však nutné „pohrát si“ s položkami v „global settings“ nebo učinit příslušná opatření již na předchozí úrovni, tj. při tvorbě modelu obvodu. Řada problémů s konvergencí bývá způsobena chybami při modelování. Zejména tzv. behaviorální modelování (ABM), kdy zjednodušeně popisujeme vnější chování složitějších bloků bez vazby na vnitřní pochody uvnitř, může být častým zdrojem nestandardního chování simulačního programu.

Jednou ze základních analýz v PSPICE, založené na algoritmickém hledání stejnosměrných řešení, je analýza DC neboli stejnosměrná analýza, kterou využijeme zejména pro modelování stejnosměrných charakteristik obvodů, tj. charakteristik měřených metodou „bod po bodu“ v stejnosměrném ustáleném stavu. PSPICE může generovat několik typů výsledků DC analýzy, jejichž rozdělení je na str. 55.

6.5 Řešené příklady

Příklad 6.1 Bistabilní klopný obvod

Nalezněte všechny stejnosměrné pracovní body bistabilního klopného obvodu s tranzistory podle obrázku.

Řešení:

flipflop

Vbat 1 0 10V

R1 1 2 1k

R2 1 3 1k

R3 2 5 56k

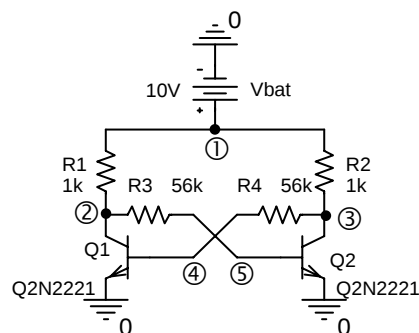
R4 3 4 56k

Q1 2 4 0 Q2N2221

Q2 3 5 0 Q2N2221

.lib bipolar.lib

.end



PSpice nalezne tzv. nestabilní stejnosměrný pracovní bod, kdy oba tranzistory jsou v aktivním režimu (v praxi se nemůže udržet): $V(2) = V(3) = 4.7792V$.

K nalezení stabilního pracovního bodu, odpovídajícího otevřenému Q1 a uzavřenému Q2, přidáme příkaz .NODESET, např.

```
.nodeset V(3)=10V
```

Pak vyjde $V(2)=0.1244V$, $V(3)=9.8370V$

Vyzkoušejte obdobně nastavit pracovní bod pro uzavřený Q1 a otevřený Q2.

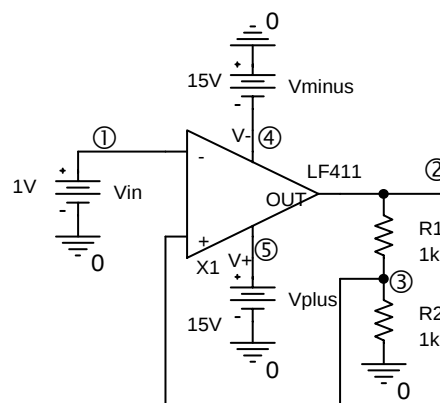
Příklad 6.2 Komparátor s hystezí

Nalezněte všechny stejnosměrné pracovní body invertujícího komparátoru s hystezí.

V praxi může nastat pouze saturace OZ, neboť v obvodu působí kladná zpětná vazba. Kladné saturaci odpovídá napětí $V(2)$ cca $+14V$, záporné saturaci napětí $V(2)$ cca $-14V$.

V obvodu může nastat další teoretický stav – nestabilní rovnováha, kdy diferenční napětí je nulové, z toho plynoucí napětí $V(3)=1V$ a $V(4)=2V$. Tento stav, který je běžně pozorovatelný u záporné zpětné vazby (při přehozených vývodech OZ), je však v tomto zapojení trvale neudržitelný.

Vyzkoušejte, který pracovní bod nalezne PSpice bez příkazu .NODESET. Pak naleznete všechny stavy.



Řešení:

invert.komparator s OZ

Vin 1 0 1V

R1 2 3 1k

R2 3 0 1k

X1 3 1 5 4 2 LF411

Vplus 5 0 15V

Vminus 0 4 15V

.lib

.end

Bez příkazu .NODESET je nalezen nestabilní pracovní bod (!).

S příkazy .NODESET $V(2)=15V$, případně .NODESET $V(2)=-15V$, jsou nalezeny stabilní pracovní body, kdy $V(2)=14.2960V$, případně $V(2)=-14.2960V$.

Příklad 6.3 Lineární zesilovač s operačním zesilovačem

Přehod'te vstupní svorky OZ z předchozího příkladu 6.2, takže dostanete lineární zesilovač se zesílením 2. Zjistěte výstupní napětí. Zjistěte, zda je možné příkazem .NODESET nalézt další řešení.

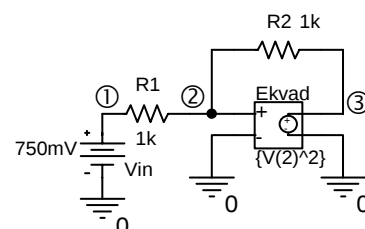
Řešení:

Výstupní napětí bude nyní 2V, žádné další řešení neexistuje (jde o lineární úlohu s jediným řešením, zpětná vazba je záporná, obvod je stabilní).

Příklad 6.4: Obvod s kvadrátorem a kladnou zpětnou vazbou

Vypočtete napětí na výstupu kvadrátoru. Zjistěte, zda je možné příkazem .NODESET nalézt další řešení.

Obvod má dvě stejnosměrná řešení: $V(3)=250mV$ a $V(3)=2.25V$. Ověřte si to.



Řešení:

obvod s kvadrátorem

Vin 1 0 750mV

R1 1 2 1k

R2 2 3 1k

Ekvad 3 0 VALUE={V(2)^2}

.end

PSpice nalezne řešení $V(3)=250\text{mV}$. Další řešení nalezne např. po použití příkazu `.NODESET V(2)=2V`. Vyzkoušejte, že nastavování $V(3)$ nemá efekt (jedná se o výstupní napětí řízeného zdroje).

Příklad 6.5: Obvod s kvadrátorem a kladnou zpětnou vazbou

Pokuste se nalézt výstupní napětí z příkladu 6.4, je-li vstupní napětí $V(1)=1\text{V}$.

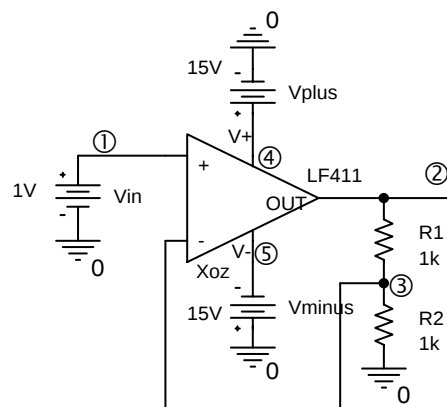
Řešení:

Obvod má nyní jediné řešení $V(3)=1\text{V}$. Toto řešení je velmi citlivé na parametry součástek. Při vstupním napětí větším než 1V dokonce řešení (v rámci tohoto modelu obvodu) neexistuje!

PSpice bude mít nyní problémy s konvergencí řešení. Odpomoc – modifikace globálních podmínek simulace. Změníme parametr `ABSTOL` ze standardního 10^{-12} na 10^{-10} a potvrdíme „OK & resume simulation“. Podrobnosti později. Částečně si lze pomoci i příkazem `.NODESET`, ale musíme dobře odhadnout konečné řešení, např. `.NODESET V(2)=0.9V`.

Příklad 6.6: Napěťová převodní charakteristika lineárního zesilovače s OZ

Zjistěte stejnosměrnou napěťovou převodní charakteristiku zesilovače, tj. graf závislosti výstupního napětí na vstupním napětí, v rozsahu vstupního napětí od -10V do $+10\text{V}$. Z charakteristiky odvoďte střídavé zesílení, saturační úrovně a rozsah vstupního napětí, v němž se zesilovač chová jako lineární prvek.



Řešení:

Vin 1 0 1V

R1 2 3 1k

R2 3 0 1k

Xoz 1 3 4 5 2 LF411

Vplus 4 0 15V

Vminus 5 0 -15V

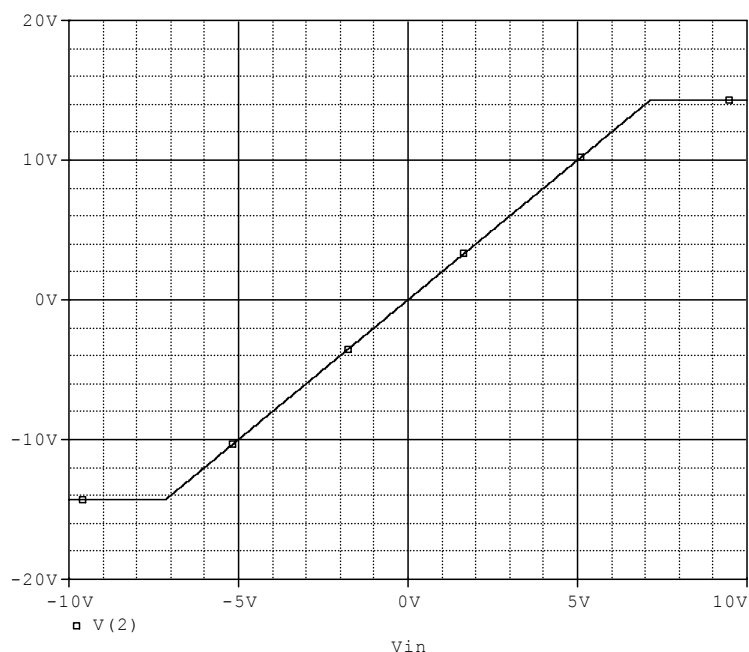
.dc Vin -10 10 0.01 ; rozmítání Vin od -10V do +10V s krokem 0.01V

.probe ; vygenerování souboru *.dat pro PROBE a spuštění PROBE

.lib

.end

Po otevření prázdného okna PROBE přidáme křivku (Add Trace), odpovídající napětí $V(2)$.



Výsledky: saturační úrovně -14.296V, +14.296V, hranice lineárního režimu pro V(1) jsou -7.15V a +7.15V (lze změřit pomocí kurzorů – „Toggle Cursor“, jemně krokovat po vypočtených bodech „Cursor Point“. Pak střídavé zesílení vychází cca $14.296/7.15=1.9994$. Podle teorie má vyjít 2.

Zesílení lze i automaticky změřit: Trace/Measurement, vybereme SlewRate_Rise, Eval, Name of trace to search: V(2). OK. Objeví se výsledek 1.99998 (rozdíl oproti předchozímu výpočtu je způsoben zaokrouhlováním mezivýsledků).

Další možnost automatického měření: Trace/Evaluate measurement...

Poznámka: Zkuste nahradit příkaz .PROBE příkazem .PROBE V(2). Význam – viz přednášky.

Příklad 6.7: Sít' převodních charakteristik lineárního zesilovače s OZ

Získejte charakteristiky zesilovače z příkladu 6.6 pro odpory R1 z řady 100ohmů, 1kohm, 10kohmů.

Řešení:

neinvertující zesilovác s OZ

Vin 1 0 1V

R1 2 3 {R1}

R2 3 0 1k

Xoz 1 3 4 5 2 LF411

Vplus 4 0 15V

Vminus 5 0 -15V

.dc Vin -10 10 0.01

.param R1 1k

.step param R1 list 100 1k 10k

.probe

.lib

.end

Příklad 6.8: *Sít' převodních charakteristik lineárního zesilovače s OZ*

Získejte charakteristiky zesilovače z příkladu 6.6 pro symetrické napájecí napětí (10, 11, 12, 13, 14, 15)V.

Řešení:

neinvertující zesilovavac s OZ

Vin 1 0 1V

R1 2 3 1k

R2 3 0 1k

Xoz 1 3 4 5 2 LF411

Vplus 4 0 {Vnap}

Vminus 5 0 {-Vnap}

.dc Vin -10 10 0.01

.param Vnap 15V

.step param Vnap 10 15 1; lineární krokování od 10V do 15V s krokem 1

.probe

.lib

.end

Ověřte, jak u tohoto operačního zesilovače souvisí napájecí napětí se saturačním napětím.

Příklad 6.9: *Sít' převodních charakteristik lineárního zesilovače s OZ*

Vyřešte předchozí příklad 6.8 s využitím vnořeného rozmítání v příkazu .DC.

Řešení:

neinvertující zesilovavac s OZ

Vin 1 0 1V

R1 2 3 1k

R2 3 0 1k

Xoz 1 3 4 5 2 LF411

Vplus 4 0 {Vnap}

Vminus 5 0 {-Vnap}

.dc Vin -10 10 0.01 **param Vnap 10 15 1**

.param Vnap 15V

.probe

.lib

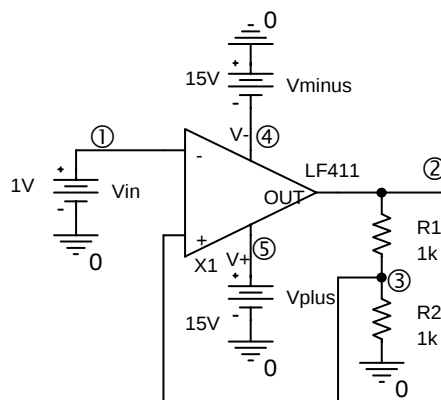
.end

Příklad 6.10: *Převodní charakteristika komparátoru s hysterezí*

Nakreslete stejnosměrnou převodní charakteristiku komparátoru s hysterezí z příkladu 6.2 při změně vstupního napětí a) z -10V do +10V, b) z +10V do -10V.

Řešení:

a)



```

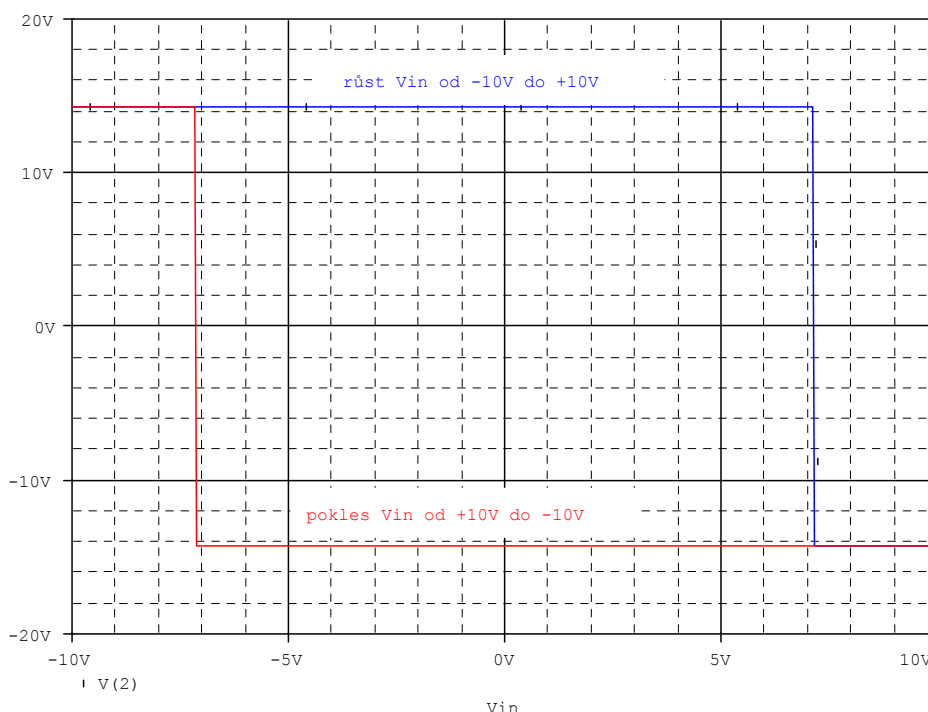
invert.komparator s OZ
Vin 1 0 1V
R1 2 3 1k
R2 3 0 1k
X1 3 1 5 4 2 LF411
Vplus 5 0 15V
Vminus 4 0 -15V
.dc Vin -10 10 0.01
.probe
.lib
.end

```

b)

Modifikujeme příkaz .DC:

.DC Vin 10 -10 0.01



Z výsledků simulací je vidět hystereze. Příkaz .DC nemůže být použit vícekrát v rámci jednoho vstupního souboru, takže běžně nelze při stejnosměrné analýze vykreslit hysterezní charakteristiku.

Překlápěcí úrovně vstupního napětí jsou rovny polovině hysterezního napětí, tedy asi $\pm 7.14V$.

Příklad 6.11: Převodní charakteristika komparátoru s hysterezí – selhání PSPICE

Nakreslete stejnosměrnou převodní charakteristiku komparátoru s hysterezí z příkladu 6.2 při změně vstupního napětí a) z -7V do +7V, b) z +7V do -7V.

Řešení:

```

invert.komparator s OZ
Vin 1 0 1V
R1 2 3 1k

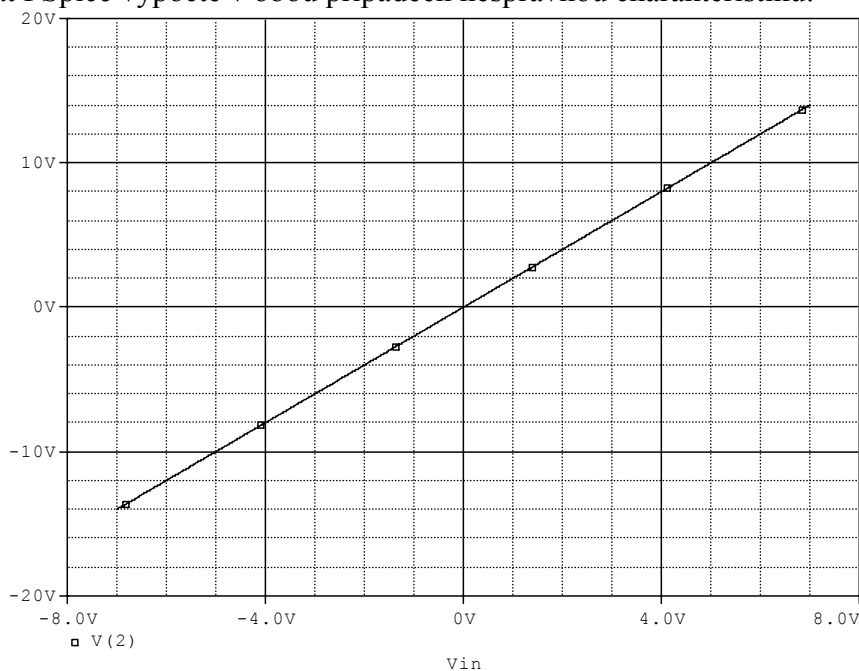
```

```

R2 3 0 1k
X1 3 1 5 4 2 LF411
Vplus 5 0 15V
Vminus 4 0 -15V
.dc Vin -7 7 0.01; v dalším pokusu změníme na .dc Vin 7 -7 0.01
.probe
.lib
.end

```

Tentokrát PSpice vypočte v obou případech nesprávnou charakteristiku:



Na vině je nalezení nestabilního (tedy nesprávného) pracovního bodu hned při výpočtu prvního bodu křivky. Při vstupním napětí -7V, resp. +7V jsou totiž tři možná řešení. Při vstupním napětí -10V, resp. +10V (předchozí příklady) je jediné možné řešení – saturace. Obrázek tak vlastně ukazuje část převodní charakteristiky jiného obvodu – lineárního zesilovače se zesílením 2 z příkladu 33.

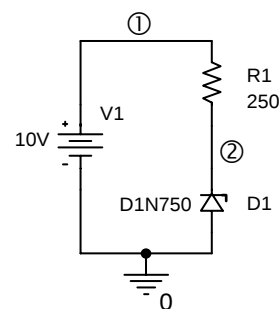
Vyřešení problému – kombinace s příkazem .NODESET:

.NODESET V(2)=15V pro a), .NODESET V(2)=-15V pro b).

Příklad 6.12: Stabilizátor se Zenerovou diodou – netypické využití příkazu .PRINT

Pomocí příkazu .PRINT vypíšte ve výstupním souboru napětí na R1, výkon na R1 a výkon na diodě.

Poznámka: příkaz .PRINT funguje jen při některé ze základních analýz, např. .DC. Pomocí .PRINT lze přímo zobrazit pouze napětí a proudy, nikoliv tedy výkony. Pomůžeme si např. řízeným zdrojem proudu typu G, zapojeným oběma vývody na referenční uzel, jeho proud vypočteme tak, aby byl roven sledované veličině (třeba výkonu), a následně tento proud zobrazíme příkazem .PRINT.



Řešení:

Stabilizátor se Zenerovou diodou

*

V1 1 0 10V

R1 1 2 250

D1 0 2 D1N750

.DC V1 10 10 1; nastaví se jen jedna hodnota V1=10V; krok nesmí být 0

.print DC V(R1) i(GvykonR) i(GvykonD)

GvykonR 0 0 value={V(1,2)^2/250}; výpočet výkonu na R1

GvykonD 0 0 value={-V(2)*i(V1)}; výpočet výkonu na D1

.lib

.END

Výsledky: Napětí na R1 je 5.297V, výkon na R1 je 112.2mW, výkon na diodě je 99.65mW.

Příklad 6.13: Stabilizátor se Zenerovou diodou – netypické využití příkazu .PRINT

Získejte tabulku napětí na R1, napětí na D1, výkonů na R1 a výkonů na D1 pro vstupní napětí od 0V do 10V po 1V.

Řešení:

stabil

V1 1 0 10V

R1 1 2 250

D1 0 2 D1N750

.DC V1 0 10 1

.print DC V(R1) V(2) i(GvykonR) i(GvykonD)

GvykonR 0 0 value={V(1,2)^2/250}

GvykonD 0 0 value={-V(2)*i(V1)}

.lib

*

.END

Výsledek:

V1	V(R1)	V(2)	I(GvykonR)	I(GvykonD)
0.000E+00	2.842E-23	-2.842E-23	0.000E+00	0.000E+00
1.000E+00	3.547E-05	1.000E+00	5.032E-12	1.419E-07
2.000E+00	4.595E-04	2.000E+00	8.447E-10	3.675E-06
3.000E+00	5.978E-03	2.994E+00	1.429E-07	7.159E-05
4.000E+00	6.741E-02	3.933E+00	1.818E-05	1.060E-03
5.000E+00	4.589E-01	4.541E+00	8.422E-04	8.335E-03
6.000E+00	1.371E+00	4.629E+00	7.523E-03	2.539E-02
7.000E+00	2.341E+00	4.659E+00	2.191E-02	4.362E-02
8.000E+00	3.322E+00	4.678E+00	4.413E-02	6.216E-02
9.000E+00	4.308E+00	4.692E+00	7.423E-02	8.085E-02
1.000E+01	5.297E+00	4.703E+00	1.122E-01	9.965E-02
1.100E+01	6.288E+00	4.712E+00	1.581E-01	1.185E-01
1.200E+01	7.280E+00	4.720E+00	2.120E-01	1.374E-01

1.300E+01	8.273E+00	4.727E+00	2.738E-01	1.564E-01
1.400E+01	9.267E+00	4.733E+00	3.435E-01	1.754E-01
1.500E+01	1.026E+01	4.739E+00	4.212E-01	1.945E-01

Obvod začíná stabilizovat výstupní napětí při vstupním napětí cca od 5V nahoru. Při vyšších hodnotách vstupního napětí je výkon na R1 větší než výkon na diodě.

Výpis hodnot bývá nepřehledný, mnohdy dáme přednost grafům v PROBE.

Tip: pokud chcete zobrazit čísla na více míst za desetinnou tečkou, umístíme do vstupního souboru příkaz .OPTIONS, např.

.OPTIONS NUMDGT=6

Příklad 6.14: Stabilizátor se Zenerovou diodou – práce s PROBE, návaznost na předchozí příklady

Získejte v PROBE dva obrázky: závislost výkonů na R1 a D1 na vstupním napětí (obrázek 1) a závislost výstupního napětí na vstupním napětí (obrázek 2).

Řešení:

stabil

V1 1 0 10V

R1 1 2 250

D1 0 2 D1N750

.DC V1 0 15 0.1 ;je zmenšen krok, aby křivky byly dostatečně hladké

GvykonR 0 0 value={V(1,2)^2/250}

GvykonD 0 0 value={-V(2)*i(V1)}

.probe

.lib

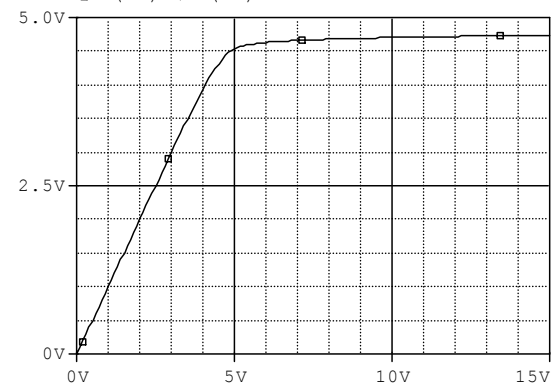
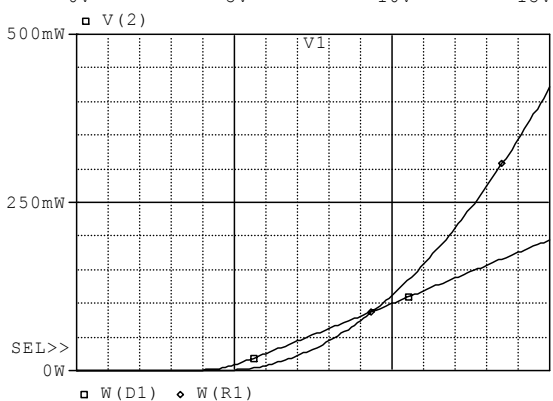
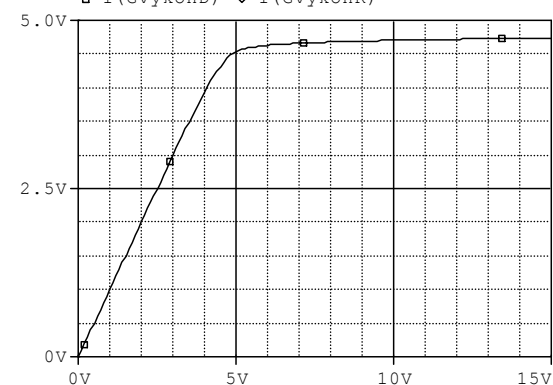
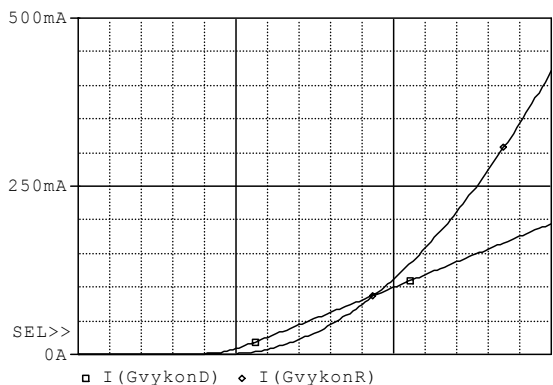
*

.END

Nejprve definujeme spodní obrázek přidáním křivky V(2). Potom přidáme další obrázek (Plot/Add Plot to Window) a definujeme v něm křivky I/(GvykonD) a I(GvykonR).

Ze spodního obrázku je zřejmé, že obvod nestabilizuje napětí, pokud vstupní napětí poklesne pod cca 5V. Horní obrázek ukazuje, že pro vstupní napětí menší než cca 9.5V je výkon na diodě větší než výkon na rezistoru, pro větší vstupní napětí je tomu naopak.

Horní obrázek sice „ukazuje“ proudy v mA, ve skutečnosti se ale jedná o výkony v mW.



V1

Příklad 6.15: Stabilizátor se Zenerovou diodou – práce s PROBE, návaznost na předchozí příklady

Zobrazte výkony na rezistoru a diodě bez mezivýpočtů GvykonR a GvykonD, jen s využitím možností PROBE.

Řešení:

stabil

V1 1 0 10V

R1 1 2 250

D1 0 2 D1N750

.DC V1 0 15 0.1

.probe

.lib

.END

V PROBE zavedeme do obrázku dvě křivky pomocí těchto matematických operací (například):

$$(V(1)-V(2))*I(R1)$$

$$V(2)*I(R1)$$

nebo ještě pohodlněji

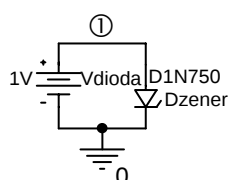
$$W(R1)$$

$$W(D1)$$

Příklad 6.16: Ampérvoltová charakteristika diody

Zobrazte ampérvoltovou charakteristiku diody D1N750 pro napětí z intervalu od -4.8V do +0.8V.

Řešení:



Zenerka

Dzener 1 0 D1N750

Vdioda 1 0 1V

.DC Vdioda -4.8 0.8 0.01

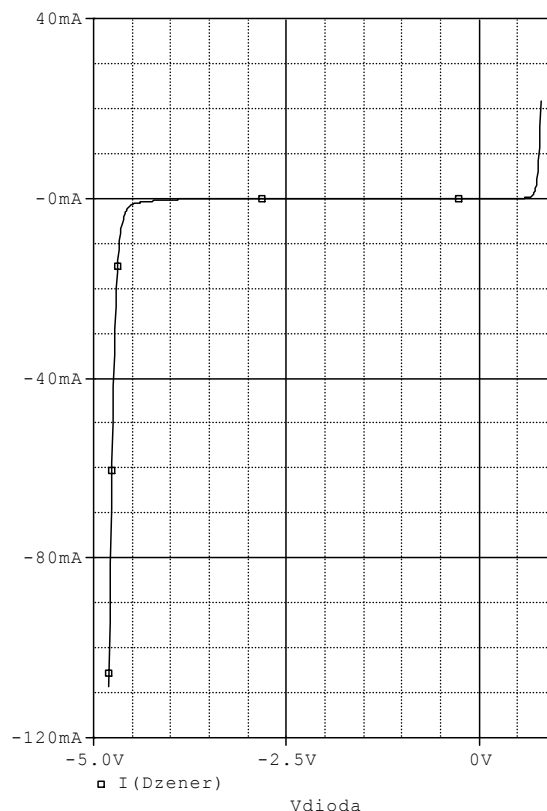
.probe

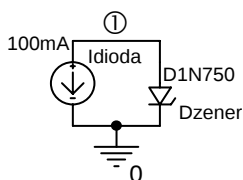
.lib

*

.END

Problém: zdrojem napětí nelze dobře regulovat proud diodou. Lepší by bylo otvírat diodu zdrojem proudu, který budeme rozmnítat např. od -100mA do +100mA:





Zenerka

Dzener 1 0 D1N750

Idioda 1 0 100mA

.DC Idioda -100m 100m 1m

.probe

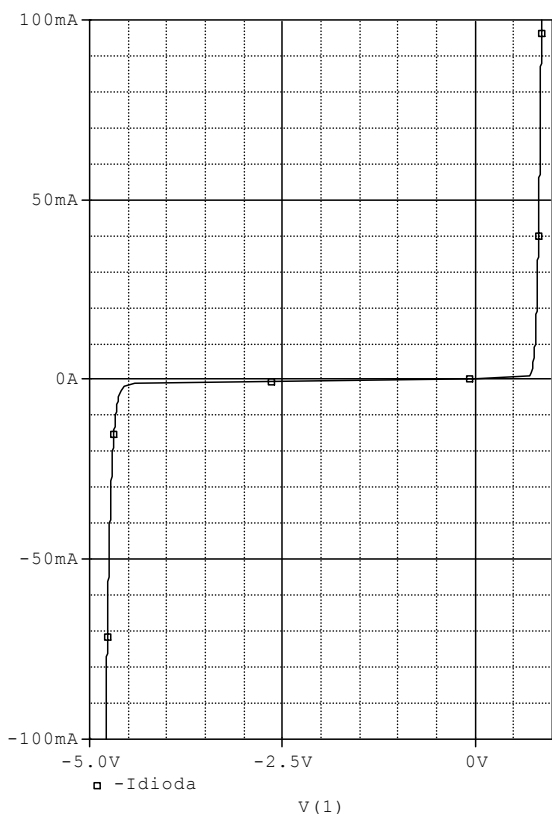
.lib

*

.END

PROBE zobrazuje na vodorovnou osu standardně rozmítanou veličinu. Je třeba „přehodit osy“ a upravit jejich měřítka, vše v okně „Plot/Axis Settings“.

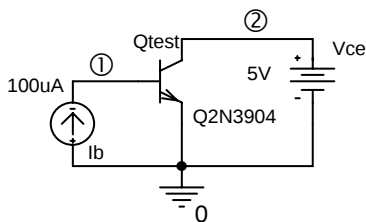
Otázka: proč jsou obě charakteristiky mírně odlišné pro napětí cca od -4.5V do 0V? (Je to v důsledku toho, že nejprve bylo krokováno napětí a pak proud. Krok byl krokován po 1mA, takže v závěrné oblasti diody bylo použito málo bodů k jejímu vykreslení. Zmenšení kroku by vše napravilo).



Příklad 6.17: Sít' výstupních charakteristik bipolárního tranzistoru

Zobrazte sít' výstupních charakteristik tranzistoru Q2N3904. Rozsah napětí kolektor-emitor od 0V do 10V, proud báze od 0 do 100μA.

Řešení:



AV char

Qtest 2 1 0 Q2N3904

Vce 2 0 5V

Ib 0 1 100uA

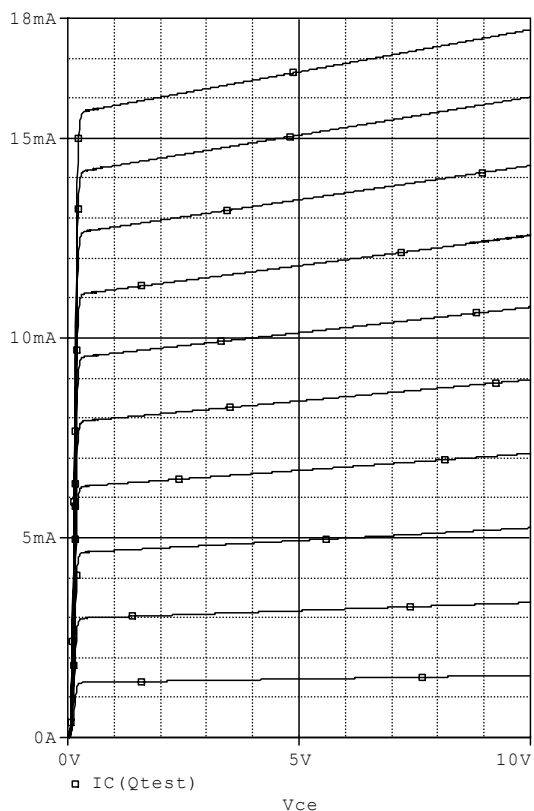
.DC Vce 0V 10V 0.01V Ib 0A 100uA 10uA

.probe

.lib

*

.END

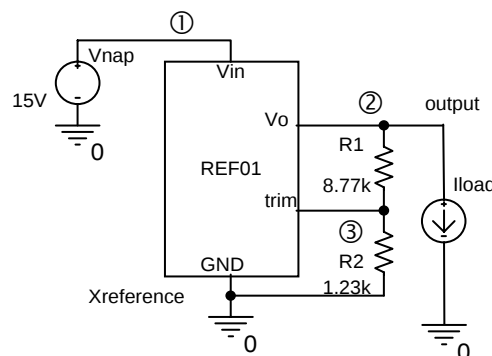


Příklad 6.18: *Zatěžovací charakteristika zdroje s napětovou referencí REF01*

Změřte zatěžovací charakteristiku zdroje napětí 10V s napětovou referencí REF01, tj. závislost výstupního napětí na zatěžovacím proudu.

Řešení:

Tento obvod byl řešen v Př. 5.8. Níže je použit podobvod REF-01/AD z knihovny profesionální verze OrCadPSpice10.



zdroj 10V

*

Vnap 1 0 15V

R1 2 3 8.77k

R2 3 0 1.23k

Iload 2 0 20mA

Xreference 1 0 3 2 REF-01/AD

.DC Iload 0 20mA 0.1mA

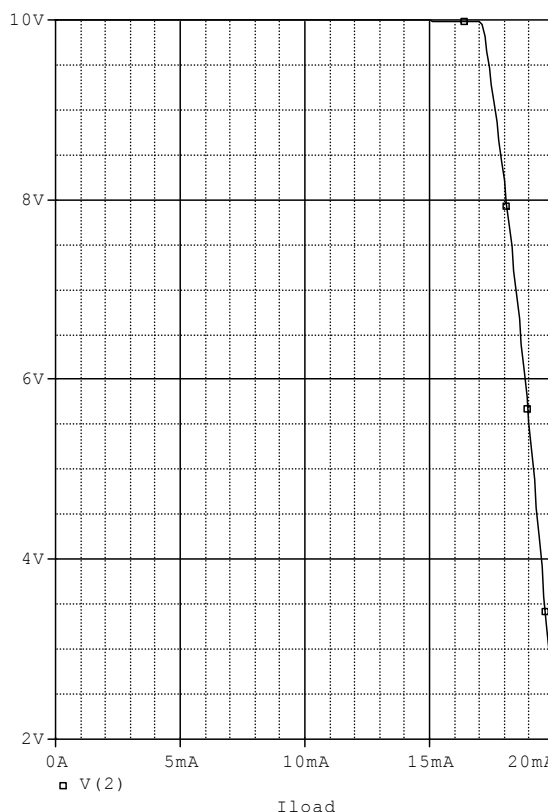
.PROBE

*

.lib

.end

Z výsledků simulace je zřejmé, že daný zdroj spolehlivě pracuje do odběrného proudu cca 17mA.



Příklad 6.19: *Zobecněná DC analýza tranzistorového zesilovače*

Zjistěte závislost stejnosměrného kolektorového napětí a kolektorového proudu na teplotě. Teplotu rozmítejte od nuly do 100 stupňů Celsia. Z křivek odečtěte změnu kolektorového napětí na jeden stupeň v mV/°C a procentuální změnu kolektorového proudu na jeden stupeň (nominální hodnotu IC uvažujte při teplotě 27 stupňů Celsia).

Pozn.: Model tranzistoru je např. v knihovně ebipolar.lib profesionální verze OrCadPSpice 10, nebo si jej získejte např. z Internetu.

Řešení:

zesilDC

*

Vin in 0 AC 1

Cv in baze 5u

R2 baterie baze 100k

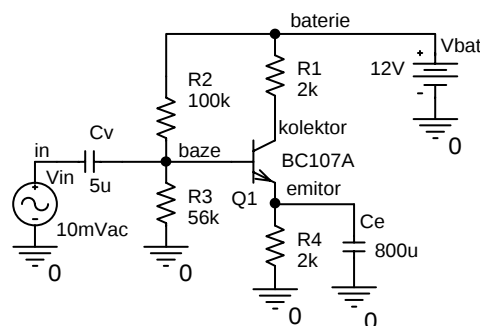
R3 baze 0 56k

R1 baterie kolektor 2k

R4 emitor 0 2k

Ce emitor 0 800u

Q1 kolektor baze emitor BC107A



```
Vbat baterie 0 12V
.lib ebipolar.lib
.DC TEMP 0 100 1
.probe Ic(Q1) Vc(Q1)
.end
```

Z grafů odečteme:

Při 0°C je napětí na kolektoru 8.8092V, při 100°C je napětí 8.5123V. Tedy teplotní součinitel je

$$-2.969\text{mV}/^{\circ}\text{C}.$$

Při 27°C je proud kolektoru 1.6411mA, při 97°C je proud 1.7387mA. Na 1°C to dělá cca nárůst o 0.085%. Pracovní bod je poměrně dobře teplotně stabilizován.

Příklad 6.20: Zobecněná DC analýza tranzistorového zesilovače

Zjistěte závislost kolektorového a emitorového napětí na odporu kolektoru, který rozmítejte od 1kohmu do 10kohmů.

Řešení:

.... Viz příklad 6.19

R1 baterie kolektor {Rc}

...

```
.param Rc 2k
```

```
.DC param Rc 1k 10k 100
```

```
.probe V([kolektor]) V([emitor])
```

```
.end
```

6.6 Kontrolní otázky ke kapitole 6

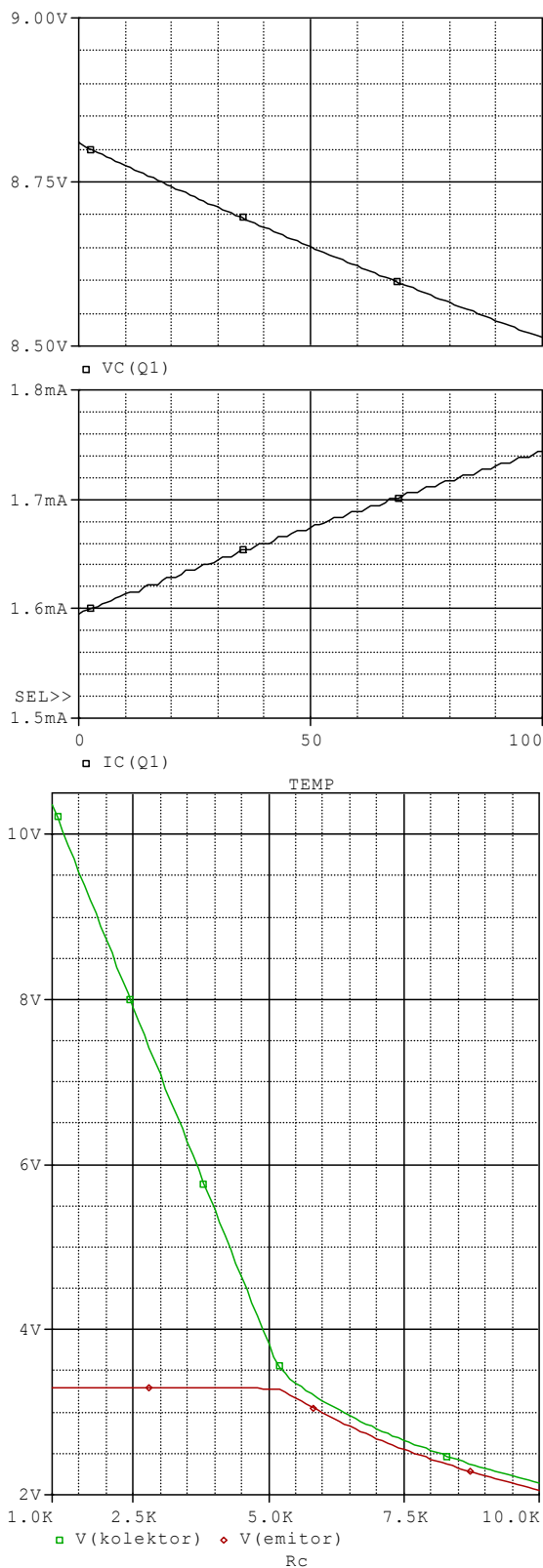
Pokud dospějete k závěru, že na některou z níže uvedených otázek nedovedete vyčíst úplnou odpověď z tohoto elektronického textu, pokuste se dočerpát dodatečné informace z dalších zdrojů.

Správné odpovědi naleznete na konci učebního textu.

1. Stejnoseměrný pracovní bod se v PSPICE hledá pomocí metody

a) iterační, b) analytické, c) půlení intervalu

2. Nelineární obvod může mít: a) jen jedno stejnosměrné řešení, b) jen jedno nebo dvě stejnosměrná řešení, c) více stejnosměrných řešení.



3. Příkaz .NODESET definuje **a)** nastavení uzlových napětí jako výsledek hledání pracovního bodu, **b)** počáteční odhad stejnosměrného řešení před první iterací, **c)** nastavení nulových uzlových napětí a proudů induktory před aplikací Newtonovy-Raphsonovy metody.

4. Označte správně napsaný příkaz:

a) .nodeset Vbe(Q1) 1V, **b)** .NODESET (IL=1A), **c)** .nodeset I(L) -2mA

5. Konverguje-li stejnosměrné řešení k relativně velkým hodnotám napětí a proudů v obvodu, pak je pro přesnost nalezeného řešení rozhodující kritérium: **a)** abstol, **b)** reltol, **c)** kombinace abstol a vntol.

6. V průběhu analýzy .DC simulační program zahlásí problémy s konvergencí. Pak je vhodné pokusit se o: **a)** zvětšení parametru ITL1, **b)** zvětšení parametru ITL2, **c)** zmenšení parametru ITL2.

7. V PSPICE se stejnosměrné napětí mezi uzlem UZEL a referenčním uzlem označí takto:

a) V(uzel), **b)** U(uzel, 0), **c)** v([uzel])

8. Rozhodněte, která syntaxe je správná:

a) .DC Idc 100mA -100mA 1mA, **b)** .DC LIN Ix -100m 100m -1m, **c)** .DC Rx 1k 10k 1k

9. Příkazem .DC můžeme krokovat:

a) jen napětí a proudy nezávislých zdrojů, **b)** napětí libovolného uzlu, i když k němu není připojen zdroj, **c)** teplotu.

10. Příkaz .PRINT slouží k vytištění do výstupního souboru:

a) textového řetězce, který následuje za příkazem, **b)** libovolných napětí a proudů, které jsou definovány tímto příkazem a vypočteny na základě analýzy DC, **c)** jen napětí a proudů zdrojů v obvodu.

6.7 Neřešené příklady

Příklad 6.21: *Modelování kvadrátoru z příkladu 6.4*

Modelujte kvadrátor z příkladu 6.4 polynomiálním zdrojem.

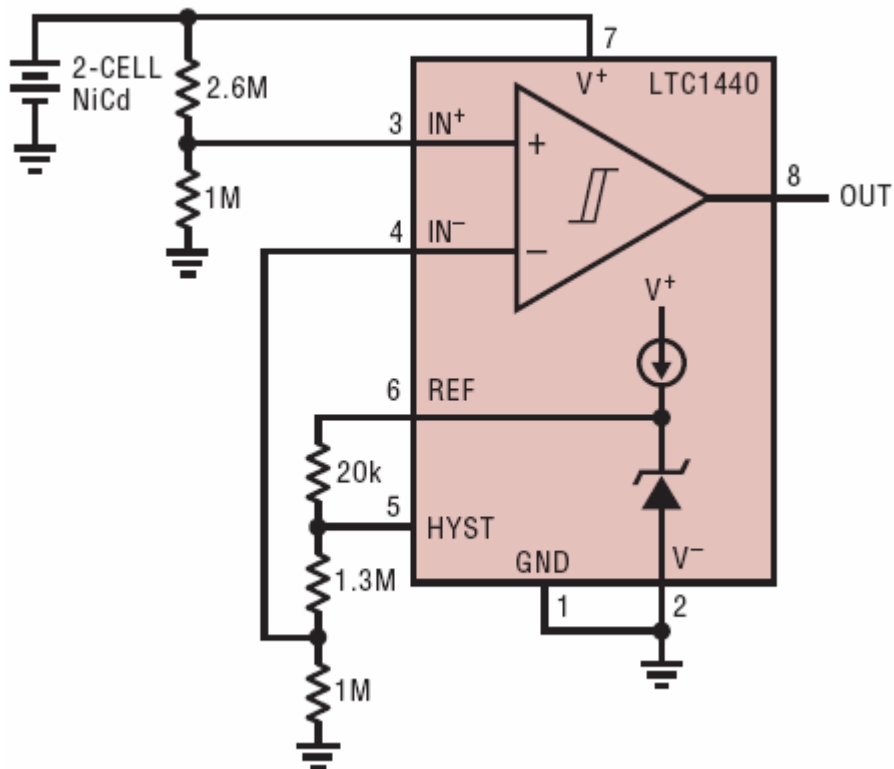
Příklad 6.22: *Modelování zdroje s napětovou referencí REF01 z příkladu 6.18*

Zjistěte, jak se budou zatěžovací charakteristiky měnit při vstupních napětích od 10V do 20V.

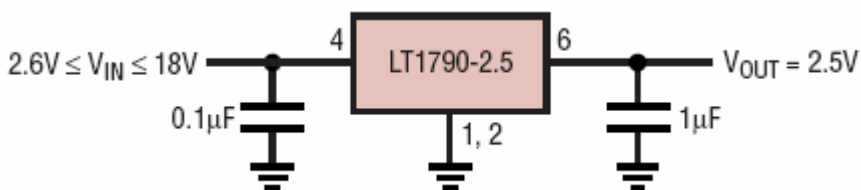
Návod: .DC Iload 0 20m 0.1m Vnap 10 20 2

Příklad 6.23: *Obvod pro hlídání napětí baterie*

Zjistěte závislost stejnosměrného výstupního napětí (OUT) na stejnosměrném vstupním napětí (2-CELL NiCd) pro vstupní napětí rostoucí i klesající. Obvod LTC1440 je nízkopříkonový komparátor a v tomto zapojení slouží jako detektor poklesu napětí NiCd akumulátorové baterie pod kritickou mez.

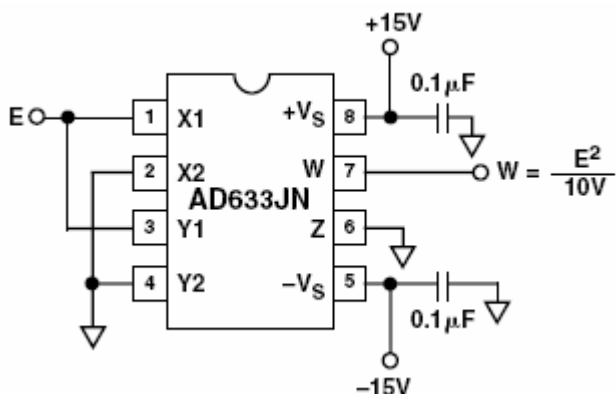
**Příklad 6.24:** *Nízkopříkonová napěťová reference LT1790*

Zjistěte závislost stejnosměrného výstupního napětí (VOUT) na stejnosměrném vstupním napětí (VIN) pro vstupní napětí od 0 do 18V. Obvod LT1790-2.5 je napěťová reference se zanedbatelným příkonem pro bateriově napájené přístroje. Změřte závislost výkonu ze vstupního zdroje na napětí tohoto zdroje.

**Příklad 6.25:** *Kvadrátor*

s analogovou násobičkou AD633

Ověřte pomocí DC analýzy, zda se pro vstupní napětí od -10V do +10V obvod na obrázku chová jako kvadrátor, tj. zda platí vzorec mezi vstupním a výstupním napětím, uvedený v obrázku.



7 Analýza TRANSIENT neboli časová analýza

Cíle kapitoly:

- ✚ Ukázat možnosti časové analýzy ve smyslu „inteligentního osciloskopu“.
- ✚ Vysvětlit souvislosti mezi časovou analýzou a hledáním počátečního stejnosměrného řešení v obvodu.
- ✚ Seznámit se syntaxí příkazu .TRAN a se způsobem modelování počátečních podmínek akumulacních prvků.
- ✚ Vysvětlit podrobně mechanismy časové analýzy v PSPICE a z toho pramenící zásady pro správné používání příkazů .IC, .NODESET a SKIPBP.
- ✚ Rozebrat možnosti modelování zdrojů s požadovanými časovými průběhy napětí a proudů.

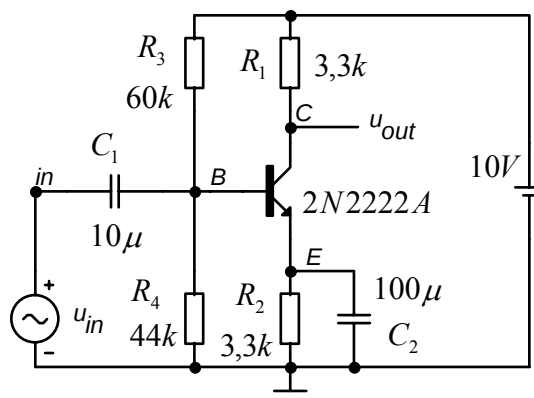
7.1 Cíle časové analýzy = analýzy signálů v obvodu

Základní cíle časové analýzy jsou dva:

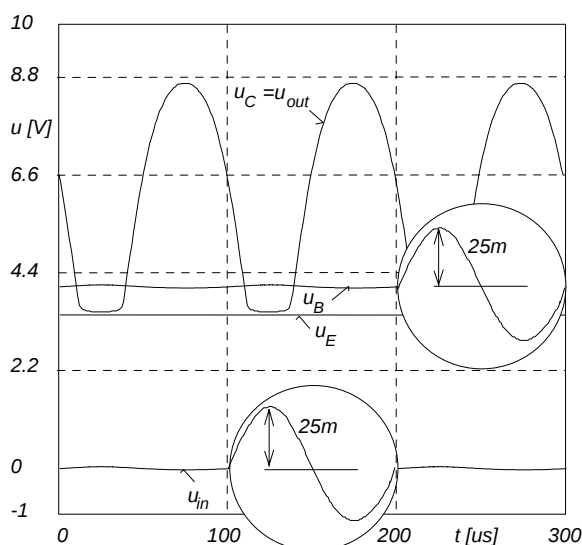
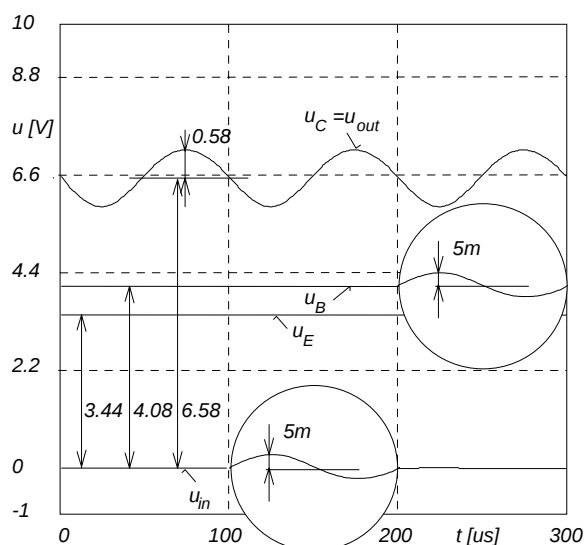
- napodobování činnosti „inteligentního osciloskopu“,
- různé způsoby dalšího zpracování analyzovaných signálů (zejména FFT).

„Inteligentní“ osciloskop zobrazuje časové průběhy napětí, proudu a všeho, co lze popsat vzorci, ve všech možných stavech obvodu (přechodové, ustálené, jednorázové děje..).

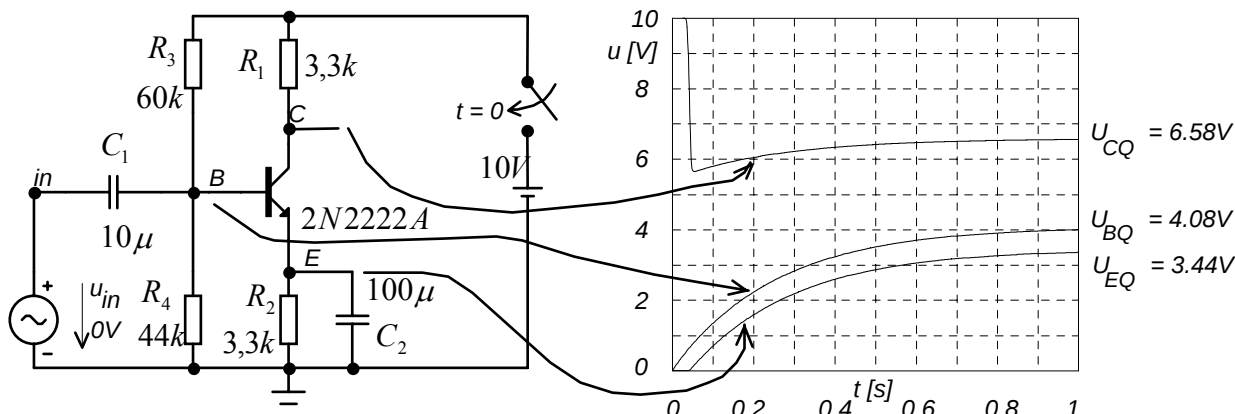
Uvažujme tranzistorový zesilovač na obrázku. Analýza TRANSIENT běžně umožňuje následující operace v ukázce. Některé z nich by bylo nesnadné, ne-li nemožné realizovat v laboratorních podmínkách.



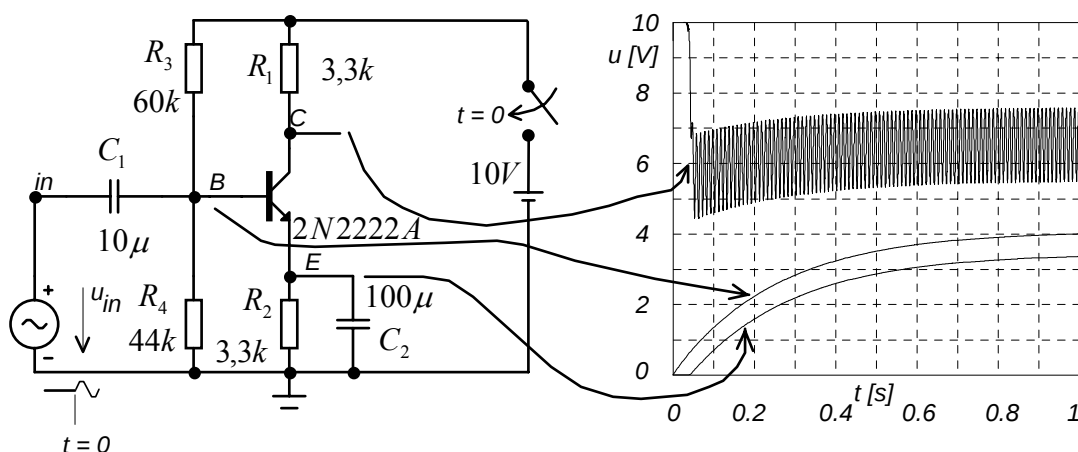
1. Vypočte se ss pracovní bod, **2.** Spustí se časová analýza při působení střídavého vstupního napětí.



1. Potlačí se výpočet ss pracovního bodu, 2. Spustí se časová analýza při neuvažování střídavého vstupního napětí.



1. Potlačí se výpočet ss pracovního bodu, 2. Spustí se časová analýza při působení střídavého vstupního napětí.



V prvním případě jsme simulovali zpracování střídavého signálu zesilovačem, který již má nastavený stejnosměrný pracovní bod. Rychlost sledovaných dějů závisí na rychlosti zesilovaného signálu. V našem případě je opakovací perioda $100\mu s$.

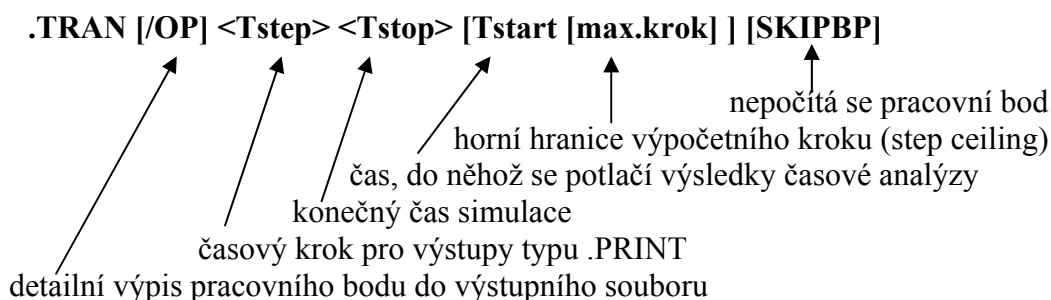
Druhý případ představuje přechodné děje, k nimž dochází po připojení napájecí baterie. Jde o náběh stejnosměrných poměrů do ustáleného stavu. V konkrétním příkladu toto ustalování může trvat řádově sekundy.

Třetí případ je kombinací předchozích dvou a představuje situaci, kdy současně se zapnutím napájecího zdroje přivedeme na vstup zesilovače rychlý signál. Z hlediska simulačního programu se jedná o náročnou úlohu, neboť pomalý děj se prolíná s rychlým.

PSPICE může simulovat interakci obvodu a signálu velmi komplexně. Je však na uživateli, nakolik má o možnostech analýzy přehled a jak jich dovede využít. Využívání potenciálů PSPICE je vázáno i na určité zásady při modelování součástek, zejména zdrojů a akumulacích prvků typu C a L.

7.2 Příkaz .TRAN

Zjednodušená syntaxe časové analýzy je následující:



Místo slova SKIPBP (Skip Bias Point) je možno uvést UIC (Use Initial Conditions).

Výpočetní krok je proměnný, mění se v průběhu analýzy podle toho, jak strmé jsou generované průběhy. Položka **max. krok** udává „horní zarážku“ pro tyto změny. Čím menší je toto číslo, tím detailněji budou křivky generovány, ale za cenu prodloužení výpočtů a větších nároků na paměť. Přednastavená hodnota je menší z čísel $(\text{Stop}-\text{Tstart})/50$ nebo $2 \cdot \text{Tstep}$ (v PSPICE je však vždy první z uvedených čísel). Pokud obvod neobsahuje akumulární prvky, nastaví se automaticky **max. krok** na **Tstep**.

Položka **Tstep** nemá vliv na data generovaná pro PROBE. Uplatní se pro výpisy dat např. příkazem .PRINT nebo při Fourierové transformaci (body se získají polynomiální interpolací z vypočtených bodů).

Pozn.: pracovní bod počítaný v analýze Transient se obecně liší od pracovního bodu počítaného při stejnosměrné analýze (příkazy .OP, .DC, nebo bez příkazu):

- Klasický pracovní bod se počítá z atributů DC zdrojů
- Pracovní bod se při analýze Transient počítá z počátečních hodnot zdrojů.

Na čem závisí signály generované obvodem:

- na počátečních podmínkách v obvodu (na počátečních napětích v uzlech a proudech induktory)
- na tom, zda je nastaven pracovní bod nebo ne
- na vstupních signálech
- na samotném obvodu

7.3 Zadávání počátečních podmínek pro časovou analýzu

Zadávání počátečních podmínek – IC (Initial Conditions) je možné třemi způsoby:

- v rámci definice prvků typu C a L
- příkazem .IC
- čtením ze souboru

Rozšířené zadávání prvků C a L:

C<jméno> <+ svorka> <- svorka> <kapacita> [IC=<počáteční hodnota napětí>]
L<jméno> <+ svorka> <- svorka> <indukčnost> [IC=<počáteční hodnota proudu>]

Počáteční napětí na C je orientováno od svorky + ke svorce -.

Počáteční proud induktorem teče od svorky + do svorky - vnitřkem induktoru.

Chybí-li specifikace IC, platí implicitní nulové hodnoty.

Příklady:

Celektrolyt 6 2 1uF IC=15V ;počáteční napětí na kapacitou je 15V od uzlu 6 do uzlu 2
 Ltlumivka 44 21 1mH IC=2.4A ;počáteční proud 2.4A teče induktorem od uzlu 44 do uzlu
 + 21
 Cx vstup baze 10uF ;kapacitor je na počátku časové analýzy vybitý

Zadávání počátečních podmínek příkazem .IC:

.IC <V(uzel1 [,uzel2])=<počáteční napětí>

.IC <I(inductor)= <počáteční proud>

Příkazy lze sdružovat do jednoho řádku.

Příklad:

.IC V(5)=4V V(12,3)=3V I(Lx)=-25mA

Zadávání počátečních podmínek ze souboru:

.LOADBIAS <jméno souboru>

Soubor lze získat na základě předchozí analýzy obvodu s použitím příkazu .SAVEBIAS.

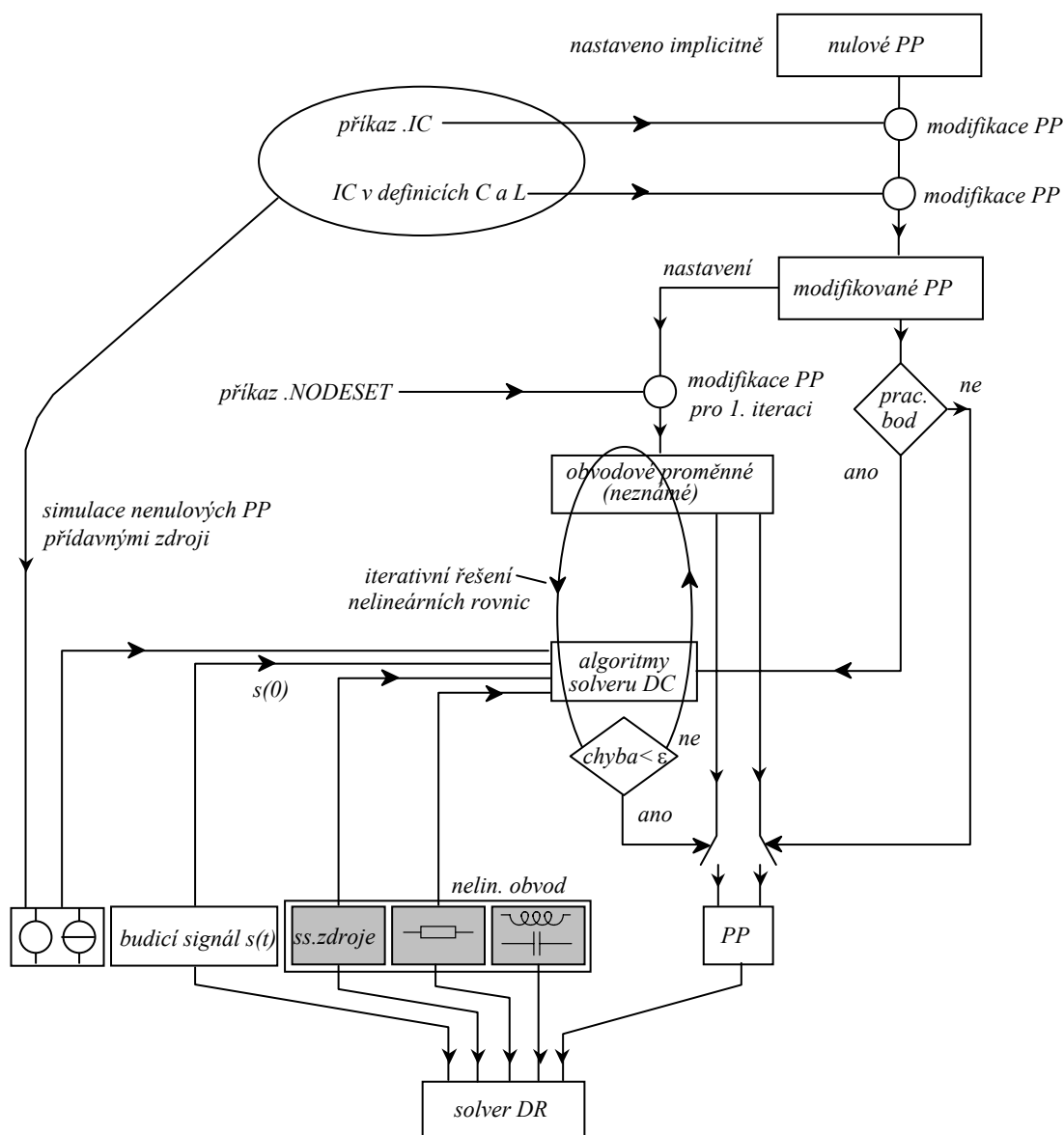
Soubor je textový a obsahuje příkazy typu .NODESET, tj. výčet hodnot všech uzlových napětí a proudů induktorů v obvodu (tzv. stavové proměnné). V případě nutnosti je lze editací souboru přepsat na příkazy typu .IC (viz dále).

Tvorba textového souboru: syntaxe příkazu .SAVEBIAS je bohatá, v příkladech uvedeny jsou nejzákladnější varianty:

.SAVEBIAS prac_bod.txt OP	; stavové proměnné odpovídající
+	ss pracovnímu bodu
.SAVEBIAS PP.txt TRAN	; stavové proměnné v prvním bodě
+	analýzy Transient
.SAVEBIAS xyz.tra TRAN TIME=10ms	; stavové proměnné v čase
+	bezprostředně následující za 10ms
.SAVEBIAS MKO.obv TRAN NOSBCKT TIME=10ms	; stavové proměnné v čase
+	bezprostředně následující za 10ms;
+	v souboru nebudou údaje o
+	vnitřních proměnných podobvodů

Takto lze ukládat napětí uzlů a proudy induktorů v obvodu pro analýzy OP, DC a TRAN. Podrobnosti viz manuál.

7.4 Mechanismus působení příkazů .IC a .NODESET v analýze Transient



Rozdíly v působení příkazů .IC a .NODESET:

.IC způsobí, že při případném výpočtu pracovního bodu budou nastavené počáteční hodnoty napětí a proudů „drženy“ až do konce výpočtu. Ostatní „volné“ veličiny se dopočtou. Tedy – časová analýza začne s počátečními hodnotami, uvedenými v příkazu .IC.

.NODESET je jen iterační násada pro výpočet pracovního bodu. Na konci výpočtu mohou být úplně jiné hodnoty.

Příkaz .IC má přednost před .NODESET, pokud jsou použity současně.

IC v definici C a L má přednost před příkazem .IC.

Z toho plyne praktická metoda hledání ustálených stavů v obvodu:

- uskuteční se analýza
- poslední stav v obvodu se uloží do souboru příkazem .SAVEBIAS

- v souboru se příkaz .NODESET nahradí za .IC
- analýza se opakuje s příkazem .LOADBIAS, musí se zakázat výpočet pracovního bodu, atd.

Pozn.: V MicroCapu se ustálené stavy hledají elegantně v režimu LEAVE.

7.5 Typické kombinace (ne)povolení výpočtu pracovního bodu a implicitních/vlastních počátečních podmínek

PP	výpočet pracovního bodu	
	povolit	zakázat
nulové	<i>I.</i> - Obvod je již připojen k napájecím zdrojům, řešíme jeho reakci na vstupní signály (nejčastější typ úlohy).	<i>IV.</i> - Rozběh oscilátorů a generátorů kmitů. - Analýza pasivních obvodů s nulovými počátečními stavy. - Hledání stejnosměrných pracovních bodů u obvodů, kde klasický výpočet selhává.
modifikované pomocí .IC, IC	<i>II.</i> - Řešení přechodných dějů se známými PP v obvodech, které obsahují např. IO s nastaveným ss pracovním bodem.	<i>V.</i> - Ustálené kmity autonomních obvodů. - Start analýzy z přesně definovaných stavů obvodu. - Řešení přechodných dějů v obvodech s nenulovými počátečními podmínkami. - Postupný přechod obvodu do ustáleného stavu sledem opakovaných analyzačních běhů.
modifikované pomocí .NODESET	<i>III.</i> - Hledání pracovních bodů, které leží blízko zadaných počátečních podmínek, a časová analýza vycházející z těchto bodů.	<i>VI.</i> - Nemá smysl.

7.6 Generování signálů v PSPICE

V PSPICE lze signály o požadovaných časových průbězích vytvářet dvěma základními způsoby:

- použitím zdrojů signálů, tj. nezávislých a závislých (řízených) zdrojů napětí a proudů
- použitím STIMULUS EDITORU

V tomto učebním textu se omezíme na první metodu. Práci se stimulus editorem je možné nastudovat z dokumentace programu.

7.6.1 Nezávislé zdroje napětí a proudu jako zdroje signálů

- mají shodnou syntaxi; zdroj napětí dává napětí orientované od svorky + do svorky -, zdroj proudu poskytuje proud, který teče vnitřkem zdroje od svorky + do svorky -.

V <jméno zdroje> <+ uzel> <- uzel> [specifikace vlastností]

I <jméno zdroje> <+ uzel> <- uzel> [specifikace vlastností]

Specifikace vlastností zdroje:

[[DC] <hodnota>][AC <amplituda> [<fáze>]] [stimulus=<jméno stimulu>][<def. čas. průběhu>]

↑
↑
↑
↑
specifikace časového průběhu jen pro analýzu Transient
podrobnosti viz manuál
amplituda a fáze signálu pro analýzu AC

V nebo I konstantního signálu nebo hodnota pro výpočet DC pracovního bodu

Definice časového průběhu:

PULSE (<seznam parametrů>)	impulzní průběhy
SIN (<seznam parametrů>)	harmonický signál (zobecněný)
EXP (<seznam parametrů>)	exponenciální průběhy
PWL (<seznam parametrů>)	po částech lineární signál (graf ve tvaru lomené čáry)
SFFM (<seznam parametrů>)	kmitočtově modulovaný signál

Podrobnosti viz manuál.

Proměnné Tstop (max. čas analýzy) a Tstep (časový krok) jsou definovány příkazem .TRAN.

Je třeba zadat alespoň první dva parametry. Chybějící se nastaví z přednastavených hodnot.

Závorky nejsou povinné.

PULSE(<v1> <v2> <td> <tr> <tf> <pw> <per>)

formát	atributy				časový průběh
PULSE	<u>název</u>	<u>popis</u>	<u>jednotka</u>	<u>přednast.</u>	
	v1 (i1)	poč. hodnota	V (A)	-	
	v2 (i2)	výška impulzu	V (A)	-	
	td	zpoždění	s	0	
	tr	doba náběhu	s	Tstep	
	tf	doba doběhu	s	Tstep	
	pw	šířka impulzu	s	Tstop	
	per	perioda	s	Tstop	

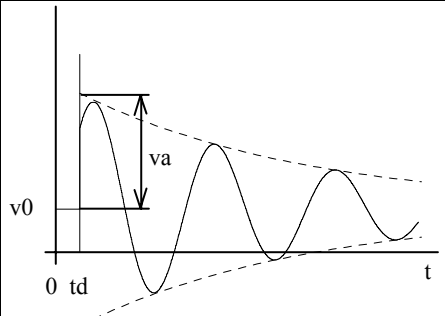
Příklady:

VTTL 1 0 DC 1 **PULSE(0 5 0 1n 1n 10u 20u)**; obdélníkový průběh se střídou prakticky 1:1

ISAW in out **PULSE(0 1 0 1m 0.1m 0 1.1m)** ; pilovitý průběh

SIN(<v0> <va> <freq> <td> <df> <phase>)

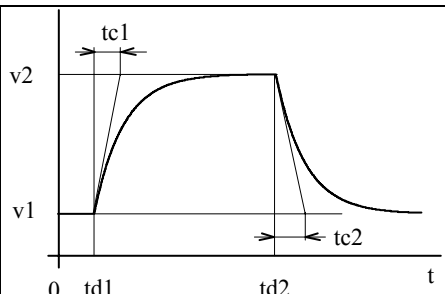
SIN	název	popis	jednotka	přednast.
	v0 (i0)	posunutí	V (A)	-
	va (ia)	amplituda	V (A)	-
	freq	kmitočet	Hz	1/Tstop
	td	zpoždění	s	0
	df	tlumicí faktor	1/s	0
	phase	poč. fáze	stupně	0
	df = 1/časová konstanta tlumení, může být i záporná nulová poč. fáze znamená sinusový signál.			


Příklady:

VSINUS 1 0 SIN(0 1) ; sinusoida o amplitudě 1V; v simulačním čase se vykreslí jedna
+ perioda
Ix 3 4 SIN 1 1 1k 0 1k 90 ; kosinusovka se ss složkou 1V, kmitočtem 1kHz a tlumicí
+ časovou konstantou 1ms

EXP(<v1> <v2> <td1> <tc1> <td2> <tc2>)

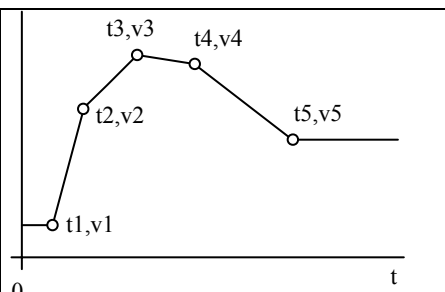
EXP	název	popis	jednotka	přednast.
	v1 (i1)	poč. hodnota	V (A)	-
	v2 (i2)	špičk. hodnota	V (A)	-
	td1	začátek náběhu	Hz	0
	tc1	τ náběhu	s	Tstep
	td2	konec náběhu	s	td1+Tstep
	tc2	τ doběhu	s	Tstep


Příklad:

Vrc out 0 EXP 0 1 0 1m 10m ; simulace nabíjení kapacitou přes rezistor na napětí 1V z 0V,
+ s časovou konstantou 1ms; v čase 10ms se kapacitor vybije
+ s časovou konstantou Tstep

PWL(t1,v1) (t2,v2)

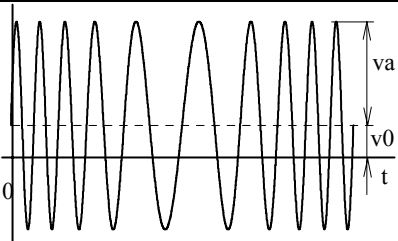
PWL	Časový průběh signálu je tvořen lomenou čarou, která spojuje body o zadaných souřadnicích. Signál „vlevo“ od prvního bodu a „vpravo“ od posledního bodu na časové ose je prodloužením vertikálních souřadnic krajních bodů.
-----	---


Příklad:

Vtroj out 0 PWL (0,0) (1m,1) (2m,0) ; trojúhelníkový impulz trvající 2ms.

Čárky ani závorky nejsou povinné. Další (značné) možnosti tohoto formátu viz manuál.

SFFM(<v0> <va> <freq> <mi> <fm>)

SFFM	<u>název</u>	<u>popis</u>	<u>jednotka</u>	<u>přednast.</u>	
	v0 (i0)	posunutí	V (A)	-	
	va (ia)	amplituda	V (A)	-	
	freq	kmitočet nosné	Hz	1/Tstop	
	mi	index FM= $\Delta F/fm$	-	0	
	<u>fm</u>	<u>modulační kmit.</u>	Hz	<u>1/Tstop</u>	
	ΔF je kmitočtový zdvih.				

Příklad:

Vsirena 1 0 SFFM 0 1 1k 200 1 ; napětí představující zvuk sirény o základním kmitočtu 1kHz, který je rozmítán „nahoru“ a „dolů“ o 200Hz periodicky jedenkrát za sekundu.

7.6.2 Závislé (řízené) zdroje jako zdroje signálů

Ke generování signálů jsou vhodné zejména řízené zdroje, jejichž napětí, resp. proud lze definovat vzorcem, v němž figuruje proměnná TIME (simulační čas).

Příklad:

EGPSK 1 0 value={ $\sin(2*3.14159*1k*time+V(2))$ } ; napětí z modulátoru PSK, kde počáteční fáze je modulována napětím uzlu 2

7.7 Shrnutí kapitoly 7

Pomocí časové analýzy, neboli analýzy TRANSIENT, můžeme zkoumat časové průběhy signálů na vstupech, výstupech a uvnitř obvodu. Tato analýza se aktivuje příkazem .TRAN.

Při této analýze bychom měli respektovat skutečnost, že sledované časové průběhy závisí nejen na budících signálech a na obvodu, ale i na tom, v jakém stavu se obvod nacházel na počátku simulace, například zda kondenzátory v obvodu byly vybity či nikoliv a zda induktory tekly počáteční proud. Proto má uživatel možnost definovat tzv. počáteční podmínky (Initial Conditions, IC) simulace. Pokud se tyto podmínky nedefinují, platí implicitní nastavení, totiž nulový výchozí stav. Existují celkem tři způsoby predefinování implicitního nastavení, a to příkazem .IC, klíčovým slovem IC v definici kapacitoru a induktoru v netlistu, a čtením počátečních podmínek ze souboru příkazem .LOADBIAS. Pro úplné pochopení příkazu .IC je vhodné popřemýšlet o jeho rozdílech oproti již dříve zmíněnému příkazu .NODESET.

Výsledky analýzy TRANSIENT lze dále modifikovat použitím, příp. nepoužitím klíčového slova SKIPBP (přeskočení, tj. neprovedení výpočtu stejnosměrného pracovního bodu před vlastní časovou analýzou) v rámci příkazu .TRAN. Typické příklady v praxi používaných kombinací povolení/nepovolení výpočtu pracovního bodu a různých počátečních podmínek jsou uvedeny v části 7.5.

7.8 Řešené příklady

Příklad 7.1: Přechodný děj v obvodu RLC

Zjistěte časový průběh napětí na R, L a C a proudu tekoucího obvodem od okamžiku připojení stejnosměrného zdroje k obvodu. Uvažujte nulové počáteční podmínky. Časový průběh sledujte do času 50us.

Řešení:

Příklad 7.1 - RLC transient

Vin 1 0 10V

R 1 2 800

L 2 3 2.53mH

C 3 0 1nF

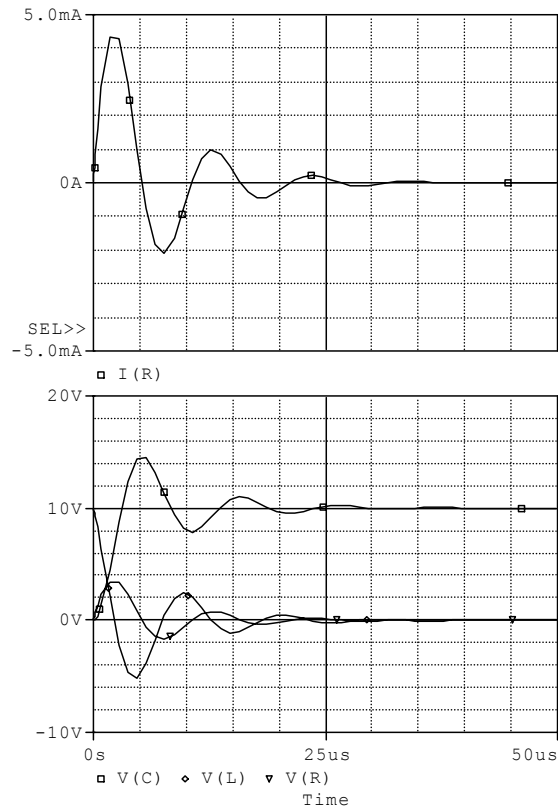
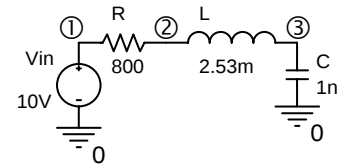
.TRAN 1u 50u SKIPBP

.PROBE V(R) V(L) V(C) I(R)

.end

SKIPBP znamená zákaz výpočtu pracovního bodu před vlastní analýzou Transient. Zkuste vyblokovat – SPICE najde přímo stejnosměrný ustálený stav. Zkuste zaměnit za ekvivalent – UIC (Use Initial Conditions).

Všimněte si nepřesně vykreslených časových průběhů v oblastech „špiček“ průběhů.



Příklad 7.2: přesnější vykreslení časového průběhu pomocí parametru „step ceiling“

Pokuste se lépe vykreslit detaily křivek z příkladu 7.1 pomocí parametru „step ceiling“.

Řešení:

Příklad 7.2 - RLC transient

Vin 1 0 10V

R 1 2 800

L 2 3 2.53mH

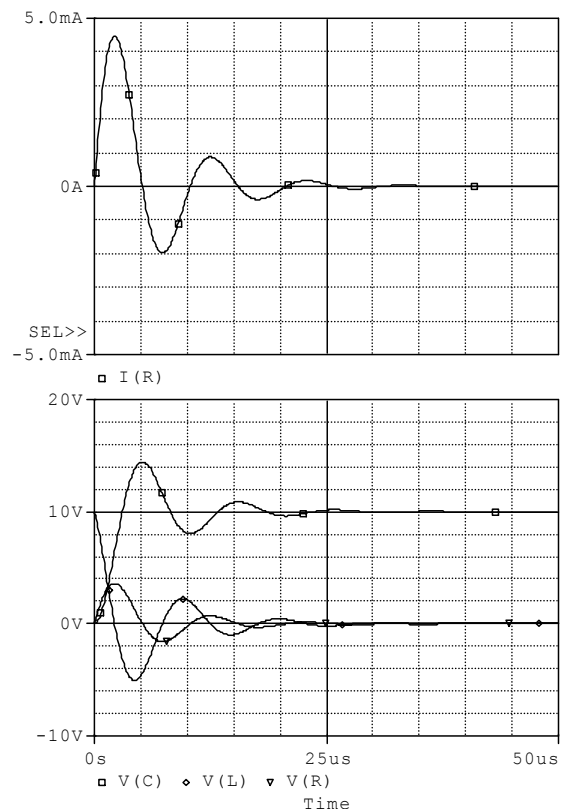
C 3 0 1nF

.TRAN 1u 50u 0 0.1u SKIPBP

.PROBE V(R) V(L) V(C) I(R)

.end

Dvě vložená čísla značí: čas, do něhož je potlačen výstup výsledků simulace, a „step ceiling“, t.j. maximální povolený krok časové analýzy. Bez uvedení těchto čísel platí implicitní hodnoty 0, Tstop/50, neboli 0, 1us.



Příklad 7.3: *Přechodný děj v obvodu bez vnějšího buzení*

Po ustálení přechodných dějů v obvodu z příkladu 7.2 bude napětí na kapacitě 10V a obvodem nepoteče proud. Určete přechodný děj, který vznikne, když vstupní napětí náhle klesne na nulu.

Pozn.: Místo zdroje s nulovým napětím je možno použít zkrat.

Řešení:

Příklad 7.3 - RLC transient

Vin 1 0 0V

R 1 2 800

L 2 3 2.53mH

C 3 0 1nF IC=10V

.TRAN 1u 50u 0 0.1u SKIPBP

.PROBE V(R) V(L) V(C) I(R)

.end

Příklad 7.4: *Použití příkazu .IC*

Příklad 7.3 vyřešte s využitím příkazu .IC.

Řešení:

Příklad 7.4 - RLC transient

Vin 1 0 0V

R 1 2 800

L 2 3 2.53mH

C 3 0 1nF

.IC V(3)=10V

.TRAN 1u 50u 0 0.1u SKIPBP

.PROBE V(R) V(L) V(C) I(R)

.end

Poznámky:

Ověřte si, že daný příkaz .IC V(3)=10V vede na stejné výsledky jako například příkazy

.IC V(3)=10V I(L)=0A

.IC V(3)=10V V(2)=0V

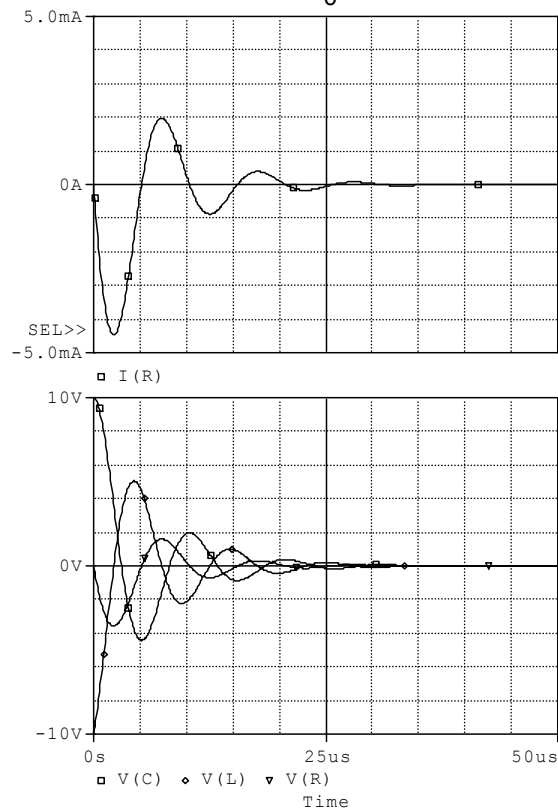
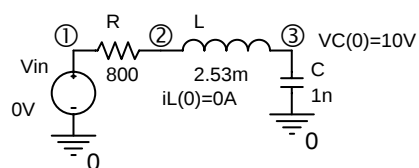
.IC V(3)=10V V(2)=10V

Pokuste se vysvětlit „nesrovnalost“ v posledních dvou příkazech.

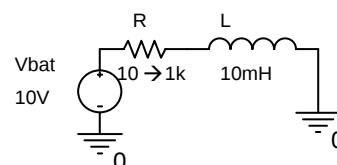
Pokuste se vysvětlit, proč po uvedení příkazu

.IC V(3)=10V I(L)=0A

můžeme odstranit slovo SKIPBP a výsledek analýzy se nezmění.

**Příklad 7.5:** *Přechodný děj v RL obvodu*

Při odporu 10Ω teče obvodem v ustáleném stavu proud 1A. Nakreslete přechodný děj, který nastane, když se náhle odpor zvětší na 1kΩ (simulujeme rozpojení spínače). Sledujte napětí na L a proud obvodem.



Řešení:

Příklad 7.5 - RL transient

Vin 1 0 0V

R 1 2 1k

L 2 0 10mH IC=1A

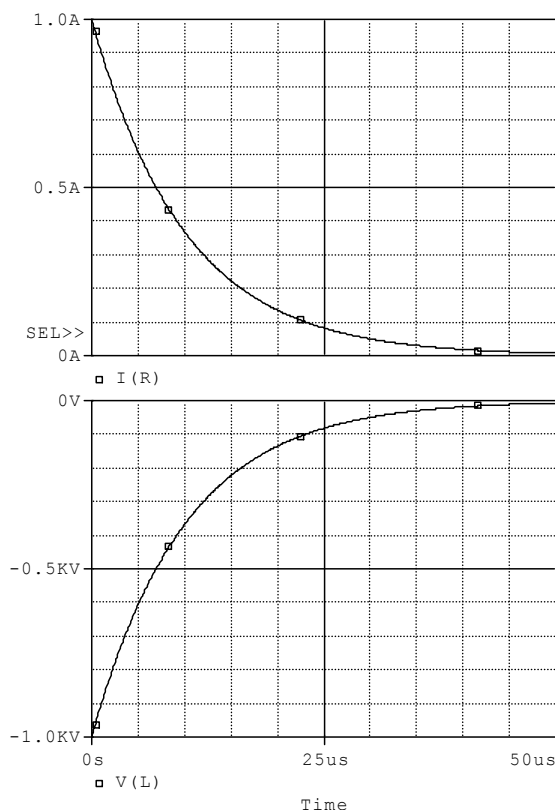
.TRAN 1u 50u SKIPBP

.PROBE V(R) V(L) I(R)

.end

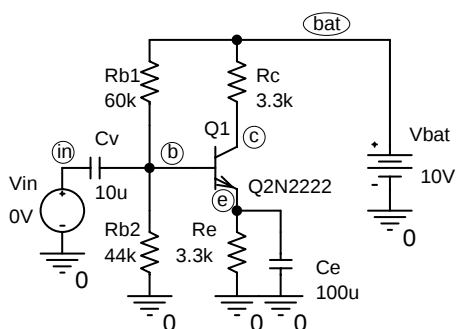
Proč je počáteční proud induktorem nastaven na 1A?

Na cívce se objeví záporná napěťová špička 1kV. Vysvětlete.



Příklad 7.6: Přechodný děj v tranzistorovém zesilovači

Zobrazte přechodný děj v tranzistorovém zesilovači po jeho připojení k napájecí baterii. Před připojením baterie je zesilovač v nulovém energetickém stavu. Vstupní signál je nulový.



Řešení:

Příklad 7.6 - zesilovav

Vin in 0 0V

Cv in b 10u

Rb1 bat b 60k

Rb2 b 0 44k

Rc bat c 3.3k

Re e 0 3.3k

Ce e 0 100u

Vbat bat 0 10V

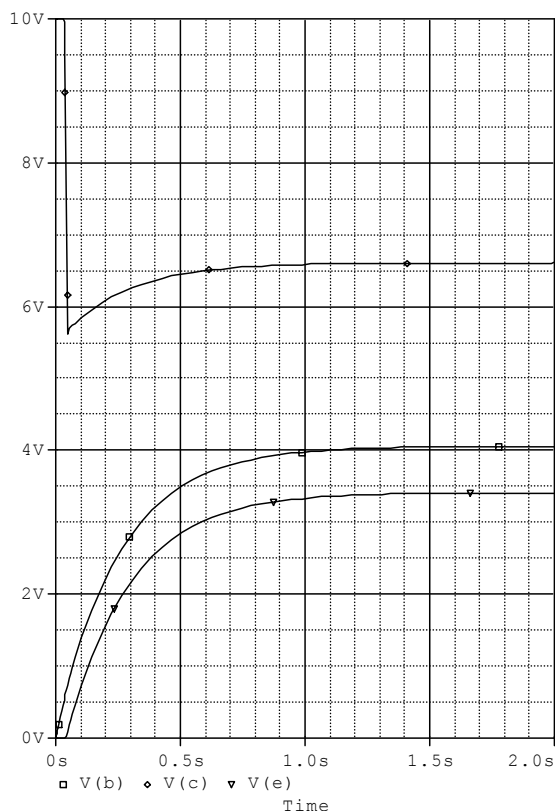
Q1 c b e Q2N2222

.TRAN 1u 2 SKIPBP

.PROBE

.lib

.end



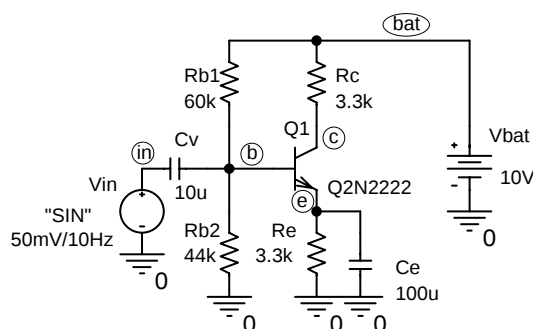
Závěr: Trvá skoro dvě sekundy, než je stejnosměrný pracovní bod „připraven“.

Ověřte si, že po blokování slova SKIPBP začne analýza TRANSIENT přímo z pracovního bodu, vypočteného iterační procedurou (je potlačen přechodný děj).

Příklad 7.7: Přechodný děj

v tranzistorovém zesilovači

Zobrazte přechodný děj v tranzistorovém zesilovači po jeho připojení k napájecí baterii. Před připojením baterie je zesilovač v nulovém energetickém stavu. Současně s baterií se připojí na vstup signál, který je sinusový o amplitudě 50mV a kmitočtu 10Hz.



Řešení:

Příklad 7.7 - zesilovác

Vin in 0 SIN 0 50mV 10Hz

Cv in b 10u

Rb1 bat b 60k

Rb2 b 0 44k

Rc bat c 3.3k

Re e 0 3.3k

Ce e 0 100u

Vbat bat 0 10V

Q1 c b e Q2N2222

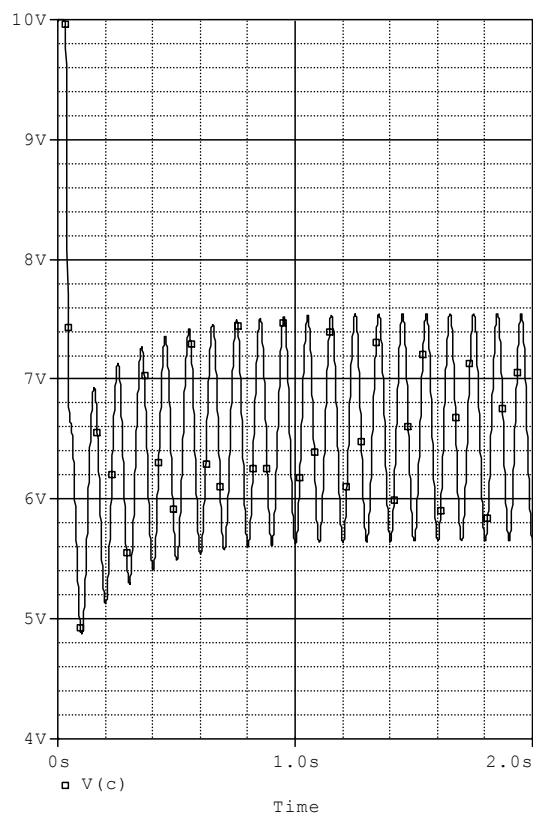
.TRAN 1u 2 0 1m SKIPBP; zjemnění kroku

.PROBE

.lib

.end

Z časového průběhu odečteme rozkmit napětí na kolektoru cca 1.898V špička-špička, což je amplituda 0.949V. To představuje střídavé zesílení na kmitočtu 10Hz cca $0.949V/50mV=18.98$.



Příklad 7.8: Potlačení pomalého přechodného děje v tranzistorovém zesilovači

Potlačte „pomalý“ přechodný děj náběhu zesilovače do stejnosměrného pracovního bodu povolením jeho výpočtu před analýzou Transient.

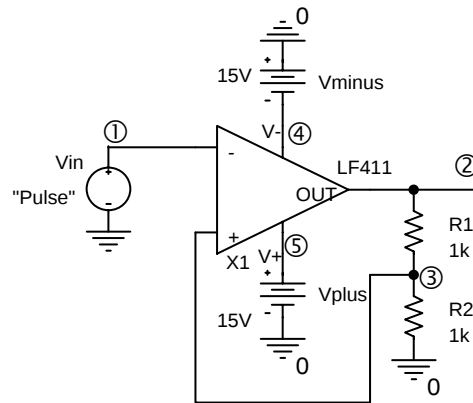
Řešení: Upravíme pouze příkaz .TRAN:

.TRAN 1u 2 0 1m

Příklad 7.9: Zobrazení hysterezní charakteristiky komparátoru

Na vstup komparátoru s hysterezí připojte zdroj symetrických trojúhelníkových kmitů, napětí od -10V do +10V, kmitočet 1Hz. Prozkoumejte průběh výstupního napětí komparátoru a zobrazte hysterezní charakteristiku komparátoru $U_2=f(U_1)$.

Řešení:



Příklad 7.9 - komparator

Vin 1 0 PULSE -10 10 0 0.5 0.5 1p 1

+

+

Xkomp 3 1 5 4 2 LF411

Vplus 5 0 15V

Vminus 0 4 15V

R1 2 3 1k

R2 3 0 1k

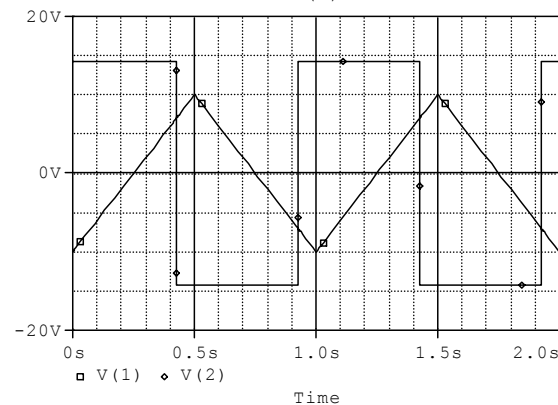
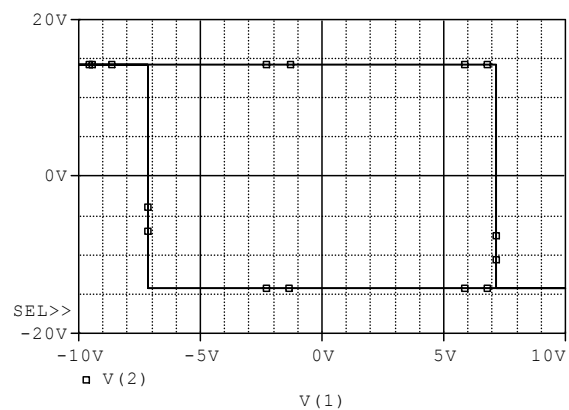
.TRAN 1m 2 ; vykreslí se 2 opakovací
+ periody, počítání pracovního
+ bodu povoleno

.PROBE

.lib

.end

; místo 1p má být teoreticky nula (šířka impulsu);
pak ale PSPICE místo nuly dosadí implicitní
hodnotu Tstop.



V PROBE zobrazíme V(1) a V(2). Pak založíme další obrázek, do něhož umístíme hysterezní charakteristiku. Protože na vodorovnou osu nyní vyneseme V(1), nikoliv čas, je třeba zadat

Plot/Unsynchronize X Axis

Náměty na další práci:

Vysvětlete, jak PSPICE dospěl k prvnímu bodu převodní charakteristiky. Jaký vliv na analýzu bude mít klíčové slovo SKIPBP v příkaze .TRAN?

Zkuste „zrychlit“ rozmítání vstupního signálu, např.

Vin 1 0 PULSE -10 10 0 5u 5u 1p 10u

.TRAN 1n 20u

Přesvědčte se o tom, že se nyní výrazně uplatní mezní rychlost přeběhu OZ a převodní charakteristika již bude silně deformovaná (již to nebude stejnosměrná charakteristika, kterou jsme chtěli získat).

Pokuste se realizovat rozmítání komparátoru zdrojem typu „PWL“.

Příklad 7.10: *Astabilní klopný obvod s komparátorem s hysterezi*

V příkladu 7.9 odstraňte zdroj vstupního signálu a komparátor doplňte zpětnovazebními součástkami R a C podle obr. Jde o generátor obdélníkových a pilovitých kmitů o kmitočtu

$$f \doteq 0.455 / (RC) \doteq 98.6 \text{ Hz.}$$

Prostudujte možnosti rozběhu generátoru do provozního režimu.

Řešení:

Příklad 7.10 - funkční generator

Xkomp 3 1 5 4 2 LF411

Vplus 5 0 15V

Vminus 0 4 15V

R1 2 3 1k

R2 3 0 1k

R 2 1 100k

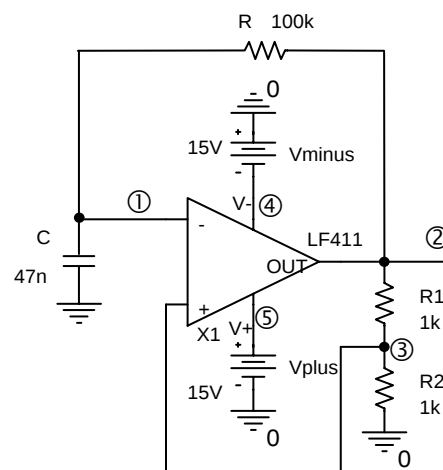
C 1 0 47nF

.TRAN 1n 60m 0 100u SKIPBP

.PROBE

.lib

.end



Změřte opakovací periodu kmitů a z ní odvoďte kmitočet. Možné správné výsledky: 10.336ms, 96.75Hz. ZV důsledku numerických chyb mírně závisí na pořadí měřené periody. Pokuste se periodu změřit i pomocí měřicích funkcí Period, příp. Period_XRange.

Náměty na samostatnou práci:

Ověřte si, jaké problémy má PSpice s nasazováním kmitů, pokud mu ponecháme přednastavenou hodnotu max. časového kroku (Step ceiling), tj.

.TRAN 1n 60m SKIPBP

Obvod nyní nabíhá do ustálených kmitů podstatně déle (je třeba zvětšit simulační čas). Simulátoru můžeme „pomoci“ například nastavením počátečního napětí na kapacitou na 100mV.

Pokuste se rozběhnout generátor při povolení výpočtu pracovního bodu:

.TRAN 1n 60m 0 100u

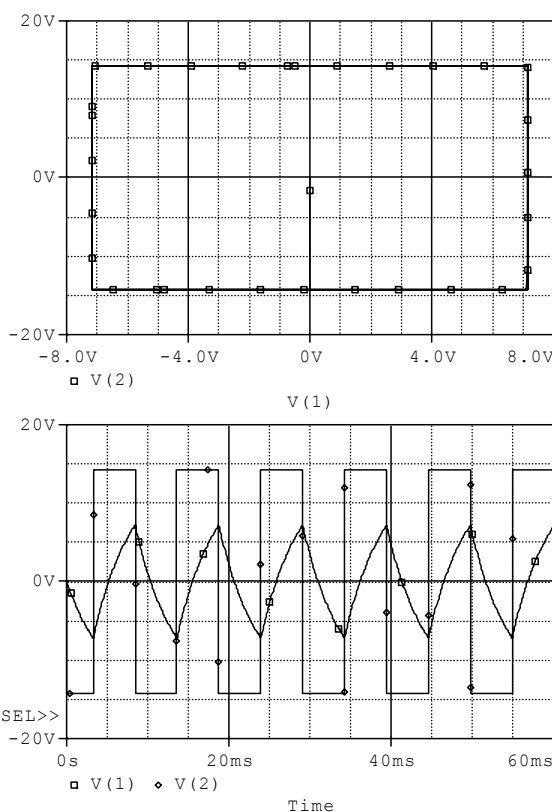
Postupně zkoušejte tyto nastavené počáteční podmínky a přemýšlejte o získaných výsledcích:

.IC V(2)=15V

.IC V(2)=15V V(1)=0V

.IC V(2)=-15V

.IC V(2)=-15V V(1)=0V



Příklad 7.11: *Tranzistorový oscilátor*

Prostudujte rozběh oscilátoru s tranzistorem MOSFET z nulových počátečních podmínek.
Pozn.: Prostudujte syntaxi zadávání tranzistoru MOSFET.

Řešení:

Příklad 7.11 - RC oscilator s MOSFETem

M1 D G 0 0 M2N6661

Vbat bat 0 10V

R1 D 1 10k

R2 1 2 100k

R3 2 G 1meg

C1 1 0 28.4nF

C2 2 0 2.84nF

C3 G 0 284pF

R4 bat D 100

.TRAN 1n 20m 0 20u SKIPBP

.PROBE

.lib

.end

Ověřte, že kmitočet generovaných kmitů je asi 971Hz.

Příklad 7.12: *Tranzistorový oscilátor, výpočet ustáleného stavu*

Pomocí příkazů `.SAVEBIAS` a `.LOADBIAS` dosáhněte přímo simulace ustálených kmitů oscilátoru bez mezivýpočtů přechodného děje.

Řešení:

1. Vytvoření souboru s výsledky simulace na konci analyzačního běhu analýzy Transient:

Příklad 7.12 - RC oscilator s MOSFETem

.....
.SAVEBIAS 712.ope TRAN TIME=20ms

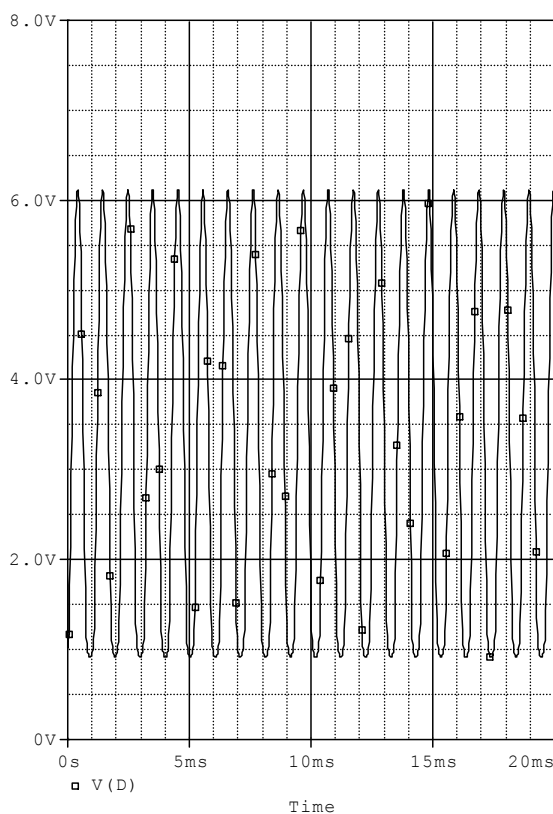
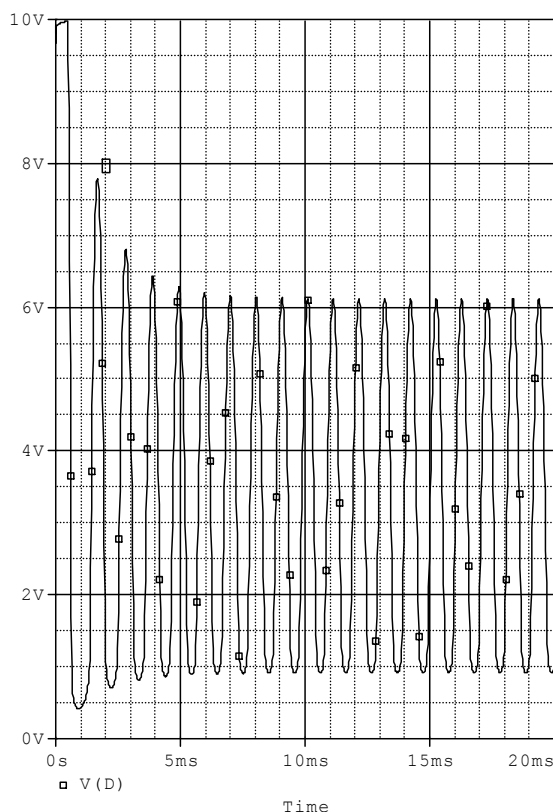
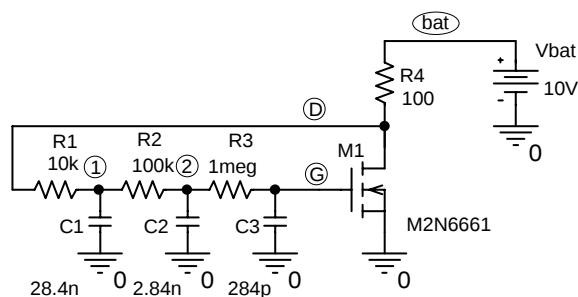
.TRAN 1n 20m 0 20u SKIPBP

.....

2. Editace vzniklého souboru záměnou `.NODESET` za `.IC`

3. Spuštění simulace s modifikovaným příkazem:

.LOADBIAS 712.ope



Poznámka: příkaz .IC v externím souboru je možno překopírovat do našeho vstupního souboru. Pak již k simulaci nebudeme externí soubor potřebovat (a tím ani příkaz .LOADBIAS):

Příklad 7.12 - RC oscilator s MOSFETem

M1 D G 0 0 M2N6661

Vbat bat 0 10V

R1 D 1 10k

R2 1 2 100k

R3 2 G 1meg

C1 1 0 28.4nF

C2 2 0 2.84nF

C3 G 0 284pF

R4 bat D 100

.IC

+ V(1) = 2.0676387445

+ V(2) = 3.1072366239

+ V(D) = 1.0196013703

+ V(G) = 3.4454667233

+ V(bat) = 10.0000000000

.TRAN 1n 20m 0 20u SKIPBP

.PROBE

.lib

.end

Příklad 7.13: Odezva rychlého operačního zesilovače na napěťový skok

Simulujte odezvu zesilovače na skokovou změnu vstupního napětí z 0V na 1V pro odpory $R1=R2=$ (1000, 400, 300, 200) ohmů, nejprve bez a pak s uvažováním parazitní kapacity 10pF.

Uvažte, že AD8001 je velmi rychlý OZ typu CFA s dobou ustálení jednotek až desítek ns a že dynamické vlastnosti zesilovače budou záviset nejen na poměru $R1/R2$, ale i na absolutních velikostech odporů. Čím menší odpory, tím větší náchylnost k nestabilitám.

Řešení:

Zesilovác s AD8001

Vin in 0 PWL (0,0) (1n,0)(1.1n 1)

;jednotkový skok z 0V na 1V v čase 1ns,
+ nástupná hrana 0.1ns

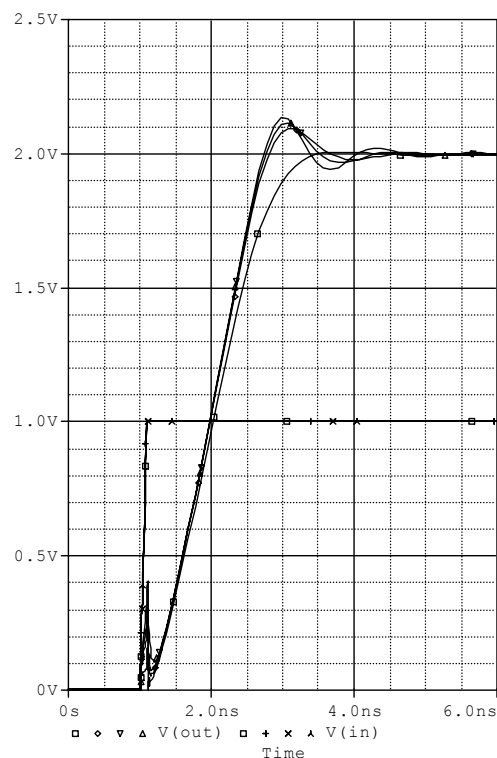
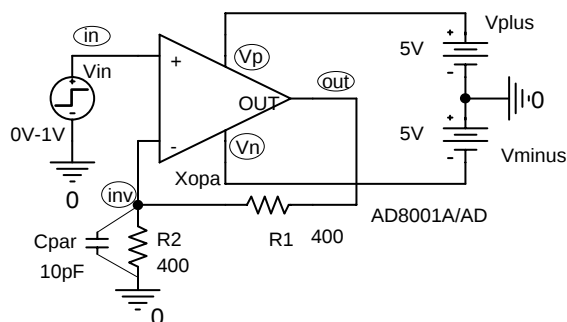
Vplus Vp 0 5V

Vminus Vn 0 -5V

R1 out inv {Rx}

R2 inv 0 {Rx}

*Cpar inv 0 10p



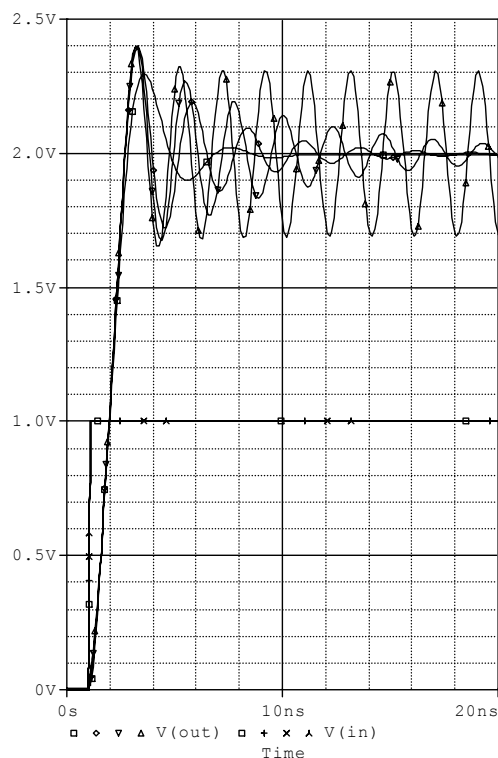
```

X in inv Vp Vn out AD8001a/AD
.param Rx 10
.step param Rx list 1k 400 300 200
.TRAN 1p 6n 0 100p ;časová analýza do 6ns
+                      s max. časovým krokem
+                      100ps
.probe
.lib
.end

```

Při $R_1=R_2=1k$ ohm se výstupní napětí ustálí bez přeskoku za cca 3.5ns. Při $R_1=R_2=200$ ohmů je zesilovač potenciálně nestabilní, protože např. při parazitní kapacitě 10pF dojde k netlumeným oscilacím.

Pozn.: Při simulaci vlivu parazitní kapacity zvolte delší simulační čas, např. 20ns.



7.9 Kontrolní otázky ke kapitole 7

1. Je-li zadán příkaz

TRAN 0 50m 4M,

pak maximální výpočetní krok (step ceiling) v průběhu analýzy TRANSIENT bude

a) 4ms, b) 12.5ms, c) 1ms

2. Příkaz

Cx out 3 1u IC=2

znamená, že napětí 2V je na počátku analýzy TRANSIENT mezi uzly

a) 3 a 1, b) out a 3, c) out a 0

3. K simulaci rozběhu oscilátoru z nulových počátečních podmínek použijeme příkaz

a) .TRAN 0 100u, b) .tran 1U 100U 50U, c) .tran 1u 100u SKIPBP

4. Proud induktorem L1, tekoucí z uzlu 1 do uzlu 2, bude v čase $t=0$ analýzy TRANSIENT:

a) 2mA, b) 5mA, c) nelze předpovědět

L1 1 2 1m IC=2m
.IC I(L1)=5m

5. Napětí na kapacitoru Cy bude v čase $t=0$ analýzy TRANSIENT:

a) 0V, b) 2V, c) 5V

Cy 1 2 1u IC=2
.IC V(Cy)=5V
.NODESET V(1,2)=0V

6. Napětí na kapacitoru Cy bude v čase $t=0$ analýzy TRANSIENT:

a) 5V, b) -5V, c) nelze předpovědět

Cy 1 2 1u
.NODESET V(1,2)=5V

7. Jednocestně usměrněný sinusový signál lze modelovat nezávislým zdrojem V s atributem
a) SIN, b) PUL, c) zdrojem V nelze modelovat

8. PSPICE vypočte hodnotu napětí mezi uzly 1 a 0 v analýze TRANSIENT

Ex 1 0 value={3*V(1)-5}

a) 2.5V, b) 3V, c) 5V

9. K simulaci vybíjení kapacitoru přes rezistor použijeme následující modelování kapacitoru:

a) Cx 1 2 1u IC=1V, b) C a b 1u .IC 1V, c) Cnabity c d 1u 1v(time=0)

10. Pracovní bod (Operating Point) se při analýze TRANSIENT počítá z:

a) atributů DC zdrojů signálů, b) počátečních hodnot zdrojů v čase 0, c) atributů AC zdrojů signálů.

11. Příkaz .IC ovlivňuje výpočet stejnosměrného pracovního bodu:

a) jen jako iterační násada, b) vůbec neovlivňuje, c) napětí a proudy definované příkazem .IC jsou „drženy“ po celou dobu výpočtu pracovního bodu.

12. Výpočetní krok je v analýze .TRAN

a) vždy konstantní, b) vždy proměnný, c) vždy proměnný s výjimkou, kdy se ve vstupním souboru objeví příkaz .FOUR; pak jsou časové průběhy vypočteny s fixním krokem

13. Příkaz .SAVEBIAS znamená uložení pracovního bodu do souboru s využitím příkazu

a) .IC, b) .NODESET, c) .INC

14. Signál o grafu ve tvaru obecné lomené čáry lze získat z nezávislého zdroje s atributem

a) PUL, b) SFFM, c) PWL

7.10 Neřešené příklady

Příklad 7.14: *Přechodný děj v RLC obvodu z příkladu 7.1*

Ve srovnání s výsledky z příkladu 7.1 se pokuste lépe vykreslit detaily křivek zpřesněním analýzy, konkrétně snížením chybového kritéria RELTOL (standardně nastaveno na 0.001).

Příklad 7.15: *Nabíjení kapacitoru z baterie*

Analyzujte přechodný děj, který vznikne po připojení vybitého kapacitoru o $C=100\mu\text{F}$ ke zdroji napětí o vnitřním odporu 10Ω a vnitřním napětí 10V. Změřte, za jak dlouho se kapacitor nabije na 63,2 procent vnitřního napětí zdroje. Srovnajte tento čas s časovou konstantou nabíjení.

Příklad 7.16: *Nabíjení kapacitoru ze zdroje proudu*

Analyzujte přechodný děj, který vznikne po připojení vybitého kapacitoru o $C=100\mu\text{F}$ k ideálnímu zdroji proudu 1mA . Proved'te analýzu TRANSIENT v délce trvání 1s .

Příklad 7.17: *Nabíjení kapacitoru z baterie s nenulovými počátečními podmínkami*

Analyzujte přechodný děj, který vznikne po připojení kapacitoru o $C=100\mu\text{F}$ ke zdroji napětí o vnitřním odporu 10Ω a vnitřním napětí 10V . Do jednoho obrázku umístěte nabíjecí křivky pro počáteční napětí kapacitoru 0V , 5V , 10V a 15V .

Příklad 7.18: *Modelování zdroje obdélníkových impulzů*

Namodelujte v PSPICE zdroj napětí o vnitřním odporu 10 ohmů . Vnitřní napětí se periodicky opakuje a je tvořeno impulzy o úrovních 0V a 10V , s šířkou impulzu 1ms a s šířkou mezery 1ms . V PROBE zobrazte 5 opakovacích period signálu.

Příklad 7.19: *Nabíjení kapacitoru ze zdroje impulzů*

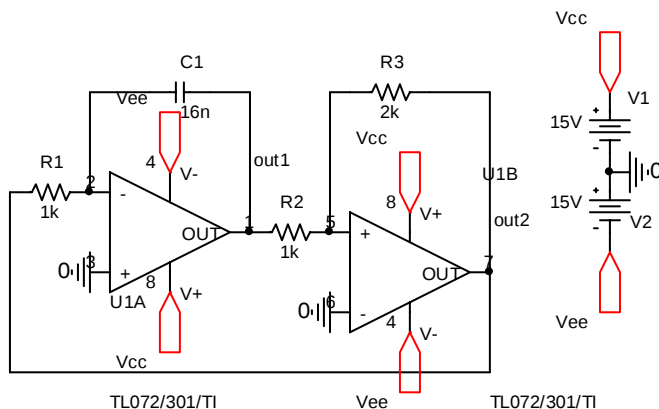
Zdrojem impulzů z příkladu 7.18 nabíjejte kapacitor o $C=100\mu\text{F}$ z nulového počátečního napětí. V PROBE zobrazte vnitřní napětí zdroje a napětí na kapacitoru. Prozkoumejte přechodný děj a odhadněte periodický ustálený stav, k němuž obvod spěje.

Příklad 7.20: *Nalezení periodického ustáleného stavu v RC obvodu z příkladu 7.19*

Pomocí příkazů `.LOADBIAS` a `.SAVEBIAS` simulujte přímo periodický ustálený stav v obvodu z příkladu 7.19.

Příklad 7.21: *Nasazování kmitů ve funkčním generátoru*

Napište vstupní soubor pro studium nasazování kmitů v generátoru se dvěma operačními zesilovači TL072. Zobrazte časové průběhy napětí na výstupech out1 a out2. Změřte kmitočet generovaných signálů.

**Příklad 7.22:** *Modelování signálu vzorcem*

Napište vstupní soubor pro zdroj napětí o časovém průběhu ideálně dvoucestně usměrněného sinusového signálu. Původní signál má amplitudu 10V a kmitočet 1kHz . K modelování použijte zdroj typu E. Usměrněný signál zobrazte v PROBE.

8 Analýza AC neboli kmitočtová analýza

Cíle kapitoly:

- ✚ Ukázat AC analýzu jako softwarový ekvivalent „inteligentního obvodového analyzátoru“.
- ✚ Vysvětlit, jak postupuje simulátor při analýze AC. Objasnit, že AC analýza jakožto analýza malosignálová je analýza linearizovaného modelu obvodu.
- ✚ Seznámit se syntaxí příkazu .AC a s významem atributů AC zdrojů signálů.
- ✚ Vysvětlit, jak jsou v PSPICE reprezentovány střídavá napětí a proudy, a jaké funkce v PSPICE a v PROBE jsou často používány při zpracování výsledků AC analýzy.
- ✚ Seznámit se způsoby modelování kmitočtových závislostí přenosů řízených zdrojů.
- ✚ Ukázat způsob použití příkazu .PRINT pro výsledky analýzy AC.

8.1 Cíle kmitočtové analýzy

Hlavním cílem analýzy AC je analýza kmitočtových závislostí obvodových veličin v harmonickém ustáleném stavu (HUS).

Jestliže při analýze TRANSIENT napodobuje PSPICE funkci „inteligentního osciloskopu“, pak při analýze AC je imitována činnost „inteligentního obvodového analyzátoru“, tj. přístroje pro snímání kmitočtových charakteristik obvodů pracujících v lineárním režimu.

Další příbuzné analýzy v „Spice-like“ simulátorech jsou tyto:

- šumová analýza (.NOISE),
- analýza nelineárního zkreslení (.DISTO, ne v PSpice)
- analýza nulových bodů a pólů (.PZ, jen SPICE 3),
- různé způsoby dalšího zpracování analyzovaných signálů (zejména IFFT).

8.2 Jak postupuje simulátor při kmitočtové analýze

V první fázi je vypočten stejnosměrný pracovní bod. Zdroje signálů jsou za tímto účelem nahrazeny jejich DC atributy. Nalezený pracovní bod můžeme dopředu ovlivnit příkazy .NODESET a .IC (viz též obrázek níže).

V druhé fázi je provedena linearizace obvodu v okolí nalezeného pracovního bodu.

V třetí fázi jsou budicí signály nahrazeny harmonickými signály o amplitudách a počátečních fázích, které se odvodí z AC atributů zdrojů signálů. Pak je provedena analýza linearizovaného obvodu v harmonickém ustáleném stavu pomocí fázorového počtu. Kmitočet signálů všech zdrojů je centrálně řízen příkazem .AC.

SPICE nemá možnost zakázat výpočet pracovního bodu. Tento se tedy počítá **vždy** před linearizací obvodu. Možnost zákazu má např. MicroCap.

IAC 3 5 DC 10mA AC 5mA 0 ;stejně jako u předchozího příkladu
 Vbat 1 2 5V ;harmonický signál o nulové amplitudě; DC atribut 5V se
 + uplatní při výpočtu stejnosměrného pracovního bodu
 Vsinus x y AC 10 SIN 0 1 5k ;harmonický signál o amplitudě 10V a poč. fázi 0; nulová
 + stejnosměrná složka vstupuje do výpočtu pracovního bodu;
 + časový průběh (sinusový o amplitudě 1V a kmitočtu 5kHz) se
 + uplatní jen při analýze Transient a nemá s analýzou AC nic
 + společného
 Kmitočet signálů z těchto zdrojů je stanoven „centrálně“ příkazem .AC.

V paměti je každý signál obvodu při analýze .AC reprezentován jako komplexní číslo. Jeho modul znamená amplitudu signálu a jeho argument počáteční fázi ve stupních.

8.5 Reprezentace proměnných při analýze AC v PSPICE

V nebo VM nebo I nebo IM	modul napětí V, resp. proudu I
VP nebo IP	počáteční fáze napětí V, resp. proudu I
VR nebo IR	reálná část napětí V, resp. proudu I
VI nebo II	imaginární část napětí V, resp. proudu I
VDB nebo IDB	modul napětí, resp. proudu v decibelech

Příklady:

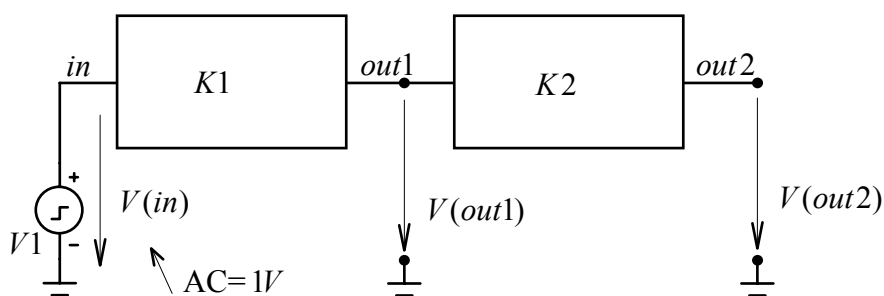
V(2)	modul napětí mezi uzlem 2 a referenčním uzlem
VM(x,z)	modul napětí mezi uzly x a z
VI(R12)	imaginární část napětí na rezistoru R12
VDB(D1)	modul napětí mezi anodou a katodou diody D1 v decibelech

8.6 Některé funkce v PSPICE a v PROBE pro analýzu AC

Funkce PSPICE a PROBE, aplikovatelné ve vzorcích na proměnné V a I při analýze AC:

M()	modul
P()	argument (počáteční fáze)
R()	reálná část
IMG()	imaginární část
DB()	decibely

Jestliže je v obvodu jediný zdroj signálu s nenulovým atributem „AC“, připojený například mezi uzel 1 a uzel referenční, pak zápisu $V(2)$ přísluší napětí, které se objeví mezi uzlem č. 2 a zemí, vyvolané působením zdroje signálu. Zápisem $V(2)/V(1)$ definujeme přenos napětí z uzlu 1 do uzlu 2. Pokud je amplituda signálového zdroje jednotková a počáteční fáze 0, pak zápis $V(2)$ je ekvivalentní zápisu $V(2)/V(1)$. Toho se často využívá: jestliže obvod budíme jediným „střídavým“ zdrojem, pak k zadání požadavku na analýzu kmitočtové charakteristiky stačí zapsat $V(Y)$, kde Y je jméno výstupního uzlu.



$$V(out1) = K1$$

$$V(out2) = K1.K2$$

$$V(out2)/V(out1) = K2$$

Jestliže ovšem v obvodu působí více signálových zdrojů, pak simulátor počítá napětí a další obvodové veličiny pomocí principu superpozice jako součet příspěvků jednotlivých zdrojů. V tomto případě tedy nelze analyzovat kmitočtové charakteristiky přenosů z jednoho uzlu do druhého.

8.7 Závislé (řízené) zdroje pro transformaci signálů v analýze AC

(vysvětleno na zdrojích typu E)

Bloky s kmitočtově závislými vlastnostmi, modelovanými vzorci:

E<název> <výstup+> <výstup-> LAPLACE {<výraz>}[=] {<výraz_s>}

musí obsahovat napětí nebo proud v obvodu

představuje vzorec obvodové funkce, obsahující Laplaceův operátor s

výstupní napětí= {<výraz>}*{<výraz_s>}

Příklad:

Vin 1 0 AC 1

.param f0 1k pi 3.14159

Elap 2 0 LAPLACE {V(1)} {2*pi*f0/(s+2*pi*f0)} ;modelování filtru 1. řádu typu DP, vstup 1-0, výstup 2-0

Bloky s kmitočtově závislými vlastnostmi, modelovanými tabulkami kmitočtových závislostí:

E<název> <výstup+> <výstup->

+ FREQ {<výraz>}[=] [KEYWORD] <<f>,<m>,<p>>* [DELAY=<zpoždění>]

trojice čísel, jejichž význam je dán klíčovým slovem KEYWORD

zpoždění bloku, které ovlivní výslednou fázovou kmitočtovou charakteristiku

výstupní napětí= {<výraz>}*{modelovaná kmitočtová závislost}

KEYWORD	f	m	p
není uvedeno	kmitočet	modul v db	fáze ve stupních
DB	kmitočet	modul v db	fáze ve stupních
MAG	kmitočet	modul	fáze ve stupních
DEG	kmitočet	modul v db	fáze ve stupních
RAD	kmitočet	modul v db	fáze v radiánech
R_I	kmitočet	reálná část	imaginární část

Příklad:

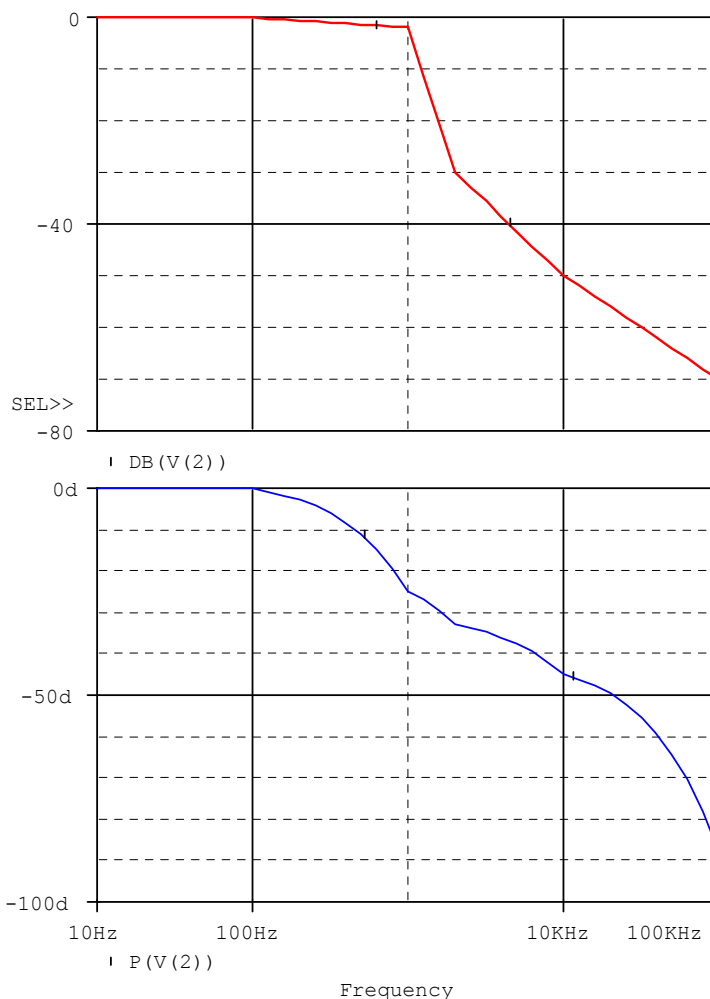
Vin 1 0 AC 1

Elap 2 0 FREQ {V(1)}= (10,0,0)
 +(100,0,0) (1k,-2,-25) (2k,-30,-33)
 +(10k,-50,-45) (100k,-70,-88)

.AC DEC 10 10 100k

.probe

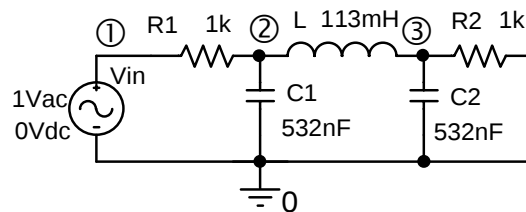
.end



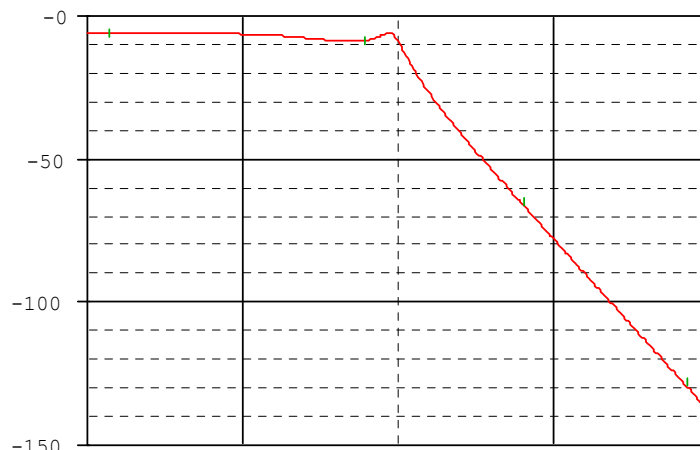
8.8 Použití příkazu .PRINT v analýze AC

.AC
 .PRINT AC <veličina1,veličina2..>

Příklad:



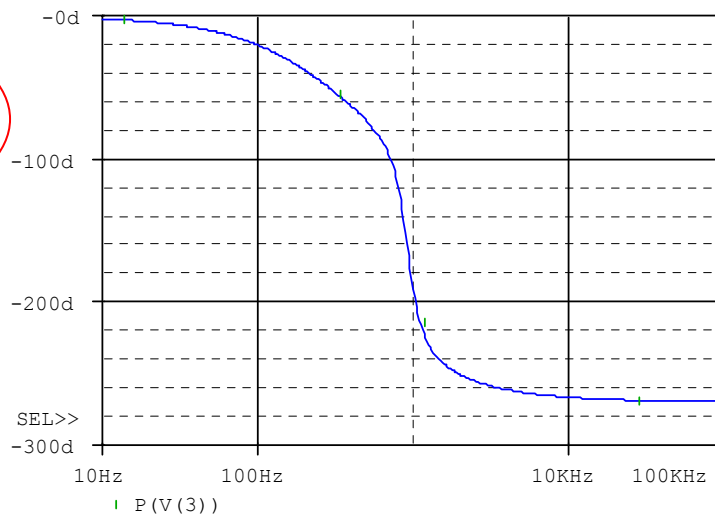
```
*
Vin 1 0 AC 1
R1 1 2 1k
C1 2 0 532n
L 2 3 113m
C2 3 0 532n
R2 3 0 1k
.AC LIN 1 1k 2k
.print AC V(3) VP(3)
.print AC VR(3) VI(3)
.end
```



FREQ	V(3)	VP(3)
1.000E+03	3.573E-01	1.689E+02

FREQ	VR(3)	VI(3)
1.000E+03	-3.506E-01	6.865E-02

```
.....
.AC DEC 100 10 100k
.probe
.end
```



Frequency

8.9 Shrnutí kapitoly 8

Analýza AC znamená analýzu linearizovaného obvodu v harmonickém ustáleném stavu. Uživatel musí vědět, že PSPICE nejprve nalezne stejnosměrný pracovní bod obvodu, poté provede linearizaci obvodu v tomto pracovním bodu, a nakonec nahradí nezávislé zdroje v obvodu jejich AC atributy. Vznikne tak model pro řešení analýzou AC. Příkazem `.AC` zadáme způsob rozmitání kmitočtu, tj. odkud, kam a po jakých krocích. Tento kmitočet je v průběhu analýzy přiřazen současně všem zdrojům signálu. Tím je zaručeno, že pro konkrétní kmitočet je vždy řešen obvod v harmonickém ustáleném stavu. Řešení probíhá pomocí klasického názorového počtu, takže každé napětí a proud je reprezentováno komplexním číslem, v jehož modulu a argumentu je informace o amplitudě a počáteční fázi na daném kmitočtu. Je vhodné naučit se jednoduchou syntaxi, platnou v PSPICE, pro zápis reálné a imaginární složky, amplitudy a počáteční fáze napětí a proudu. Pak je možné tyto veličiny sledovat např. pomocí příkazu `.PRINT` nebo v `PROBE`.

Důležitou pomůckou pro behaviorální modelování přenosových vlastností obvodů mohou být řízené zdroje se syntaxí typu `LAPLACE` nebo `FREQ`.

8.10 Řešené příklady

Příklad 8.1 Střídavá analýza Wienova článku

Zjistěte střídavá napětí a proudy u všech součástek Wienova článku, je-li kmitočet střídavého vstupního napětí 1kHz, amplituda 1V a počáteční fáze 0. Zjistěte amplitudy a počáteční fáze těchto veličin a jejich reálné a imaginární složky. Nakreslete fázorové diagramy.

Řešení:

Wienův clanek na 1kHz

Vin 1 0 Ac 1

R1 1 2 1k

C1 2 3 159nF

R2 3 0 1k

C2 3 0 159nF

.AC LIN 1 1k 1k ;AC analýza proběhne na jediném kmitočtu 1kHz

`.print AC V(R1) VP(R1) V(C1) VP(C1) V(R2) VP(R2)` ; výstup modulu a fáze jednotlivých napětí

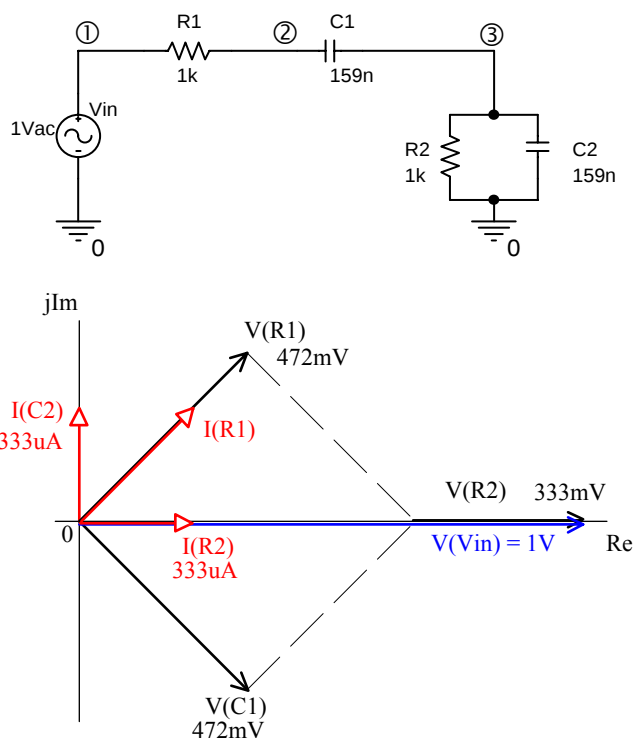
`.print AC VR(R1) VI(R1) VR(C1) VI(C1) VR(R2) VI(R2)` ; výstup reálných a imag. složek napětí

`.print AC I(R1) IP(R1) I(R2) IP(R2) I(C2) IP(C2)` ; výstup modulu a fáze jednotlivých proudů

`.print AC IR(R1) II(R1) IR(R2) II(R2) IR(C2) II(C2)` ; výstup reálných a imag. složek proudů

*

`.end`



Výsledky z výstupního souboru (zkráceno):

FREQ	V(R1)	VP(R1)	V(C1)	VP(C1)	V(R2)	VP(R2)
1.000E+03	4.712E-01	4.501E+01	4.716E-01	-4.499E+01	3.333E-01	3.720E-02

FREQ	VR(R1)	VI(R1)	VR(C1)	VI(C1)	VR(R2)	VI(R2)
1.000E+03	3.331E-01	3.332E-01	3.336E-01	-3.334E-01	3.333E-01	2.164E-04

FREQ	I(R1)	IP(R1)	I(R2)	IP(R2)	I(C2)	IP(C2)
1.000E+03	4.712E-04	4.501E+01	3.333E-04	3.720E-02	3.330E-04	9.004E+01

FREQ	IR(R1)	II(R1)	IR(R2)	II(R2)	IR(C2)	II(C2)
1.000E+03	3.331E-04	3.332E-04	3.333E-04	2.164E-07	-2.162E-07	3.330E-04

Příklad 8.2 Použití příkazu *.PRINT*, Wienův článek z příkladu 8.1

Do výstupního souboru vytiskněte tabulku závislosti amplitudy a počáteční fáze výstupního napětí V(3) na kmitočtu v kmitočtových bodech 100Hz, 1kHz a 10kHz.

Výsledky:

FREQ	V(3)	VP(3)
1.000E+02	9.658E-02	7.316E+01
1.000E+03	3.333E-01	3.720E-02
1.000E+04	9.676E-02	-7.313E+01

Závěr: Zdá se, že v okolí kmitočtu 1kHz má Wienův článek největší přenos $1/3$ a nulový fázový posuv mezi vstupním a výstupním napětím. Tento kmitočet je dán vzorcem $1/(2\pi RC)$.

Řešení:

```
Wienuv clanek na 1kHz
Vin 1 0 Ac 1
R1 1 2 1k
C1 2 3 159nF
R2 3 0 1k
C2 3 0 159nF
.AC DEC 1 100 10k
.print AC V(3) VP(3)
.end
```

Příklad 8.3 Zobrazení kmitočtové charakteristiky Wienova článku z příkladu 8.1

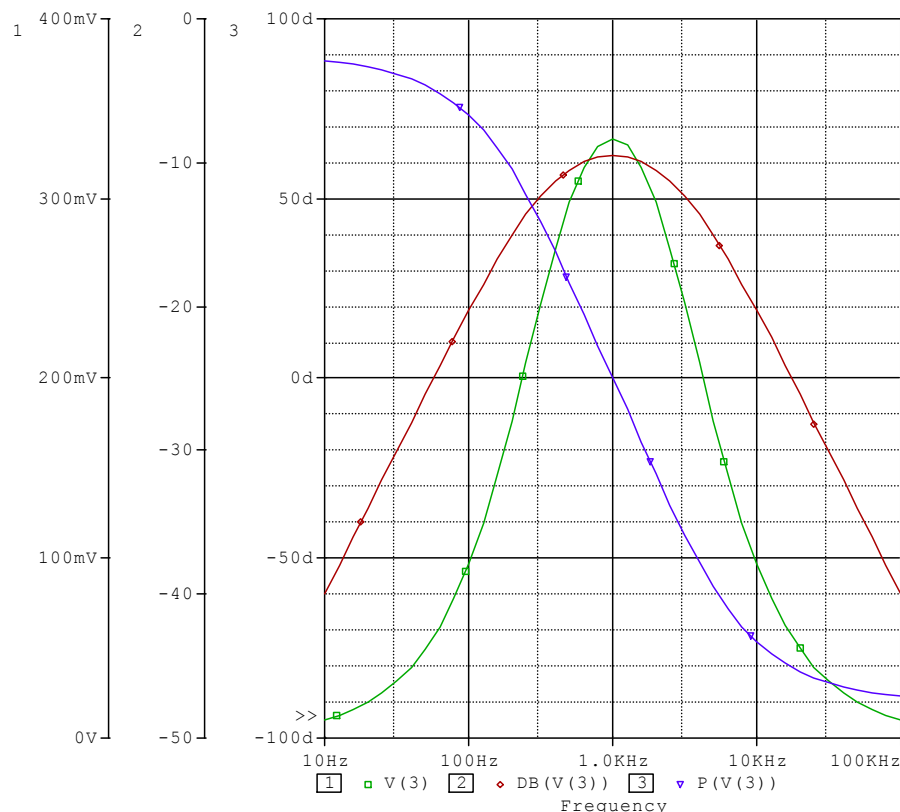
Zobrazte amplitudovou a fázovou kmitočtovou charakteristiku Wienova článku ve frekvenčním rozsahu od 10Hz do 100kHz (logaritmická frekvenční osa, logaritmické rozmítání kmitočtu).

Řešení:

Wienuv clanek, kmitoctove charakteristiky

```
Vin 1 0 Ac 1
R1 1 2 1k
C1 2 3 159nF
R2 3 0 1k
C2 3 0 159nF
.AC DEC 10 10 100k
.probe V(3)
.end
```

Pozn.: v PROBE je amplitudová charakteristika zobrazena jednak jako klasický přenos napětí a jednak v decibelech. Každá křivka má svou vlastní osu Y (Plot/Add Y Axis).

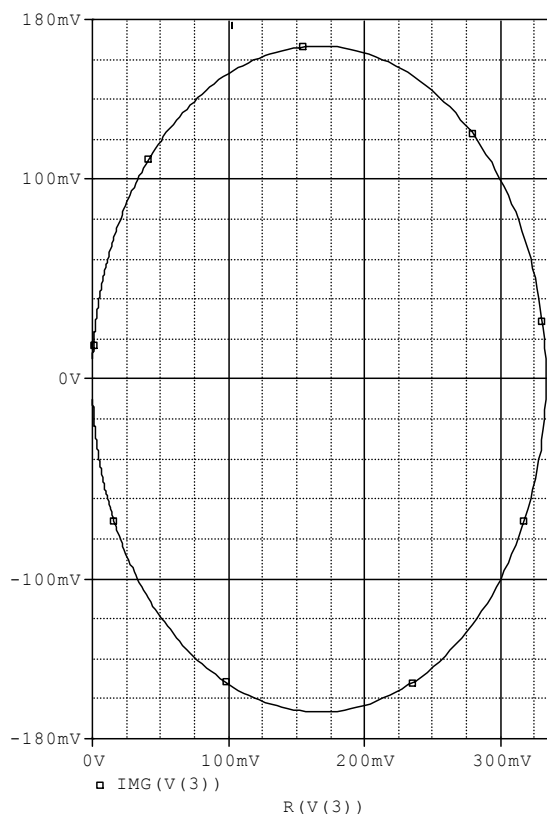


Příklad 8.4 Komplexní kmitočtová charakteristika Wienova článku z příkladu 8.1

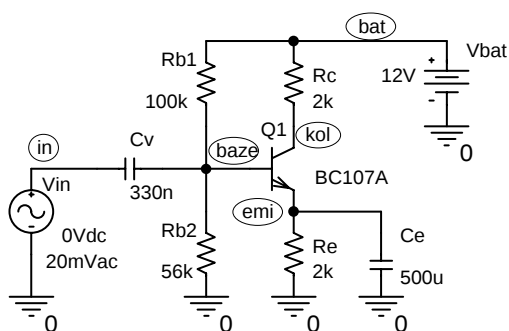
Zobrazte komplexní kmitočtovou charakteristiku Wienova článku z příkladu 8.1.

Řešení: V PROBE je třeba nastavit na vodorovnou osu zobrazování reálné části přenosu a na svislou osu zobrazování imaginární části přenosu.

Pro vykreslení hladké křivky je vhodné zvýšit počet kroků na dekádu na cca 100.



Příklad 8.5 AC analýza tranzistorového zesilovače



Zjistěte stejnosměrná a střídavá napětí ve všech uzlech zesilovače, je-li kmitočet vstupního signálu 10kHz.

Řešení:

```
Tran. zesilovac
Vin in 0 AC 20mV
Cv in baze 330nF
Rb1 bat baze 100k
Rb2 baze 0 56k
Rc bat kol 2k
Re emi 0 2k
Ce emi 0 500uF
Vbat bat 0 12V
Q kol baze emi BC107A
.AC LIN 1 10k 10k
.print AC V([in]) V([baze]) V([kol]) V([emi]) V([bat])
.lib
.end
```

Výsledky:

Ss pracovní bod:

```
NODE VOLTAGE  NODE VOLTAGE  NODE VOLTAGE  NODE VOLTAGE  NODE VOLTAGE
( in) 0.0000   ( bat) 12.0000   ( emi) 3.2979   ( kol) 8.7207   ( baze) 3.9738
```

Střídavý signál – amplitudy:

FREQ	V(in)	V(baze)	V(kol)	V(emi)	V(bat)
1.000E+04	2.000E-02	1.997E-02	2.464E+00	3.942E-05	1.000E-30

Střídavý signál na kolektoru nebude ořezán, protože ss napětí na kolektoru 8.7207V je „vzdáleno“ od napětí baterie asi 3.28V a amplituda na kolektoru je 2.464V.

Střídavé zesílení je asi $2.464/0.02=123.2$.

Přesvědčte se, že fázový posuv napětí na kolektoru oproti vstupnímu napětí je asi -179 stupňů, tj. prakticky 180 stupňů (invertováno).

Návod: .print VP(v[kol]).

Příklad 8.6 *Kmitočtové charakteristiky zesilovače z příkladu 8.5*

Analyzujte amplitudovou a fázovou kmitočtovou charakteristiku zesilovače z příkladu 8.5 v kmitočtovém rozsahu 100Hz až 100MHz.

Řešení:

```
Tran. zesilovac
Vin in 0 AC 1V
Cv in baze 330nF
Rb1 bat baze 100k
Rb2 baze 0 56k
Rc bat kol 2k
Re emi 0 2k
```

```

Ce emi 0 500uF
Vbat bat 0 12V
Q kol baze emi BC107A
.AC DEC 10 100 100meg
.probe V([kol])
.lib
.end

```

Ověřte si, že na kmitočtu 10kHz je zesílení asi 41.813 dB, tj. zesílení 123.2, a fázový posuv -178.956 stupňů. Porovnejte s výsledky z příkladu 8.5.

Příklad 8.7 *Kmitočtové charakteristiky zesilovače z příkladu 8.5*

Zobrazte amplitudové kmitočtové charakteristiky zesilovače s výstupem na kolektoru a na emitoru pro blokovací emitorovou kapacitu 1pF, 1uF a 500uF.

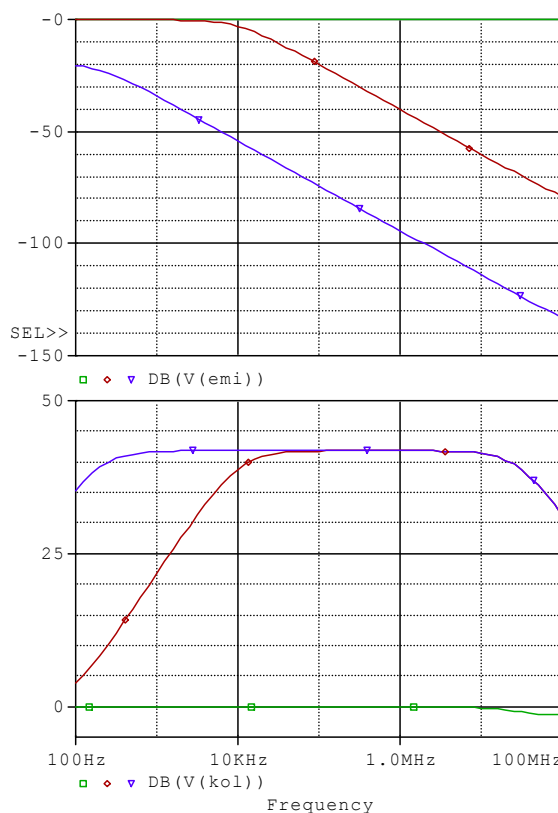
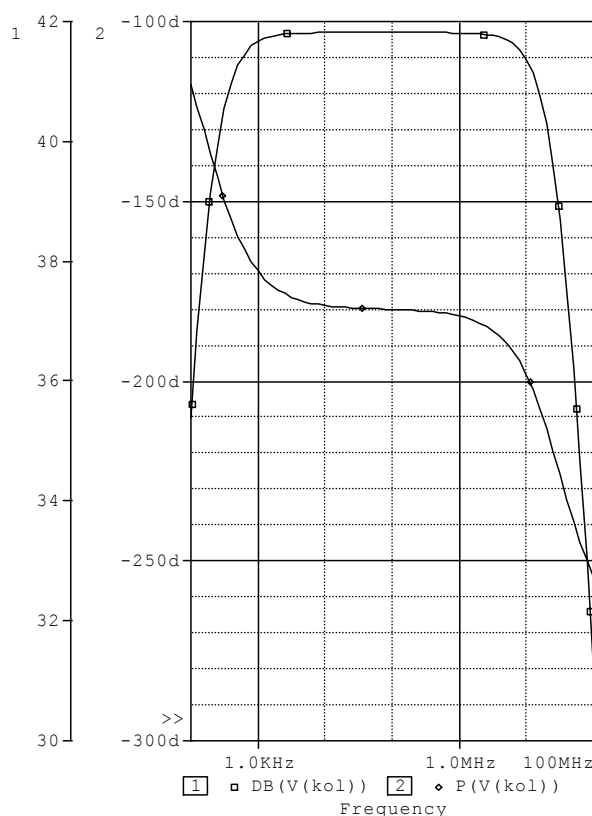
Řešení:

```

Tran. zesilovac
Vin in 0 AC 1V
Cv in baze 330nF
Rb1 bat baze 100k
Rb2 baze 0 56k
Rc bat kol 2k
Re emi 0 2k
Ce emi 0 {Cx}
Vbat bat 0 12V
Q kol baze emi BC107A
.param Cx 500u
.step param Cx list 1p 1u 500u
.AC DEC 10 100 100meg
.probe V([kol]) V([emi])
.lib
.end

```

Při zanedbatelné kapacitě C_e se zesilovač chová z hlediska emitoru jako sledovač (zesílení +1) a z hlediska kolektoru jako invertor (zesílení -1), a to až do desítek MHz. Při růstu C_e roste zesílení na kolektoru, neboť je blokována záporná zpětná vazba přes R_e .



Příklad 8.8 *Kmitočtové charakteristiky operačního zesilovače typu 741*

Zobrazte amplitudovou a fázovou kmitočtovou charakteristiku operačního zesilovače typu 741.

Řešení:

Operacni zesilovac 741

Vin in 0 AC 1V

Vplus Vp 0 15V

Vminus Vn 0 -15V

X in 0 Vp Vn out uA741

.AC DEC 10 0.1 100meg

.probe

.lib

.end

Ověřte tyto číselné údaje:

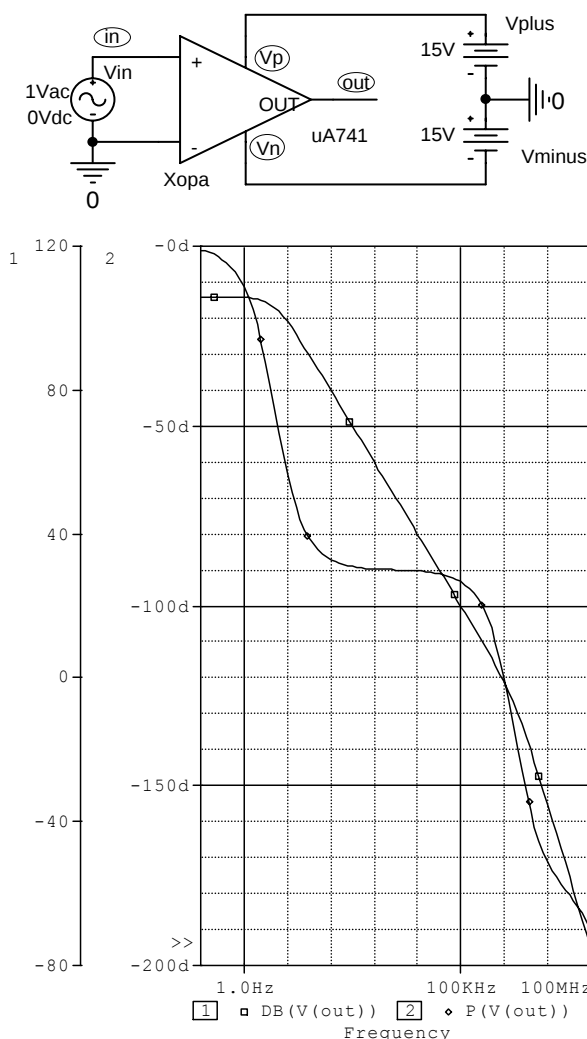
Zesílení na nízkých kmitočtech je asi 106dB, tj. 200000.

Kmitočet třidecibellového poklesu zesílení je asi 5Hz. Fázový posuv výstupního napětí oproti vstupnímu je na tomto kmitočtu asi 45 stupňů.

Tranzitní kmitočet, tj. kmitočet, při kterém poklesne zesílení na 0 dB, je asi 900kHz.

Mezi kmitočtem třidecibellového poklesu a kmitočtem tranzitním klesá zesílení se strmostí 20 db/dekádu neboli 6 db/oktávu.

Za tranzitním kmitočtem se nachází druhý lomový kmitočet, který způsobuje dvojnásobně rychlý pokles zesílení 40 db na dekádu neboli 12 db na oktávu.



Poznámka: strmost poklesu zesílení lze měřit v PROBE například měřicí funkcí Swing_XRange.

Příklady:

Swing_XRange(db(V(out)),10meg,100meg)

(měření strmosti 40 db/dekádu za kmitočtem 2. lomu)

Swing_XRange(db(V(out)),10meg,20meg)

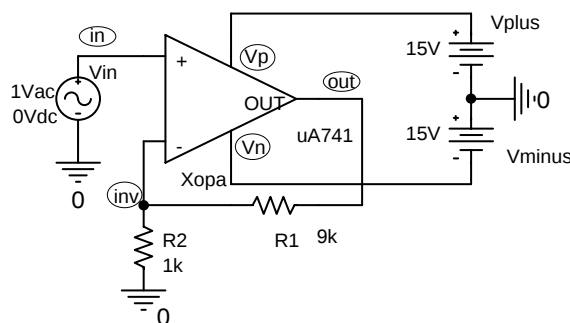
(měření strmosti 12 db/oktávu za kmitočtem 2. lomu)

Swing_XRange(db(V(out)),1k,10k)

(měření strmosti 20 db/dekádu před kmitočtem 2. lomu).

Příklad 8.9 *Kmitočtové charakteristiky neinvertujícího zesilovače s OZ*

Nakreslete amplitudové kmitočtové charakteristiky zesilovače pro $R_2=1k$, $R_1=1m$, $1k$, $10k$, $100k\Omega$.



Řešení:

```
Operacni zesilovac 741
Vin in 0 AC 1V
Vplus Vp 0 15V
Vminus Vn 0 -15V
R1 out inv {Rx}
R2 inv 0 1k
X in inv Vp Vn out ua741
.param Rx 10k
.step param Rx list 1m 1k 10k 100k
.AC DEC 10 100 100meg
.probe
.lib
.end
```

Ověřte, že stejnosměrné zesílení je dáno vzorcem

$$A_0 = 1 + R_1/R_2$$

a že kmitočet třidecibelového poklesu je

$$GBW/A_0,$$

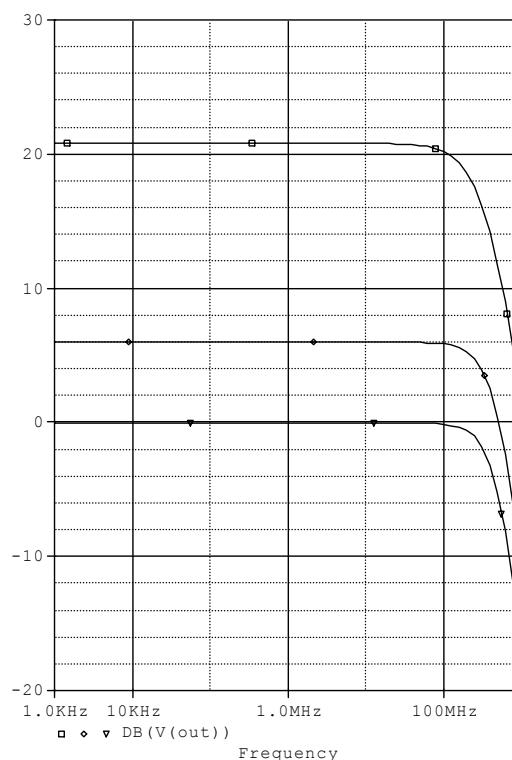
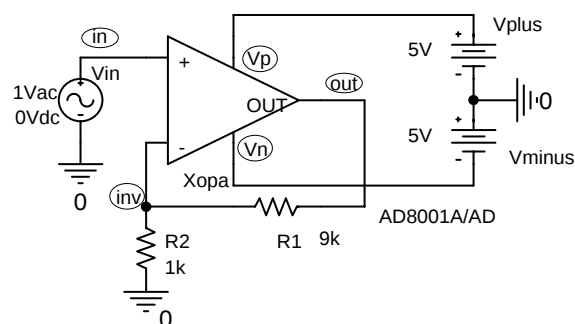
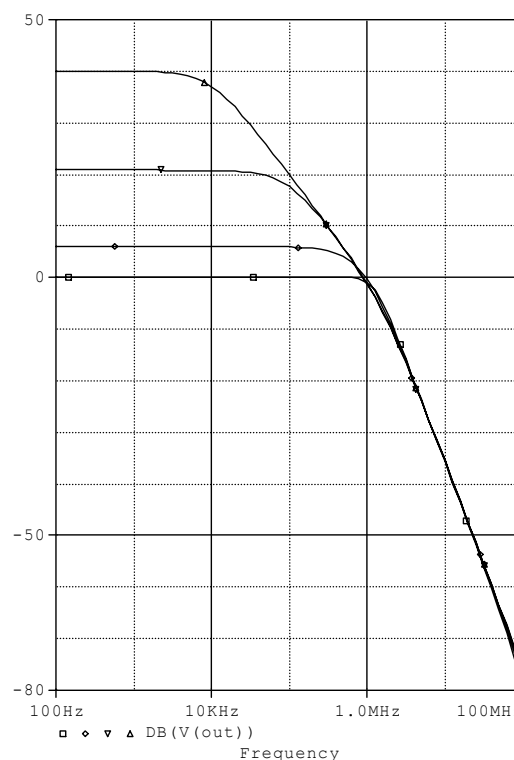
kde GBW je asi 1MHz.

Příklad 8.10 *Kmitočtové charakteristiky zesilovače s rychlým obvodem CFA typu AD8001*

Nakreslete amplitudové kmitočtové charakteristiky zesilovače s CFA operačním zesilovačem AD8001 pro odpory $R_1 = 1k$, $R_2 = 100, 1K, 1G$.

Řešení:

```
Operacni zesilovac AD8001
Vin in 0 AC 1V
Vplus Vp 0 5V
Vminus Vn 0 -5V
R1 out inv 1k
R2 inv 0 {Rx}
X in inv Vp Vn out AD8001A/AD
.param Rx 10k
.step param Rx list 100 1k 1G
.AC DEC 10 1k 1000meg
.probe
.lib
.end
```



Poznámky:

Tento operační zesilovač je „rychlejší“ než uA741, má daleko vyšší tranzitní kmitočet. Způsob řízení zesílení odpory je jiný než u napětového OZ. Řídíme-li zesílení odporem R2, pak je zachována šířka pásma zesilovače.

Ověřte, že stejnosměrné zesílení je dáno vzorcem

$$A_0 = 1 + R_1/R_2$$

Příklad 8.11 *Optimalizace zpětnovazebního obvodu pro AD8001*

Zapojte operační zesilovač AD8001 jako neinvertující zesilovač se zesílením 2. Změřte jeho amplitudovou kmitočtovou charakteristiku v kmitočtovém rozsahu od 10MHz do 400MHz pro odpory R1=R2= (649, 698, 750) ohmů.

Řešení:

Operacni zesilovac AD8001

Vin in 0 AC 1V

Vplus Vp 0 5V

Vminus Vn 0 -5V

R1 out inv {Rx}

R2 inv 0 {Rx}

X in inv Vp Vn out AD8001a/AD

.param Rx 10k

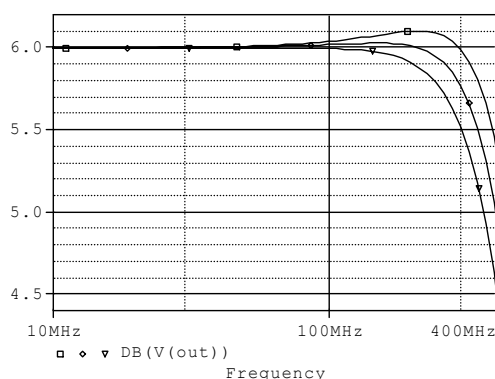
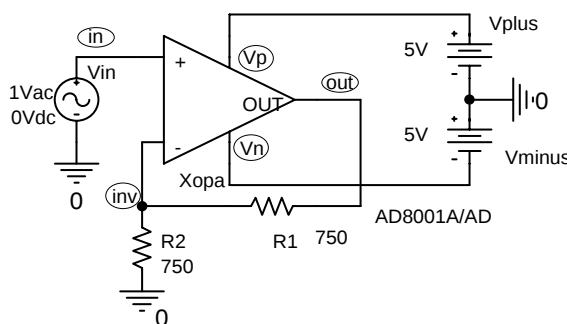
.step param Rx list 649 698 750

.AC OCT 10 10meg 400meg

.probe

.lib

.end



Poznámka: U operačních zesilovačů CFA záleží nejen na poměrech odporů R1 a R2, ale i na jejich absolutních velikostech. Je třeba optimalizovat zejména zpětnovazební odpor R1. Čím menší R1, tím větší šířka pásma, ale tím větší překmitý v kmitočtové charakteristice a náchylnost k nestabilitě.

Hodnoty odporů jsou převzaty z katalogového listu AD8001. Zkontrolujte s výsledky simulace.

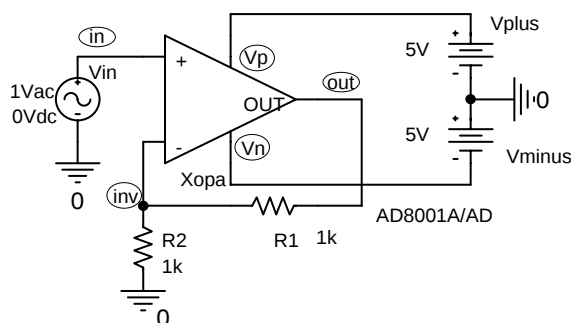
Příklad 8.12 *Zjišťování stability z kmitočtových charakteristik*

Analyzujte amplitudovou a fázovou kmitočtovou charakteristiku otevřené smyčky zesilovače pro

$$R_1=R_2=(1000, 400, 300, 200) \text{ ohmů}$$

a pro parazitní kapacitu Cpar=10pF.

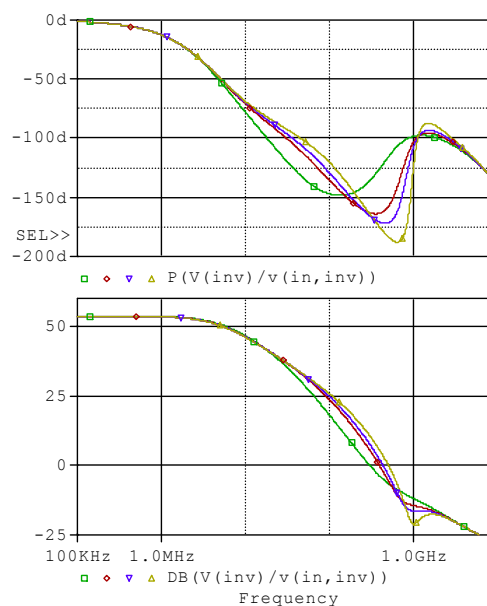
Z charakteristik odečtěte fázovou bezpečnost pro všechny 4 případy a rozhodněte o potenciální stabilitě zesilovače. Porovnejte s výsledky z příkladu 8.11.



Poznámka: Fázová bezpečnost je doplněk fázového posunu mezi výstupním a vstupním napětím zesilovače do 180 stupňů na kmitočtu, při němž je zesílení 1, tj. 0 db. Čím je fázová bezpečnost větší, tím lépe pro stabilitu obvodu.

Řešení:

```
Operacni zesilovac AD8001
Vin in 0 AC 1V
Vplus Vp 0 5V
Vminus Vn 0 -5V
R1 out inv {Rx}
R2 inv 0 {Rx}
Cpar inv 0 10p
X in inv Vp Vn out AD8001a/AD
.param Rx 200
.step param Rx list 1000 400 300 200
.AC OCT 100 100k 10G
.probe
.lib
.end
```



Evaluate	Measurement	1	2	3	4
☒	PhaseMargin(db(v(inv)/v(in,inv)),P(v(i...	44.06162	16.36399	8.08168	-3.50735

Fázovou bezpečnost změříme buď pomocí kurzorů nebo lépe pomocí měřicí funkce PhaseMargin. Pro odpory 200 ohmů je již fázová bezpečnost záporná, což znamená nestabilitu.

Poznámka: Přenos tzv. otevřené smyčky (Open Loop Gain) je přenos od vstupních svorek OZ k výstupu děliče R1-R2, tj. $V(\text{inv})/V(\text{in}, \text{inv})$.

Ověřte si, že při parazitní kapacitě 7pF a méně je již fázová bezpečnost vždy kladná pro všechny uvažované odpory.

8.11 Kontrolní otázky ke kapitole 8

1. Před analýzou AC počítá PSPICE stejnosměrný pracovní bod:

a) vždy bez výjimky, b) uživatel může výpočet zakázat slovem SKIPBP, c) u lineárních obvodů se pracovní bod nepočítá, protože není třeba provádět následnou linearizaci obvodu

2. Příkaz

.AC LIN 2 1k 10k

znamená provedení AC analýzy na kmitočtech:

a) 1kHz, b) 1kHz, 10kHz, c) 1kHz, 5.5kHz, 10kHz

3. Příkaz

.AC DEC 1 1k 10k

znamená provedení AC analýzy na kmitočtech:

a) 1kHz, b) 1kHz, 10kHz, c) je syntakticky nesprávně

4. Při analýze AC se nezávislý zdroj v obvodu nahradí:

a) vždy zdrojem harmonického signálu o jednotkové amplitudě, **b)** vždy zdrojem harmonického signálu o amplitudě, dané atributem AC, **c)** nenahrazuje se, uvažuje se původní zdroj

5. Amplitudu vstupního napětí pro střídavou analýzu AC nastavíme:

a) atributem AC nezávislého zdroje V, **b)** specifikací SIN u nezávislého zdroje V, **c)** ve vzorci VALUE u řízeného zdroje E.

6. Při analýze AC počítá PSpice amplitudu napětí V(1) z výrazu

a) VR(1), **b)** VMAG(1), **c)** V(1)

7. Při analýze AC interpretuje PSpice výraz VR(1) jako

a) střídavé napětí na odporu R1, **b)** stejnosměrné napětí uzlu 1, **c)** reálnou část napětí uzlu 1.

8. Zdroj o definici

Vsinus 2 10 AC 10 90 SIN 0 1 5k

bude mít při analýze AC amplitudu

a) 1V, **b)** 2V, **c)** 10V

9. Slovo LAPLACE v syntaxi řízených zdrojů SPICE se u analýzy AC:

a) uplatňuje, protože modeluje kmitočtovou závislost přenosu, **b)** neuplatňuje, projeví se u statistické analýzy, **c)** vůbec se v syntaxi řízených zdrojů nevyskytuje.

10. Příkaz

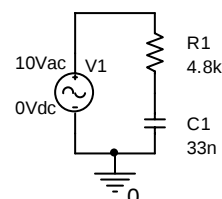
.print AC V(3) VP(3)

a) musí, **b)** nesmí, **c)** může být doprovázen příkazem .AC.

8.12 Neřešené příklady

Příklad 8.13 *Střídavá analýza RC článku*

Zjistěte velikost a počáteční fázi napětí na R a na C při kmitočtu (100, 1000, 10000) Hz. Na vstupu je zdroj napětí o amplitudě 10V a počáteční fázi 0.

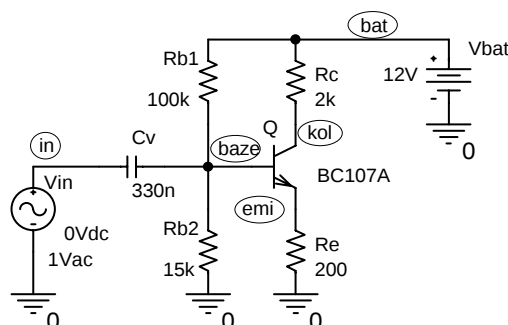


Příklad 8.14 *Kmitočtové charakteristiky RC článku*

Sestrojte logaritmické amplitudové a fázové kmitočtové charakteristiky RC článku z příkladu 8.13, je-li výstupní napětí bráno buď z rezistoru, nebo z kapacitoru.

Příklad 8.15 *Kmitočtové charakteristiky tranzistorového zesilovače*

Nakreslete amplitudové a fázové kmitočtové charakteristiky tranzistorového zesilovače v rozsahu kmitočtů od 1Hz do 100MHz pro tři různé kapacity Cv: 3.3nF, 33nF a 330nF. Zajímejte se pouze o napěťové zesílení ze vstupu na kolektor.



Příklad 8.16 *Určení vstupní impedance v zesilovači z příkladu 8.15*

Zjistěte kmitočtovou závislost impedance v obvodu z příkladu 8.15 mezi bází a zemí, avšak bez uvažování vlivu RB1, RB2 a Cv. Jde nám o vstupní impedanci tranzistoru, která bude modifikována působením záporné zpětné vazby přes Re.

Návod: Musíte odstranit odpory Rb1, Rb2 a kapacitor Cv a obvod doplnit tak, aby byl zachován původní stejnosměrný pracovní bod. Souřadnice tohoto bodu naleznete v řešení příkladu 8.15.

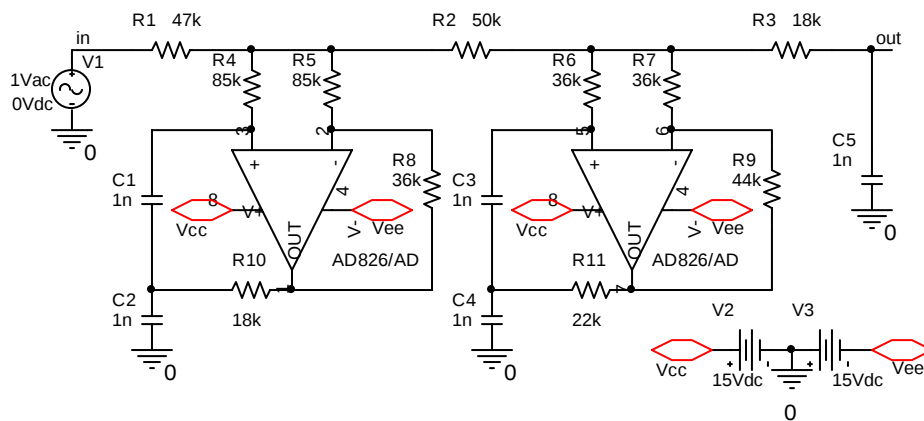
Příklad 8.17 *Určení výstupní impedance celého zesilovače z příkladu 8.15*

Určete kmitočtovou závislost výstupní impedance zesilovače z příkladu 8.15, tj. impedanci naměřenou mezi svorkami kol a 0, za předpokladu nepůsobení střídavého vstupního napětí, tj. za předpokladu „střídavého zkratu“ vstupních svorek.

Návod: Nyní je potřebné budit střídavým signálem výstupní svorky, ale tak, aby nebyl narušen nastavený stejnosměrný pracovní bod.

Příklad 8.18 *Kmitočtová charakteristika aktivního filtru*

Na obrázku je schéma aktivního filtru typu dolní propust podle Bauerovy aproximace, s mezním kmitočtem 3.4kHz. Jsou použity operační zesilovače AD826. Nakreslete amplitudovou a fázovou kmitočtovou charakteristiku v kmitočtovém rozsahu od 500Hz do 50kHz.



Příklad 8.19 Behaviorální modelování filtru z příkladu 8.18

Přenosová funkce filtru z příkladu 8.18 má v případě ideálních operačních zesilovačů tvar

$$\frac{V(out)}{V(in)} = K \frac{(p^2 + A_1^2)(p^2 + A_2^2)}{(p + B)[(p + R_1)^2 + I_1^2][(p + R_2)^2 + I_2^2]}$$

$$K = -2919.75817532289$$

$$A_1 = 35533.4527259350, A_2 = 25565.4996282456$$

$$B = -14241.7235360065$$

$$R_1 = -6944.76097738481, R_2 = -1360.43294756782$$

$$I_1 = 17530.2667416564, I_2 = 21771.7554665257$$

Sestavte behaviorální model filtru a srovnajte průběhy kmitočtových charakteristik s charakteristikami filtru z příkladu 8.18.

9 Příloha: Simulace v OrCadPSpice 10 na úrovni vstupních souborů

(bez schématického editoru)

Spouštěný program: PSpice.exe (ikona PSpice AD Demo), nikoliv Capture CIS Demo.

Základní metody tvorby vstupního souboru:

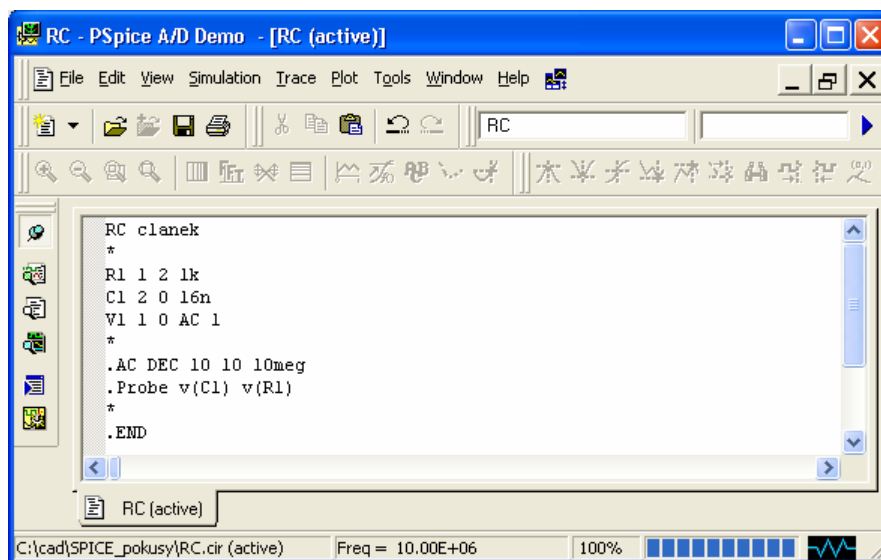
A) Tvorba vstupního souboru externím programem.

1. V jakémkoliv textovém manažeru, např. v NotePadu, vytvoříme vstupní soubor pro simulaci. Jeho přípona bude cir.
2. V prostředí jakéhokoliv „file manageru“ (Windows Commander, Průzkumník..) kliknutím na soubor spustíme PSpice (pokud jsou soubory s příponou „cir“ asociovány s PSpicem). Nebo použijeme jiný postup, viz 3:
3. Spustíme PSpice.exe, zvolíme „File/Open simulation“, v okně „Open Simulation“ nastavíme masku „Soubory typu“ na „Circuit Files (*.cir)“ a vybereme náš vstupní soubor.

B) Tvorba vstupního souboru přímo v PSpice

1. Spustíme PSpice.exe, zvolíme „File/New/Text File“ (alternativou je horká klávesa Ctrl N).
2. Vytvoříme vstupní soubor a uložíme jej volbou „File/Save As. Maska typu souboru může být nastavena jakákoliv, avšak název souboru musíme napsat úplný, t.j. s příponou cir!!!
3. Zvolíme „File/Open simulation“, v okně „Open Simulation“ nastavíme masku „Soubory typu“ na „Circuit Files (*.cir)“ a vybereme náš vstupní soubor.

Připravenost programu k simulaci na základě vstupního souboru je indikována modrou šipkou na horní liště (ikona pro spouštění simulace).



Práce s několika soubory:

Často potřebujeme pracovat s několika textovými soubory „současně“. Například tvoříme vstupní soubor pro simulaci a potřebujeme založit svou vlastní novou knihovnu nebo editovat jinou knihovnu na disku.

V prostředí PSpice není možné mít současně otevřených více vstupních souborů typu *.cir. Pokus o otevření dalšího vstupního souboru vede na deaktivaci souboru předchozího.

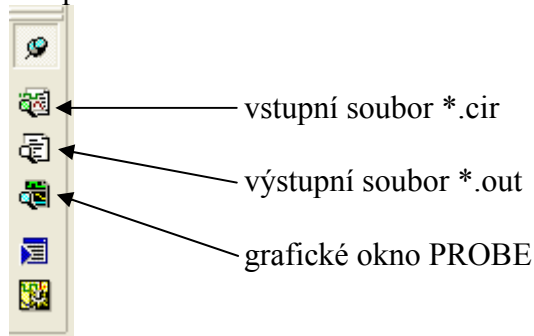
Další soubor jiného typu než vstupní, např. knihovnu, otevřeme volbou „File/Open“. Zvolíme příslušnou masku (v případě nutnosti „All Files“) a vybereme daný soubor. Chceme-li založit nový soubor, který na disku ještě není, provedeme volbu „File/New/Text File“ a po jeho vytvoření jej uložíme pod úplným názvem, tedy i s příponou.

Mezi otevřenými soubory přecházíme klikáním na jejich záložky podobně jako na listy v Excelu. PSpice sice nezobrazuje přípony v názvech souborů, avšak vstupní soubor poznáme podle toho, že je za ním připojena poznámka „(active)“. Kromě toho je název „aktivní úlohy“, která je připravena k simulaci, napsán v okénku na horní liště (viz název „RC“ na obrázku na předchozí straně).

Simulace:

Aktivuje se kliknutím na „modrou šipku“ (viz obrázek na předchozí straně) nebo volbou „Simulation/Run“.

Pokud jsou ve vstupním souboru instrukce pro výstup dat např. do „Probe“, otevře se příslušný další list. Příslušné výstupy simulace je možno zobrazovat a volit pomocí levého sloupce ikon:



Při práci se simulační úlohou často pracujeme tak, že po prozkoumání výsledků simulace, chybových hlášení atd. se vracíme do okna vstupního souboru či do okna knihovny, provedeme určité úpravy a poté opět spouštíme simulaci. Před spuštěním simulace se nás program zeptá, zda chceme uložit provedené změny. Proto je vhodné vždy před kliknutím na „modrou šipku“ kliknout na ikonu s disketou.

Struktura vstupního souboru: VSTUPNÍ SOUBOR (* .CIR)

Hlavička
Popis obvodu = netlist
Příkazy pro řízení simulace
.END

PŘÍKLAD: RLC.CIR

RLC obvod
* Vzdroj 1 0 4.5V Ls 1 2 10mH Rs 2 3 100 Rp 3 0 3.3k Cp 3 0 220nF *
.TRAN 1u 1m SKIPBP .PROBE V(Cp) V(Ls)
* .end

10 Správné výsledky ze vstupního testu

1 c, d, c, c

2 c, b, a, c

3 c, a, c, d

4 b, b

5 c, a, b

6 c, c, b

7 d, a, a

8 a, a, b

9 c, b, b

10 a, a, c

11 b, a, d

12 b, a, d

13 c, b, c

14 c, a

15 c

11 Odpovědi na kontrolní otázky

číslo otázky	číslo kapitoly			
	4	5	6	7
1	b	a	a	c
2	b	b	c	b
3	a	b	b	c
4	b	a	c	a
5	c	c	b	b
6	a*)	a	b	c
7	c	c	c	c
8	b	b	b	a
9	b		c	a
10	b		b	b
11	a			c
12	b			b
13	b			a
14				c

*) v příkazu .step musí být slovo param před r, při krokování nesmí R1 ani R2 nabýt nulové hodnoty, toto se musí ošetřit dvěma zásahy do vstupního souboru

12 Výsledky neřešených příkladů

12.1 Kapitola 4

Příklad 4.13 *Dělič napětí z příkladu 4.1*

- 1) Slovo DC není v syntaxi povinné
- 2) Záměnou pořadí uzlů dojde k přepólování zdroje napětí a tím i k změně znamének všech napětí v obvodu, která jsou od napětí zdroje odvozena.
- 3) Viz význam příkazu .OP.

Příklad 4.14 *Zatížený dělič napětí*

Odporovy delic

R1 1 2 4k

R2 2 0 1k

Rx 2 0 1k

V1 1 0 DC 10V

.END

Výstupní napětí nyní vyjde 1.1111V.

Příklad 4.15 *Modelování potenciometru*

Potenciometricky delic

.param alfa 0.1

R1 1 2 {5k*alfa}

R2 2 0 {5k*(1-alfa)}

V1 1 0 DC 10V

.step param alfa list 0.1 0.5 0.9

.END

Výstupní napětí vyjde (9, 5, 1)V.

Příklad 4.16 *Obvod s diodou z příkladu 4.4*

Po „otočení“ bude dioda pólovaná v závěrném směru, nepoteče jí proud a napětí na ní bude prakticky 10V.

Obvod s diodou

R1 1 2 4k

D1 0 2 obycejna_dioda

V1 1 0 10V

```
.model obycejna_dioda D
.END
```

Příklad 4.17 *Obvod s diodou z příkladu 4.6*

```
Obvod s diodou
R1 1 2 4k
D1 2 0 obycejna_dioda
V1 1 0 10V
.temp 27 37
.model obycejna_dioda D
.END
```

Napětí na diodě vyjde 0.6770V při 27 stupních Celsia a 0.6600V při 37 stupních Celsia. Teplotní koeficient napětí je $(0.6600-0.6770)/(37-27)=-1.7\text{mV}/^{\circ}\text{C}$.

Příklad 4.18 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

```
Stabilizator se Zenerovou diodou
R1 1 2 250
D1 0 2 D1N750
V1 1 0 10V
Rz 2 0 1k
.model D1N750 D(Is=880.5E-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p M=.5516
+ Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=4.7 Ibv=20.245m Nbv=1.6989
+ Ibvl=1.9556m Nbv1=14.976 Tbv1=-21.277u)
.END
```

Bez Rz je výstupní napětí 4.7031V, s Rz mírně poklesne na 4.69V. Krokováním Rz lze zjistit, že při Rz pod 200 ohmů již výstupní napětí rychle klesá.

Příklad 4.19 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

```
Stabilizator se Zenerovou diodou
R1 1 2 250
D1 0 2 D1N750
V1 1 0 10V
.temp 27 37
.model D1N750 D(Is=880.5E-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p M=.5516
+ Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=4.7 Ibv=20.245m Nbv=1.6989
+ Ibvl=1.9556m Nbv1=14.976 Tbv1=-21.277u)
.END
```

Výstupní napětí vyjde 4.7031V při 27 stupních a 4.7020V při 37 stupních. Teplotní součinitel vychází $-2,34 \cdot 10^{-5}/^{\circ}\text{C}$.

Příklad 4.20 *Stabilizátor se Zenerovou diodou z příkladu 4.8*

Stabilizátor se Zenerovou diodou

```
R1 1 2 250
D1 0 2 D1N750
V1 1 0 {V1}
.param V1 10V
.step param V1 list 10 20
.model D1N750 D(Is=880.5E-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p M=.5516
+      Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=4.7 Ibv=20.245m Nbv=1.6989
+      Ibv1=1.9556m Nbv1=14.976 Tbv1=-21.277u)
.END
```

Výstupní napětí vychází 4.7031V a 4.7620V pro vstupní napětí 10V a 20V. Činitel stabilizace vychází asi 170.

Příklad 4.21 *Jednostupňový tranzistorový zesilovač z příkladu 4.10*

```
zesilovac
Vin 1 0 AC 1
Cv1 1 2 10uF
Rb1 6 2 16.7k
Rb2 2 pom 3.7k
Vpom pom 0 0V
QT 3 2 5 obycejny_tranzistor
Rc 6 3 1k
Re 5 0 200
Cv2 3 4 10u
Rz 4 0 5k
Vbat 6 0 10V
.model obycejny_tranzistor NPN
.OP
.end
```

Proud zdrojem Vpom vychází 454.9μA.

Příklad 4.22 *Jednostupňový tranzistorový zesilovač z příkladu 4.10*

```
zesilovac
Vin 1 0 AC 1
Cv1 1 2 10uF
Rb1 6 2 16.7k
Rb2 2 0 3.7k
QT 3 2 5 obycejny_tranzistor
Rc 6 3 1k
Re 5 0 200
Cv2 3 4 10u
```

```

Rz 4 0 5k
Vbat 6 0 10V
.temp 20 30
.model obycejny_tranzistor NPN
.OP
.end

```

Analýzou vychází $I_{c20}=4.28\text{mA}$ a $I_{c30}=4.33\text{mA}$, tedy činitel teplotní stabilizace je 0.117.

12.2 Kapitola 5

Příklad 5.10 Použití funkce TABLE

```

E1 3 0 value={table(V(1,2), -13/A, -13, 13/A,13)}

```

Příklad 5.11 Neinvertující zesilovač s TL084

```

zesilovac s TL084
Vin in 0 1V
V1 Vplus 0 15V
V2 0 Vminus 15V
X in R1R2 Vplus Vminus out TL084
R1 out R1R2 4.7k
R2 R1R2 0 1k
.lib
.end

```

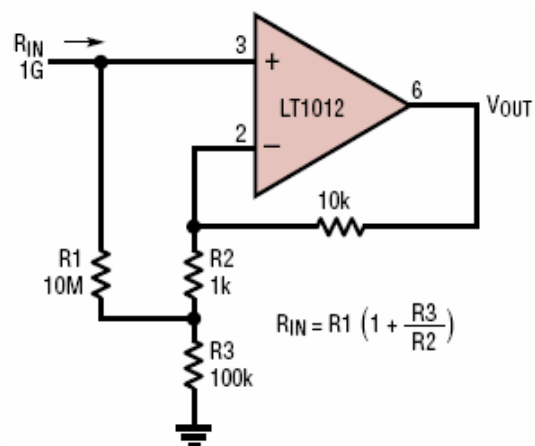
Výstupní napětí vychází 5.6997V. Teoretická hodnota pro ideální operační zesilovač je $1+R_1/R_2=5.7\text{V}$.

Příklad 5.12 Násobič odporu s LT1012

```

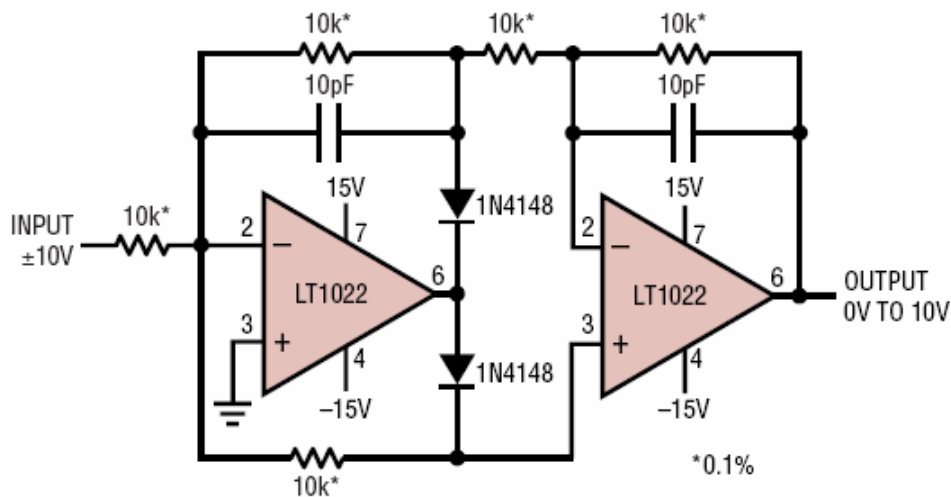
nasobic odporu s LT1012
Vin in 0 1V
V1 Vplus 0 15V
V2 0 Vminus 15V
X in R2R4 Vplus Vminus out LT1012/LT
R1 in R2R3 10meg
R2 R2R4 R2R3 1k
R3 R2R3 0 100k
R4 R2R4 out 10k
.lib
.end

```



Proud zdrojem V_{in} vychází $9.952 \cdot 10^{-10} \text{ A}$, takže vstupní odpor je $1.0048 \text{ G}\Omega$. Teoretická hodnota je $1.01 \text{ G}\Omega$.

Příklad 5.13 Přesný usměrňovač s LT1022



presny usmernovac s LT1022

Vin in 0 0.1V; v druhem kroku zamenime za -0.1V

V1 Vplus 0 15V

V2 0 Vminus 15V

X1 0 in1 Vplus Vminus out1 LT1022/LT

X2 in2 in3 Vplus Vminus out2 LT1022/LT

D1 R2R4 out1 D1N4148

D2 out1 in2 D1N4148

R1 in in1 10k

R2 in1 R2R4 10k

R3 in1 in2 10k

R4 R2R4 in3 10k

R5 in3 out2 10k

C1 in1 R2R4 10p

C2 in3 out2 10p

.lib

.end

Napětí na výstupu out2 vychází 0.1001V a 0.1003V pro vstupní napětí 0.1V a -0.1V.

12.3 Kapitola 6

Příklad 6.21: Modelování kvadrátoru z příkladu 6.4

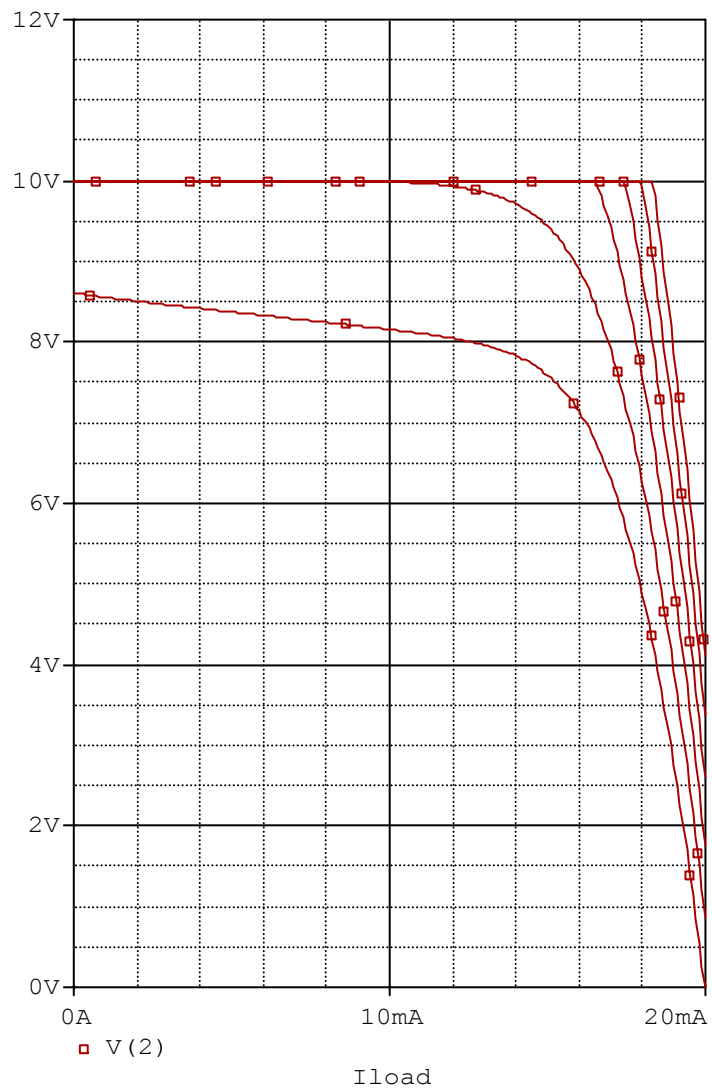
Ekvad 3 0 POLY(1) 2 0 0 0 1

Příklad 6.22: *Modelování zdroje s napětovou referencí REF01 z příkladu 6.18*

```

zdroj 10V
Vnap 1 0 15V
R1 2 3 8.77k
R2 3 0 1.23k
Iload 2 0 20mA
Xreference 1 0 3 2 REF-01/AD
.DC Iload 0 20m 0.1m
+Vnap 10 20 2
.PROBE
.lib
.end

```

**Příklad 6.23:** *Obvod pro hlídání napětí baterie*

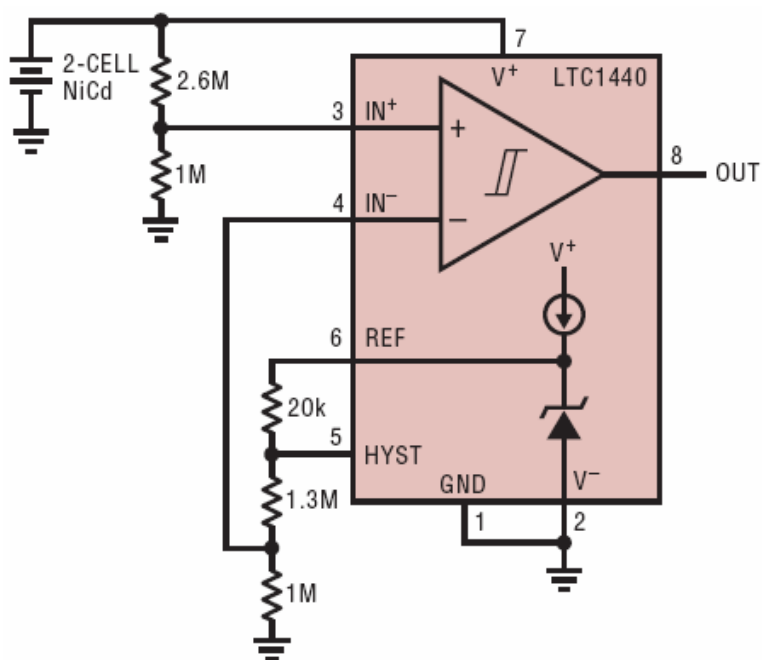
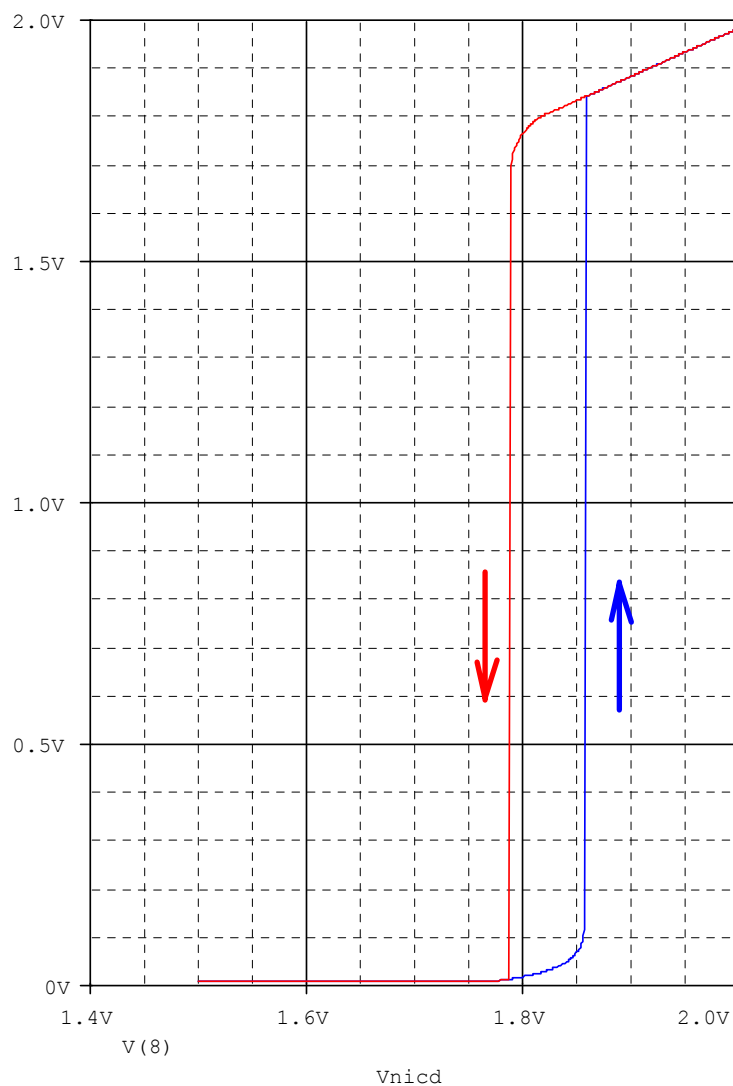
```

hlidac baterie s LTC1440
Vn1cd 7 0 2V
X 3 4 7 0 8 0 6 5 LTC1440/LT
R1 7 3 2.6MEG
R2 3 0 1MEG
R3 6 5 20k
R4 5 4 1.3MEG
R5 4 0 1MEG
.DC Vn1cd 1.5 2 0.001; simulace nabijeni baterie, modra krivka
*.DC Vn1cd 1.5 2 0.001; simulace vybijeni baterie, cervena krivka
.PROBE
.lib
.end

```

Při poklesu napětí baterie pod 1.785V se na výstupu objeví nízká úroveň – pokyn k dobíjení.

Při vzrůstu napětí baterie nad 1.86V se výstup přepne na vysokou úroveň – pokyn k ukončení nabíjení.



Příklad 6.24: *Nízkopříkonová napěťová reference LT1790*

reference s LT1790-2.5

Vin in 0 10V

X in 0 out LT1790-25

C1 in 0 0.1u

C2 out 0 0.1u

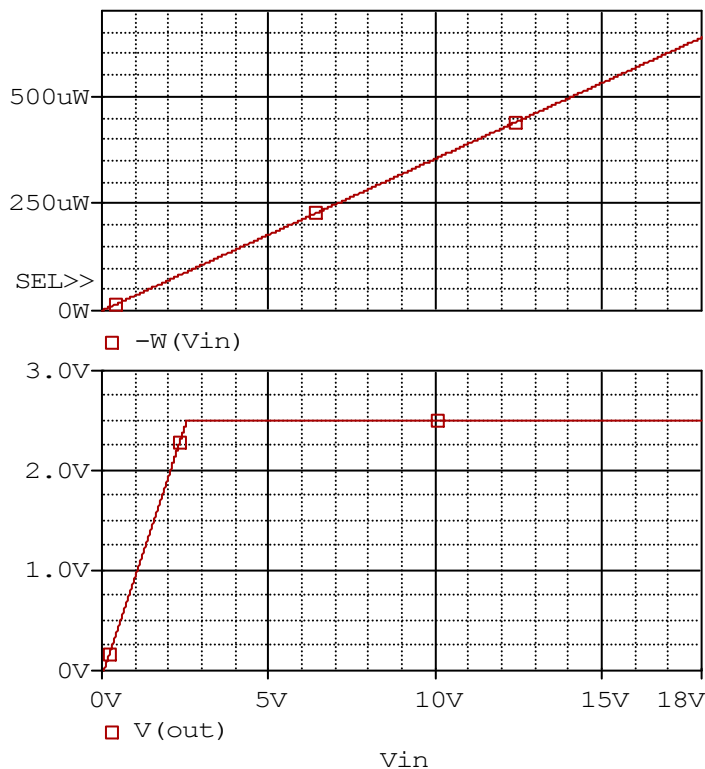
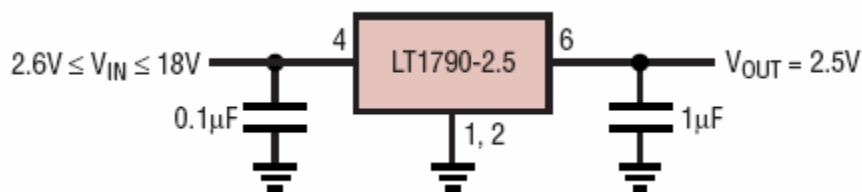
.dc Vin 0 18 0.01

.lib LTC.lib

.probe

.end

Výstupní napětí je velmi přesně 2.5V pro vstupní napětí od 2.58V a větší. Příkon při výstupu naprázdno je velmi malý, řádově se jedná o stovky mikrowattů.

**Příklad 6.25:** *Kvadrátor s analogovou násobičkou AD633*

kvadrator s AD633

Vin X1 0 1V

X X1 0 X1 0 Vneg 0 W Vpos AD633/AD

Vpos Vpos 0 15V

Vneg Vneg 0 -15V

C1 Vpos 0 0.1uF;zbytecne..

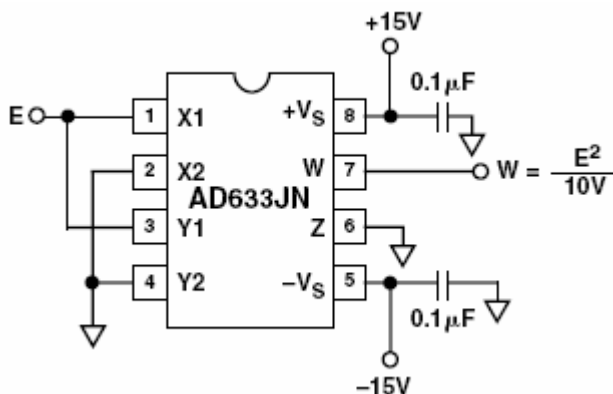
C2 Vneg 0 0.1uF;zbytecne..

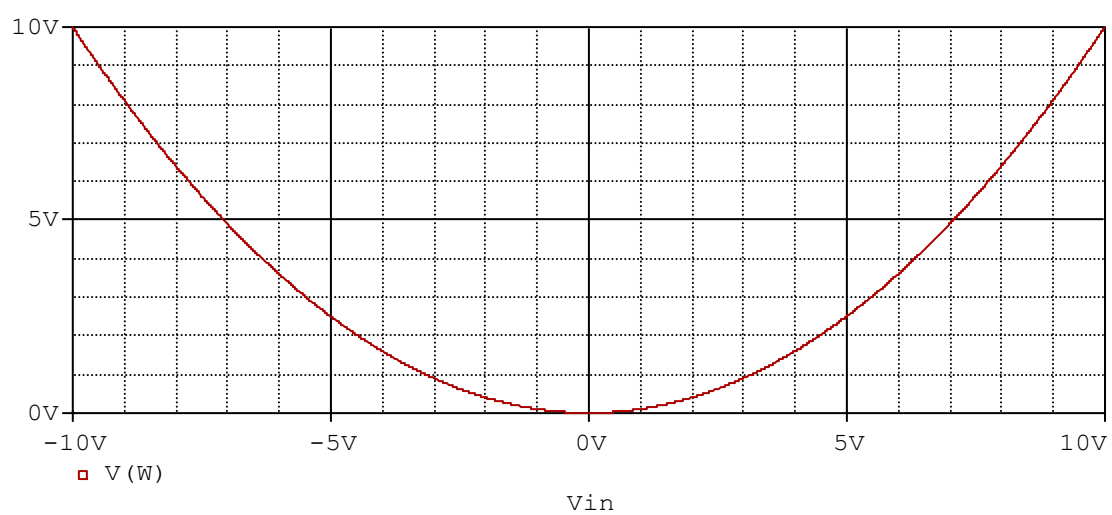
.DC Vin -10 10 0.01

.PROBE

.lib

.end





12.4 Kapitola 7

Příklad 7.14: Přechodný děj v RLC obvodu z příkladu 7.1

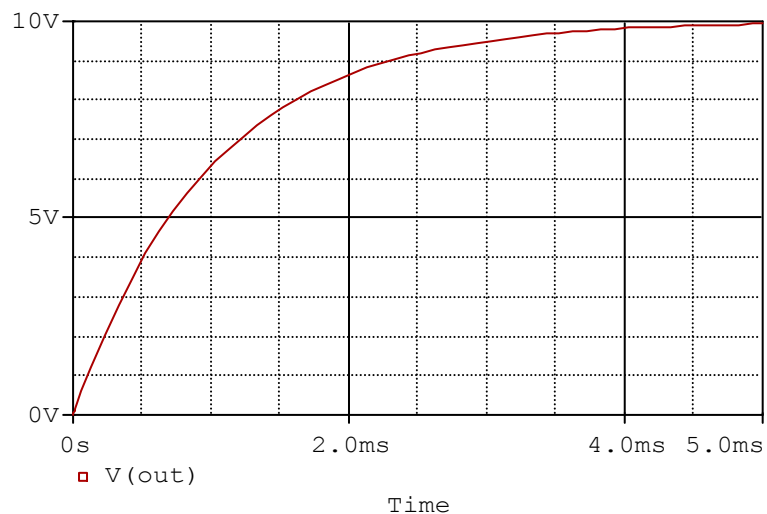
Analýzu inicializujeme bez modifikace „step ceiling“, viz př. 7.2, ale přidáme příkaz

.OPTIONS RELTOL 1e-6

Příklad 7.15: Nabíjení kapacitoru z baterie

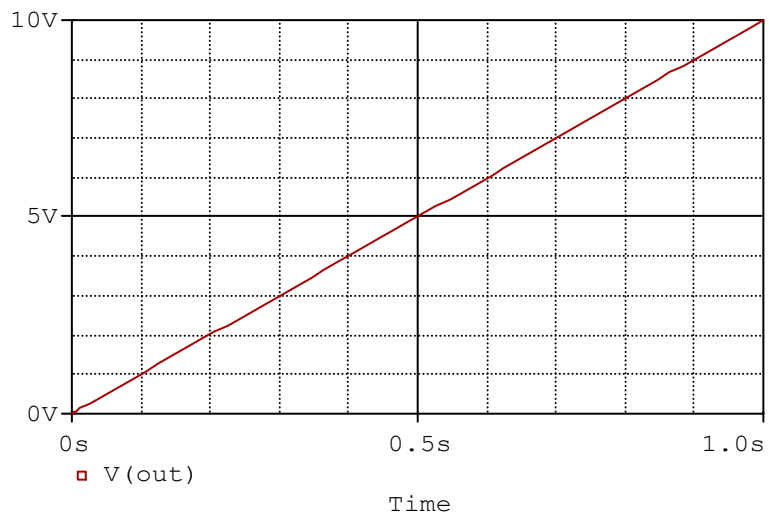
```
nabijeni kapacitoru
Vin in 0 10V
Ri in out 10
C out 0 100u
.tran 0 5m skipbp
.probe
.end
```

Kapacitor se nabije na 6.3V za 1ms, což je časová konstanta RiC.



Příklad 7.16: Nabíjení kapacitoru ze zdroje proudu

```
nabijeni kapac. ze zdroje I
Iin 0 out 1mA
Rpom out 0 1T
*Rpom definuje ss cestu
+mezi uzly out a 0
C out 0 100u
.tran 0 1 skipbp
.probe
.end
```

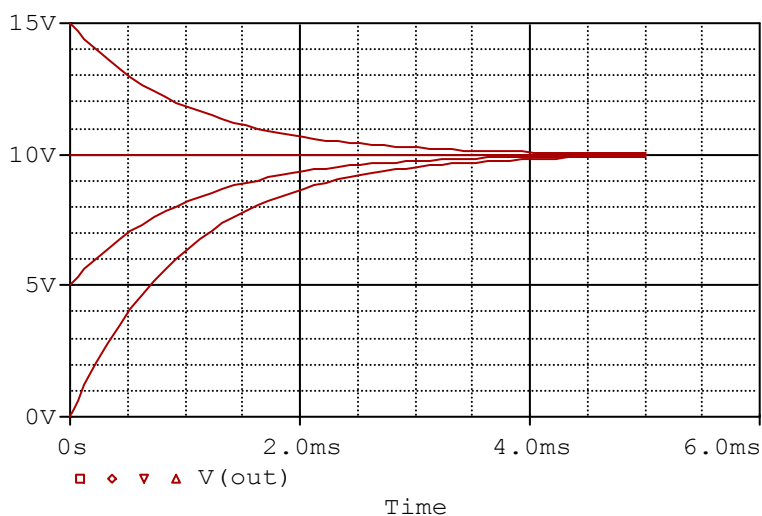


Příklad 7.17: *Nabíjení kapacitoru z baterie s nenulovými počátečními podmínkami*

```

nabíjení kapacitoru
Vin in 0 10V
Ri in out 10
C out 0 100u
.param Vini 0V
.IC V(out)={Vini}
.step param Vini list 0 5 10 15
.tran 0 5m skipbp
.probe
.end

```



Pozor – v PSPICE nefunguje krokování počátečních podmínek definovaných v rámci modelu akumulčního prvku. Ověřte si na níže uvedeném příkladu.

nabíjení kapacitoru – špatný způsob krokování počátečních podmínek

```

Vin in 0 10V
Ri in out 10
C out 0 100u IC={Vini}
.param Vini 0V
.step param Vini list 0 5 10 15
.tran 0 5m skipbp
.probe
.end

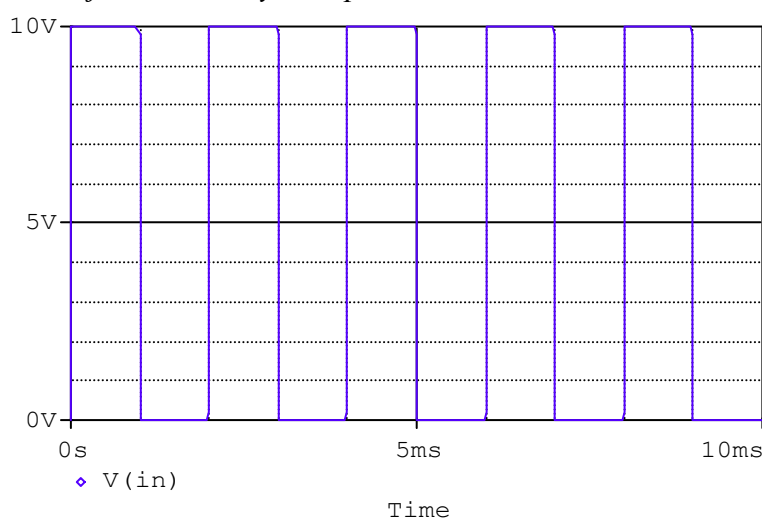
```

Příklad 7.18: *Modelování zdroje obdélníkových impulzů*

```

Zdroj impulsu
Vin in 0 PULSE
+0 10 0 1p 1p 1m 2m
.tran 0 10m
.probe
.end

```



Příklad 7.19: *Nabíjení kapacitoru ze zdroje impulzů*

nabíjení kapacitoru z impulzního zdroje

Vin in 0 PULSE 0 10 0 1p 1m 2m

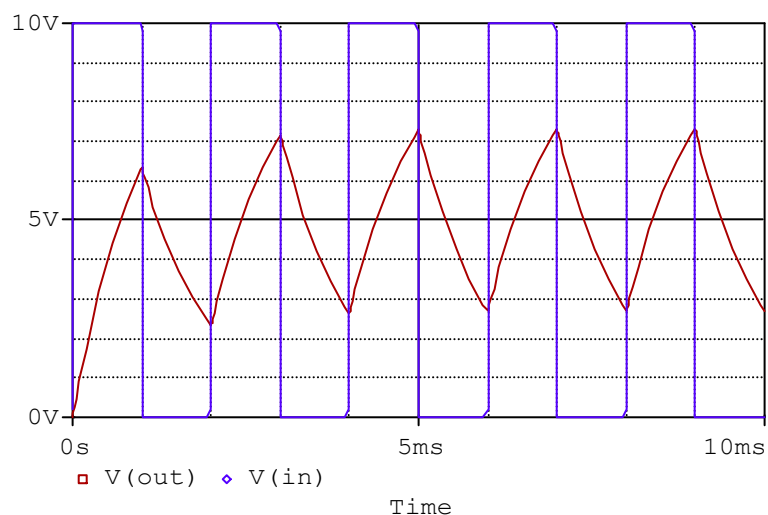
Ri in out 10

C out 0 100u

.tran 0 10m skipbp

.probe

.end



V čase 10ms již prakticky dochází k ustalování do periodického ustáleného stavu. Napětí na kapacitoru kolísá cca mezi hodnotami 2.6881V a 7.3107V.

Příklad 7.20: *Nalezení periodického ustáleného stavu v RC obvodu z příkladu 7.19*

Vstupní soubor z příkladu 7.19 doplníme o příkaz

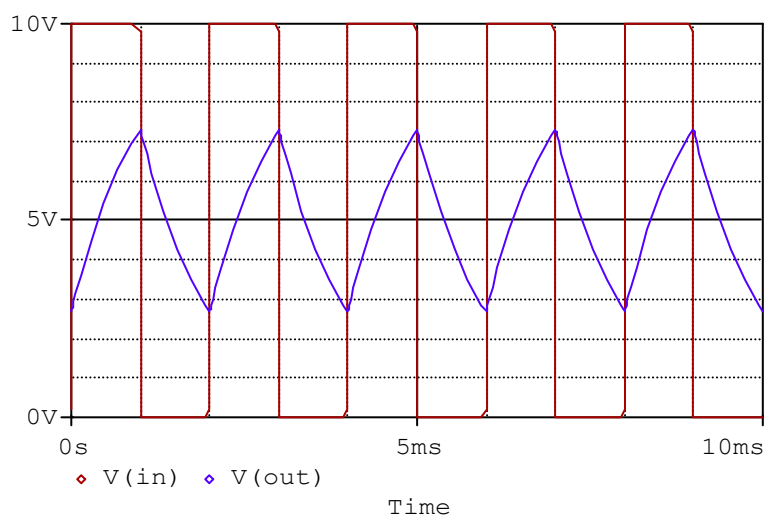
.SAVEBIAS ic.tra

+TRAN TIME=10ms

a spustíme analýzu. Na disku nalezneme vygenerovaný soubor ic.tra. Vyhledáme v něm příkaz .NODESET a zaměníme jej na .IC. Uložíme změnu. Poté modifikujeme ve vstupním souboru řádek s příkazem .SAVEBIAS takto:

.LOADBIAS ic.tra

Pak spustíme opět analýzu.



Příklad 7.21: Nasazování kmitů ve funkčním generátoru

funkční generator

X1 0 in1 Vcc Vee out1 TL072/301/TI

X2 in2 0 Vcc Vee out2 TL072/301/TI

Vpos Vcc 0 15V

Vneg 0 Vee 15V

R1 in1 out2 1k

R2 out1 in2 1k

R3 in2 out2 2k

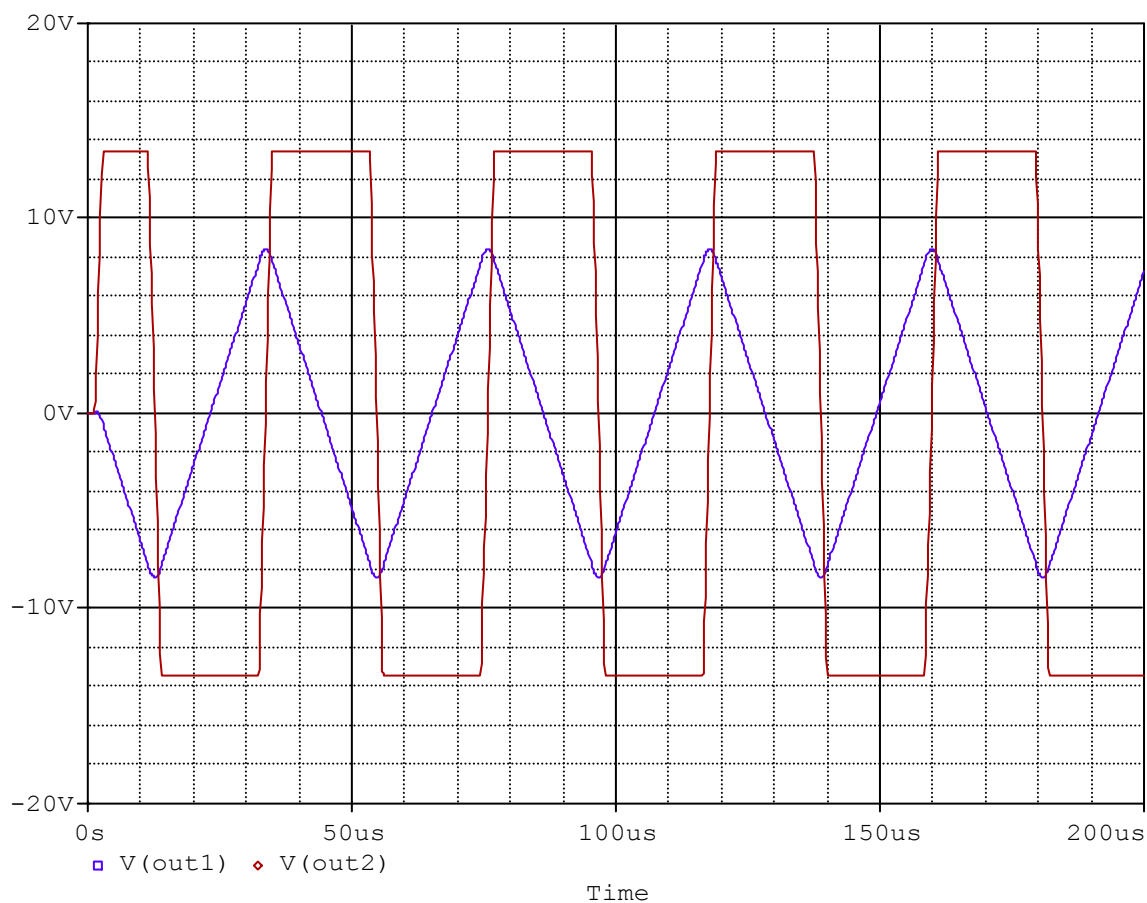
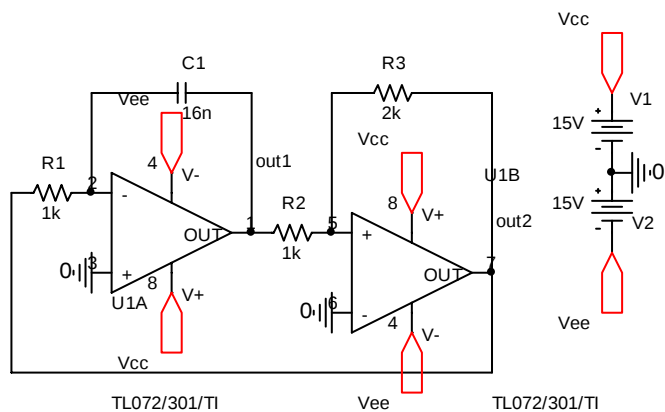
C1 in1 out1 16n

.tran 0 200u 0 200n

.lib

.probe

.end



Evaluate	Measurement	Value
☒	1/Period_XRange(V(out1),120u,200u)	23.77992k

Kmitočet je asi 23.8kHz. Podle teorie by měl být v ideálním případě

$$0.455/(R1C1)=28.4\text{kHz.}$$

Příklad 7.22: *Modelování signálu vzorcem*

zdroj dvoucestne usmerneneho sinusoveho napeti

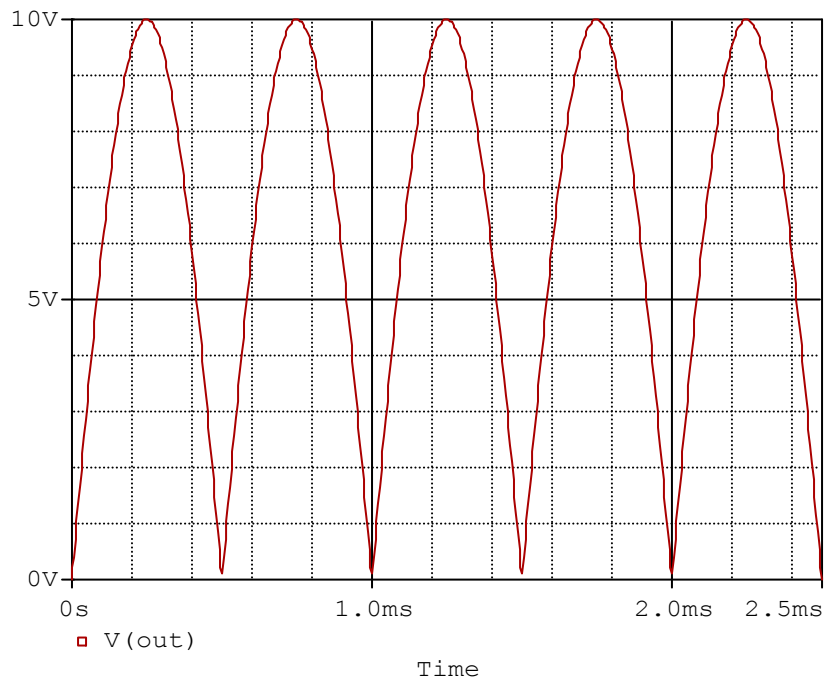
.param pi=3.141592 F=1k

Erect out 0 VALUE={10*abs(sin(2*pi*F*time))}

.tran 0 2.5m 0 5u

.probe

.end

**12.5 Kapitola 8****Příklad 8.13** *Střídavá analýza RC článku*

RC clánek

Vin in 0 Ac 10

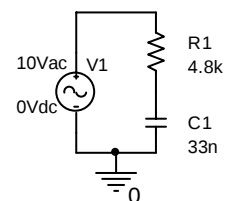
R in out 4.8kOhm

C out 0 33nF

.AC DEC 1 100 10k

.print AC V(R) VP(R) V(C) VP(C)

.end

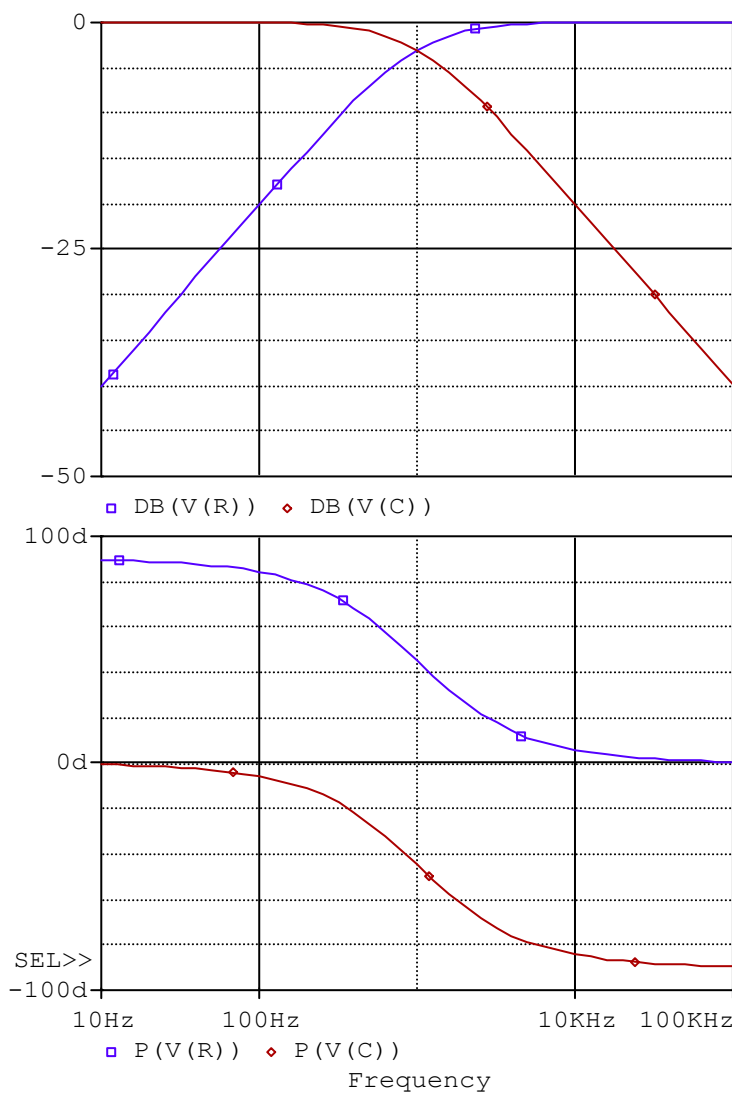


FREQ	V(R)	VP(R)	V(C)	VP(C)
1.000E+02	9.904E-01	8.432E+01	9.951E+00	-5.684E+00
1.000E+03	7.054E+00	4.514E+01	7.088E+00	-4.486E+01
1.000E+04	9.950E+00	5.738E+00	9.997E-01	-8.426E+01

Ověřte si ručním výpočtem, že kmitočet třídécibellového poklesu je zhruba 1kHz. Na tomto kmitočtu jsou velikosti napětí na R a C stejné a jsou rovny V_{in} lomeno odmocnina ze dvou, tj. asi 7.07V. Fázový posuv napětí na R a C oproti vstupu je +45 stupňů a -45 stupňů. Pro rostoucí kmitočet napětí na R roste (horní propust) a na C klesá (dolní propust).

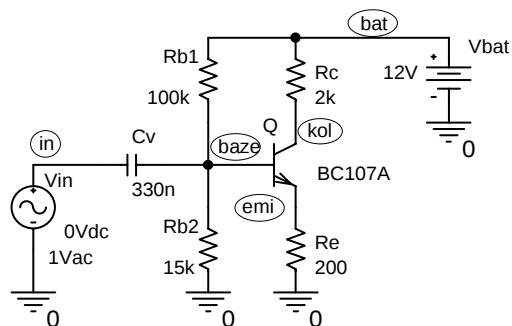
Příklad 8.14 Kmitočtové charakteristiky RC článku

```
RC clanek
Vin in 0 Ac 1
R in out 4.8kOhm
C out 0 33nF
.AC DEC 10 10 100k
.probe v(R) v(C)
.end
```

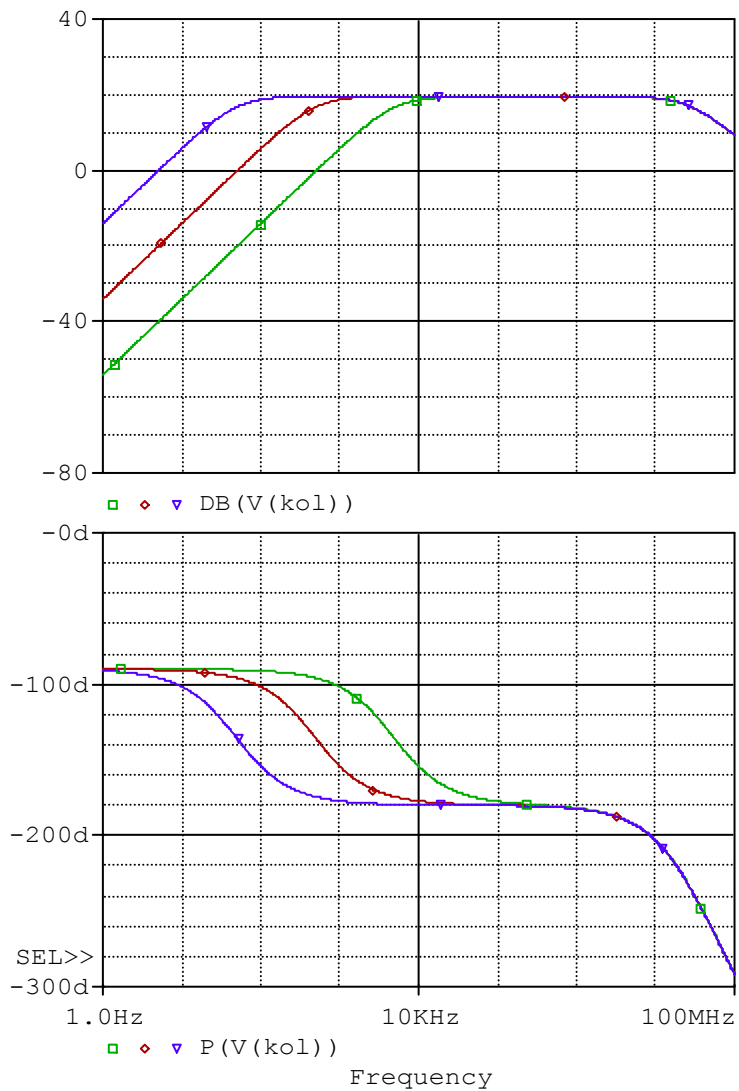


Příklad 8.15 Kmitočtové charakteristiky tranzistorového zesilovače

```
tranzistorovy zesilovac
Vin in 0 AC 1
Cv in baze {Cv}
Rb1 bat baze 100k
Rb2 baze 0 15k
Rc bat kol 2k
Re emi 0 200
Vbat bat 0 12V
Q kol baze emi BC107A
.param Cv {330nF}
```



```
.step DEC param Cv 3.3n 330n 1
.AC DEC 100 1 100meg
.probe
.lib
.end
```

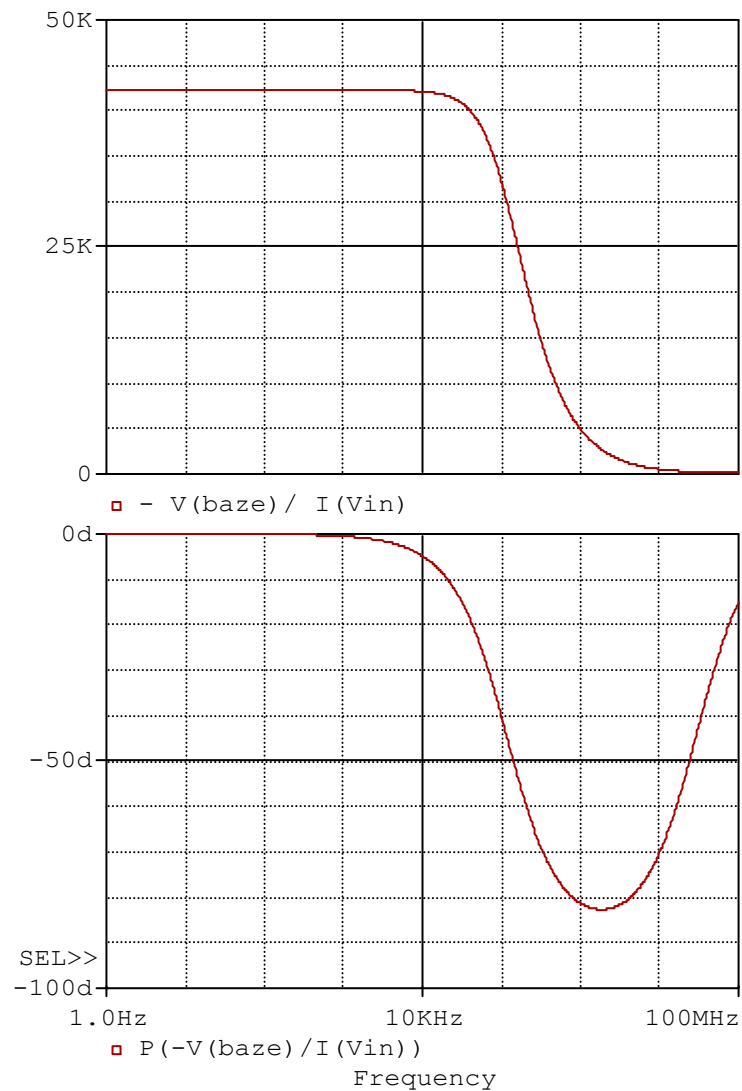


SS pracovní bod:

	NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE	
(in)	0.0000	(bat)	12.0000	(emi)	.6497	(kol)	5.5370	(baze)	1.3433

Příklad 8.16 *Určení vstupní impedance v zesilovači z příkladu 8.15*

```
tranzistorovy zesilovac
Vin baze 0 DC 1.3433V AC 1
Rc bat kol 2k
Re emi 0 200
Vbat bat 0 12V
Q kol baze emi BC107A
.AC DEC 100 1 100meg
.probe
.lib
.end
```



Vstupní impedance na nízkých kmitočtech má odporový charakter, odpor je asi $42.3\text{k}\Omega$. Je to daleko více než je součet vstupního odporu samotného tranzistoru h_{11e} a odporu R_e . Pro tento případ platí vzorec ze zpětnovazební teorie $h_{11e} + h_{21e} \cdot R_e$.

Příklad 8.17 *Určení výstupní impedance celého zesilovače z příkladu 8.15*

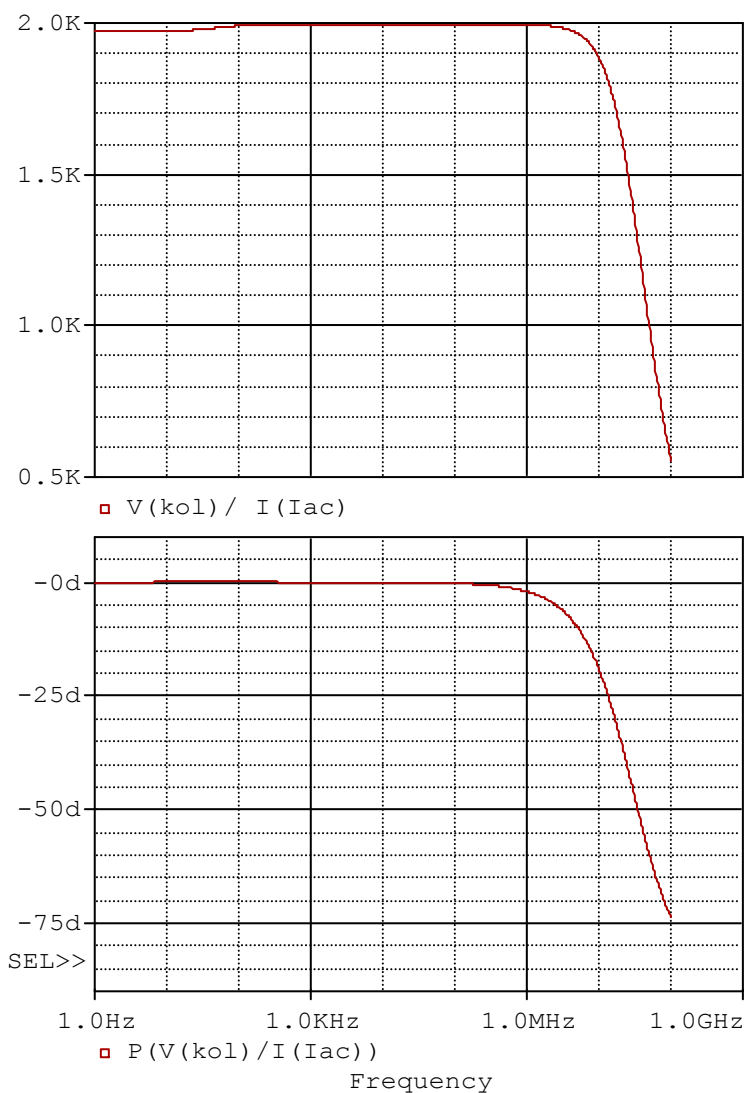
```

tranzistorovy zesilovac
Vin in 0 AC 0
Cv in baze 330nF
Rb1 bat baze 100k
Rb2 baze 0 15k
Rc bat kol 2k
Re emi 0 200
Vbat bat 0 12V
Q kol baze emi BC107A
Iac 0 kol AC=1
.AC DEC 100 1 100meg
.probe
.lib
.end

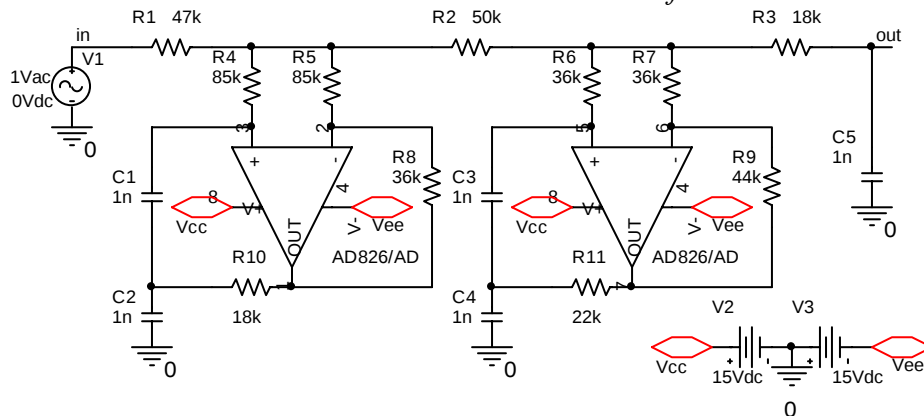
```

Pro určování výstupní impedance je třeba budit zesilovač do výstupu, a to zdrojem proudu, který by nezpůsoboval změnu nastaveného pracovního bodu. Původní střídavý zdroj na vstupu nyní nesmí působit.

Výstupní impedance je o něco menší než odpor R_c , tedy $2\text{k}\Omega$. paralelně k němu je výstupní odpor tranzistoru, modifikovaný odporem R_e . Na nízkých kmitočtech se uplatňuje nepůsobení vazební kapacity, na vysokých kmitočtech zase pokles zesilovací schopnosti tranzistoru.



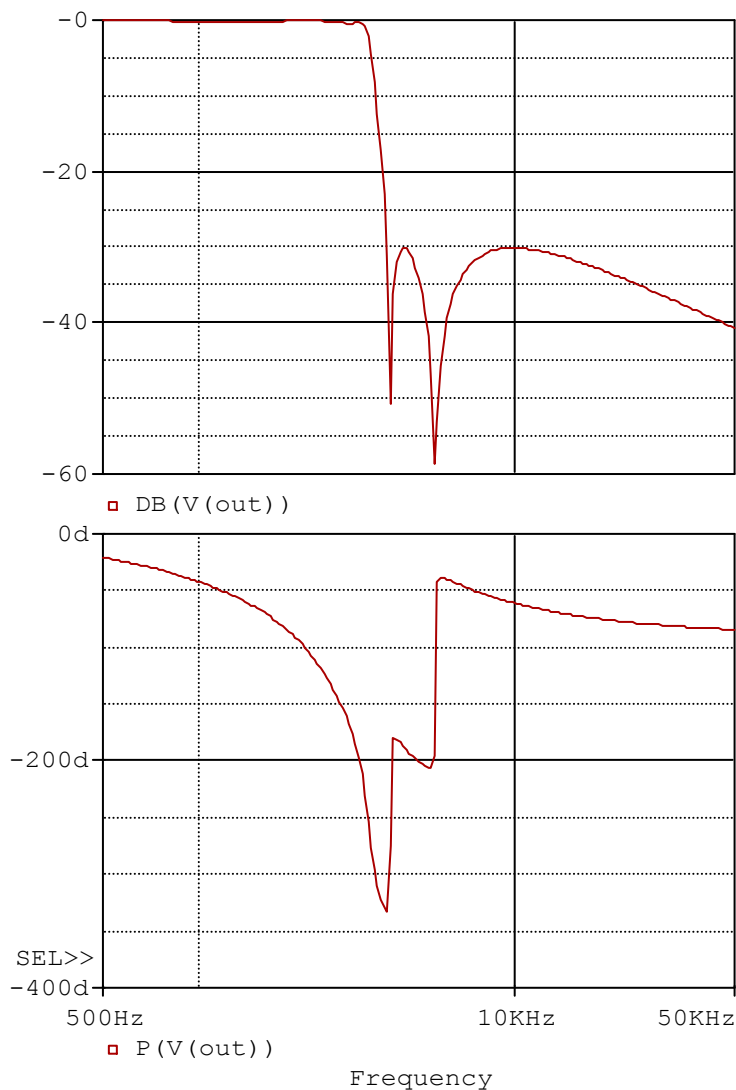
Příklad 8.18 Kmitočtová charakteristika aktivního filtru



```

Aktivni filtr Cauer DP 3.4kHz
Vin in 0 AC 1
X1 c d Vcc Vee h AD826/AD
X2 e f Vcc Vee j AD826/AD
Vplus Vcc 0 15V
Vminus 0 Vee 15V
R1 in a 47k
R2 a b 50k
R3 b out 18k
R4 a c 85k
R5 a d 85k
R6 b e 36k
R7 b f 36k
R8 d h 36k
R9 f j 44k
R10 g h 18k
R11 i j 22k
C1 c g 1n
C2 g 0 1n
C3 e i 1n
C4 i 0 1n
C5 out 0 1n
.AC DEC 100 500 50k
.probe
.lib
.end

```



Příklad 8.19 Behaviorální modelování filtru z příkladu 8.18

```

behavioralni model filtru
Vin in 0 AC 1
Efiltr out2 0 LAPLACE
+{V(in)} {K*(s^2+A1^2)*(s^2+A2^2)/((s+B)*((s+R1)^2+I1^2)*((s+R2)^2+I2^2))}
.param K= -2919.75817532289
.param A1= {35533.4527259350} A2= 25565.4996282456
.param B= -14241.7235360065
.param R1= -6944.76097738481 R2= -1360.43294756782
.param I1= 17530.2667416564, I2= 21771.7554665257
*
.probe
.end

```

Kmitočtové charakteristiky jsou prakticky shodné s charakteristikami filtru z příkladu 8.18.

13 Seznam použité literatury

- [1] Vladimirescu, A. The SPICE Book. John Willey & Sons, Inc., 1994.
- [2] Kielkowski, R. Inside SPICE. McGraw-Hill, 1998.
- [3] Kundert, K.S. The Designer's Guide to SPICE@SPECTRE®. Kluwer Academic Publishers, 1995.
- [4] Elektronická dokumentace k OrcadPSpice 10: pspcref.pdf, psp_sug.pdf
- [5] Kejhar, M. a kol. Program SPICE v příkladech. Skriptum ČVUT Praha, 1995.
- [6] Láníček, R. Simulační programy pro elektroniku. BEN technická literatura, 2000, 113 s.
- [7] Biolek, Z. Úvod do SPICE pomocí programu MicroCap. Učební texty SPŠE v Rožnově p.R. SENSIT HOLDING s.r.o., 2004, 34 s.
- [8] Biolek, D. Řešíme elektronické obvody aneb kniha o jejich analýze. BEN technická literatura, 2004, 520 s.
- [9] Katalogové listy z Internetu firmy Analog Devices, www.analog.com.
- [10] Katalogové listy z Internetu firmy Linear Technology, www.linear.com.